aws

Entwicklerhandbuch

Amazon MQ

Amazon MQ: Entwicklerhandbuch

Table of Contents

Was ist Amazon MQ?

Amazon MQ ist ein verwalteter Message Broker-Service für Apache ActiveMQ Classic und RabbitMQ, der die Einrichtung, den Betrieb und die Wartung von Message Brokern verwaltet. Sie können einen neuen Amazon MQ-Broker mit branchenüblichen Messaging-Protokollen erstellen oder bestehende Message Broker zu Amazon MQ migrieren, ohne den Messaging-Code neu schreiben zu müssen.

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Mit einem Message Broker können Software-Anwendungen und -Komponenten mithilfe verschiedener Programmiersprachen, Betriebssysteme und formeller Messaging-Protokolle miteinander kommunizieren. Sie können Amazon MQ-Broker für die Kommunikation zwischen großen, Cloud-nativen Anwendungen und Komponenten verwenden.

Themen

Funktionen von Amazon MQ

Verwaltete Wartung und Versionsupgrades

Amazon MQ führt während Ihres geplanten Wartungsfensters Wartungs - und Versionsupgrades für einen Message Broker durch.

Überwachen Sie Makler mit CloudWatch

Amazon MQ ist in Amazon integriert, CloudWatch sodass Sie Kennzahlen für Ihre Broker und Warteschlangen anzeigen und analysieren können. Sie können Metriken über die Amazon MQ MQ-Konsole, die CloudWatch Konsole, die Befehlszeile und die API anzeigen und analysieren. Metriken werden automatisch erfasst und auf CloudWatch jede Minute übertragen.

Sicherheit

Amazon MQ bietet Verschlüsselung Ihrer Nachrichten im Ruhezustand und während der Übertragung. Verbindungen zum Broker verwenden SSL, und der Zugriff kann auf einen privaten

Endpunkt in Ihrer Amazon VPC beschränkt werden. Darüber hinaus können Sie AWS Identity and Access Management(IAM) verwenden, um die Aktionen zu steuern, die Ihre IAM-Benutzer und -Gruppen bei bestimmten Amazon MQ-Brokern ausführen können.

Quorum-Warteschlangen für RabbitMQ auf Amazon MQ

Quorum-Warteschlangen sind replizierte Warteschlangenarten, die aus einem Leader-Knoten (primäres Replikat) und Follower-Knoten (andere Replikate) bestehen. Jeder Knoten befindet sich in einer anderen Availability Zone. Wenn also ein Knoten vorübergehend nicht verfügbar ist, wird die Nachrichtenzustellung mit einem neu gewählten Leader-Replikat in einer anderen Availability Zone fortgesetzt. Quorumwarteschlangen sind nützlich für den Umgang mit giftigen Nachrichten, die entstehen, wenn eine Nachricht fehlschlägt und mehrfach in die Warteschlange gestellt wird.

Regionsübergreifende Datenreplikation für ActiveMQ auf Amazon MQ

Die regionsübergreifende Datenreplikation (CRDR) ermöglicht die asynchrone Nachrichtenreplikation vom primären Broker in einer primären AWS Region zum Replikatbroker in einer Replikatregion. Durch eine Failover-Anfrage an die Amazon-MQ-API wird der aktuelle Replikat-Broker in die Rolle des Primär-Brokers hochgestuft und der aktuelle Primär-Broker wird in die Rolle des Replikat-Brokers heruntergestuft.

Wie sehen meine ersten Schritte mit Amazon MQ aus?

Informationen zu den ersten Schritten mit ActiveMQ auf Amazon MQ finden Sie in der folgenden Dokumentation:

- Erste Schritte: Einen ActiveMQ-Broker erstellen und eine Verbindung zu ihm herstellen
- the section called “Bereitstellen eines Brokers”
- ActiveMQ-Tutorials
- the section called “Best Practices für Amazon MQ für ActiveMQ”

Lesen Sie die folgende Dokumentation, um mit RabbitMQ auf Amazon MQ zu beginnen:

- Erste Schritte: Einen RabbitMQ-Broker erstellen und eine Verbindung zu ihm herstellen
- the section called “Bereitstellen eines RabbitMQ-Brokers”
- the section called “RabbitMQ-Tutorials”
- the section called “Best Practices für Amazon MQ for RabbitMQ”

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

Weitere Informationen zu Amazon AWS CLI MQ-Befehlen finden Sie unter Amazon MQ in der AWS CLI Befehlsreferenz.

Wie kann ich Amazon MQ Feedback geben?

Wir freuen uns über Ihr Feedback zur Dokumentation und freuen uns über Ihr Feedback. Sie können die Symbole „Daumen hoch“ und „Daumen runter“ auf der rechten Seite verwenden, um Feedback einzureichen, oder Sie können das unten verlinkte Formular „Feedback geben“ verwenden.

Verwenden Sie das Amazon MQ MQ-Diskussionsforum, um das Amazon MQ MQ-Team zu kontaktieren.

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.

   Während der Anmeldung erhalten Sie einen Telefonanruf oder eine Textnachricht und müssen 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 Benutzer 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. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

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-Managementkonsoleals 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 den Root-Benutzer.

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

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center -Benutzerhandbuch.
2. Gewähren Sie einem Administratorbenutzer im IAM Identity Center Benutzerzugriff.

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

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-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 Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen 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-Managementkonsole interagieren möchten. Die Art und Weise, wie programmatischer Zugriff gewährt wird, hängt vom Benutzertyp ab, der zugreift. AWS

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

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	(Empfohlen) Verwenden Sie Konsolenanmeldeinformationen als temporäre Anmeldeinformationen, um programmatische Anfragen an AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zu den AWS CLI finden Sie unter Anmeldung für AWS lokale Entwicklung im AWS Command Line Interface Benutzerhandbuch. • Weitere Informationen finden Sie unter Anmeldung

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
		für AWS lokale Entwicklung im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch. AWS SDKs
Mitarbeiteridentität (Benutzer, die in IAM Identity Center verwaltet werden)	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. • Informationen zu AWS SDKs Tools und AWS APIs finden Sie unter IAM Identity Center-Authentifizierung im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Folgen Sie den Anweisungen unter Verwenden temporärer Anmeldeinformationen mit AWS Ressourcen im IAM-Benutzerhandbuch.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder zu signieren. AWS APIs	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu AWS CLI finden Sie unter Authentifizierung mithilfe von IAM-Benutzeranmeldeinformationen im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldeinformationen im Referenzhandbuch AWS SDKs und im Tools-Referenzhandbuch. • Weitere Informationen finden Sie unter Verwaltung von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch. 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-Managementkonsole wie Sie programmgesteuert 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 mithilfe der Amazon MQ REST API 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 dannVerbinden einer Java-Anwendung mit Ihrem Amazon MQ for ActiveMQ -Brokeroder verwenden Sie die RabbitMQ Java-Client-Bibliothek, umVerbinden Sie eine JVM-basierte Anwendung mit Ihrem Amazon MQ for RabbitMQ Broker.

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-Instance-Klasse (`m5`) und der Größe (`large`,`medium`) wird als Broker-Instance-Typ bezeichnet (z. B.`mq.m5.large`). Weitere Informationen finden Sie unter Was ist ein Amazon MQ for ActiveMQ-Broker?.

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-Managementkonsole , 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 Amazon EFS Speicher-Volume. 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 Amazon EFS. Weitere Informationen finden Sie unter Option 2: Amazon active/standby MQ-Broker für hohe Verfügbarkeit.

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

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

c. Wählen Sie Next (Weiter).

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

   a. Geben Sie den Broker-Namen ein.

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

> **Note**
>
> Im Abschnitt Zusätzliche Einstellungen können Sie auch Folgendes konfigurieren:
>
> - Konfigurationen
> - CloudWatch logs
> - Privater Zugriff
> - Wartungsfenster für Makler

   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 Passwort muss mindestens 12 Zeichen lang sein, muss mindestens 4 eindeutige Zeichen enthalten und darf keine Kommas, Doppelpunkte oder Gleichheitszeichen (,:=) enthalten.

> **Important**
>
> Fügen Sie keine persönlich identifizierbare Informationen (PII) oder andere vertrauliche oder sensible Informationen in Broker-Benutzernamen 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 ***MyBroker***.

   Notieren Sie sich auf der ***MyBroker***Seite im Bereich Connect die ActiveMQ-Webkonsolen-URL Ihres Brokers, 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
   ```

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-Instanzklasse (`m5`) und der Größe (`large`,`medium`) wird als Broker-Instance-Typ bezeichnet (z. B.`mq.m5.large`). Weitere Informationen finden Sie unter Was ist ein Amazon MQ for RabbitMQ Broker?.

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-Managementkonsole Broker zu erstellen.

Wenn Sie einen Amazon MQ for RabbitMQ-Broker erstellen, befolgen Sie die Best Practices für die Brokereinrichtung für RabbitMQ, um die Broker-Leistung zu maximieren und die Effizienz des Nachrichtendurchsatzes zu optimieren.

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 Amazon EBS-Speicher-Volume. 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 keine persönlich identifizierbare Informationen (PII) oder andere vertrauliche oder sensible Informationen in Brokernamen hinzu. Broker-Namen sind für andere Dienste zugänglich, einschließlich Logs. AWS CloudWatch Brokernamen sind nicht für private oder sensible Daten gedacht.

b. Wählen Sie den Broker-Instanztyp (z. B. mq.m7g.large). Weitere Informationen finden Sie unter Broker instance types.

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.
   - 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 keine persönlich identifizierbare Informationen (PII) oder andere vertrauliche oder sensible Informationen in Broker-Benutzernamen hinzu. Broker-Benutzernamen sind für andere Dienste zugänglich, einschließlich Logs. AWS CloudWatch Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

> **Note**
>
> Im Abschnitt Zusätzliche Einstellungen können Sie auch Folgendes konfigurieren:
>
> - Konfigurationen
> - CloudWatch logs
> - Privater Zugriff

- Wartungsfenster für Makler

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 ***MyBroker***.

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

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

   Beachten Sie auch dieSicherer AMQP-Endpunkt. Es folgt ein Beispiel für einamqpsEndpunkt Zuweisen auf Listener-Port5671.

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

Verwalten eines Amazon MQ -Brokers

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

Themen

Herstellen einer Verbindung mit Amazon MQ

Sie können von anderen AWS Services aus über Service- und Broker-Endpunkte eine Verbindung zu Amazon MQ herstellen.

Service-Endpunkte

Die folgenden Verbindungsmethoden werden für die Amazon MQ-Service-API verwendet:

Domains	Verbindungsmethode
`mq.region.amazonaws.com`	IPv4
`mq.region.api.aws`	Dual-Stack (IPv4 und IPv6)
`mq-fips.region.amazonaws.com`	FIPS nur mit IPv4

Domains	Verbindungsmethode
`mq-fips.region.api.aws`	FIPS mit Dual-Stack

Broker-Endpunkte

Die folgenden Verbindungsmethoden werden für Amazon MQ-Broker verwendet:

Domains	Verbindungsmethode
`brokerId.mq.region.amazonaws.com`	IPv4
`brokerId.mq.region.on.aws`	Dual-Stack (IPv4 und IPv6) **Note** Amazon MQ für ActiveMQ-Broker unterstützen Dual-Stack nicht.

Stellen Sie mithilfe von Dual-Stack IPv4 - (und IPv6) Endpunkten eine Connect zu Amazon MQ her

Dual-Stack-Endpunkte unterstützen sowohl den als auch den Datenverkehr. IPv4 IPv6 Wenn Sie eine Anfrage an einen Dual-Stack-Endpunkt stellen, wird die Endpunkt-URL in eine Adresse oder eine IPv4 Adresse aufgelöst. IPv6 Weitere Informationen zu Dual-Stack- und FIPS-Endpunkten finden Sie im SDK-Referenzhandbuch.

Amazon MQ unterstützt regionale Dual-Stack-Endpunkte, was bedeutet, dass Sie die AWS Region als Teil des Endpunktnamens angeben müssen. Dual-Stack-Endpunktnamen verwenden die folgende Namenskonvention:. `mq.region.api.aws` Beispielsweise ist der Dual-Stack-Endpunktname für die Region `eu-west-1` `mq.eu-west-1.api.aws`.

Die vollständige Liste der Amazon MQ MQ-Endpunkte finden Sie in der AWS Allgemeinen Referenz.

Connect zu Amazon MQ her mit AWS PrivateLink

AWS PrivateLinkEndpunkte für die Amazon MQ MQ-API mit Unterstützung IPv4 und IPv6 Bereitstellung privater Konnektivität zwischen virtuellen privaten Clouds (VPCs) und der Amazon MQ MQ-API, ohne dass Ihr Datenverkehr dem öffentlichen Internet ausgesetzt wird.

> **Note**
>
> Support für PrivateLink ist nur für den Amazon MQ API-Endpunkt verfügbar, nicht für den Broker-Endpunkt. Weitere Informationen zur privaten Verbindung mit einem Broker-Endpunkt finden Sie unterConfiguring a private Amazon MQ broker.

Für den Zugriff auf die Amazon MQ MQ-API PrivateLink müssen Sie zunächst einen VPC-Schnittstellen-Endpunkt in der spezifischen VPC erstellen, von der aus Sie eine Verbindung herstellen möchten. Wenn Sie den VPC-Endpunkt erstellen, verwenden Sie den Dienstnamen `com.amazonaws.region.mq` oder `com.amazonaws.region.mq-fips` für FIPS-Endpunkte.

Wenn Sie Amazon MQ über die AWS CLI oder das SDK aufrufen, müssen Sie die Endpunkt-URL angeben, um den Dual-Stack-Domainnamen zu verwenden: `mq.region.api.aws` oder. `mq-fips.region.api.aws` PrivateLink für Amazon MQ unterstützt nicht den Standard-Domainnamen, der auf `amazonaws.com` endet. Weitere Informationen finden Sie unter Dual-Stack- und FIPS-Endpunkte im SDK-Referenzhandbuch.

Das folgende CLI-Beispiel zeigt, wie Sie die `describe-broker-engine-type` in der Region Asien-Pazifik (Sydney) über einen Amazon MQ VPC-Endpunkt aufrufen.

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

Weitere Möglichkeiten zur Konfiguration des Endpunkts in der CLI finden Sie unter Verwenden von Endpunkten in der AWS CLI

Sie können den Benutzerzugriff auf die VPC-Endpoints auch mithilfe von VPC-Endpunktrichtlinien festlegen. Weitere Informationen finden Sie unter Steuern des Zugriffs auf VPC-Endpoints mithilfe von Endpunktrichtlinien.

Authentifizierung und Autorisierung für Amazon MQ-Broker

Amazon MQ bietet mehrere Authentifizierungs- und Autorisierungsmethoden, um Ihre Messaging-Infrastruktur gemäß den Anforderungen Ihres Unternehmens zu schützen.

Authentifizierung und Autorisierung für Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ unterstützt die folgenden Authentifizierungs- und Autorisierungsmethoden:

Einfache Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer intern im RabbitMQ-Broker gespeichert und über die Webkonsole oder die Management-API verwaltet. Berechtigungen für Vhosts, Exchanges, Warteschlangen und Themen werden direkt in RabbitMQ konfiguriert. Dies ist die Standardmethode. Weitere Informationen finden Sie unter Einfache Authentifizierung und Autorisierung.

OAuth 2.0 Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer und ihre Berechtigungen von einem externen OAuth 2.0-Identitätsanbieter (IdP) verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen für Vhosts, Exchanges, Warteschlangen und Themen werden über das Scope-System des OAuth 2.0-Anbieters zentralisiert. Dies vereinfacht die Benutzerverwaltung und ermöglicht die Integration in bestehende Identitätssysteme. Weitere Informationen finden Sie unter Authentifizierung und Autorisierung OAuth 2.0.

IAM-Authentifizierung und -Autorisierung

Bei dieser Methode authentifizieren sich Broker-Benutzer mithilfe von AWS IAM-Anmeldeinformationen über den IAM-Outbound-Federation. IAM-Anmeldeinformationen werden verwendet, um JWT-Token vom AWS Security Token Service (STS) abzurufen, und diese JWT-Token dienen als 2.0-Token für die Authentifizierung. OAuth Diese Methode nutzt die bestehende OAuth 2.0-Unterstützung in Amazon MQ für RabbitMQ, wo sie als 2.0-Identitätsanbieter AWS fungiert. OAuth Die Benutzerauthentifizierung wird von AWS IAM abgewickelt, während die Ressourcenberechtigungen für Vhosts, Exchanges, Warteschlangen und Themen über in RabbitMQ konfigurierte IAM-Richtlinien und Bereichsaliase verwaltet werden. Weitere Informationen finden Sie unter IAM-Authentifizierung und -Autorisierung.

LDAP-Authentifizierung und -Autorisierung

Bei dieser Methode werden Broker-Benutzer und ihre Berechtigungen von einem externen LDAP-Verzeichnisdienst verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen werden über den LDAP-Server zentralisiert, sodass Benutzer mit ihren vorhandenen Verzeichnisdienstanmeldedaten auf RabbitMQ zugreifen können. Weitere Informationen finden Sie unter LDAP-Authentifizierung und -Autorisierung.

HTTP-Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer und ihre Berechtigungen von einem externen HTTP-Server verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen werden über den HTTP-Server zentralisiert, sodass Benutzer über ihren eigenen Authentifizierungs- und Autorisierungsanbieter auf RabbitMQ zugreifen können. Weitere Informationen zu dieser Methode finden Sie unter HTTP-Authentifizierung und Autorisierung.

Authentifizierung mit SSL-Zertifikaten

Amazon MQ unterstützt Mutual TLS (mTLS) für RabbitMQ-Broker. Das SSL-Authentifizierungs-Plugin verwendet Client-Zertifikate von mTLS-Verbindungen, um Benutzer zu authentifizieren. Bei dieser Methode werden Broker-Benutzer mithilfe von X.509-Clientzertifikaten anstelle von Benutzernamen und Kennwörtern authentifiziert. Das Zertifikat des Clients wird anhand einer vertrauenswürdigen Zertifizierungsstelle (CA) validiert, und der Benutzername wird aus einem Feld im Zertifikat extrahiert, z. B. dem Common Name (CN) oder dem Subject Alternative Name (SAN). Diese Methode bietet eine starke Authentifizierung, ohne dass Anmeldeinformationen über das Netzwerk übertragen werden müssen. Weitere Informationen finden Sie unter SSL-Zertifikatsauthentifizierung.

> **Note**
>
> RabbitMQ unterstützt mehrere Authentifizierungs- und Autorisierungsmethoden, die gleichzeitig verwendet werden können. Sie können beispielsweise sowohl OAuth 2.0 als auch die einfache (interne) Authentifizierung aktivieren. Weitere Informationen finden Sie im OAuth 2.0-Tutorial-Abschnitt zur Aktivierung sowohl der OAuth 2.0-Authentifizierung als auch der einfachen (internen) Authentifizierung sowie in der Dokumentation zur RabbitMQ-Zugriffskontrolle.
>
> Amazon MQ empfiehlt, beim Testen von Authentifizierungskonfigurationen einen internen Benutzer zu erstellen. Auf diese Weise kann die Zugriffskonfiguration mithilfe der RabbitMQ-Management-API validiert werden. Weitere Informationen finden Sie unter Zugriffsvalidierung.

Authentifizierung und Autorisierung für Amazon MQ for ActiveMQ

Amazon MQ for ActiveMQ unterstützt die folgenden Authentifizierungs- und Autorisierungsmethoden:

Einfache Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer über die Amazon MQ MQ-Konsole oder API erstellt und verwaltet. Benutzern können spezifische Berechtigungen für den Zugriff auf Warteschlangen, Themen und die ActiveMQ Web Console zugewiesen werden. Weitere Informationen zu dieser Methode finden Sie unter ActiveMQ-Broker-Benutzer erstellen.

LDAP-Authentifizierung und -Autorisierung

Bei dieser Methode authentifizieren sich Broker-Benutzer anhand der auf Ihrem LDAP-Server gespeicherten Anmeldeinformationen. Über den LDAP-Server können Sie Benutzer hinzufügen, löschen und ändern sowie Themen und Warteschlangen Berechtigungen zuweisen, sodass eine zentrale Authentifizierung und Autorisierung gewährleistet ist. Weitere Informationen zu dieser Methode finden Sie unter ActiveMQ-Broker mit LDAP integrieren.

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` `X`Bezeichnet in Amazon MQ MQ-Implementierungen die Hauptversion, `Y` steht für die Nebenversion und `Z` gibt die Patch-Versionsnummer an. Amazon MQ unterstützt zwei Arten von Upgrades:

- Hauptversions-Upgrade: Tritt auf, wenn sich die Versionsnummern der Haupt-Engine ändern. Beispielsweise wird ein Upgrade von RabbitMQ Version 3.13 auf Version 4.2 als Hauptversions-Upgrade betrachtet.
- Unterversion-Upgrade: Tritt auf, wenn sich nur die Versionsnummer der Neben-Engine ändert. Zum Beispiel ein Upgrade von Version 3. 11 auf Version 3. 12 wird als geringfügiges Versionsupgrade betrachtet.

Sie können Ihren Broker jederzeit manuell auf die nächste unterstützte Haupt- oder Nebenversion aktualisieren. Amazon MQ verwaltet das Upgrade auf die neueste unterstützte Patch-Version für

alle Broker während des geplanten Wartungsfensters. Sowohl manuelle als auch automatische Versionsupgrades erfolgen während des geplanten Wartungsfensters oder nach dem Neustart Ihres Brokers. Amazon MQ aktualisiert Ihren Broker auf die nächste Nebenversion, wenn der Support für die aktuelle Nebenversion ausläuft.

Manuelles Upgraden der Engine-Version

Sie können die Engine-Version eines Brokers mithilfe der AWS-Managementkonsole, der oder der AWS CLI Amazon MQ MQ-API aktualisieren.

AWS-Managementkonsole

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

1. Klicken Sie auf der Seite Broker-Details auf Edit.
2. UNDERTechnische Daten, für-Broker-Engine-VersionWählen Sie in der Dropdown-Liste die neue Versionsnummer .
3. Scrollen Sie ans Seitenende und wählen SieÄnderungen im Zeitplan.
4. 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**
   >
   > Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern ist jeweils nur ein Knoten ausgefallen, während der Broker neu gestartet wird.

5. 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 Update-Broker-CLI-Befehl und spezifizieren Sie die folgenden Parameter wie im Beispiel dargestellt an.

   - `--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`.
   - `--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**
   >
   > Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern ist jeweils nur ein Knoten ausgefallen, während der Broker neu gestartet wird.

Amazon MQ-API

So upgraden Sie die Engine-Version eines Brokers über die Amazon MQ-API:

1. Verwenden Sie die API-Operation UpdateBroker. 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 RebootBrokerAPI-Vorgang, 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**
>
> Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern ist jeweils nur ein Knoten ausgefallen, während der Broker neu gestartet wird.

Upgrade eines Amazon MQ-Broker-Instance-Typs

> **Important**
>
> `mq.m7g.x`Instances sind nur für Amazon MQ für RabbitMQ-Broker verfügbar. Amazon MQ for ActiveMQ-Broker verwenden nur Instances. `mq.m5.x`

Die kombinierte Beschreibung der Broker-Instance-Klasse (`m7g`) und der Größe (`large`) wird als Broker-Instance-Typ bezeichnet (z. B.). `mq.m7g.large` 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 (`mq.m7g.medium`) werden nur zum Testen der Anwendungsleistung empfohlen. Wir empfehlen größere Broker-Instance-Typen (`mq.m7g.large` und höher) für die Produktion von Clients und Warteschlangen, hohen Durchsatz, Nachrichten im Speicher und redundante Nachrichten.

Wir empfehlen ein Upgrade auf einen größeren Instance-Typ (d. h. von `micro` bis`large`), wenn Sie Leistungsprobleme haben oder wenn Sie von einer Test- zu einer Produktionsumgebung wechseln. Um Ihren Instance-Typ zu aktualisieren, können Sie die AWS-Managementkonsole AWS CLI, oder die Amazon MQ MQ-API verwenden.

AWS-Managementkonsole

Gehen Sie wie folgt vor, um mithilfe der AWS-Managementkonsole auf einen größeren Instance-Typ umzusteigen:

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. Wählen Sie unter Spezifikationen für Broker-Instanztyp den neuen Instanztyp aus der Dropdownliste aus.
5. Wählen Sie unten auf der Seite die Option Änderungen planen aus.
6. Klicken Sie auf derÄnderungen für Broker-Seite, fürWann Änderungen angewendet werdenWählen Sie eine der folgenden Optionen .

   - Wählen Sie Nach dem nächsten Neustart, wenn Amazon MQ das Upgrade während des nächsten geplanten Wartungsfensters abschließen soll.
   - Wählen Sie Sofort, wenn Sie den Broker neu starten und den Instance-Typ sofort aktualisieren möchten.

   > **Important**
   >
   > Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern ist jeweils nur ein Knoten ausgefallen, während der Broker neu gestartet wird.

7. Wählen Sie Anwenden, um die Anwendung der Änderungen abzuschließen.

AWS CLI

Um den Instanztyp eines Brokers zu aktualisieren, verwenden Sie den AWS CLI

1. Verwenden Sie den CLI-Befehl modify-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.
   - `--host-instance-type`— Die Engine-Versionsnummer für die Broker-Engine, auf die das Upgrade durchgeführt wird.

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

2. (Optional) Verwenden Sie den CLI-Befehl reboot-broker, um Ihren Broker neu zu starten, wenn Sie den Instance-Typ 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**
>
> Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern ist jeweils nur ein Knoten ausgefallen, während der Broker neu gestartet wird.

Amazon MQ-API

Um den Instance-Typ eines Brokers mithilfe der Amazon MQ MQ-API zu aktualisieren

1. Verwenden Sie die API-Operation UpdateBroker. 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 verfügbaren Amazon MQ MQ-Endpunkten finden Sie unter Amazon MQ 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
   ```

   Geben Sie `host-instance-type` in der Payload der Anfrage den Instance-Typ an, auf den der Broker das Upgrade durchführen soll.

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

2. (Optional) Verwenden Sie den RebootBrokerAPI-Vorgang, 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**
>
> Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern ist jeweils nur ein Knoten ausgefallen, während der Broker neu gestartet wird.

Amazon MQ für ActiveMQ-Speichertypen

Amazon MQ für ActiveMQ unterstützt Amazon Elastic File System (EFS) und Amazon Elastic Block Store (EBS). Standardmäßig verwenden ActiveMQ-Broker Amazon EFS für Broker-Speicher. Verwenden Sie Amazon EFS, um die Vorteile der hohen Haltbarkeit und Replikation über mehrere Availability Zones hinweg zu nutzen. Verwenden Sie Amazon EBS, um die Vorteile der niedrigen Latenz und des hohen Durchsatzes zu nutzen.

> **Important**
>
> - Sie können Amazon EBS nur mit dem`mq.m5`Broker-Instance-Typ.
> - 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 einzelnen Availability Zone und unterstützt den ActiveMQ Aktiv/Standby-Bereitstellungsmodus nicht.

Unterschiede zwischen Speichertypen

Die folgende Tabelle gibt einen kurzen Überblick über die Unterschiede zwischen In-Memory-, Amazon EFS- und Amazon EBS-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	Keine
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 über mehrere Server innerhalb einer einzelnen AZ repliziert.

Konfiguration eines privaten Amazon MQ-Brokers

Ein privater Broker ist nicht öffentlich zugänglich und kann nicht von außerhalb Ihrer VPC aufgerufen werden. Bevor Sie einen privaten Broker konfigurieren, sollten Sie sich die folgenden Informationen zu VPCs Subnetzen und Sicherheitsgruppen ansehen:

- VPCs
  - Die Subnetze und Sicherheitsgruppen eines Brokers müssen sich in derselben VPC befinden.
  - Wenn Sie einen privaten Broker verwenden, werden Ihnen möglicherweise IP-Adressen angezeigt, die Sie nicht mit Ihrer VPC konfiguriert haben. Dies sind IP-Adressen aus der Amazon MQ MQ-Infrastruktur, für die keine Aktion erforderlich ist.
- Subnets
  - Wenn sich Subnetze innerhalb einer gemeinsam genutzten VPC befinden, muss die VPC demselben Konto gehören, das den Broker erstellt hat.
  - Wenn keine Subnetze bereitgestellt werden, werden die Standardsubnetze in der Standard-VPC verwendet.
  - Sobald der Broker erstellt wurde, können die verwendeten Subnetze nicht mehr geändert werden.
  - Bei Clustern und active/standby Brokern müssen sich die Subnetze in unterschiedlichen Availability Zones befinden.
  - Bei Single-Instance-Brokern können Sie angeben, welches Subnetz verwendet werden soll. Der Broker wird dann innerhalb derselben Availability Zone erstellt.
- Sicherheitsgruppen
  - Wenn keine Sicherheitsgruppe angegeben wird, werden die Standardsicherheitsgruppen in der Standard-VPC verwendet.
  - Einzelinstanzen, Cluster und active/standby Broker benötigen mindestens eine Sicherheitsgruppe (z. B. die Standardsicherheitsgruppe).

    > **Note**
    >
    > Öffentliche RabbitMQ-Broker verwenden keine Subnetze oder Sicherheitsgruppen.

  - Sobald der Broker erstellt wurde, kann die verwendete Sicherheitsgruppe nicht mehr geändert werden. Die Sicherheitsgruppen selbst können immer noch geändert werden.

Konfiguration eines privaten Brokers im AWS-Managementkonsole

Um einen privaten Broker zu konfigurieren, beginnen Sie mit der Erstellung eines neuen Brokers in der AWS-Managementkonsole. Gehen Sie dann im Abschnitt Netzwerkeinstellungen wie folgt vor, um die Konnektivität Ihres Brokers zu konfigurieren:

1. Wählen Sie Private Access für Ihren Broker. Um eine Verbindung zu einem privaten Broker herzustellen, können Sie IPv4 IPv6, oder Dual-Stack (IPv4 und IPv6) verwenden. Weitere Informationen finden Sie unter Connecting to Amazon MQ.
2. Wählen Sie als Nächstes die Option Standard-VPC, Subnetz (s) und Sicherheitsgruppe (n) verwenden oder Existierende VPC, Subnetz (e) und Sicherheitsgruppe (n) auswählen. Wenn Sie die standardmäßige oder vorhandene VPC, Subnetze oder Sicherheitsgruppe (n) nicht verwenden möchten, müssen Sie eine neue erstellen, um eine Verbindung zum Private Broker herzustellen.

> **Note**
>
> Für den privaten Broker-Zugriff entspricht die Verbindungsmethode dem ausgewählten IP-Typ des Subnetzes. Sobald der Broker erstellt wurde, kann der VPC-Endpunkt nicht mehr geändert werden und hat immer den IP-Typ der ausgewählten Subnetze. Wenn Sie einen neuen IP-Typ verwenden möchten, müssen Sie einen neuen Broker erstellen.

> **Note**
>
> Amazon MQ for ActiveMQ verwendet keine VPC-Endpunkte. Wenn Sie zum ersten Mal einen ActiveMQ-Broker erstellen, stellt Amazon MQ ein elastic network interface (ENI) in der VPC bereit. Sicherheitsgruppen werden in der ENI platziert und können sowohl für öffentliche als auch für private Broker verwendet werden.

Zugriff auf die Amazon MQ Broker-Webkonsole ohne öffentlichen Zugriff

Wenn Sie den öffentlichen Zugriff für Ihren Broker deaktivieren, kann die AWS Konto-ID, mit der der Broker erstellt wurde, auf den privaten Broker zugreifen. Wenn Sie den öffentlichen Zugriff für Ihren Broker deaktivieren, müssen Sie die folgenden Schritte ausführen, um auf die Broker-Webkonsole zuzugreifen.

1. Erstellen Sie eine Linux-EC2-Instance in `public-vpc` (bei Bedarf mit einer öffentlichen IP).
2. Um zu überprüfen, ob Ihre VPC korrekt konfiguriert ist, stellen Sie eine `ssh`-Verbindung mit der EC2-Instance her, und führen Sie den Befehl `curl` mit der URI Ihres Brokers aus.
3. Erstellen Sie auf Ihrem Rechner einen `ssh`-Tunnel zur EC2-Instance unter Verwendung des Pfads zu Ihrer Datei mit dem privaten Schlüssel und der IP-Adresse Ihrer öffentlichen EC2-Instance. Beispiel:

   ```
   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 `localhost` an.
   - Als Port `8080`.
   - Entfernen Sie alle vorhandenen URL-Muster.
   - Geben Sie als URL-Muster `*.mq.*.amazonaws.com*` an.
   - Geben Sie als Verbindungstyp `HTTP(S)` an.

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

> **Important**
>
> Wenn Sie einen privaten Broker verwenden, werden Ihnen möglicherweise IP-Adressen angezeigt, die Sie nicht mit Ihrer VPC konfiguriert haben. Dies sind IP-Adressen aus der RabbitMQ on Amazon MQ MQ-Infrastruktur, für die keine Aktion erforderlich ist.

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 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 active/standby Modus führt Amazon MQ Wartungsarbeiten für eine Instanz nach der anderen durch, und mindestens eine Instanz bleibt verfügbar. Darüber hinaus können Sie ein Netzwerk von Brokern mit Wartungsfenstern konfigurieren, die im Laufe der Woche variieren. 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 die AWS-Managementkonsole AWS CLI, oder die Amazon MQ MQ-API verwenden.

Planen Sie das Broker-Wartungsfenster mithilfe der AWS-Managementkonsole

Um das Broker-Wartungsfenster anzupassen, verwenden Sie AWS-Managementkonsole

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 Beginnzeit-Optionen werden in UTC+0-Zeitzone konfiguriert.

5. Wählen Sie als Nächstes Änderungen planen aus. Wählen Sie dann Nach dem nächsten Neustart oder Sofort. Wenn Sie Nach dem nächsten Neustart wählen, wird das Wartungsfenster sofort aktualisiert, ohne den Broker neu zu starten. Wenn Sie Sofort wählen, wird der Broker sofort neu gestartet.
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 Update-Broker-CLI-Befehl und spezifizieren Sie die folgenden Parameter wie im Beispiel dargestellt an.

   - `--broker-id` - Die eindeutige ID, die Amazon MQ für die Broker-Instance generiert. Sie können die ID von Ihrem Broker ARN analysieren. Beispielsweise angesichts des 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`.
   - `--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, im Land/Stadt-Format, oder das 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 Beschreibung-BrokerCLI-Befehl, um zu überprüfen, ob das Wartungsfenster erfolgreich aktualisiert wurde.

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

Planen Sie das Broker-Wartungsfenster mithilfe der Amazon MQ MQ-API

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

1. Verwenden Sie die API-Operation UpdateBroker. 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 verfügbaren Amazon MQ MQ-Endpunkten finden Sie unter Amazon MQ 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 DescribeBrokerAPI-Vorgang, 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
```

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-Managementkonsole 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. Wählen Sie auf der ***MyBroker***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-Managementkonsole 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***? Geben Sie in das Dialogfeld ein `delete` und wählen Sie dann Löschen.

   Das Löschen eines Brokers dauert ca. 15 Minuten.

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	Description
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_REQUIRED`	Der Broker läuft, befindet sich aber in einem heruntergestuften Zustand und erfordert sofortiges Handeln. Anweisungen zur Behebung des Problems finden Sie, indem Sie den erforderl

Konsole	API	Description
		ichen Code für die Aktion aus der Liste Fehlerbehebung auswählen.

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. Sie können Tags zur Kostenzuweisung verwenden, um Ihre AWS Rechnung so zu 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	Value (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 mithilfe der REST-API anzeigen. Weitere Informationen finden Sie in der Amazon MQ REST API-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 mithilfe der Amazon MQ REST API oder der AWS-Managementkonsole erstellen, 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 und LevelDB, werden nicht unterstützt.

Sie können auf Ihre Broker mithilfe von jeder Programmiersprache, die ActiveMQ unterstützt zugreufen, und indem Sie TLS explizit für die folgenden Protokolle aktivieren:

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

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

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-Instance-Klasse (`m5`) und

der Größe (`large`,`medium`) wird als Broker-Instance-Typ bezeichnet (z. B.`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 Amazon-EFS-Speicher-Volume.
- 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 Amazon EFS.

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 werden während der-Wartungsfensterdefiniert durch den Wochentag, die Tageszeit (im 24-Stunden-Format) und die Zeitzone (standardmäßig UTC).

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 mithilfe von jeder Programmiersprache, die ActiveMQ unterstützt zugreifen, und indem Sie TLS explizit für die folgenden Protokolle aktivieren:

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

Attribute

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

- Einen Namen (`MyBroker`)
- Eine ID (`b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Einen Amazon-Ressourcennamen (ARN) (`arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Eine ActiveMQ-Webkonsolen-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**
    >
    > Das ist ein OpenWire Endpunkt.

  - `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 active/standby Broker bietet Amazon MQ zwei ActiveMQ-Web-Konsolen URLs, aber es ist jeweils nur eine URL aktiv. 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 Abschnitt im Amazon MQ REST API Reference:

- REST-Operations-ID: Broker
- REST-Operations-ID: Broker
- REST-Operations-ID: Broker Reboot

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 Benutzer-Attribute finden Sie im folgenden Abschnitt in der Amazon MQ REST-API-Referenz:

- REST-Operations-ID: User
- REST-Operations-ID: Users

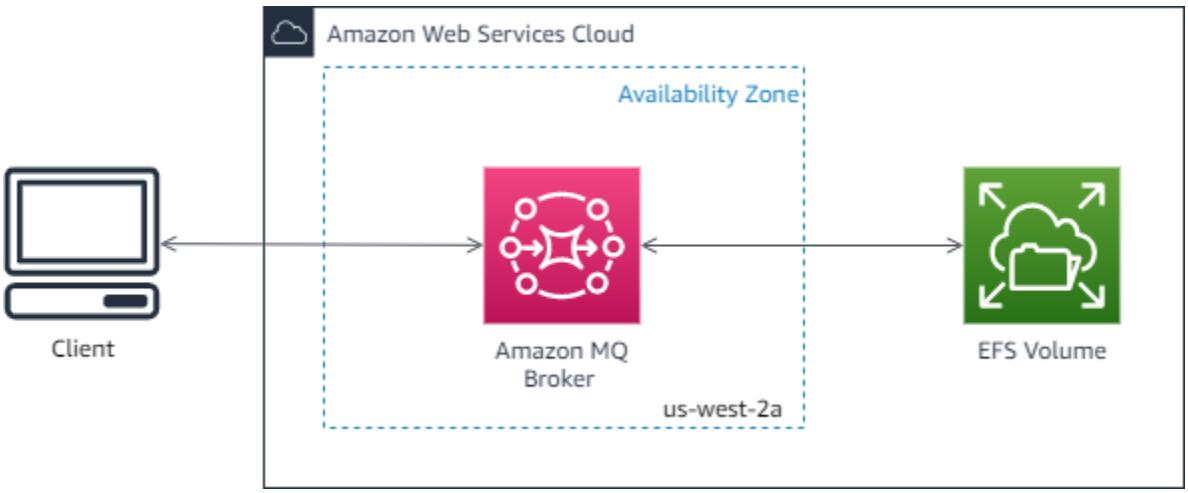
Bereitstellungsoptionen für Amazon MQ für ActiveMQ-Broker

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

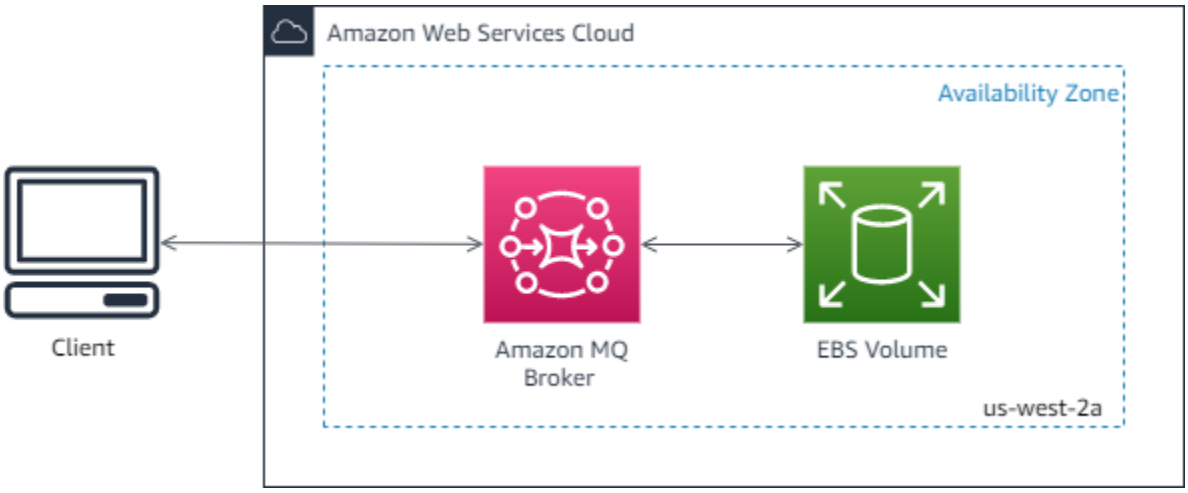
Option 1: Amazon MQ-Broker mit einer einzigen Instanz

Ein Single-Instance-Broker besteht aus einem Broker in einer Availability Zone. Der Broker kommuniziert mit Ihrer Anwendung und mit einem Amazon-EBS- oder Amazon-EFS-Speicher-Volume. Amazon EFS-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 Einzelinstanz-Broker mit Amazon EFS-Speicher, der über mehrere repliziert wird. AZs



Das folgende Diagramm veranschaulicht einen Single-Instance-Broker mit Amazon EBS-Speicher, der über mehrere Server innerhalb einer einzelnen AZ repliziert wird.



Option 2: Amazon active/standby MQ-Broker für hohe Verfügbarkeit

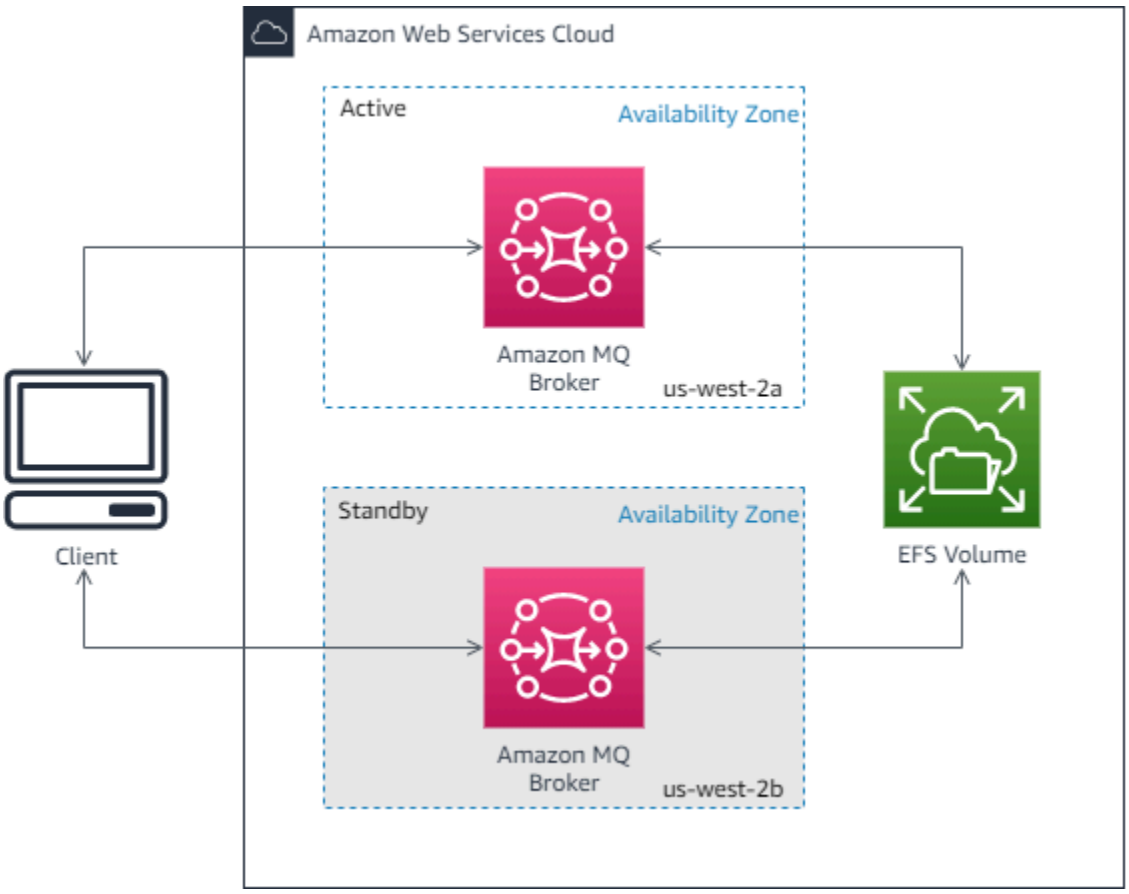
Ein aktiv/standby-Broker besteht aus zwei Brokern in zwei verschiedenen Availability Zones, die in einem redundanten Paar konfiguriert sind. Diese Broker kommunizieren synchron mit Ihrer Anwendung und mit Amazon EFS. Amazon EFS-Speichervolumes sind so konzipiert, dass sie ein Höchstmaß an Haltbarkeit 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. Wartungsfenster und Broker-Neustarts,

die Sie einleiten, führen zu einem Failover. Wenn Sie einen Broker neu starten, dauert das Failover nur wenige Sekunden.

Für einen active/standby Broker bietet Amazon MQ zwei ActiveMQ-Web-Konsolen URLs, aber es ist jeweils nur eine URL aktiv. 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. Bei Protokollendpunkten auf Wire-Level-Ebene sollten Sie Ihrer Anwendung ermöglichen, mithilfe des Failover-Transports eine Verbindung zu einem der beiden Endpunkte herzustellen.

Das folgende Diagramm zeigt einen active/standby Broker mit Amazon EFS-Speicher, der über mehrere AZs repliziert wurde.



Amazon MQ Brokernetzwerk

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

Ein Netzwerk von Brokern besteht aus mehreren gleichzeitig aktiven Einzelinstanz-Brokern oder Brokern. active/standby Durch die Einrichtung eines Netzwerks von Brokern können die

Verfügbarkeit, die Fehlertoleranz und der Lastenausgleich bei mehreren Broker-Instanzen verbessert werden.

Wie funktioniert ein Brokernetzwerk?

Ein Netzwerk von Brokern wird aufgebaut, indem ein Broker über Netzwerkkonnektoren mit einem anderen verbunden wird. Ein Netzwerkconnector stellt On-Demand-Nachrichten von einem Broker zu einem anderen bereit. Netzwerkconnectors werden in der Broker-Konfiguration entweder als Nichtduplex- oder als Duplexverbindungen konfiguriert. Bei Nicht-Duplex-Verbindungen werden Nachrichten nur von einem Broker zum anderen weitergeleitet. Bei Duplexverbindungen werden Nachrichten zwischen beiden Brokern in beide Richtungen weitergeleitet.

Wenn der Netzwerkconnector als Duplex konfiguriert ist, werden Nachrichten auch von Broker2 an Broker1 weitergeleitet.

In einem Netzwerk von Brokern können Sie sowohl Nichtduplex- als auch Duplexverbindungen verwenden. Möglicherweise möchten Sie eine Duplexverbindung zu einem anderen Broker einrichten, um den Datenverkehr zu verbessern oder eine Erhöhung des Limits zu vermeiden. Duplexverbindungen eignen sich auch für die teilweise Migration von lokalen zu von Amazon MQ verwalteten Brokern.

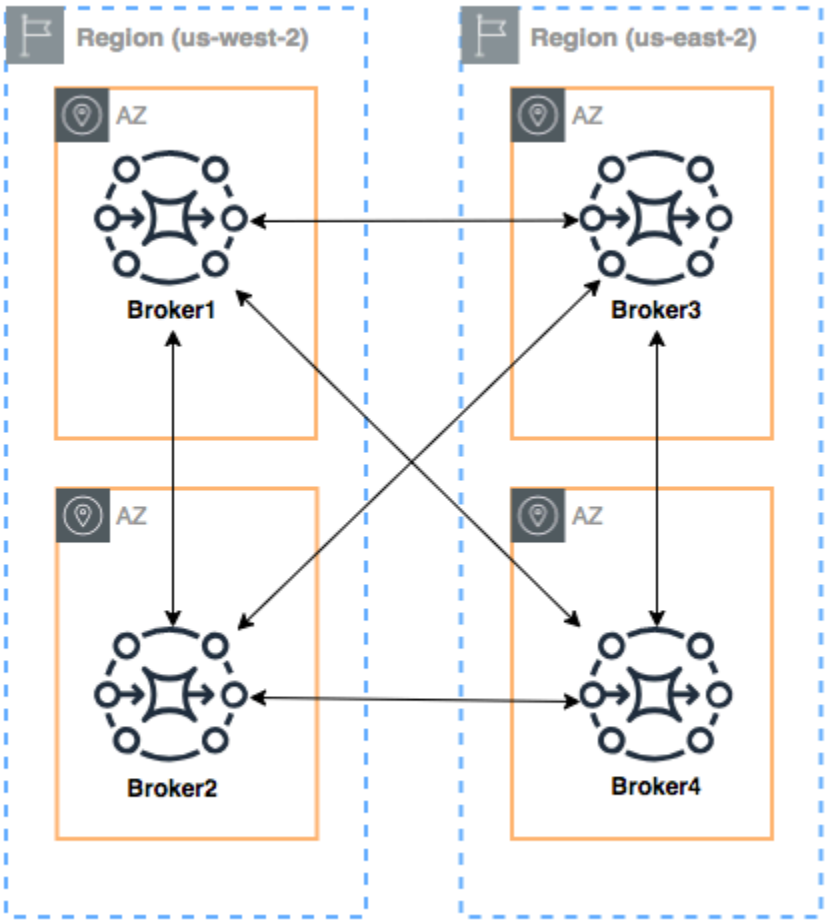
Wie geht ein Netzwerk von Brokern mit Anmeldeinformationen um?

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.

Regionsübergreifend

Um ein regionsübergreifendes AWS Broker-Netzwerk zu konfigurieren, setzen Sie Broker in diesen Regionen ein 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-
west-2.amazonaws.com:61617)"/>
    <networkConnector name="1_to_3" userName="myCommonUser" duplex="true"
```

```
        uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
    <networkConnector name="1_to_4" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-62a7fb31-d51c-466a-a873-905cd660b553-4.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

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-
west-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.

Amazon MQ für ActiveMQ-Broker-Instance-Typen

Die kombinierte Beschreibung der Broker-Instance-Klasse (`m5`) und der Größe (`large`,`medium`) wird als Broker-Instance-Typ bezeichnet (z. B.`mq.m5.large`). In der folgenden Tabelle sind die verfügbaren Amazon MQ-Broker-Instance-Typen für ActiveMQ-Broker aufgeführt.

Amazon MQ informiert Sie mindestens 90 Tage im Voraus, bevor der Support für einen Instance-Typ endet. Wir empfehlen, Ihren Broker vor diesem end-of-support Datum auf einen neuen Instance-Typ zu aktualisieren, um Störungen zu vermeiden.

> **Important**
>
> Am `t2.micro` oder `mq.m4.large` nach dem 17. März 2025 können Sie keine Broker mehr erstellen.

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Empfohlene Verwendung	Speicher	Ende des Supports bei Amazon MQ
`mq.t3.micro`	2	1	Bewertung	EFS	
`mq.m5.large`	2	8	Produktion	EFS oder EBS	
`mq.m5.xlarge`	4	16	Produktion	EFS oder EBS	
`mq.m5.2xlarge`	8	32	Produktion	EFS oder EBS	
`mq.m5.4xlarge`	16	64	Produktion	EFS oder EBS	

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

Konfigurationen für Amazon MQ für ActiveMQ Broker

Eine Konfiguration enthält alle Einstellungen für Ihren ActiveMQ-Broker im XML-Format (ähnlich wie die `activemq.xml`-Datei von ActiveMQ). 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.
> Sie können eine Konfiguration nur mithilfe der `DeleteConfiguration` API löschen. Weitere Informationen finden Sie unter Konfigurationen in der Amazon MQ API-Referenz.

Attribute

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

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

Eine vollständige Liste der Konfigurationsattribute finden Sie im folgenden Abschnitt im Amazon MQ REST API Reference:

- REST-Operations-ID: Configuration
- REST-Operations-ID: Configurations

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

- REST-Operations-ID: Configuration Revision
- REST-Operations-ID: Configuration Revisions

Verwenden von Spring XML-Konfigurationsdateien

ActiveMQ-Broker werden mittels Spring XML-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.

Die vollständige Palette der unterstützten Konfigurationsoptionen wird in den Amazon MQ-XML-Schemas angegeben. Laden Sie ZIP-Dateien der unterstützten Schemas unter Verwendung der folgenden Links herunter.

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

- amazon-mq-active-mq-5.15.16.xsd.zip

Sie können Ihre Konfigurationsdateien anhand dieses Schemas validieren und bereinigen. Amazon MQ ermöglicht Ihnen außerdem die Bereitstellung von Konfigurationen durch Hochladen von XML-Dateien. Beim Hochladen einer XML-Datei werden ungültige und nicht zulässige Konfigurationsparameter von Amazon MQ automatisch entsprechend des Schemas gelöscht und entfernt.

> **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 wie die Datei`activemq.xml` von ActiveMQ). 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.

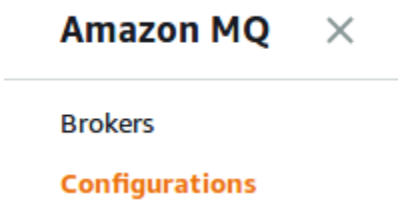
Das folgende Beispiel zeigt, wie Sie eine Amazon MQ-Broker-Konfiguration mithilfe der AWS-Managementkonsole erstellen und anwenden.

> **Important**
>
> Sie können eine Konfiguration nur mithilfe der `DeleteConfiguration` API löschen. Weitere Informationen finden Sie unter Konfigurationen in der Amazon MQ API-Referenz.

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

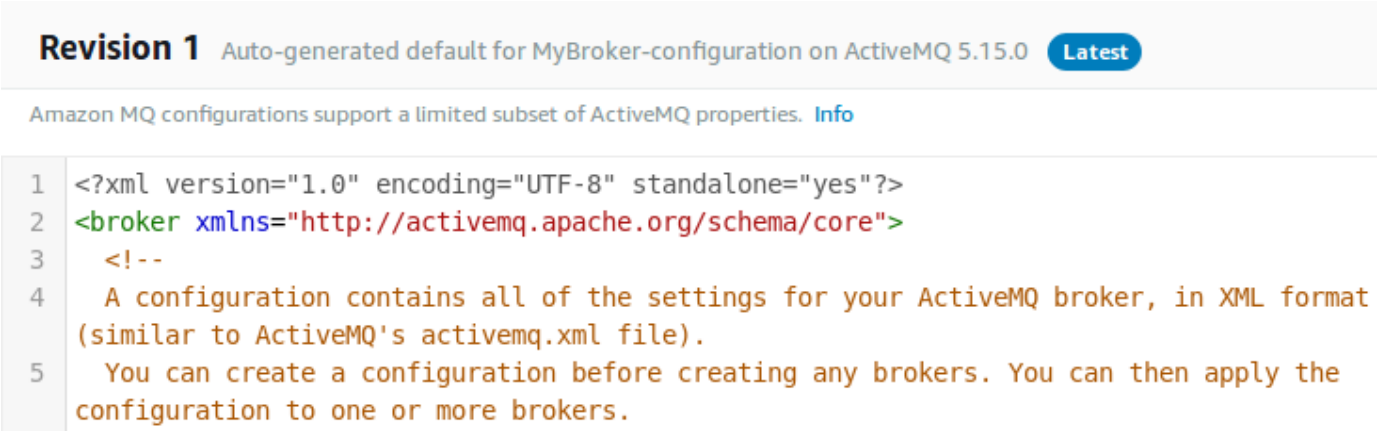
Nachdem Sie eine Broker-Konfiguration erstellt haben, müssen Sie die Konfiguration mithilfe einer Konfigurationsrevision bearbeiten.

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 der ***`MyConfiguration`***Seite werden der Broker-Engine-Typ und die Version angezeigt, die Ihre neue Konfigurationsrevision verwendet (z. B. Apache ActiveMQ 5.15.16).
2. Auf der Registerkarte Configuration details (Konfigurationsdetails) werden die Konfigurations-Revisionsnummer, die Beschreibung und die Broker-Konfiguration im XML-Format angezeigt.

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



3. Klicken Sie aufBearbeiten der KonfigurationNehmen Sie Änderungen an der XML-Konfiguration vor.
4. Wählen Sie Speichern.

   DieSpeichern der Revisionwird angezeigt.
5. (Optional) Geben Sie `A description of the changes in this revision` ein.
6. Wählen Sie Speichern.

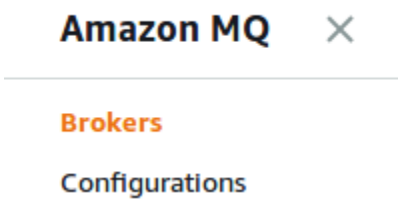
   Die neue Version der Konfiguration wird gespeichert.

> **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. Wählen Sie auf der *MyBroker* Seite Bearbeiten im Abschnitt Konfiguration eine Konfiguration und eine Revision aus und wählen Sie dann Änderungen planen aus.
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**
>
> Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern ist jeweils nur ein Knoten ausgefallen, während der Broker neu gestartet wird.

5. Wählen Sie 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.

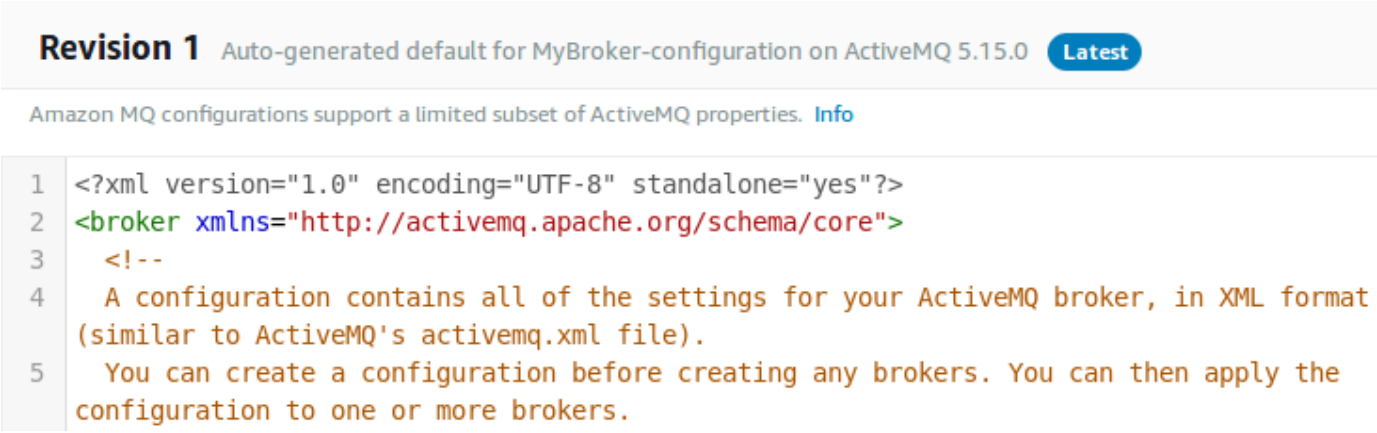
1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und klicken Sie dann auf Bearbeiten.

3. Wählen Sie auf der ***MyBroker***Seite Bearbeiten aus.
4. Wählen Sie auf der *MyBroker* Seite Bearbeiten 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 werden der Broker-Engine-Typ und die Version angezeigt, die von der Konfiguration verwendet werden (z. B. Apache ActiveMQ 5.15.8).
5. Auf der Registerkarte Configuration details (Konfigurationsdetails) werden die Konfigurations-Revisionsnummer, die Beschreibung und die Broker-Konfiguration im XML-Format angezeigt.

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



6. Klicken Sie aufBearbeiten der KonfigurationNehmen Sie Änderungen an der XML-Konfiguration vor.
7. Wählen Sie Speichern.

   DieSpeichern der Revisionwird angezeigt.

8. (Optional) Geben Sie `A description of the changes in this revision` ein.
9. Wählen Sie Speichern.

   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 XML-Konfiguration 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)

Element
discarding (Attribute)
discardingDLQBrokerPlugin (Attribute)
fileCursor
fileDurableSubscriberCursor
fileQueueCursor
filteredDestination (Attribute)
fixedCountSubscriptionRecoveryPolicy (Attribute)
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

Element
priorityNetworkDispatchPolicy
queryBasedSubscriptionRecoveryPolicy (Attribute)
queue (Attribute)
redeliveryPlugin (Attribute | untergeordnete Sammlungselemente)
redeliveryPolicy (Attribute)
redeliveryPolicyMap (untergeordnete Sammlungselemente)
retainedMessageSubscriptionRecoveryPolicy (untergeordnete Sammlungselemente)
roundRobinDispatchPolicy
sharedDeadLetterStrategy (Attribute | untergeordnete Sammlungselemente)
simpleDispatchPolicy
simpleMessageGroupMapFactory
statisticsBrokerPlugin
storeCursor
storeDurableSubscriberCursor (Attribute)
strictOrderDispatchPolicy
tempDestinationAuthorizationEntry (Attribute)
tempQueue (Attribute)
tempTopic (Attribute)
timedSubscriptionRecoveryPolicy (Attribute)
timeStampingBrokerPlugin (Attribute)

Element
topic (Attribute)
transportConnector (Attribute)
uniquePropertyMessageEvictionStrategy (Attribute)
virtualDestinationInterceptor (untergeordnete Sammlungselemente)
virtualTopic (Attribute)
vmCursor
vmDurableCursor
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 XML-Konfiguration in der Apache ActiveMQ-Dokumentation.

Element	Attribut
abortSlowAckConsumerStrategy	abortConnection
	checkPeriod
	ignoreIdleConsumers
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	maxTimeSinceLastAck

Element	Attribut
	name
abortSlowConsumerStrategy	abortConnection
	checkPeriod
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	name
authorizationEntry	admin
	queue
	read
	tempQueue
	tempTopic
	topic
	write
broker	advisorySupport
	allowTempAutoCreationOnSend
	cacheTempDestinations
	consumerSystemUsagePortion
	dedicatedTaskRunner
	deleteAllMessagesOnStartup

Element	Attribut
	keepDurableSubsActive
	enableMessageExpirationOnActiveDurableSubs
	maxPurgedDestinationsPerSweep
	maxSchedulerRepeatAllowed
	monitorConnectionSplits
	networkConnectorStartAsync
	offlineDurableSubscriberTaskSchedule
	offlineDurableSubscriberTimeout
	persistenceThreadPriority
	persistent
	populateJMSXUserID
	producerSystemUsagePortion
	rejectDurableConsumers
	rollbackOnlyOnAsyncException
	schedulePeriodForDestinationPurge
	schedulerSupport
	splitSystemUsageForProducersConsumers
	taskRunnerPriority

Element	Attribut
	timeBeforePurgeTempDestinations
	useAuthenticatedPrincipalFo rJMSXUserID
	useMirroredQueues
	useTempMirroredQueues
	useVirtualDestSubs
	useVirtualDestSubsOnCreation
	useVirtualTopics
cachedMessageGroupMapFactory	cacheSize
compositeQueue	concurrentSend
	copyMessage
	forwardOnly
	name
	sendWhenNotMatched
compositeTopic	concurrentSend
	copyMessage
	forwardOnly
	name
	sendWhenNotMatched
conditionalNetworkBridgeFilterFactory	rateDuration
	rateLimit

Element	Attribut
	`replayDelay`
	`replayWhenNoConsumers`
	`selectorAware` Unterstützt in Apache ActiveMQ 5.16.x
`constantPendingMessageLimitStrategy`	`limit`
`discarding`	`deadLetterQueue`
	`enableAudit`
	`expiration`
	`maxAuditDepth`
	`maxProducersToAudit`
	`processExpired`
	`processNonPersistent`
`discardingDLQBrokerPlugin`	`dropAll`
	`dropOnly`
	`dropTemporaryQueues`
	`dropTemporaryTopics`
	`reportInterval`
`filteredDestination`	`queue`

Element	Attribut
	selector
	topic
fixedCountSubscriptionRecoveryPolicy	maximumSize
fixedSizedSubscriptionRecoveryPolicy	maximumSize
	useSharedBuffer
forcePersistencyModeBrokerPlugin	persistenceFlag
individualDeadLetterStrategy	destinationPerDurableSubscriber
	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
	queuePrefix
	queueSuffix
	topicPrefix
	topicSuffix
	useQueueForQueueMessages
	useQueueForTopicMessages
messageGroupHashBucketFactory	bucketCount

Element	Attribut
	cacheSize
mirroredQueue	copyMessage
	postfix
	prefix
oldestMessageEvictionStrategy	evictExpiredMessagesHighWat ermark
oldestMessageWithLowestPrio rityEvictionStrategy	evictExpiredMessagesHighWat ermark
policyEntry	advisoryForConsumed
	advisoryForDelivery
	advisoryForDiscardingMessages
	advisoryForFastProducers
	advisoryForSlowConsumers
	advisoryWhenFull
	allConsumersExclusiveByDefault
	alwaysRetroactive
	blockedProducerWarningInterval
	consumersBeforeDispatchStarts
	cursorMemoryHighWaterMark
	doOptimzeMessageStorage
	durableTopicPrefetch

Element	Attribut
	enableAudit
	expireMessagesPeriod
	gcInactiveDestinations
	gcWithNetworkConsumers
	inactiveTimeoutBeforeGC
	inactiveTimoutBeforeGC
	includeBodyForAdvisory
	lazyDispatch
	maxAuditDepth
	maxBrowsePageSize
	maxDestinations
	maxExpirePageSize
	maxPageSize
	maxProducersToAudit
	maxQueueAuditDepth
	memoryLimit
	messageGroupMapFactoryType
	minimumMessageSize
	optimizedDispatch
	optimizeMessageStoreInFligh tLimit

Element	Attribut
	`persistJMSRedelivered`
	`prioritizedMessages`
	`producerFlowControl`
	`queue`
	`queueBrowserPrefetch`
	`queuePrefetch`
	`reduceMemoryFootprint`
	`sendAdvisoryIfNoConsumers`
	`sendFailIfNoSpace`
	`sendFailIfNoSpaceAfterTimeout` Unterstützt in Apache ActiveMQ 5.16.4 und höher
	`sendDuplicateFromStoreToDLQ`
	`storeUsageHighWaterMark`
	`strictOrderDispatch`
	`tempQueue`
	`tempTopic`
	`timeBeforeDispatchStarts`
	`topic`
	`topicPrefetch`

Element	Attribut
	useCache
	useConsumerPriority
usePrefetchExtension	
prefetchRatePendingMessageLimitStrategy	multiplier
queryBasedSubscriptionRecoveryPolicy	query
queue	DLQ
	physicalName
redeliveryPlugin	fallbackToDeadLetter
	sendToDlqIfMaxRetriesExceeded
redeliveryPolicy	backOffMultiplier
	collisionAvoidancePercent
	initialRedeliveryDelay
	maximumRedeliveries
	maximumRedeliveryDelay
	preDispatchCheck
	queue
	redeliveryDelay
	tempQueue
	tempTopic

Element	Attribut
	topic
	useCollisionAvoidance
	useExponentialBackOff
sharedDeadLetterStrategy	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
storeDurableSubscriberCursor	immediatePriorityDispatch
	useCache
tempDestinationAuthorizatio nEntry	admin
	queue
	read
	tempQueue
	tempTopic
	topic
	write
tempQueue	DLQ
	physicalName

Element	Attribut
tempTopic	DLQ
	physicalName
timedSubscriptionRecoveryPolicy	zeroExpirationOverride
timeStampingBrokerPlugin	recoverDuration
	futureOnly
	processNetworkMessages
	ttlCeiling
topic	DLQ
	physicalName
transportConnector	name
	updateClusterClients
	rebalanceClusterClients
	updateClusterClientsOnRemove
uniquePropertyMessageEvictionStrategy	evictExpiredMessagesHighWatermark
	propertyName
virtualTopic	concurrentSend
	local
	dropOnResourceLimit
	name
	postfix

Element	Attribut
	`prefix`
	`selectorAware`
	`setOriginalDestination`
	`transactedSend`

Attribute des übergeordneten Amazon MQ Elemente

Im Folgenden finden Sie eine detaillierte Erklärung der Attribute des übergeordneten Elements. Weitere Informationen finden Sie unter XML-Konfiguration 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 XML-Konfiguration 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`

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
	networkConnectors	networkConnector
	persistenceAdapter	kahaDB
	plugins	authorizationPlugin
		discardingDLQBrokerPlugin
		forcePersistencyModeBrokerPlugin
		redeliveryPlugin
		statisticsBrokerPlugin
		timeStampingBrokerPlugin
	systemUsage	systemUsage
	transportConnector	name
		updateClusterClients
		rebalanceClusterClients
		updateClusterClientsOnRemove
compositeQueue	forwardTo	queue
		tempQueue
		tempTopic

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
		topic
		filteredDestination
compositeTopic	forwardTo	queue
		tempQueue
		tempTopic
		topic
		filteredDestination
policyEntry	deadLetterStrategy	discarding
		individualDeadLetterStrategy
		sharedDeadLetterStrategy
	destination	queue
		tempQueue
		tempTopic
		topic
	dispatchPolicy	priorityDispatchPolicy
		priorityNetworkDispatchPolicy
		roundRobinDispatchPolicy

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
		simpleDispatchPolicy
		strictOrderDispatchPolicy
		clientIdFilterDispatchPolicy
	messageEvictionStrategy	oldestMessageEvictionStrategy
		oldestMessageWithLowestPriorityEvictionStrategy
		uniquePropertyMessageEvictionStrategy
	messageGroupMapFactory	cachedMessageGroupMapFactory
		messageGroupHashBucketFactory
		simpleMessageGroupMapFactory
	pendingDurableSubscriberPolicy	fileDurableSubscriberCursor
		storeDurableSubscriberCursor
		vmDurableCursor
	pendingMessageLimitStrategy	constantPendingMessageLimitStrategy

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
		prefetchRatePendingMessageLimitStrategy
	pendingQueuePolicy	fileQueueCursor
		storeCursor
		vmQueueCursor
	pendingSubscriberPolicy	fileCursor
		vmCursor
	slowConsumerStrategy	abortSlowAckConsumerStrategy
		abortSlowConsumerStrategy
	subscriptionRecoveryPolicy	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
		retainedMessageSubscriptionRecoveryPolicy
timedSubscriptionRecoveryPolicy		
policyMap	defaultEntry	policyEntry
	policyEntries	policyEntry
redeliveryPlugin	redeliveryPolicyMap	redeliveryPolicyMap
redeliveryPolicyMap	defaultEntry	redeliveryPolicy
	redeliveryPolicyEntries	redeliveryPolicy
retainedMessageSubscriptionRecoveryPolicy	wrapped	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
		`timedSubscriptionRecoveryPolicy`
`sharedDeadLetterStrategy`	`deadLetterQueue`	`queue`
		`tempQueue`
		`tempTopic`
		`topic`
`virtualDestinationInterceptor`	`virtualDestinations`	`compositeQueue`
		`compositeTopic`
		`virtualTopic`

Amazon MQ-Attribute

Im Folgenden finden Sie eine detaillierte Erklärung der Attribute untergeordneter Sammlungselemente. Weitere Informationen finden Sie unter XML-Konfiguration 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>
```

> **Note**
>
> Die `activemq-webconsole` Gruppe in ActiveMQ auf Amazon MQ hat Administratorberechtigungen für alle Warteschlangen und Themen. Alle Benutzer in dieser Gruppe haben Administratorzugriff.

networkConnector

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

Themen

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 Lastausgleich von Nachrichten über Verbraucher hinweg haben und in bestimmten Szenarien (z. B. bei JMS-Nachrichtenselektoren oder bei dauerhaften Themen) zu einem falschen Verhalten führen.

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 Netzwerkkonnektors ist`connector_1_to_2`, der gemeinsame Benutzername der Broker lautet`myCommonUser`, die Verbindung ist`duplex`, und dem OpenWire Endpunkt-URI wird ein Präfix vorangestellt`static:`, 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-west-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>
```

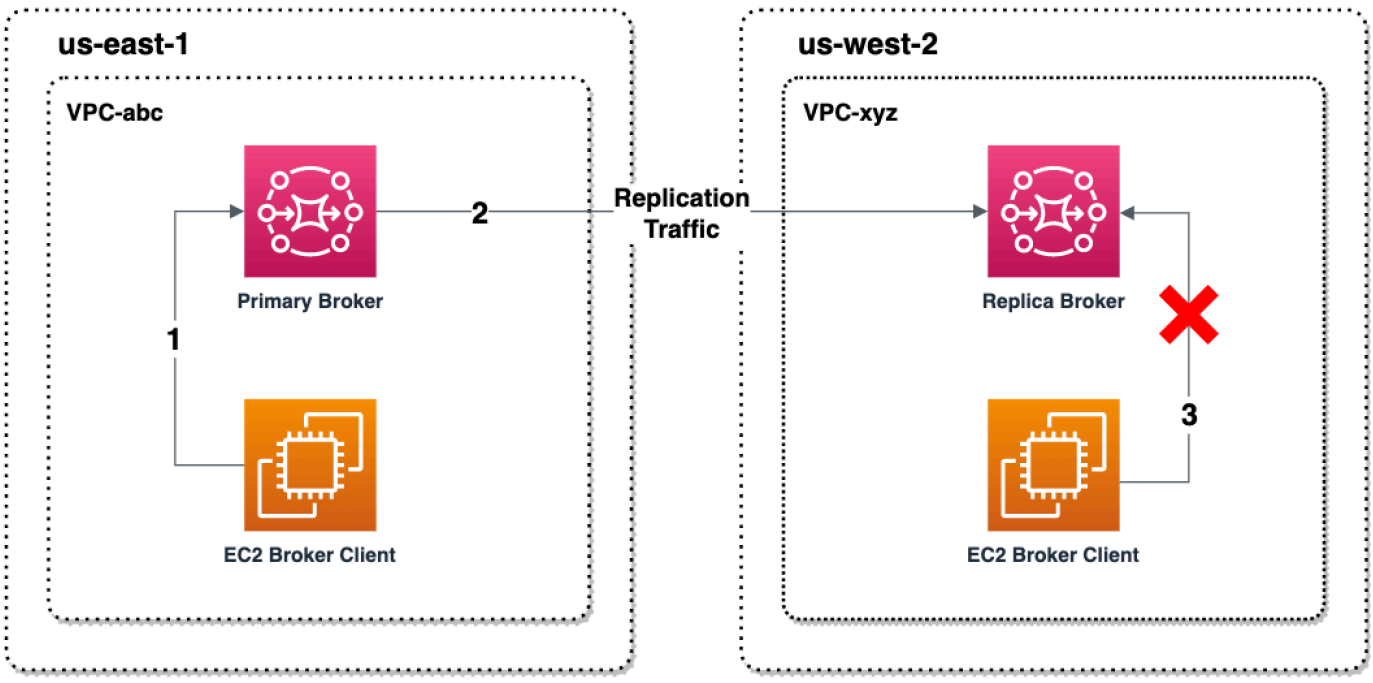
Regionsübergreifende Datenreplikation für Amazon MQ für ActiveMQ

Amazon MQ for ActiveMQ bietet eine CRDR-Funktion (Cross-Region Data Replication), die eine asynchrone Nachrichtenreplikation vom primären Broker in einer primären AWS Region zum Replikatbroker in einer Replikatregion ermöglicht. Durch eine Failover-Anfrage an die Amazon-MQ-API wird der aktuelle Replikat-Broker in die Rolle des Primär-Brokers hochgestuft und der aktuelle Primär-Broker wird in die Rolle des Replikat-Brokers heruntergestuft.

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.
> Nicht verfügbar für gespiegelte Warteschlangen.

Einen Amazon MQ-Broker für die regionsübergreifende Datenreplikation erstellen

Mit der regionsübergreifenden Datenreplikation (CRDR) können Sie bei Bedarf zwischen den Message Brokern von Amazon MQ für ActiveMQ 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 Replikat-Broker mithilfe der `Promote`-API-Operation von Amazon MQ in die Rolle des Primär-Brokers hochstufen. 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 Amazon EFS. 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 keine persönlich identifizierbare Informationen (PII) oder andere vertrauliche oder sensible Informationen in Brokernamen 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 keine persönlich identifizierbare Informationen (PII) oder andere vertrauliche oder sensible Informationen in Broker-Benutzernamen 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 für Ihre Broker einsehen. Wählen Sie das Zahnradsymbol in der oberen rechten Ecke der Tabelle Broker aus, um die Spalten „CRDR-Rolle“ und „RPO-Status“ zu deaktivieren.. Deaktivieren Sie anschließend auf der Seite Einstellungen die Option „CRDR-Rolle“ 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 zum Auswählen des Primär-Brokers einen vorhandenen Broker aus, den Sie als Primär-CRDR-Broker verwenden möchten. Wählen Sie anschließend Weiter aus.
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 keine persönlich identifizierbare Informationen (PII) oder andere vertrauliche oder sensible Informationen in Broker-Benutzernamen 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 keine persönlich identifizierbare Informationen (PII) oder andere vertrauliche oder sensible Informationen in Broker-Benutzernamen 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 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 Seite Broker 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 „Bestätigen“ 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

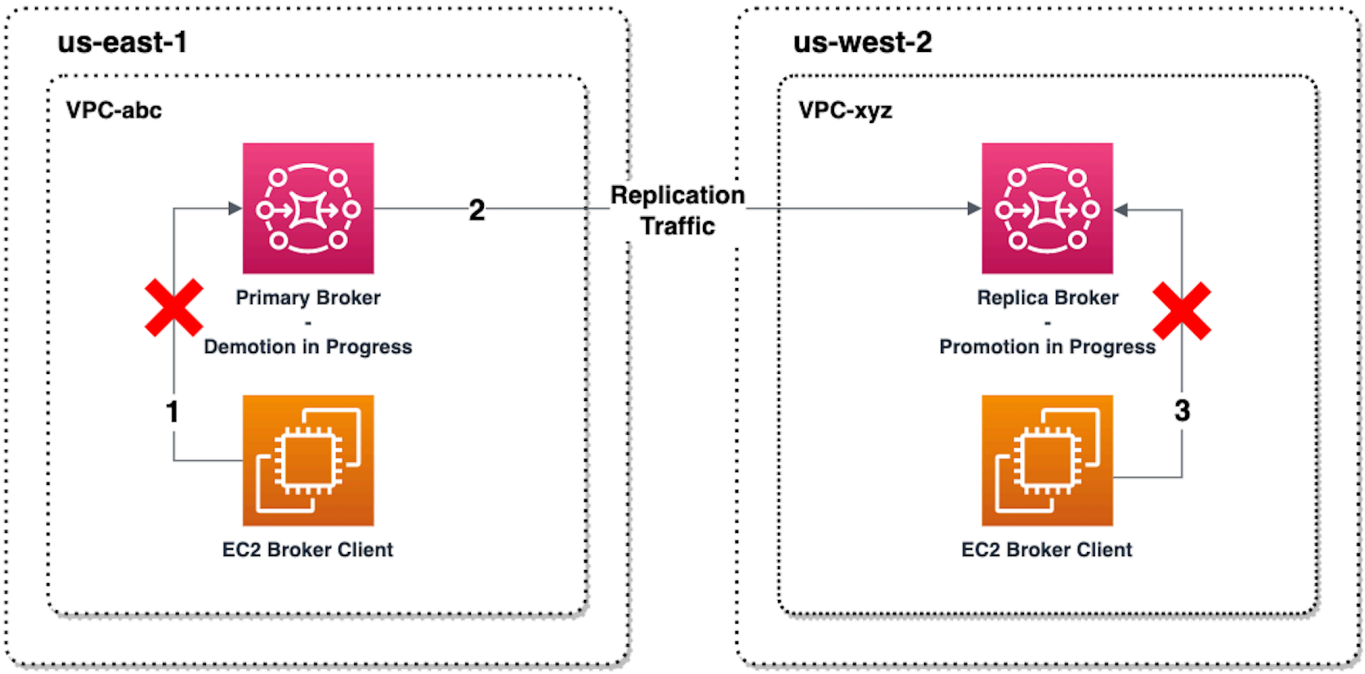
Sie können ein Switchover oder Failover initiieren, wenn Sie den Replikat-Broker in die Rolle des Primär-Brokers hochstufen möchten. Wenn Sie den Replikat-Broker hochstufen, wird der Primär-Broker in die Rolle des Replikat-Brokers heruntergestuft.

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.

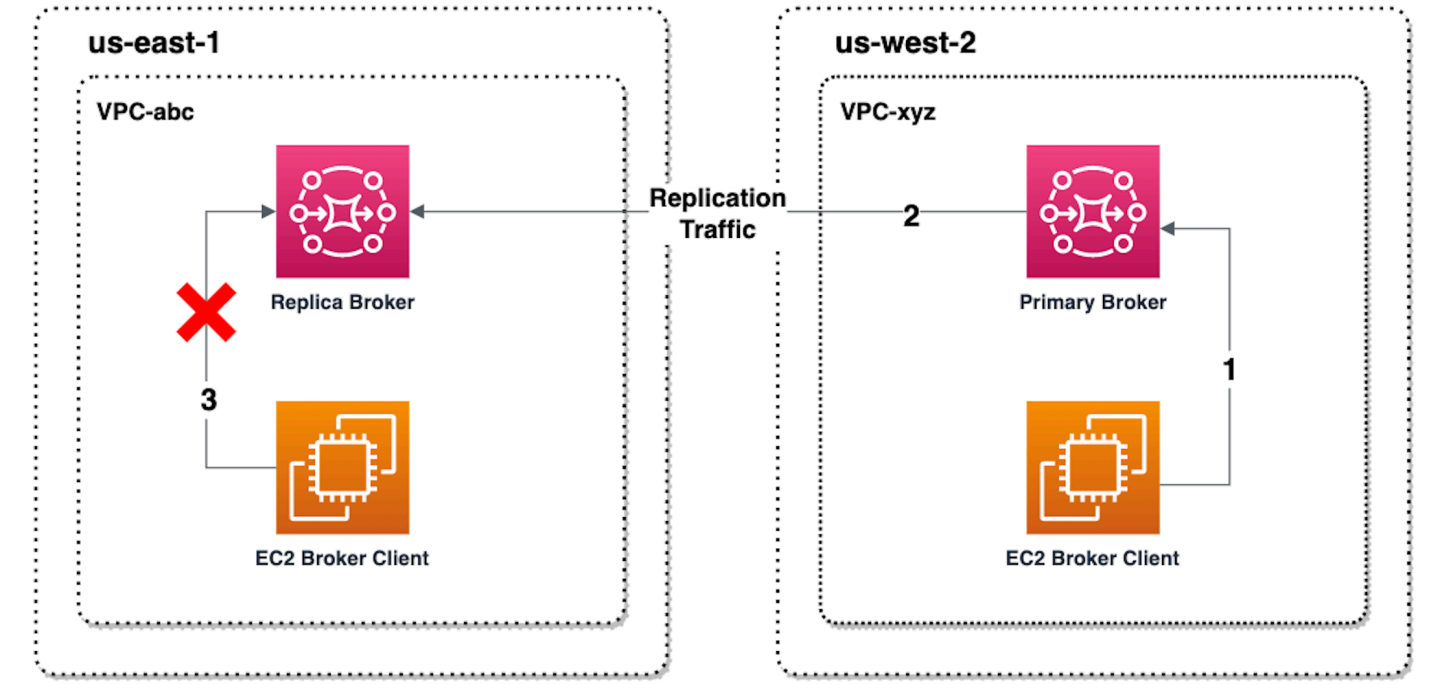
Bei einem Failover hat die Verfügbarkeit Vorrang vor der Konsistenz. Es kann nicht garantiert werden, dass Makler nach Abschluss dieses Vorgangs identische Status haben. Bei einem Failover

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. In diesem Prozess kann der Client in der VPC des primären Brokers keine weiteren Statusänderungen vornehmen, während der Vorgang ausgeführt wird, und der primäre Broker wird zu einem Replikat herabgestuft. Wenn die Replikationswarteschlange geleert ist und die beiden Broker einen identischen Status erreichen, kann der Client in der VPC des Replikat-Brokers keine Verbindung zum Replikat-Broker herstellen, bis der Failover-Vorgang abgeschlossen ist und der Replikat-Broker zum Primär-Broker hochgestuft 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.

Metriken zur regionsübergreifenden Datenreplikation 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 bei Amazon CloudWatch finden Sie unterZugreifen auf CloudWatch Metriken für Amazon MQ.

CRDR-Zeitstempel

Die folgenden Zeitstempel beschreiben, wie die in Amazon gefundenen Kennzahlen berechnet CloudWatch 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.

Schätzen Sie die Switchover-/Failover-Leistung anhand von CRDR-Metriken ab CloudWatch

Amazon MQ aktiviert standardmäßig Metriken für Ihren Broker. Sie können Ihre Broker-Metriken einsehen, indem Sie auf die CloudWatch Amazon-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-Metrik CloudWatch	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.
`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
- Schritt 3: (Optional) Connect zu einer AWS Lambda Funktion herstellen
- Einen ActiveMQ-Broker-Benutzer erstellen
- Einen ActiveMQ-Broker-Benutzer bearbeiten
- Löschen Sie einen ActiveMQ-Broker-Benutzer
- Funktionierende Beispiele 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. 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

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 in derselben VPC oder im VPCs Peered-Modus befinden. Weitere Informationen finden Sie VPCs unter Was ist Amazon VPC? im Amazon VPC-Benutzerhandbuch und Was ist VPC Peering? im Amazon VPC Peering Guide.

> **Important**
>
> Wenn Sie keinen Standard-VPC, kein Subnetz oder keine Sicherheitsgruppe haben, müssen Sie diese zuerst erstellen. Weitere Informationen finden Sie unter den folgenden Themen im Amazon VPC Benutzerhandbuch:
>
> - Erstellen einer Standard-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 von LDAP-Authentifizierung in ein Netzwerk von Brokern sicher, dass der Benutzer sowohl als ActiveMQ -Broker als auch als LDAP-Benutzer vorhanden ist.

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 Security Groups (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 Type (Typ) Custom TCP (Benutzerdefiniertes TCP).
   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-Eintrag 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 Ihre Broker privat sind und zur gleichen VPC gehören, lassen Sie bei Source (Quelle) Custom (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 Configuration details (Konfigurationsdetails) werden die Konfigurations-Revisionsnummer, die Beschreibung und die Broker-Konfiguration im XML-Format 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`. Beispiel:

```
<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` Beispiel:

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

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

f. Wählen Sie Speichern.

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. Beispiel:

```
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. Beispiel:

```
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 zu erstellen, 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

Aktivieren der VPC-Attribute

Um sicherzustellen, dass Ihr Broker innerhalb Ihrer VPC zugänglich ist, müssen Sie die`enableDnsHostnames`und`enableDnsSupport`VPC Attribute Weitere Informationen finden Sie unter DNS-Support in Ihrer VPC im Amazon-VPC-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. Notieren Sie sich auf der ***`MyBroker`***Seite im Abschnitt Verbindungen die Adressen und Ports der Webkonsolen-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 Security Groups (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. In dem Dialogfeld Edit inbound rules (Bearbeiten von Regeln für eingehenden Datenverkehr), fügen Sie eine Regel für jede URL oder jeden Endpunkt hinzu, auf den Sie öffentlich zugreifen möchten (im folgenden Beispiel wird gezeigt, wie Sie dies für eine Broker-Webkonsole tun).

   a. Klicken Sie auf Add Rule (Regel hinzufügen).
   b. Wählen Sie für Type (Typ) Custom TCP (Benutzerdefiniertes TCP).
   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 eine JMS-Pool-Connection Factory für den Nachrichtenproduzenten mit dem Endpunkt Ihres Brokers und rufen Sie dann die`createConnection`Methode gegen die Fabrik.

> **Note**
>
> Für einen active/standby Broker bietet Amazon MQ zwei ActiveMQ-Web-Konsolen URLs, aber es ist jeweils nur eine URL aktiv. 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).
> Bei Protokollendpunkten auf Wire-Level-Ebene sollten Sie Ihrer Anwendung ermöglichen, mithilfe des Failover-Transports eine Verbindung zu einem der beiden Endpunkte herzustellen.

```
// 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 Nachrichtengenerator zu erstellen und eine Nachricht zu empfangen.

1. Erstellen Sie eine JMS-Connection Factory für den Nachrichtenproduzenten mit dem Endpunkt Ihres Brokers und rufen Sie dann die`createConnection`Methode gegen die Fabrik.

```
// 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 Amazon SQS) 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**
>
> Amazon MQ unterstützt kein Serverzertifikat, das von einer privaten Zertifizierungsstelle ausgestellt wurde.

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

- AMQP
- MQTT
- MQTT über WebSocket
- OpenWire
- STOMP
- STOMP rüber 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.

> **Important**
>
> Bei der Verwendung von LDAP wird bei der Authentifizierung nicht zwischen Groß- und Kleinschreibung unterschieden, bei der Autorisierung wird jedoch zwischen Groß- und Kleinschreibung für Ihren Benutzernamen unterschieden.

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.

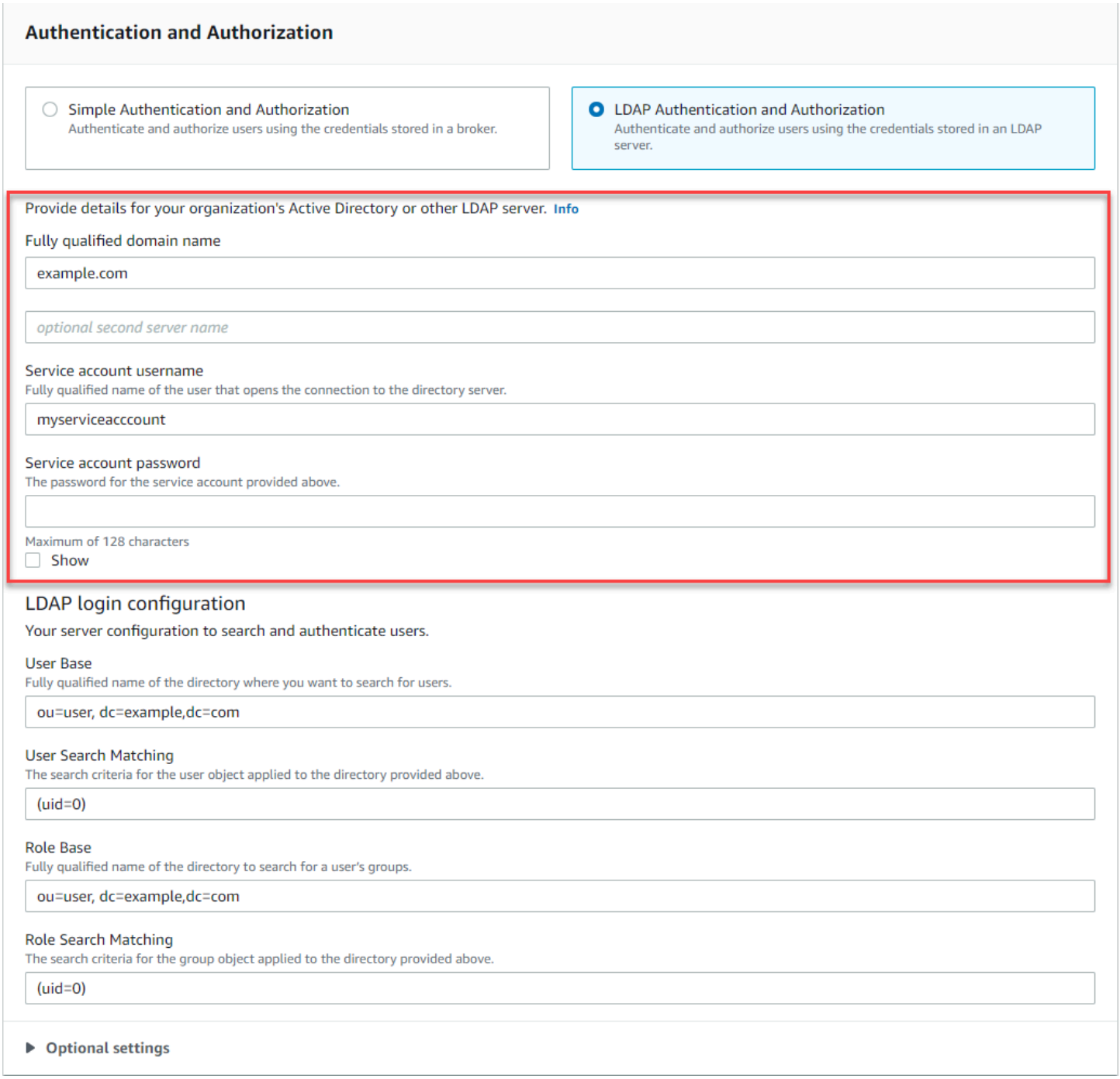
Geben Sie die folgenden Informationen zum Servicekonto ein:

- 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.

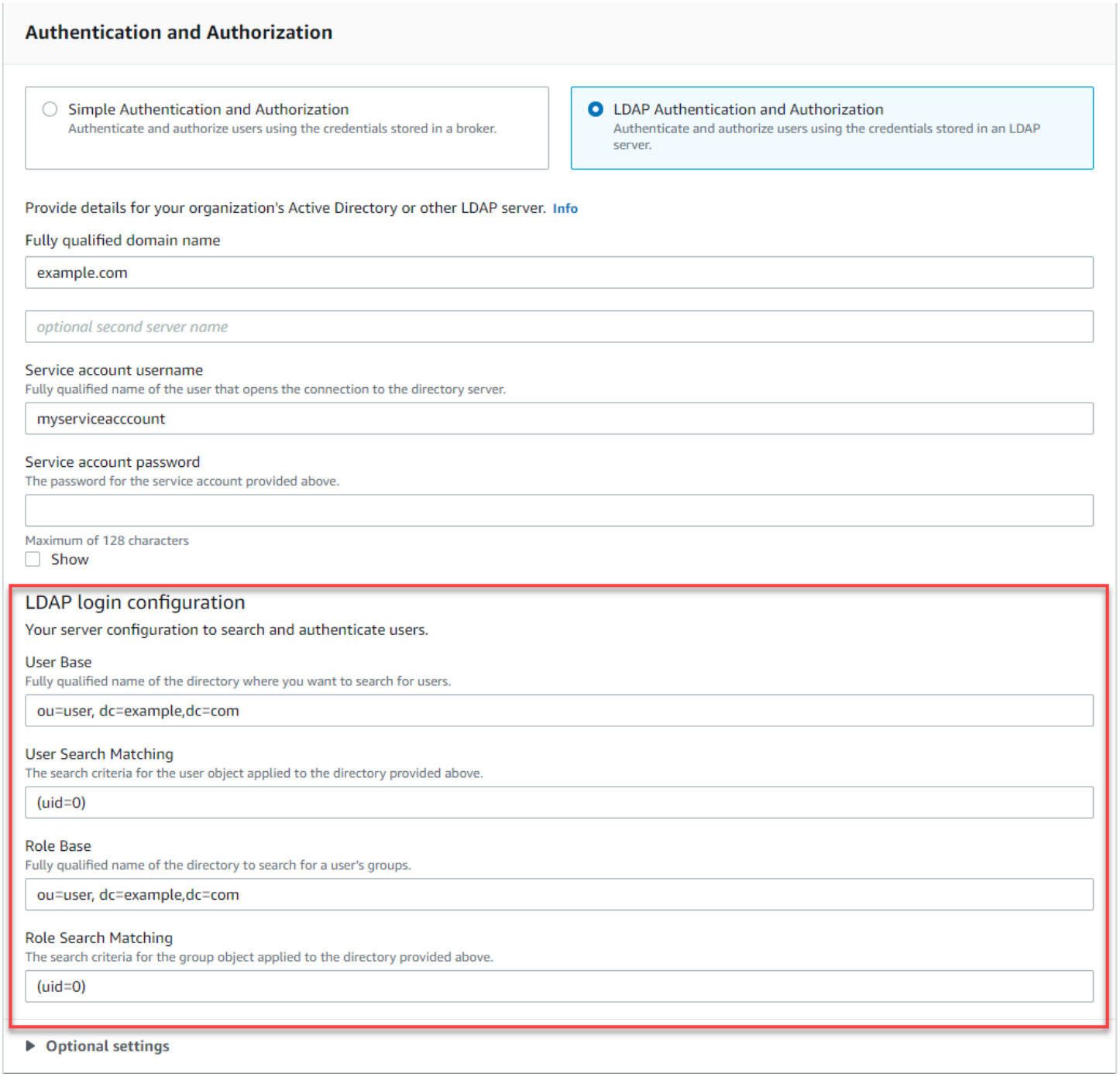


In der Konfiguration der LDAP-Anmeldung geben Sie die folgenden erforderlichen Informationen ein:

- 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 erhalten 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:

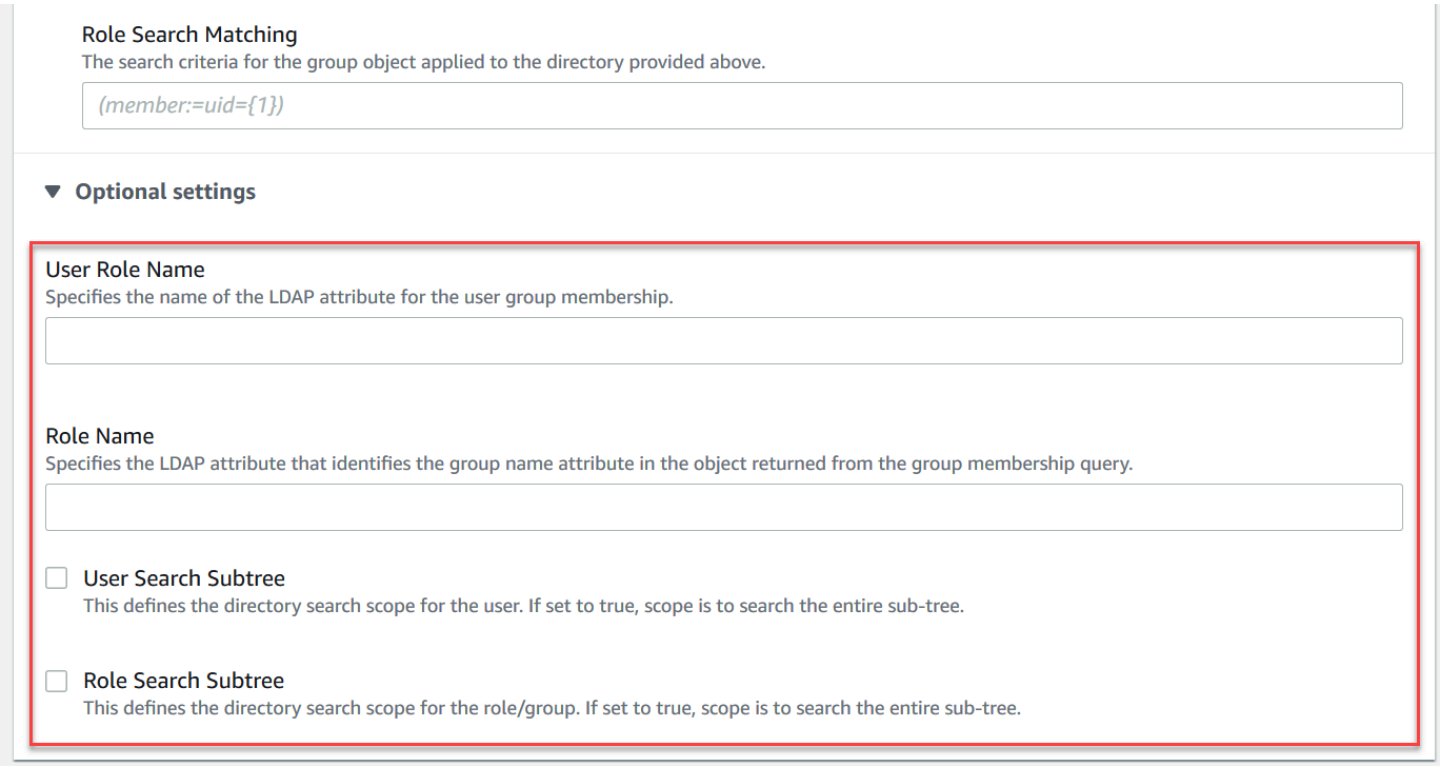
- Benutzerrollen-Name Der Name des LDAP-Attributs im Verzeichniseintrag des Benutzers für die Gruppenmitgliedschaft des Benutzers. In einigen Fällen können Benutzerrollen durch den Wert eines Attributs im Verzeichniseintrag des Benutzers identifiziert werden. Mit der `userRoleName`-Option können Sie 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.

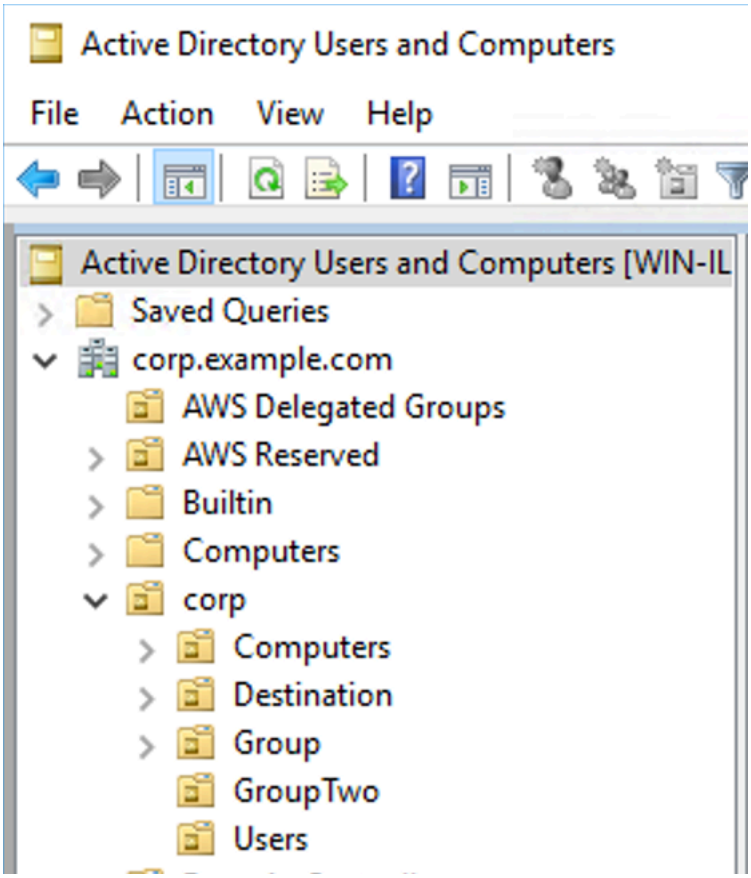
Authentifizierung

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 die Domänenkomponenten AWS Managed Microsoft AD, und `corpexample`, verwenden und `com` innerhalb dieser Organisationseinheiten `corp` vorhanden sind`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-Authentifizierung und Autorisierung wählen, wenn Sie eine Konfiguration über die erstellen AWS-Managementkonsole, oder die Option zum Erstellen einer Konfiguration über die festlegen oder die `authenticationStrategy`Eigenschaft auf setzen AWS-Managementkonsole, `LDAP` wenn Sie eine neue Konfiguration mit der Amazon MQ 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 oben genannten Attribute durch die Werte AWS-Managementkonsole, die Sie über die oder in der `ldapServerMetadata`Eigenschaft Ihrer API-Anfrage angeben.

Das folgende Beispiel illustriert die Verwendung von Verschiebungen.

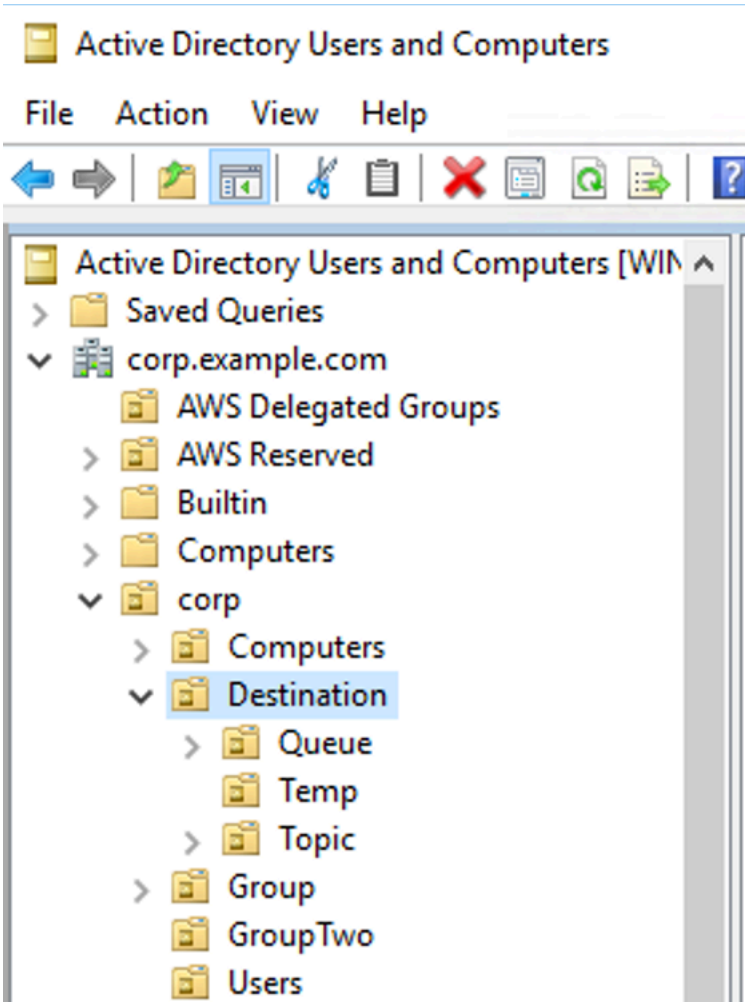
```
<authorizationPlugin>
```

```
    <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. Für das obige Beispiel mit AWS Managed Microsoft AD der Verwendung derselben Domänenkomponenten von`corp`, `example` und würden Sie also eine Organisationseinheit angeben`com`, die so benannt ist, dass `destination` sie all Ihre Zieltypen enthält. 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:

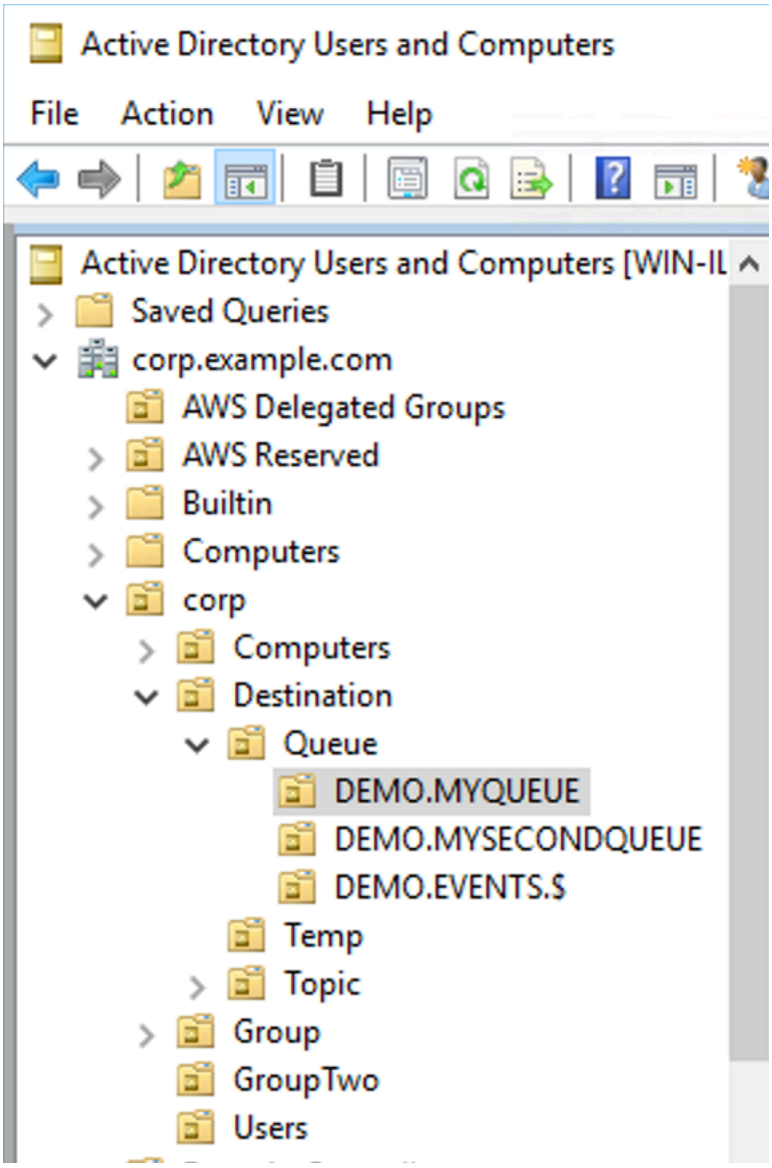
```
OU=DEMO.EVENTS.$,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```

> **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:

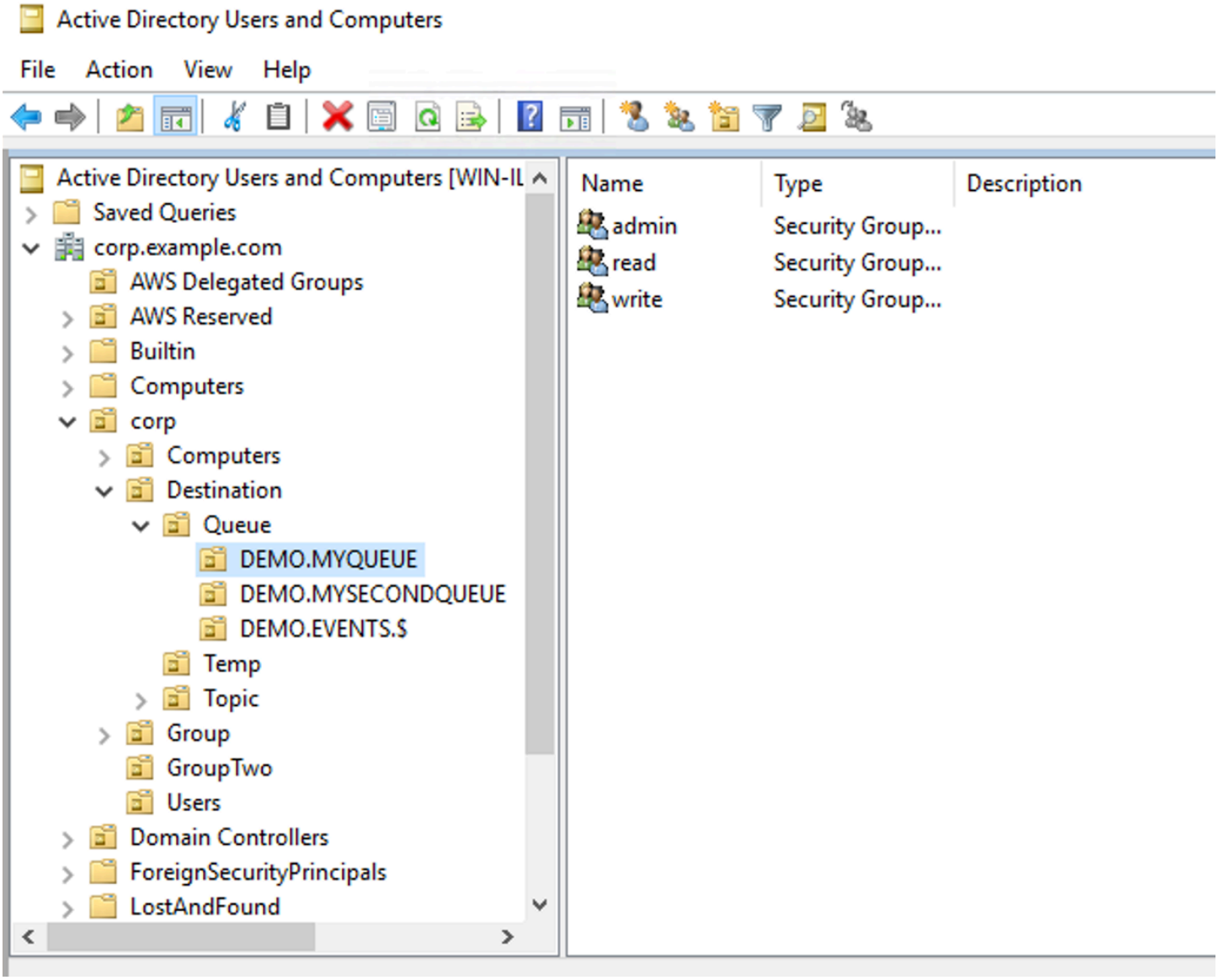
```
OU=DEMO.MYQUEUE,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



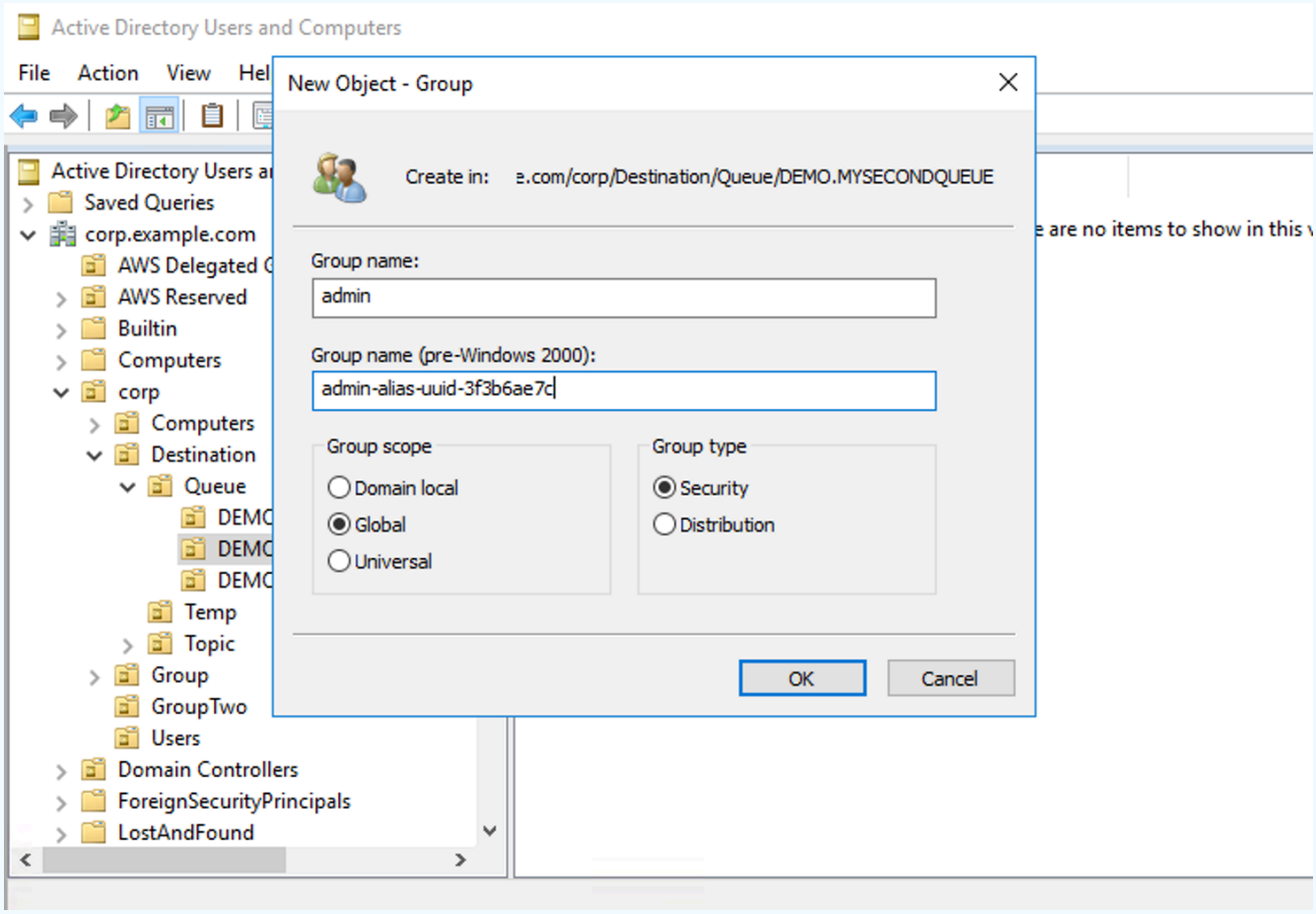
Sicherheitsgruppen

Innerhalb jeder Organisationseinheit, die ein Ziel oder einen Platzhalter darstellt, müssen Sie drei Sicherheitsgruppen erstellen. Wie bei allen Berechtigungen in ActiveMQ handelt es sich auch hier read/write/admin um Berechtigungen. 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.

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 Ereignisquellen-Zuweisung, die Sie erstellen, liest Nachrichten von Ihrem Broker in Batches und wandelt sie in eine Lambda -Payload in Form eines JSON-Objekts um.

So verbinden Sie Ihren Broker mit einer Lambda Funktion

1. Fügen Sie die folgenden IAM-Rollenberechtigungen zu der Ausführungsrolle Ihrer Lambda-Funktion hinzu.

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

   > **Note**
   >
   > Ohne die erforderlichen IAM-Berechtigungen ist Ihre Funktion nicht in der Lage, Datensätze aus Amazon MQ Ressourcen erfolgreich zu 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 unterInternet- und Servicezugriff für VPC-verbundene FunktionenimAWS Lambda Entwicklerhandbuch.
   - Erstellen Sie mithilfe eines VPC-Endpunkts eine Verbindung zwischen Ihrer Amazon Virtual Private Cloud (Amazon VPC) und Lambda. Ihre Amazon VPC muss auch eine Verbindung zu AWS -Security-Token-Service (AWS STS) und Secrets Manager Manager-Endpunkten herstellen. Weitere Informationen finden Sie unterKonfigurieren von Schnittstellen-VPC-Endpunkten für LambdaimAWS Lambda Entwicklerhandbuch.

3. Konfigurieren Sie Ihren Broker als EreignisquelleVerwendung für eine Lambda -Funktion unter Verwendung der AWS-Managementkonsole. 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.

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.

> **Note**
>
> Die `activemq-webconsole` Gruppe in ActiveMQ auf Amazon MQ hat Administratorberechtigungen für alle Warteschlangen und Themen. Alle Benutzer in dieser Gruppe haben Administratorzugriff.

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

Erstellen Sie einen neuen ActiveMQ-Broker-Benutzer

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***Seite werden im Bereich Benutzer alle Benutzer dieses Brokers aufgelistet.

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

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.

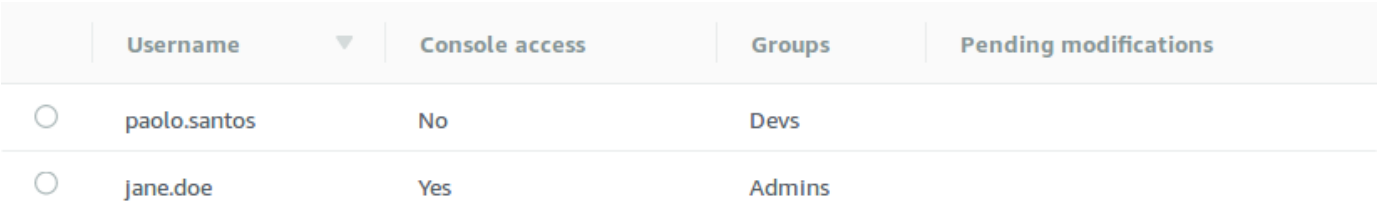
> **Important**
>
> Das Vornehmen von Änderungen an einem Benutzer wendet nicht sofort die Änderungen auf den Benutzer an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten.

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***Seite werden im Bereich Benutzer alle Benutzer dieses Brokers aufgelistet.

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

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.

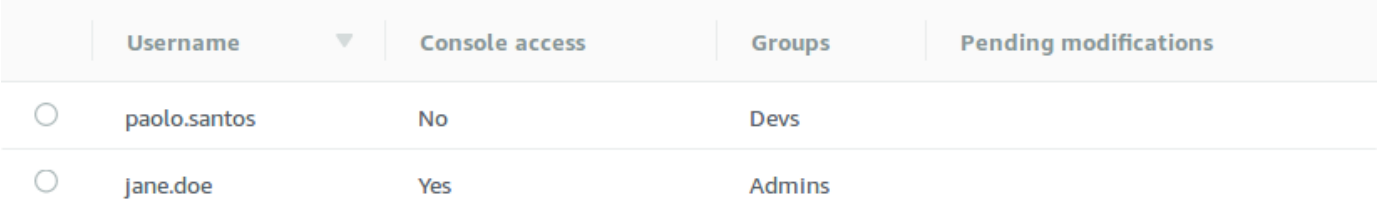
> **Important**
>
> Das Vornehmen von Änderungen an einem Benutzer wendet nicht sofort die Änderungen auf den Benutzer an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten.

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***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 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.

Funktionierende Beispiele 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.
- Mit dem MQTT Java-Beispielcode wird eine Verbindung zu einem Broker hergestellt, eine Warteschlange erstellt und eine Nachricht gesendet und empfangen.
- Der Java-Beispielcode für STOMP+WSS stellt eine Verbindung zu einem Broker her, erstellt eine Warteschlange und veröffentlicht und empfängt eine Nachricht.

Voraussetzungen

Aktivieren der VPC-Attribute

Um sicherzustellen, dass Ihr Broker innerhalb Ihrer VPC zugänglich ist, müssen Sie die`enableDnsHostnames`und`enableDnsSupport`VPC Attribute Weitere Informationen finden Sie unter DNS-Support in Ihrer VPC im Amazon-VPC-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. Notieren Sie sich auf der ***`MyBroker`***Seite im Abschnitt Verbindungen die Adressen und Ports der Webkonsolen-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 Security Groups (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. In dem Dialogfeld Edit inbound rules (Bearbeiten von Regeln für eingehenden Datenverkehr), fügen Sie eine Regel für jede URL oder jeden Endpunkt hinzu, auf den Sie öffentlich zugreifen möchten (im folgenden Beispiel wird gezeigt, wie Sie dies für eine Broker-Webkonsole tun).

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

b. Wählen Sie für Type (Typ) Custom TCP (Benutzerdefiniertes TCP).

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 STOMP-Unterstützung in der Spring Framework-Dokumentation.

MQExampleAmazon.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. 18

auf 5. 19 wird als geringfügiges Versionsupgrade betrachtet. Wenn diese Option aktiviert `autoMinorVersionUpgrade` ist, aktualisiert Amazon MQ Ihren Broker auf die neueste verfügbare Patch-Version.

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 5.19 (empfohlen)	
ActiveMQ 5.18	
ActiveMQ 5.17	16. Juni 2025
ActiveMQ 5.16	15. November 2024
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, *`engine-type`* ersetzen Sie durch `ACTIVEMQ` und *`instance-type`* durch. `mq.m5.large`

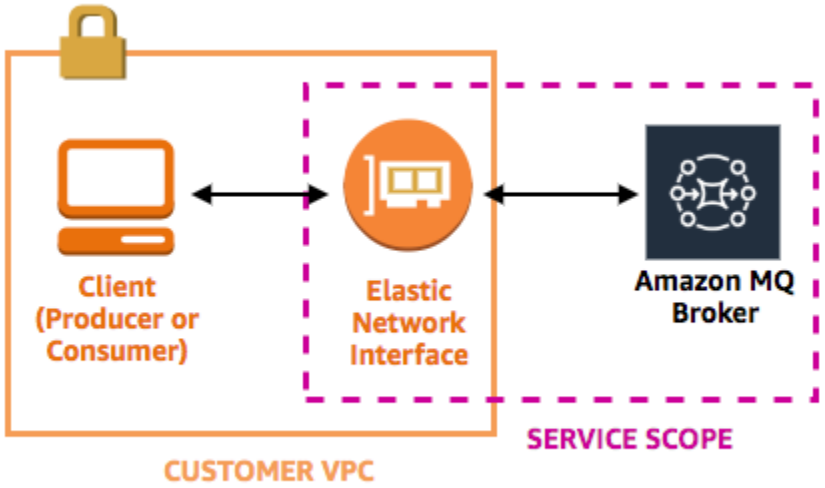
Best Practices für Amazon MQ für ActiveMQ

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 eine Elastic Network-Schnittstelle in der Virtual Private Cloud (VPC) unter Ihrem Konto bereit. Deshalb sind verschiedene EC2-Berechtigungen dafür erforderlich. Die Netzwerkschnittstelle gestattet Ihrem Client (Erzeuger oder Verbraucher), mit dem Amazon MQ-Broker zu kommunizieren. Die Netzwerkschnittstelle wird als im Service-Umfang von Amazon MQ begriffen betrachtet, obwohl sie Teil der VPC Ihres Kontos ist.

> **Warning**
>
> Sie dürfen diese Netzwerkschnittstelle nicht ändern oder löschen. Das Ändern oder Löschen der Netzwerkschnittstelle kann zu einem permanenten Verlust der Verbindung zwischen Ihrer 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. Beispiel:

```
// 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. Beispiel:

> **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. Beispiel:

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

> **Important**
>
> Bei Brokern mit mehreren Verfügbarkeitszonen kann es während Wartungsfenstern und Broker-Neustarts zu Failovers kommen. Verwenden Sie den Failover-Transport, um die Verfügbarkeit Ihres Brokers sicherzustellen.

Vermeiden Sie die Nachrichtenauswahl

Sie können mit JMS-Auswahlen Filter an Themenabonnements anfügen (um Nachrichten basierend auf ihrem Inhalt an Konsumenten weiterzuleiten). Doch die Verwendung von JMS-Auswahlen füllt den Filterpuffer des Amazon MQ-Brokers und verhindert somit das Filtern von Nachrichten.

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 Clients IPs im CIDR-Bereich **`10.0.0.0/16`**

Wenn Sie Amazon VPC-Peering zwischen der lokalen Infrastruktur und Ihrem Amazon MQ-Broker einrichten, dürfen Sie keine Client-Verbindungen im CIDR-Bereich konfigurieren. IPs `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 sowie unter Messen des Durchsatzes für Amazon MQ mithilfe der JMS-Benchmark.

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, profitieren von der höheren Speicherkapazität, der schnelleren CPU und dem schnelleren Netzwerk, 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 EFS, um die Vorteile der hohen Haltbarkeit und Replikation über mehrere Availability Zones hinweg zu nutzen. Verwenden Sie Amazon EBS, um die Vorteile der niedrigen Latenz und des hohen Durchsatzes zu nutzen. 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 mithilfe der Amazon MQ REST API oder der AWS-Managementkonsole erstellen, 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 REST API-Referenz.

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-Instance-Klasse (`m7g`) und der Größe (`large`,`medium`) wird als Broker-Instance-Typ bezeichnet (z. B.). `mq.m7g.large`

- 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 Amazon EBS-Speicher-Volume.
- 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 Bereitstellen eines RabbitMQ-Brokers.

Listener-Ports

Von Amazon MQ verwaltete RabbitMQ-Broker unterstützen die folgenden Listener-Ports für Konnektivität auf Anwendungsebene über. `amqps` Sie können diese Ports auch für Client-Verbindungen über die RabbitMQ-Webkonsole und die Management-API verwenden. Alle Verbindungen verwenden aus Sicherheitsgründen die TLS-Verschlüsselung.

- Listener-Port `5671` — Wird für sichere AMQP-Verbindungen verwendet, die über die sichere AMQP-URL hergestellt werden. Dieser Port unterstützt sowohl die AMQP 0-9-1- als auch die

AMQP 1.0-Protokolle in RabbitMQ 4. Zum Beispiel, für einen Broker mit Broker-IDb-c8352341-ec91-4a78-ad9c-a43f23d325bb, der in derus-west-2Region bereitgestellt ist, ist dies die komplette BrokeramqpsURL: b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com:5671.

- Listener-Ports 443 und 15671 — Sie können beide Listener-Ports synonym verwenden, um über die RabbitMQ-Webkonsole oder die Management-API auf einen Broker zuzugreifen. Port 443 bietet standardmäßigen HTTPS-Zugriff, während Port 15671 der traditionelle RabbitMQ-Verwaltungsport mit TLS-Verschlüsselung ist.

Attribute

Ein RabbitMQ-Broker verfügt über mehrere Attribute:

- Ein Name. Beispiel, MyBroker.
- Eine ID. Beispiel, b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9.
- Einen Amazon-Ressourcennamen (ARN). Beispiel, arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9.
- Eine URL der RabbitMQ-Webkonsole. 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 AMQP-Endpunkt. 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 Abschnitt in der Amazon MQ REST API Reference:

- REST-Operations-ID: Broker
- REST-Operations-ID: Broker
- REST-Operations-ID: Broker Reboot

Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen

RabbitMQ organisiert Versionsnummern gemäß der semantischen Versioning-Spezifikation als X.Y.Z. XBezeichnet 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 Version RabbitMQ 4.2 zu verwenden. Anweisungen zum Upgrade Ihrer Broker-Engine-Version finden Sie unter Upgrade einer Amazon MQ-Broker-Engine-Version.

Wenn Sie einen neuen Amazon MQ for RabbitMQ Broker erstellen, müssen Sie nur die Haupt- und Nebenversionsnummern angeben. Zum Beispiel RabbitMQ 4.2. Wenn Sie bei der Erstellung eines Brokers die Engine-Version nicht angeben, verwendet Amazon MQ automatisch standardmäßig die neueste Engine-Version.

> **Important**
>
> Amazon MQ unterstützt keine Streams. Das Erstellen eines Streams führt zu Datenverlust. Amazon MQ unterstützt die Verwendung der strukturierten Protokollierung in JSON nicht.

Amazon MQ unterstützt zwei Hauptversionen von RabbitMQ:

- RabbitMQ 4

  Amazon MQ unterstützt RabbitMQ 4.2 in der RabbitMQ 4-Release-Serie nur auf dem Instance-Typ mq.m7g für alle unterstützten Instance-Größen.

- RabbitMQ 3

  Amazon MQ unterstützt RabbitMQ 3.13 in der RabbitMQ 3-Release-Serie auf den Instance-Typen mq.t3, mq.m5 und mq.m7g in allen unterstützten Instance-Größen.

Unterstützte Engine-Versionen auflisten

Mit dem Befehl können Sie alle unterstützten Neben- und Hauptversionen der Engine auflisten. `describe-broker-instance-options` AWS CLI

```
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 RabbitMQ und `mq.m7g.large` Instanztyp zu filtern, ersetzen Sie sie durch und *`engine-type`* durch`RABBITMQ`. *`instance-type`* `mq.m7g.large`

RabbitMQ 4

Amazon MQ unterstützt RabbitMQ 4.2 in der RabbitMQ 4-Release-Serie nur auf dem Instance-Typ mq.m7g für alle unterstützten Instance-Größen.

> **Important**
>
> Sie können nur auf RabbitMQ 4.2 neue Broker erstellen. Vorhandene Upgrades von RabbitMQ 3.13 werden derzeit nicht unterstützt.

> **Important**
>
> Der Standard-Warteschlangentyp auf Amazon MQ für RabbitMQ 4.2-Broker ist „Quorum“. Wenn bei der Erstellung der Warteschlange kein Argument für den Warteschlangentyp angegeben wird, wird eine Quorum-Warteschlange erstellt.
> Aus Gründen der Haltbarkeit empfehlen wir dringend, Quorum-Warteschlangen auf RabbitMQ 4 zu verwenden, da klassische Warteschlangen nicht in allen Fällen garantiert dauerhaft sind.

Die folgenden Änderungen wurden in RabbitMQ 4 auf Amazon MQ eingeführt:

- AMQP 1.0 als Kernprotokoll: Weitere Informationen finden Sie unter Protokolle.
- Lokale Schaufeln: Shovels unterstützen jetzt zusätzlich zu AMQP 0-9-1 und AMQP 1.0 ein neues Protokoll namens „local“. Local Shovels basieren intern auf AMQP 1.0, verwenden aber keine separaten TCP-Verbindungen, sondern clusterinterne Verbindungen zwischen Clusterknoten und interne Verbindungen für die Veröffentlichung und Nutzung von Nachrichten. APIs Dies kann nur

für die Nutzung und Veröffentlichung innerhalb desselben Clusters verwendet werden und bietet einen höheren Durchsatz bei geringerem Ressourcenverbrauch als AMQP 0-9-1 und AMQP 1.0.

- Quorum-Warteschlangen unterstützen Nachrichtenprioritäten: Die Nachrichtenprioritäten von Quorum-Warteschlangen sind immer aktiv und erfordern keine Richtlinie, um zu funktionieren. Sobald eine Quorum-Warteschlange eine Nachricht mit einer festgelegten Priorität empfängt, wird die Priorisierung aktiviert. Quorum-Warteschlangen unterstützen intern nur zwei Prioritäten: hoch und normal. Nachrichten ohne Prioritätssatz werden der normalen Priorität zugeordnet, ebenso wie die Prioritäten 0-4. Nachrichten mit einer höheren Priorität als 4 werden der höchsten Priorität zugeordnet. Nachrichten mit hoher Priorität werden Nachrichten mit normaler Priorität im Verhältnis 2:1 vorgezogen, d. h. für jeweils 2 Nachrichten mit hoher Priorität wird die Warteschlange 1 Nachricht mit normaler Priorität zugestellt (falls verfügbar). In Quorumwarteschlangen wird daher eine Art nicht strikter Prioritätsverarbeitung nach dem Prinzip „fair share“ implementiert. Auf diese Weise wird gewährleistet, dass bei Nachrichten mit normaler Priorität stets Fortschritte erzielt werden, hohe Prioritäten jedoch in einem Verhältnis von 2:1 bevorzugt werden.
- Khepri: Khepri wird als Standard-Metadatenspeicher für RabbitMQ 4-Broker verwendet
- Mutual TLS (mTLS): Amazon MQ unterstützt Mutual TLS (mTLS) für RabbitMQ-Broker, sodass sich Kunden mithilfe von Zertifikaten authentifizieren können. Weitere Informationen finden Sie unter mTLS-Konfiguration.
- SSL-Zertifikat-Authentifizierungs-Plugin: Das SSL-Authentifizierungs-Plugin verwendet Client-Zertifikate von mTLS-Verbindungen, um Benutzer zu authentifizieren, sodass die Authentifizierung mithilfe von X.509-Clientzertifikaten anstelle von Benutzernamen und Kennwortanmeldeinformationen ermöglicht wird. Weitere Informationen finden Sie unter SSL-Zertifikatsauthentifizierung.
- HTTP-Authentifizierungs-Plugin: Das HTTP-Authentifizierungs-Backend-Plugin ermöglicht das Delegieren von Authentifizierung und Autorisierung an einen externen HTTP-Dienst. Weitere Informationen finden Sie unter HTTP-Authentifizierung und Autorisierung.
- JMS-Unterstützung: Der Broker unterstützt jetzt JMS-Workloads mit aktiviertem JMS-Themenaustausch-Plugin, sodass JMS-Anwendungen über den RabbitMQ JMS-Client eine Verbindung herstellen können.

Die folgenden Funktionen sind seit RabbitMQ 4 auf Amazon MQ veraltet

- Spiegelung klassischer Warteschlangen: Klassische Warteschlangen werden weiterhin unterstützt, ohne dass grundlegende Änderungen für Client-Bibliotheken und -anwendungen vorgenommen wurden, aber sie sind jetzt ein nicht replizierter Warteschlangentyp. Die Clients können sich

mit jedem beliebigen Knoten verbinden, um Inhalte auf allen nicht replizierten klassischen Warteschlangen zu veröffentlichen und Daten aus diesen zu nutzen. Quorum-Warteschlangen werden aus Gründen der Replikation und Datensicherheit empfohlen.

- Entfernung von Global QoS: Kunden wird empfohlen, QoS pro Verbraucher (nicht global) anstelle von Global QoS festzulegen, bei dem ein einziger gemeinsam genutzter Prefetch für einen gesamten Kanal verwendet wird.
- Support für vorübergehende, nicht exklusive Warteschlangen: Transiente Warteschlangen sind Warteschlangen, deren Lebensdauer von der Verfügbarkeit des Knotens abhängt, auf dem sie deklariert sind. In einem Single-Instance-Broker werden sie entfernt, wenn der Knoten neu gestartet wird. In einer Clusterbereitstellung werden sie entfernt, wenn der Knoten, auf dem sie gehostet werden, neu gestartet wird. Wir empfehlen die Verwendung von Queue-TTL für das automatische Löschen ungenutzter Warteschlangen im Leerlauf nach einer gewissen Zeit der Inaktivität. Exklusive Warteschlangen werden weiterhin unterstützt und werden gelöscht, sobald alle Verbindungen zur Warteschlange entfernt wurden.

Die folgenden grundlegenden Änderungen können sich auf Ihre Anwendungen auswirken, wenn Sie ein Upgrade auf RabbitMQ 4.2 auf Amazon MQ durchführen

- Standard-Warteschlangentyp: Der Standard-Warteschlangentyp auf einem RabbitMQ 4-Broker ist auf Quorum eingestellt. Wenn bei der Erstellung der Warteschlange kein Argument für den Warteschlangentyp angegeben wird, wird eine Quorum-Warteschlange erstellt.
- Das Standardlimit für die erneute Zustellung von Quorumwarteschlangen ist auf 20 festgelegt: Nachrichten, die 20 Mal oder öfter erneut zugestellt werden, werden als unleserlich markiert oder gelöscht (entfernt). Wenn 20 Zustellungen pro Nachricht ein übliches Szenario für eine Warteschlange sind, muss für solche Warteschlangen ein Ziel mit unerlaubter Nachricht oder ein höheres Limit konfiguriert werden, um Datenverlust zu vermeiden. Der empfohlene Weg, dies zu tun, ist eine Richtlinie.
- amqplib: Amqplib-Versionen des Node-JS-Clients, die älter als 0.10.7 sind, oder jede AMQP-Clientbibliothek, die frame_max < 8192 verwendet, können keine Verbindung zu RabbitMQ herstellen
- Standard-Ressourcenlimits: Amazon MQ for RabbitMQ hat Standardbeschränkungen für die Ressourcennutzung für Verbindungen, Kanäle, Verbraucher pro Kanal, Warteschlangen, Vhosts, Shovels, Exchanges und die maximale Nachrichtengröße eingeführt. Diese dienen als Leitplanken zum Schutz der Verfügbarkeit von Brokern und können mithilfe von Konfigurationen an Ihre spezifischen Anforderungen angepasst werden.

Die folgenden Funktionen werden auf RabbitMQ 4 auf Amazon MQ nicht unterstützt

- Lokaler zufälliger Austausch: Lokale zufällige Börsen werden auf Amazon MQ nicht unterstützt, da sich die Amazon MQ-Knoten hinter einem Netzwerk-Load-Balancer befinden.
- Message Interceptor: RabbitMQ-Nachrichtenabfanger werden auf Amazon MQ nicht unterstützt.
- Metriken pro Warteschlange: Amazon MQ verkauft keine RabbitMQ-Warteschlangenmetriken für RabbitMQ 4-Broker. AWS CloudWatch Amazon MQ wird weiterhin Metriken auf Brokerebene bereitstellen. AWS CloudWatch Sie können Warteschlangenmetriken mithilfe der RabbitMQ-Management-API abfragen. Wir empfehlen, Metriken für bestimmte Warteschlangen in Intervallen von einer Minute oder länger abzufragen.

Unterstützung der RabbitMQ-Version

Der unten stehende 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
4.2 (Empfohlen)	
3.13	
3.12	17. März 2025

Versionsupgrades

Sie können Ihren Broker jederzeit manuell auf die nächste unterstützte Haupt- oder Nebenversion aktualisieren. Weitere Informationen zum manuellen Upgrade Ihres Brokers finden Sie unter Upgrade einer Amazon MQ-Broker-Engine-Version.

Amazon MQ verwaltet Upgrades auf die neueste unterstützte Patch-Version für alle RabbitMQ-Broker, die Version 3.13 und höher verwenden. Sowohl manuelle als auch automatische Versions-Upgrades erfolgen während des geplanten Wartungsfensters oder nachdem SieSo starten Sie Ihren Broker neu.

> **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 zu minimieren, 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.

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 mithilfe vonjede Programmiersprache, die RabbitMQ unterstütztund durch Aktivieren von TLS für die folgenden Protokolle:

- AMQP (0-9-1)

Themen

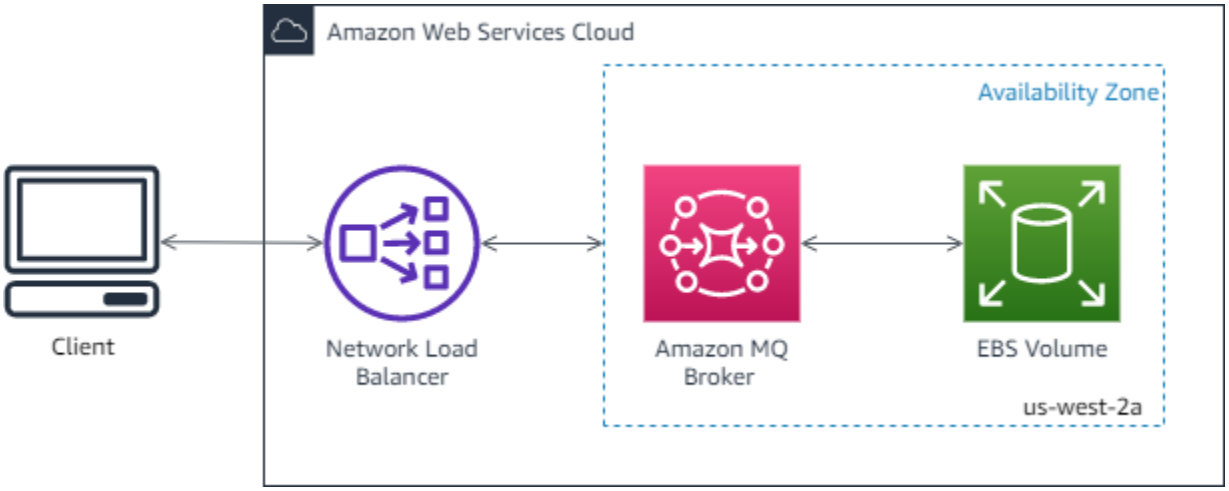
- Option 1: Einzelinstanz-Broker Amazon MQ für RabbitMQ
- Option 2: Amazon MQ für die RabbitMQ-Clusterbereitstellung

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 Amazon EBS-Speicher-Volume. 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.

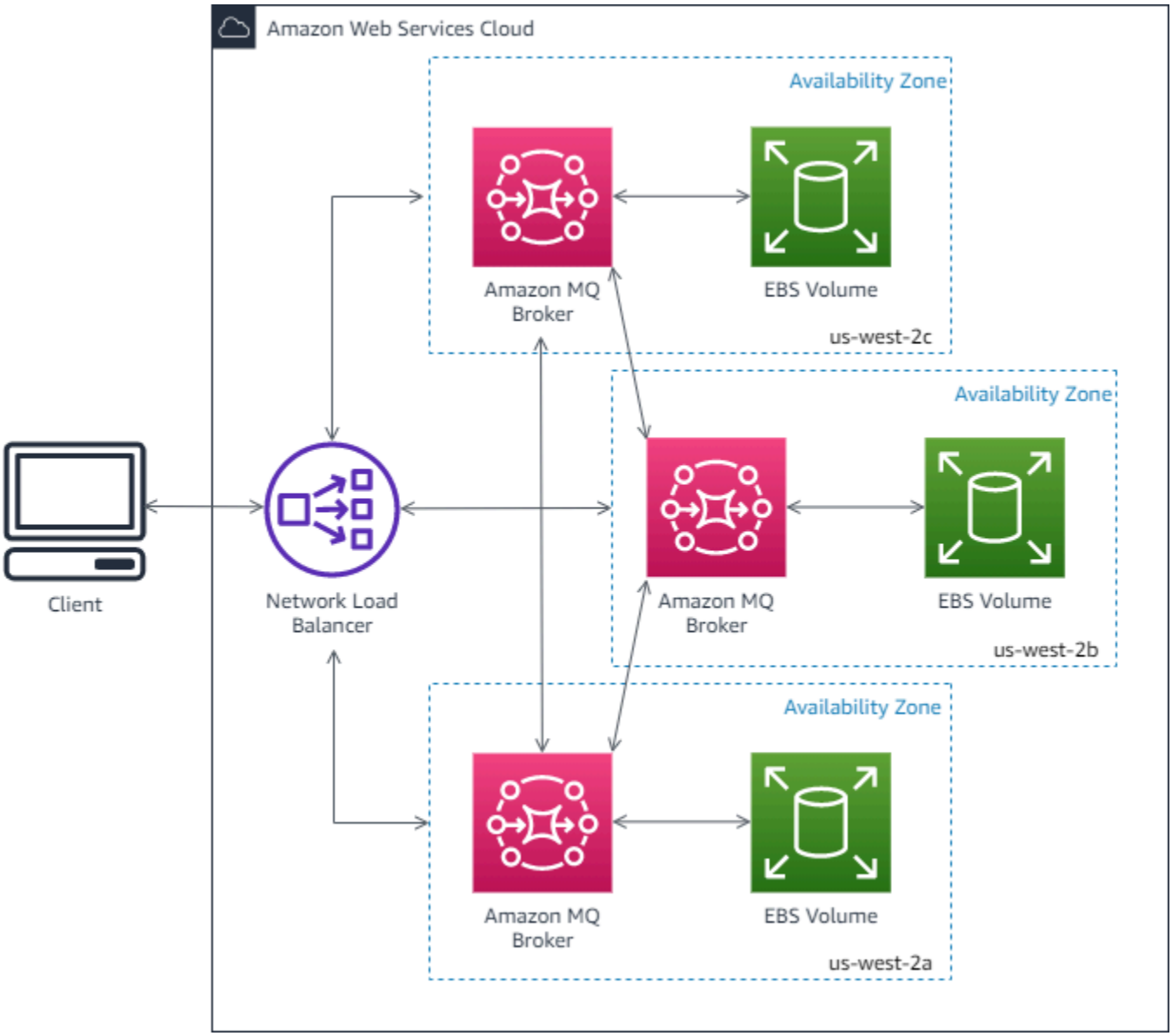
> **Note**
>
> Bei einer Cluster-Bereitstellung versucht Amazon MQ bei einem Ausfall der Availability Zone automatisch, die betroffenen RabbitMQ-Knoten in eine andere AZ zu verlagern, um die Clustergröße beizubehalten. Sobald der Ausfall behoben ist, wird der Cluster automatisch neu verteilt. AZs

> **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 “Schritt 1: 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 veranschaulicht eine RabbitMQ-Cluster-Brokerbereitstellung mit drei Knoten in drei Availability Zones (AZ), von denen jeder ein eigenes Amazon EBS-Volume und ein

freigegebener Status aufweist. Amazon EBS bietet Speicher auf Blockebene, der für niedrige Latenz und hohen Durchsatz optimiert ist.



Broker-Instance-Typen von Amazon MQ für RabbitMQ

Die kombinierte Beschreibung der Broker-Instance-Klasse (m7g) und der Größe (large, medium) wird als Broker-Instance-Typ bezeichnet (z. B. mq.m7g.large).

Wir empfehlen die Verwendung von mq.m7g-Instance-Typen sowohl für Cluster- als auch für Einzelinstanz-Bereitstellungen.

Amazon MQ informiert Sie mindestens 90 Tage im Voraus, bevor der Support für einen Instance-Typ endet. Wir empfehlen, Ihren Broker vor diesem end-of-support Datum auf einen neuen Instance-Typ zu aktualisieren, um Störungen zu vermeiden.

> **Important**
>
> Sie können einen Broker nicht von einem Instance-Typ `mq.m7g` oder auf einen `mq.m5` Instance-Typ herabstufen. `mq.t3.micro`
> Der `mq.t3.micro` Instance-Typ unterstützt keine Cluster-Bereitstellung.

Instanztypen für die Bereitstellung von M7G-Clustern

Wir empfehlen die Verwendung von `mq.m7g.x` Instance-Typen bei der Cluster-Bereitstellung. Die folgende Tabelle zeigt die verfügbaren `mq.m7g.x` Instance-Typen für die Cluster-Bereitstellung.

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerk-Basis-/Burst-Bandbreite (Gbit/s)	Empfohlene Verwendung	Speicher	Größe des Festplattenvolumens pro Knoten (GB)
mq.m7g.medium	1	4	0,52 / 12,5	Bewertung	EBS	5
mq.m7g.groß	2	8	0,937 / 12,5	Produktion	EBS	15
mq.m7g.xlarge	4	16	1,876/12,5	Produktion	EBS	25
mq. m 7 g 2 x groß	8	32	3,75 / 15,0	Produktion	EBS	45
mq.m7g.4x groß	16	64	7,5 / 15,0	Produktion	EBS	90

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerk-Basis-/Burst-Bandbreite (Gbit/s)	Empfohlene Verwendung	Speicher	Größe des Festplattenvolumens pro Knoten (GB)
mq.m7g.8x groß	32	128	15 Gigabit	Produktion	EBS	175
mq.m7g.12x groß	48	192	22,5 Gigabit	Produktion	EBS	260
mq.m7g.16x groß	64	256	30 Gigabit	Produktion	EBS	345

Instanztypen für die Bereitstellung von m7g-Einzelinstanzen

Die folgende Tabelle zeigt die verfügbaren `mq.m7g.x` Instanztypen für die Bereitstellung einer einzelnen Instanz.

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerk-Basis-/Burst-Bandbreite (Gbit/s)	Empfohlene Verwendung	Speicher	Größe des Festplattenvolumens pro Knoten (GB)
mq.m7g.medium	1	4	0,52 / 12,5	Bewertung	EBS	200
mq.m7g.groß	2	8	0,937 / 12,5	Produktion	EBS	200
mq.m7g.xlarge	4	16	1,876/12,5	Produktion	EBS	200

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerk-Basis-/Burst-Bandbreite (Gbit/s)	Empfohlene Verwendung	Speicher	Größe des Festplattenvolumens pro Knoten (GB)
mq. m 7 g 2 x groß	8	32	3,75 / 15,0	Produktion	EBS	200
mq.m7g.4x groß	16	64	7,5 / 15,0	Produktion	EBS	200
mq.m7g.8x groß	32	128	15 Gigabit	Produktion	EBS	200
mq.m7g.12 x groß	48	192	22,5 Gigabit	Produktion	EBS	200
mq.m7g.16 x groß	64	256	39 Gigabit	Produktion	EBS	200

Instanztypen für die Bereitstellung **`mq.m5`** einer einzelnen Instanz

Die folgenden Tabellen zeigen die verfügbaren `mq.m5.x` Instanztypen für die Einzelinstanzbereitstellung

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerk-Basis-/Burst-Bandbreite (Gbit/s)	Empfohlene Verwendung	Speicher	Größe des Festplattenvolumens pro Knoten (GB)
mq.t3.micro	2	1	0,064/5,0	Bewertung	EBS	20

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerk-Basis-/Burst-Bandbreite (Gbit/s)	Empfohlene Verwendung	Speicher	Größe des Festplattenvolumens pro Knoten (GB)
mq.m5. groß	2	8	0,75/10,0	Produktion	EBS	200
mq.m5.x groß	4	16	1,25 / 10,0	Produktion	EBS	200
mq.m5.2 x groß	8	32	2,5 / 10,0	Produktion	EBS	200
mq.m 5,4 x groß	16	64	5,0 / 10,0	Produktion	EBS	200

Instanztypen für die Cluster-Bereitstellung **`mq.m5`**

Die folgenden Tabellen zeigen die verfügbaren `mq.m5.x` Instanztypen für die Cluster-Bereitstellung

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerk-Basis-/Burst-Bandbreite (Gbit/s)	Empfohlene Verwendung	Speicher	Größe des Festplattenvolumens pro Knoten (GB)
mq.m5.large	2	8	0,75/10,0	Produktion	EBS	200
mq.m5.x groß	4	16	1,25 / 10,0	Produktion	EBS	200

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerk-Basis-/Burst-Bandbreite (Gbit/s)	Empfohlene Verwendung	Speicher	Größe des Festplattenvolumens pro Knoten (GB)
mq.m5.2 x groß	8	32	2,5 / 10,0	Produktion	EBS	200
mq.m 5,4 x groß	16	64	5,0 / 10,0	Produktion	EBS	200

Größenrichtlinien für Amazon MQ für RabbitMQ

Sie können den Broker-Instance-Typ wählen, der Ihre Anwendung am besten unterstützt. Berücksichtigen Sie bei der Auswahl eines Instance-Typs Faktoren, 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 `m7g.medium` werden nur zum Testen der Anwendungsleistung empfohlen. Wir empfehlen größere Broker-Instance-Typen `m7g.large` und höher oder Produktionsebenen von Clients und Warteschlangen, hohen Durchsatz, Nachrichten im Speicher und redundante Nachrichten.

> **Important**
>
> Sie können einen Broker nicht von einem Instance-Typ `mq.m5` oder auf einen `mq.m7g` Instance-Typ herabstufen. `mq.t3.micro`

Es ist wichtig, Ihre Broker zu testen, um den geeigneten Instance-Typ und die Größe für Ihre Workload-Messaging-Anforderungen zu ermitteln.

Verwenden Sie immer die Standard-Ressourcenlimits auf dem RabbitMQ 4-Broker, um die geeignete Instance-Größe für Ihre Anwendung gemäß den Best Practices von Amazon MQ zu ermitteln. Diese Standard-Ressourcenlimits basieren auf den Typen, dem `m7g` Instance-Typ und den Quorum-Warteschlangen.

- Standardressourcenlimits für die Bereitstellung von M7G-Einzelinstanzen
- Standard-Ressourcenlimits für die Bereitstellung von M7G-Clustern

Sie können den Wert eines beliebigen Grenzwerts bis zu den Höchstwerten erhöhen, die je nach Instanztyp und Bereitstellungsmodus definiert sind. Wir empfehlen jedoch dringend, die Leistung des Brokers anhand der erhöhten Werte zu testen, bevor Sie ihn in der Produktion einsetzen.

- Maximale Ressourcenlimits für die Bereitstellung einer M7G-Single-Instance
- Maximale Ressourcenlimits für die Bereitstellung von M7G-Clustern
- Maximale Ressourcenlimits für die M5-Single-Instance-Bereitstellung
- Maximale Ressourcenlimits für die Bereitstellung eines M5-Clusters
- Fehlermeldungen

> **Note**
>
> RabbitMQ 3.13-Broker verfügen nicht über standardmäßige Ressourcenlimits, wir empfehlen jedoch, die empfohlenen Standardwerte zu verwenden.

Standardmäßige Ressourcenlimits

Amazon MQ for RabbitMQ unterstützt die Konfiguration der Broker-Ressourcenlimits ab RabbitMQ 4. Wenn Sie einen Broker erstellen, wendet Amazon MQ automatisch Standardwerte auf diese Ressourcenlimits an. Diese Standardwerte dienen als Schutzmaßnahmen, um die Verfügbarkeit Ihres Brokers zu schützen und gleichzeitig den gängigen Nutzungsmustern der Kunden Rechnung zu tragen. Sie können das Verhalten Ihres Brokers anpassen, indem Sie die Werte für die Konfiguration der Grenzwerte so ändern, dass sie Ihren spezifischen Workload-Anforderungen besser entsprechen.

Bevor Sie Änderungen vornehmen, beachten Sie bitte:

Important

1. Konfigurationsänderungen können sich auf die Leistung und Verfügbarkeit des Brokers auswirken
2. Machen Sie sich mit den Auswirkungen vertraut, bevor Sie die Standardkonfigurationsoptionen ändern
3. Testen Sie zunächst Konfigurationsänderungen in Umgebungen außerhalb der Produktionsumgebung
4. Überwachen Sie den Zustand des Brokers nach der Übernahme der Änderungen

Important

Die Standardwerte und unterstützten Bereiche für diese Konfigurationen variieren je nach RabbitMQ-Version, Instanztyp und Broker-Bereitstellungsmodus.

Important

Hinweis: Das Zuordnen oder Erstellen eines Brokers mit Konfigurationswerten außerhalb des unterstützten Bereichs führt zu einer Fehlerantwort.

Die für RabbitMQ 4.2-Broker geltenden Standard-Ressourcenlimits sind

- Die Standard-Ressourcenlimits für die Bereitstellung von m7g-Einzelinstanzen
- Standard-Ressourcenlimits für die Bereitstellung von M7G-Clustern

Standard-Ressourcenlimits

Important

Amazon MQ für RabbitMQ 3-Broker, die Standardeinstellung ist mit dem maximalen Ressourcenlimit konfiguriert und Amazon MQ bietet keine Möglichkeit, die Konfiguration des Ressourcenlimits zu überschreiben.

Standardwerte für Single-Instance-Broker

Instance-Typ	Verbindungen pro Knoten	Kanäle pro Knoten	Verbraucher pro Kanal	Queues (Warteschlangen)	Geister	Schaufeln	Börsen	Nachrichtengröße in Byte
mq.m7g.medium	100	500	10	500	10	30	500	524288
mq. 7 g groß	1.500	4.500	10	1.000	50	50	1.000	524288
mq.m7g.xarge	3,000	9 000	10	2.000	100	100	2.000	524288
mq. m 7 g 2 x groß	6.000	18 000	10	4.000	150	200	4.000	524288
mq. m 7 g 4 x groß	12.000	36.000	10	8 000	200	400	8 000	524288
mq. 7 g x 8 x groß	24.000	72.000	10	16,000	250	800	16,000	524288
mq. m 7 g, 12 x groß	36.000	108.000	10	24.000	300	1.200	24.000	524288
mq. m 7 g x 16 x groß	48.000	144.000	10	32 000	350	1.600	32 000	524288

Standardwerte für Cluster-Broker

Instance-Typ	Verbindungen pro Knoten	Kanäle pro Knoten	Verbraucher pro Kanal	Queues (Warteschlangen)	Geister	Schaufeln	Börsen	Nachrichtengröße in Byte
mq.m7g.medium	100	300	10	100	10	10	100	524288
mq. 7 g groß	500	1500	10	1.000	50	30	1.000	524288
mq.m7g.xarge	1000	3000	10	2.000	100	60	2.000	524288
mq. m 7 g 2 x groß	2000	6 000	10	4.000	150	120	4.000	524288
mq. m 7 g 4 x groß	4000	12.000	10	8 000	200	240	8 000	524288
mq. 7 g x 8 x groß	8000	24.000	10	16,000	250	480	16,000	524288
mq. m 7 g, 12 x groß	12000	36.000	10	24.000	300	720	24000	524288
mq. m 7 g x 16 x groß	16,000	48.000	10	32 000	350	960	32 000	524288

Maximales Ressourcenlimit von Amazon MQ für RabbitMQ

Größenrichtlinien für M7G mit Quorum-Warteschlangen für die Bereitstellung einzelner Instanzen

Die folgende Tabelle zeigt die maximalen Grenzwerte für jeden Instance-Typ für Single-Instance-Broker.

Instance-Typ	Verbindungen	Kanäle	Verbraucher pro Kanal	Queues (Warteschlangen)	Geister	Schaufeln	Börsen	Nachrichtengröße in Byte
mq.m7g.medium	300	900	1.000	2.500	10	150	12500	134217728
mq.m7g.groß	5,000	15 000	1.000	20 000	1500	250	100 000	134217728
mq.m7g.xarge	10.000	30 000	1.000	30 000	1.500	500	150.000	134217728
mq. m 7 g 2 x groß	20 000	60 000	1.000	40 000	1.500	1.000	200 000	134217728
mq.m7g.4 groß	40 000	120.000	1.000	60 000	1.500	2000	300,000	134217728
mq.m7g.8 groß	80 000	240.000	1.000	80 000	1.500	4000	400 000	134217728
mq. m 7 g, 12 x groß	120.000	360 000	1.000	100 000	1.500	6.000	500 000	134217728
mq.m7g.1 x groß	160 000	480.000	1.000	120.000	1.500	8 000	600.000	134217728

Größenrichtlinien für M7G mit Quorum-Warteschlangen für die Clusterbereitstellung

Die folgende Tabelle zeigt die maximalen Grenzwerte für jeden Instance-Typ für Cluster-Broker.

Instance-Typ	Verbindungen pro Knoten	Kanäle pro Knoten	Verbraucher pro Kanal	Queues (Warteschlangen)	Geister	Schaufeln	Börsen	Nachrichtengröße in Byte
mq.m7g.medium	300	900	1.000	500	10	50	500	134217728
mq.m7g.groß	5,000	15 000	1.000	10.000	1.500	150	50 000	134217728
mq.m7g.xarge	10.000	30 000	1.000	15 000	1.500	300	75 000	134217728
mq.m7g.2x groß	20 000	60 000	1.000	20 000	1.500	600	100 000	134217728
mq.m7g.4 groß	40 000	120.000	1.000	30 000	1.500	1200	150.000	134217728
mq. 7 g. 8 x groß	80 000	240.000	1.000	40 000	1.500	2.400	200 000	134217728
mq. m 7 g, 12 x groß	120.000	360 000	1.000	50 000	1.500	3.600	250 000	134217728
mq.m7g.1x groß	160 000	480.000	1.000	60 000	1.500	4.800	300,000	134217728

Maximale Ressourcenlimits für die M5-Single-Instance-Bereitstellung

Die folgende Tabelle zeigt die maximalen Grenzwerte für jeden Instance-Typ für Single-Instance-Broker.

Instance-Typ	Verbindungen	Kanäle	Verbraucher pro Kanal	Queues (Wartesch langen)	Geister	Schaufeln
m5.large	5,000	15 000	1.000	30 000	1500	250
m5.xlarge	10.000	30 000	1.000	60 000	1500	500
m5.2xlarge	20 000	60 000	1.000	120.000	1500	1.000
m5.4xlarge	40 000	120.000	1000	240.000	1.000	2.000

Maximale Ressourcenlimits für die Bereitstellung von M5-Clustern

Die folgende Tabelle zeigt die maximalen Grenzwerte für jeden Instance-Typ für Cluster-Broker.

Instance-Typ	Queues (Wartesch langen)	Verbraucher pro Kanal	Schaufeln
m5.large	10.000	1.000	150
m5.xlarge	15 000	1.000	300
m5.2xlarge	20 000	1.000	600
m5.4xlarge	30 000	1.000	1200

Pro Knoten gelten die folgenden Verbindungs- und Kanalbeschränkungen:

Instance-Typ	Verbindungen	Kanäle
m5.large	5000	15 000
m5.xlarge	10.000	30 000
m5.2xlarge	20 000	60 000
m5.4xlarge	40 000	120.000

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. Wenn Sie die Grenzwerte überschreiten, können Sie eine neue Verbindung zu einem anderen Knoten herstellen und es erneut versuchen, oder Sie können die Instanzgröße aktualisieren, um die maximalen Grenzwerte zu erhöhen

Fehlermeldungen

Die folgenden Fehlermeldungen werden zurückgegeben, wenn die Grenzwerte überschritten werden. Alle Werte basieren auf den Grenzwerten für **`m7.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 (5000) 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')
```

Maximale Nachrichtengröße

```
(406, 'PRECONDITION_FAILED - message size 524289 is larger than configured
max size 524288')
```

Austausch

```
(406, "PRECONDITION_FAILED - cannot declare exchange 'limit_test_3' in
vhost '/': exchange limit of 10 is reached")
```

> **Note**
>
> Die folgenden Fehlermeldungen verwenden das HTTP Management API-Format.

Warteschlange

```
{"error":"bad_request","reason":"cannot declare queue 'my_queue': queue
limit in cluster (10,000) is reached"}]
```

Schaufel

```
{"error":"bad_request","reason":"Validation failed\n\ncomponent shovel is
limited to 150 per node\n"}
```

Gespenst

```
{"error":"bad_request","reason":"cannot create vhost 'my_vhost': vhost
limit of 1500 is reached"}
```

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 unterschiedliche Broker-Richtlinien und Vhost-Limits für Amazon MQ for RabbitMQ 3 und RabbitMQ 4. Die Unterschiede werden in den folgenden Unterabschnitten ausführlich erörtert.

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:

Amazon MQ für RabbitMQ 3:

- Einzelne Instance – `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`

- Cluster-Bereitstellung — && `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ` `AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Amazon MQ für RabbitMQ 4:

- Einzelne Instance – `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Cluster-Bereitstellung — && `AWS-DEFAULT-POLICY-CLUSTER` `AWS-DEFAULT-QUORUM-QUEUES-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. Diese Einstellungen gelten nur für Amazon MQ für RabbitMQ 3 und sind nicht für RabbitMQ 4 konfiguriert.
> 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.

Einstellungen sowohl bei RabbitMQ 3- als auch bei RabbitMQ 4-Brokern

- **`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

Spezifische Einstellungen für RabbitMQ 3

- **`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 das Beschränken von Warteschlangen wird auch eine übermäßige CPU-Auslastung für die Wartung von 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

> **Important**
>
> `max-queues`und `max-connections` gelten nur für Amazon MQ für RabbitMQ 3.

> **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	Bereitste llungsmodus	**`max-length`**	**`max-queues`**	**`max-conne ctions`**
mq.m7g.medium	Single-Instance	–	2.500	100
	Cluster	500 000	100	100
mq.m7g. groß	Single-Instance	–	20 000	5,000
	Cluster	8.000.000	10.000	5,000
mq. 7 g. x groß	Single-Instance	–	30 000	10.000
	Cluster	9.000.000	15 000	10.000
qm 7 g 2 x groß	Single-Instance	–	40 000	20 000
	Cluster	10 000 000	40 000	20 000
mq.m7g.4xgroß	Single-Instance	–	60 000	40 000
	Cluster	12.000.000	30 000	40 000
mq. 7 g. 8 x groß	Single-Instance	–	80 000	80 000
	Cluster	20,000,000	40 000	80 000

Instance-Typ	Bereitste llungsmodus	**max-length**	**max-queues**	**max-conne ctions**
mq.m7g.12x groß	Single-Instance	–	100 000	120.000
	Cluster	30.000.000	20 000	120.000
mq. 7 g. 16 x groß	Single-Instance	–	120.000	160 000
	Cluster	40.000.000	50 000	160 000

Instance-Typ	Bereitste llungsmodus	**max-length**	**max-queues**	**max-conne ctions**
t3.micro	Single-Instance	–	500	500
m5.large	Single-Instance	–	20 000	4.000
m5.large	Cluster	8.000.000	10.000	15 000
m5.xlarge	Single-Instance	–	30 000	8 000
m5.xlarge	Cluster	9.000.000	10.000	20 000
m5.2xlarge	Single-Instance	–	60 000	15 000
m5.2xlarge	Cluster	10 000 000	10.000	40 000
m5.4xlarge	Single-Instance	–	150.000	30 000
m5.4xlarge	Cluster	12.000.000	10.000	100 000

Konfiguration eines RabbitMQ-Brokers

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-1234a5b678cd-901e-2fgh-3i45j6k178l9)

Eine vollständige Liste der Konfigurationsattribute finden Sie im folgenden Abschnitt im Amazon MQ REST API Reference:

- REST-Operations-ID: Configuration
- REST-Operations-ID: Configurations

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

- REST-Operations-ID: Configuration Revision
- REST-Operations-ID: Configuration Revisions

Topics

- RabbitMQ-Broker-Konfigurationen erstellen und anwenden
- Eine Konfigurationsrevision von Amazon MQ für RabbitMQ bearbeiten
- Konfigurierbare Werte für RabbitMQ auf Amazon MQ
- ARN-Unterstützung in der RabbitMQ-Konfiguration

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.

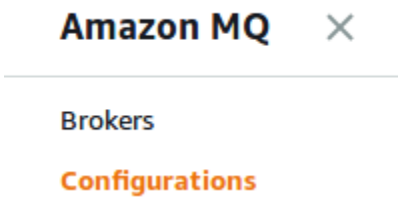
Das folgenden Beispiele zeigen, wie Sie eine RabbitMQ-Broker-Konfiguration mithilfe der AWS-Managementkonsole erstellen und anwenden.

> **Important**
>
> Sie können eine Konfiguration nur mithilfe der `DeleteConfiguration` API löschen. Weitere Informationen finden Sie unter Konfigurationen in der Amazon MQ API-Referenz.

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 der ***MyConfiguration***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 Speichern.

   DieSpeichern der Revisionwird angezeigt.
5. (Optional) Geben Sie `A description of the changes in this revision` ein.
6. Wählen Sie Speichern.

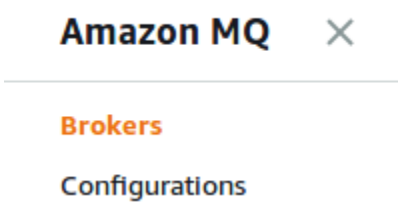
   Die neue Version der Konfiguration wird gespeichert.

> **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. Wählen Sie auf der *MyBroker* Seite Bearbeiten im Abschnitt Konfiguration eine Konfiguration und eine Revision aus und wählen Sie dann Änderungen planen aus.
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**
>
> Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern ist jeweils nur ein Knoten ausgefallen, während der Broker neu gestartet wird.

5. Wählen Sie 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. Wählen Sie auf der ***MyBroker***Seite Bearbeiten aus.

4. Wählen Sie auf der *MyBroker* Seite Bearbeiten 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 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 Speichern.

   DieSpeichern der Revisionwird angezeigt.

8. (Optional) Geben Sie `A description of the changes in this revision` ein.
9. Wählen Sie Speichern.

   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.

Konfigurierbare Werte

Sie können den Wert der folgenden Broker-Konfigurationsoptionen festlegen, indem Sie die Broker-Konfigurationsdatei in der ändern AWS-Managementkonsole.

Zusätzlich zu den in der folgenden Tabelle beschriebenen Werten unterstützt Amazon MQ zusätzliche Broker-Konfigurationsoptionen in Bezug auf Authentifizierung und Autorisierung sowie Ressourcenlimits. Weitere Informationen zu diesen Konfigurationsoptionen finden Sie unter

- OAuth 2.0-Konfiguration
- LDAP-Konfiguration
- HTTP-Konfiguration
- SSL-Konfiguration
- mTLS-Konfiguration
- ARN-Unterstützung
- Ressourcenlimits
- SSL-Konfiguration des AMQP-Clients

Konfiguration	Standardwert	Empfohlener Wert	Werte	Anwendbare Versionen	Description
consumer_timeout	1800000 ms (30 Minuten)	1800000 ms (30 Minuten)	0 bis 2.147.483.647 ms. Amazon MQ unterstützt auch den Wert 0, was „unendlich“ bedeutet.	Alle Versionen	Ein Timeout bei der Lieferbestätigung für Verbraucher, um festzustellen, wann Verbraucher keine Lieferungen verpassen.
Herzschlag	60 Sekunden	60 Sekunden	60 bis 3600 Sekunden	Alle Versionen	Definiert die Zeit, bevor eine

Konfiguration	Standardwert	Empfohlener Wert	Werte	Anwendbare Versionen	Description
					Verbindung von RabbitMQ als nicht verfügbar angesehen wird.
management.restrictions.operator_policy_changes.disabled	true	true	true, false	Alle Versionen	Deaktiviert das Vornehmen von Änderungen an den Betreiberrichtlinien. Wenn Sie diese Änderung vornehmen, wird Ihnen dringend empfohlen, die HA-Eigenschaften in Ihre eigenen Betreiberrichtlinien aufzunehmen.

Konfiguration	Standardwert	Empfohlener Wert	Werte	Anwendbare Versionen	Description
quorum_queue.property_equivalence.relaxed_checks_on_redeclaration	true	true	true, false	Alle Versionen	Wenn dieser Wert auf TRUE gesetzt ist, vermeidet Ihre Anwendung beim erneuten Deklarieren einer Quorum-Warteschlange eine Kanalausnahme.
secure.management.http.headers.enabled	true	true	true, false	Alle Versionen	Aktiviert unveränderbare HTTP-Sicherheitsheader.

Konfiguration der Empfangsbestätigung für Verbraucher

Sie können consumer_timeout so konfigurieren, dass erkannt wird, wenn Verbraucher keine Lieferungen verpassen. Wenn der Verbraucher innerhalb des Timeout-Werts keine Bestätigung sendet, wird der Kanal geschlossen. Wenn Sie beispielsweise den Standardwert 1800000 Millisekunden verwenden und der Verbraucher innerhalb von 1800000 Millisekunden keine Empfangsbestätigung sendet, wird der Kanal geschlossen. Amazon MQ unterstützt auch den Wert 0, was „unendlich“ bedeutet.

Heartbeat konfigurieren

Sie können ein Heartbeat-Timeout konfigurieren, um herauszufinden, wann Verbindungen unterbrochen oder ausgefallen sind. Der Heartbeat-Wert definiert das Zeitlimit, bis eine Verbindung als ausgefallen betrachtet wird.

Konfiguration von Betreiberrichtlinien

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 Betreiberrichtlinien über die AWS-Managementkonsole oder die 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.

Konfiguration von gelockerten Prüfungen bei der Warteschlangendeklaration

Wenn Sie Ihre klassischen Warteschlangen auf Quorumwarteschlangen migriert, aber Ihren Client-Code nicht aktualisiert haben, können Sie beim erneuten Deklarieren einer Quorumwarteschlange eine Kanalausnahme vermeiden, indem Sie quorum_queue.property_equivalence.relaxed_checks_on_redeclaration auf true setzen.

Konfiguration von HTTP-Sicherheitsheadern

Die secure.management.http.headers.enabled-Konfiguration aktiviert die folgenden 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 Verwaltungs-Plugin in einen Frame auf ihrer eigenen Website einbetten, um andere zu täuschen
- Strict-Transport-Security: max-age=47304000; includeSubDomains: zwingt Browser dazu, HTTPS zu verwenden, wenn sie über einen längeren Zeitraum (1,5 Jahre) weitere Verbindungen zur Website und ihren Subdomains herstellen.

Für Amazon MQ for RabbitMQ-Broker, die mit Versionen 3.10 und höher erstellt wurden, ist secure.management.http.headers.enabled standardmäßig auf true gesetzt. Sie können diese HTTP-Sicherheitsheader aktivieren, indem Sie secure.management.http.headers.enabled auf true setzen. Wenn Sie diese HTTP-Sicherheitsheader deaktivieren möchten, setzen Sie secure.management.http.headers.enabled auf false.

Konfiguration der Authentifizierung und Autorisierung 2.0 OAuth

Informationen zu den OAuth 2.0-Konfigurationsoptionen und zur Einrichtung der OAuth 2.0-Authentifizierung für Ihre Broker finden Sie unter Unterstützte OAuth 2.0-Konfigurationen und Verwenden der OAuth 2.0-Authentifizierung und -Autorisierung.

Konfiguration der LDAP-Authentifizierung und -Autorisierung

Informationen zu den LDAP-Konfigurationsoptionen und zur Einrichtung der LDAP-Authentifizierung für Ihre Broker finden Sie unter Unterstützte LDAP-Konfigurationen und. Verwendung der LDAP-Authentifizierung und -Autorisierung

Konfiguration der HTTP-Authentifizierung und -Autorisierung

Informationen zu den Konfigurationswerten für die HTTP-Authentifizierung und zur Einrichtung der HTTP-Authentifizierung für Ihre Broker finden Sie unter HTTP-Authentifizierung und -Autorisierung und. Verwendung der HTTP-Authentifizierung und -Autorisierung

> **Note**
>
> Das HTTP-Authentifizierungs-Plugin ist nur für Amazon MQ für RabbitMQ Version 4 und höher verfügbar.

Konfiguration der SSL-Zertifikatsauthentifizierung

Informationen zu den Konfigurationswerten für die SSL-Zertifikatsauthentifizierung und zur Einrichtung der SSL-Zertifikatsauthentifizierung für Ihre Broker finden Sie unter SSL-Zertifikatsauthentifizierung und. Verwendung der SSL-Zertifikatsauthentifizierung

> **Note**
>
> Das SSL-Zertifikat-Authentifizierungs-Plugin ist nur für Amazon MQ für RabbitMQ Version 4 und höher verfügbar.

Konfiguration von mTLS

Amazon MQ for RabbitMQ unterstützt Mutual TLS (mTLS) für sichere Verbindungen zu verschiedenen Endpunkten und externen Diensten. mTLS bietet erhöhte Sicherheit, da sich sowohl Client als auch Server mithilfe von Zertifikaten authentifizieren müssen.

> **Note**
>
> Die Verwendung von privaten Zertifizierungsstellen für mTLS ist nur für Amazon MQ for RabbitMQ Version 4 und höher verfügbar.

> **Important**
>
> Amazon MQ for RabbitMQ erzwingt die Verwendung von AWS ARNs für Zertifikate und private Schlüsseldateien. Weitere Informationen finden Sie unter ARN-Unterstützung in der RabbitMQ-Konfiguration.

Auf dieser Seite

- AMQP-Endpunkt
- RabbitMQ-Verwaltungs-Plugin
- RabbitMQ 2.0-Plugin OAuth
- RabbitMQ HTTP-Authentifizierungs-Plugin
- RabbitMQ LDAP-Plugin
- AMQP-Client-Verbindungen

AMQP-Endpunkt

Konfigurieren Sie mTLS für Client-Verbindungen zum AMQP-Endpunkt. Dies wird bei der SSL-Zertifikatsauthentifizierung verwendet. Informationen zu unterstützten Konfigurationen finden Sie unterAuthentifizierung mit SSL-Zertifikaten.

RabbitMQ-Verwaltungs-Plugin

Konfigurieren Sie mTLS für Verbindungen zur RabbitMQ-Verwaltungsoberfläche.

> **Note**
>
> Strict mTLS wird für die Management-API nicht unterstützt.

Unterstützte Konfigurationen

`aws.arns.management.ssl.cacertfile`

Zertifizierungsstellendatei zur Überprüfung von Client-Zertifikaten, die eine Verbindung zur Verwaltungsschnittstelle herstellen.

`management.ssl.verify`

Peer-Verifizierungsmodus. Unterstützte Werte:`verify_none`, `verify_peer`

`management.ssl.depth`

Maximale Tiefe der Zertifikatskette für die Überprüfung.

`management.ssl.hostname_verification`

Modus zur Überprüfung des Hostnamens. Unterstützte Werte:`wildcard`, `none`

Nicht unterstützte SSL-Optionen

Die folgenden SSL-Konfigurationswerte werden nicht unterstützt:

Vollständige Liste anzeigen

- `management.ssl.cert`
- `management.ssl.client_renegotiation`
- `management.ssl.dh`
- `management.ssl.dhfile`
- `management.ssl.fail_if_no_peer_cert`
- `management.ssl.honor_cipher_order`
- `management.ssl.honor_ecc_order`
- `management.ssl.key.RSAPrivateKey`
- `management.ssl.key.DSAPrivateKey`
- `management.ssl.key.PrivateKeyInfo`
- `management.ssl.log_alert`
- `management.ssl.password`
- `management.ssl.psk_identity`
- `management.ssl.reuse_sessions`
- `management.ssl.secure_renegotiate`
- `management.ssl.versions.$version`
- `management.ssl.sni`

RabbitMQ 2.0-Plugin OAuth

Konfigurieren Sie mTLS für Verbindungen von Amazon MQ zum OAuth 2.0-Identitätsanbieter. Informationen zu unterstützten Konfigurationen finden Sie unter. OAuth 2.0 Authentifizierung und Autorisierung

RabbitMQ HTTP-Authentifizierungs-Plugin

Konfigurieren Sie mTLS für Verbindungen von Amazon MQ zum HTTP-Authentifizierungsserver. Informationen zu unterstützten Konfigurationen finden Sie unter. HTTP-Authentifizierung und Autorisierung

RabbitMQ LDAP-Plugin

Konfigurieren Sie mTLS für Verbindungen von Amazon MQ zum LDAP-Server. Informationen zu unterstützten Konfigurationen finden Sie unter. LDAP-Authentifizierung und -Autorisierung

AMQP-Client-Verbindungen

Konfigurieren Sie die TLS-Peer-Verifizierung für AMQP-Clientverbindungen, die von Federation und Shovel verwendet werden. Weitere Informationen finden Sie unter SSL-Konfiguration des AMQP-Clients.

> **Important**
>
> Amazon MQ unterstützt derzeit nicht die Konfiguration von Client-Zertifikaten für AMQP-Clientverbindungen. Aus diesem Grund können Federation and Shovel keine Verbindung zu MTLS-fähigen Brokern herstellen, für die eine Client-Zertifikatsauthentifizierung erforderlich ist.

Konfiguration des Ressourcenlimits

Amazon MQ for RabbitMQ unterstützt die Konfiguration von Broker-Ressourcenlimits ab RabbitMQ 4. Wenn Sie einen Broker erstellen, wendet Amazon MQ automatisch Standardwerte auf diese Ressourcenlimits an. Diese Standardwerte dienen als Schutzmaßnahmen, um die Verfügbarkeit Ihres Brokers zu schützen und gleichzeitig den gängigen Nutzungsmustern der Kunden Rechnung zu tragen. Sie können das Verhalten Ihres Brokers anpassen, indem Sie die Werte für die Konfiguration der Grenzwerte so ändern, dass sie Ihren spezifischen Workload-Anforderungen besser entsprechen. Weitere Informationen zu den zulässigen Standard- und Höchstwerten finden Sie unterthe section called “Richtlinien zur Größenbestimmung”.

Ressourcennamen und Konfigurationsschlüssel

Ressourcenname	Konfigurationsschlüssel
Connection (Verbindung)	`connection_max`
Kanal	`channel_max_per_node`
Warteschlange	`cluster_queue_limit`

Ressourcenname	Konfigurationsschlüssel
Vhost	`vhost_max`
Schaufel	`runtime_parameters.limits.shovel`
Exchange	`cluster_exchange_limit`
Verbraucher pro Kanal	`consumer_max_per_channel`
Maximale Nachrichtengröße	`max_message_size`

Wie überschreibt man Ressourcenlimits

Sie können Ressourcenlimits mithilfe der Amazon MQ MQ-API und der Amazon MQ MQ-Konsole überschreiben.

Das folgende Beispiel zeigt, wie Sie das Standardlimit für die Anzahl der Warteschlangen mithilfe von überschreiben können: AWS CLI

```
aws mq update-configuration --configuration-id <config-id> --data "$(echo
 "cluster_queue_limit=500" | base64 --wrap=0)"
```

Bei einem erfolgreichen Aufruf wird eine Konfigurationsrevision erstellt. Sie müssen die Konfiguration Ihrem RabbitMQ-Broker zuordnen und den Broker neu starten, um die Überschreibung anzuwenden. Weitere Einzelheiten finden Sie unter RabbitMQ Broker Configurations

Fehler beim Überschreiben des Ressourcenlimits

Das Zuordnen oder Erstellen eines Brokers mit Konfigurationswerten außerhalb des unterstützten Bereichs führt zu einer Fehlerantwort, die der folgenden ähnelt:

```
Configuration Revision N for configuration:cluster_queue_limit has limit: of value:
 100000000 larger than maximum allowed limit:5000
```

ARN-Unterstützung in der RabbitMQ-Konfiguration

Amazon MQ for RabbitMQ unterstützt AWS ARNs die Werte einiger RabbitMQ-Konfigurationseinstellungen. Dies wird durch das RabbitMQ-Community-Plugin rabbitmq-aws ermöglicht. Dieses Plugin wurde von Amazon MQ entwickelt und verwaltet und kann auch in selbst gehosteten RabbitMQ-Brokern verwendet werden, die nicht von Amazon MQ verwaltet werden.

⚠ Wichtige Überlegungen

- Die vom aws-Plugin abgerufenen aufgelösten ARN-Werte werden zur Laufzeit direkt an den RabbitMQ-Prozess übergeben. Sie werden nicht an anderer Stelle auf dem RabbitMQ-Knoten gespeichert.
- Amazon MQ for RabbitMQ erfordert eine IAM-Rolle, die von Amazon MQ für den Zugriff auf die Konfiguration übernommen werden kann. ARNs Dies wird durch Einstellung konfiguriert. `aws.arns.assume_role_arn`
- Benutzer, die anrufen CreateBroker oder UpdateBroker APIs über eine Broker-Konfiguration verfügen, die eine IAM-Rolle enthält, müssen über die entsprechenden `iam:PassRole` Berechtigungen verfügen.
- Die IAM-Rolle muss in demselben AWS Konto wie der RabbitMQ-Broker vorhanden sein. Alle ARNs in der Konfiguration enthaltenen Elemente müssen in derselben AWS Region wie der RabbitMQ-Broker vorhanden sein.
- Amazon MQ fügt globale bedingte IAM-Schlüssel hinzu `aws:SourceAccount` und `aws:SourceArn` wenn die IAM-Rolle übernommen wird. Diese Werte müssen in der IAM-Richtlinie verwendet werden, die der Rolle für den Schutz verwirrter Stellvertreter zugewiesen ist.

Auf dieser Seite

Unterstützte Schlüssel

Erforderliche IAM-Rolle

`aws.arns.assume_role_arn`

ARN für die IAM-Rolle, die Amazon MQ für den Zugriff auf andere AWS Ressourcen annimmt. Erforderlich, wenn eine andere ARN-Konfiguration verwendet wird.

AMQP-Endpunkt

Schlüssel zur Konfiguration	Description
`aws.arns.ssl_options.cacertfile`	Zertifizierungsstellendatei für SSL/TLS Client-Verbindungen. Amazon MQ erfordert die Verwendung von Amazon S3 oder die Speicherung des Zertifikats.

RabbitMQ-Verwaltungs-Plugin

Schlüssel zur Konfiguration	Description
`aws.arns.management.ssl.cacertfile`	Zertifizierungsstellendatei für Verbindungen mit Verwaltungsschnittstellen. SSL/TLS Amazon MQ erfordert die Verwendung von Amazon S3 oder die Speicherung des Zertifikats.

RabbitMQ 2.0-Plugin OAuth

Schlüssel zur Konfiguration	Description
`aws.arns.auth_oauth2.https.cacertfile`	Zertifizierungsstellendatei für OAuth 2.0-HTTPS-Verbindungen. Amazon MQ erfordert die Verwendung von Amazon S3 oder die Speicherung des Zertifikats.

RabbitMQ HTTP-Authentifizierungs-Plugin

Schlüssel zur Konfiguration	Description
`aws.arns.auth_http.ssl_options.cacertfile`	Zertifizierungsstellendatei für HTTP-Authentifizierungsverbindungen. SSL/TLS Amazon MQ erfordert die Verwendung von Amazon S3 oder die Speicherung des Zertifikats.
`aws.arns.auth_http.ssl_options.certfile`	Zertifikatsdatei für gegenseitige TLS-Verbindungen zwischen Amazon MQ und dem HTTP-Authentifizierungsserver. Amazon MQ erfordert die Verwendung von Amazon S3 oder die Speicherung des Zertifikats.
`aws.arns.auth_http.ssl_options.keyfile`	Private Schlüsseldatei für gegenseitige TLS-Verbindungen zwischen Amazon MQ und dem HTTP-Authentifizierungsserver. Amazon MQ erfordert die Verwendung AWS Secrets Manager zum Speichern des privaten Schlüssels.

RabbitMQ LDAP-Plugin

Schlüssel zur Konfiguration	Description
`aws.arns.auth_ldap.ssl_options.cacertfile`	Zertifizierungsstellendatei für LDAP-Verbindungen. SSL/TLS Amazon MQ erfordert die Verwendung von Amazon S3 oder die Speicherung des Zertifikats.
`aws.arns.auth_ldap.ssl_options.certfile`	Zertifikatsdatei für gegenseitige TLS-Verbindungen zwischen Amazon MQ und dem LDAP-Server. Amazon MQ erfordert die Verwendung von Amazon S3 oder die Speicherung des Zertifikats.
`aws.arns.auth_ldap.ssl_options.keyfile`	Private Schlüsseldatei für gegenseitige TLS-Verbindungen zwischen Amazon MQ und dem LDAP-Server. Amazon MQ erfordert die Verwendung AWS Secrets Manager zum Speichern des privaten Schlüssels.

Schlüssel zur Konfiguration	Description
`aws.arns.auth_ldap.dn_lookup_bind.password`	Passwort für die LDAP-DN-Suchverbindung. Amazon MQ erfordert die Verwendung AWS Secrets Manager , um das Passwort als Klartextwert zu speichern.
`aws.arns.auth_ldap.other_bind.password`	Passwort für andere LDAP-Verbindungen. Amazon MQ erfordert die Verwendung AWS Secrets Manager , um das Passwort als Klartextwert zu speichern.

Beispiele für IAM-Richtlinien

Beispiele für IAM-Richtlinien, einschließlich Richtliniendokumente für die Übernahme von Rollen und Dokumente zu Rollenrichtlinien, finden Sie in der CDK-Beispielimplementierung.

Anweisungen Verwendung der LDAP-Authentifizierung und -Autorisierung zur Einrichtung AWS Secrets Manager und zu Amazon S3 S3-Ressourcen finden Sie unter.

Bestätigung des Zugriffs

Zur Fehlerbehebung in Szenarien, in denen ARN-Werte nicht abgerufen werden können, unterstützt das aws-Plugin einen RabbitMQ-Verwaltungs-API-Endpunkt, der aufgerufen werden kann, um zu überprüfen, ob Amazon MQ die Rolle erfolgreich übernehmen und lösen kann. AWS ARNs Dadurch entfällt die Notwendigkeit, die Broker-Konfiguration zu aktualisieren, den Broker mit der neuen Konfigurationsrevision zu aktualisieren und den Broker neu zu starten, um Konfigurationsänderungen zu testen.

> **Note**
>
> Für die Verwendung dieser API ist ein vorhandener RabbitMQ-Administratorbenutzer erforderlich. Amazon MQ empfiehlt, zusätzlich zu anderen Zugriffsmethoden Testbroker mit einem internen Benutzer zu erstellen. Weitere Informationen finden Sie unter Aktivieren von OAuth 2.0 und einfacher (interner) Authentifizierung. Dieser Benutzer kann dann für den Zugriff auf die Validierungs-API verwendet werden.

> **Note**
>
> Obwohl das aws-Plugin die Übergabe einer neuen Rolle als Eingabe an die Validierungs-API unterstützt, wird dieser Parameter von Amazon MQ nicht unterstützt. Die für die Validierung verwendete IAM-Rolle sollte dem Wert von `aws.arns.assume_role_arn` in der Broker-Konfiguration entsprechen.

Verwandte Quarantänestatus des Brokers

Informationen zu den Quarantänestatus von Brokern im Zusammenhang mit ARN-Supportproblemen finden Sie unter:

- RABBITMQ_INVALID_ASSUMEROLE
- RABBITMQ_INVALID_ARN_LDAP
- RABBITMQ_UNGÜLTIG_ARN

Beispielszenario

- Der Broker `b-f0fc695e-2f9c-486b-845a-988023a3e55b` wurde so konfiguriert, dass er die IAM-Rolle für den Zugriff auf geheime Daten verwendet `<role>` AWS Secrets Manager `<arn>`
- Wenn die Amazon MQ zur Verfügung gestellte Rolle keine Leseberechtigung für das AWS Secrets Manager Geheimnis hat, wird der folgende Fehler in den RabbitMQ-Protokollen angezeigt:

  ```
  [error] <0.254.0> aws_arn_config: {handle_assume_role,{error,{assume_role_failed,"AWS
   service is unavailable"}}}
  ```

  Darüber hinaus wechselt der Broker in den Quarantänestatus. `INVALID_ASSUMEROLE` Weitere Informationen finden Sie unter INVALID_ASSUMEROLE.
- LDAP-Authentifizierungsversuche schlagen mit dem folgenden Fehler fehl:

  ```
  [error] <0.254.0> LDAP bind failed: invalid_credentials
  ```
- Korrigieren Sie die IAM-Rolle mit den richtigen Berechtigungen
- Rufen Sie den Validierungsendpunkt auf, um zu überprüfen, ob RabbitMQ jetzt auf das Geheimnis zugreifen kann:

```
curl -4su 'guest:guest' -XPUT -H 'content-type: application/json' <broker-endpoint>/
api/aws/arn/validate -d '{"assume_role_arn":"arn:aws:iam::<account-id>:role/<role-
name>","arns":["arn:aws:secretsmanager:<region>:<account-id>:secret:<secret-name>"]}'
 | jq '.'
```

SSL-Konfiguration des AMQP-Clients

Federation und Shovel verwenden AMQP für die Kommunikation zwischen Upstream- und Downstream-Brokern. Standardmäßig ist die TLS-Peer-Verifizierung für AMQP-Clients in Amazon MQ für RabbitMQ 4 aktiviert. Mit dieser Einstellung führen Federation- und Shovel-AMQP-Clients, die auf Amazon MQ-Brokern ausgeführt werden, eine Peer-Verifizierung durch, wenn sie Verbindungen mit dem Upstream-Broker herstellen.

AMQP-Clients, die auf Amazon MQ-Brokern laufen, unterstützen dieselben Zertifizierungsstellen wie Mozilla. Wenn Sie ACM nicht verwenden, verwenden Sie ein Zertifikat, das von einer Zertifizierungsstelle ausgestellt wurde, die in der Mozilla Included CA Certificate List aufgeführt ist. Wenn Ihr lokaler Broker Zertifikate von anderen Zertifizierungsstellen verwendet, schlägt die SSL-Überprüfung fehl. Sie können die TLS-Peer-Verifizierung für diese Anwendungsfälle deaktivieren.

> **Important**
>
> Amazon MQ unterstützt derzeit nicht die Konfiguration von Client-Zertifikaten für AMQP-Clientverbindungen. Aus diesem Grund können Federation and Shovel keine Verbindung zu MTLS-fähigen Brokern herstellen, für die eine Client-Zertifikatsauthentifizierung erforderlich ist.

> **Important**
>
> Auf Amazon MQ für RabbitMQ 3 sind die SSL-Eigenschaften von AMQP-Clients mit den RabbitMQ-Standardeinstellungen (verify_none) konfiguriert. Amazon MQ für RabbitMQ 3 unterstützt das Überschreiben dieser Standardeinstellungen nicht.

> **Note**
>
> Mit der `verify_peer` Standardeinstellung können Sie Federation- und Shovel-Verbindungen zwischen zwei beliebigen Amazon MQ-Brokern einrichten. Dies unterstützt jedoch nicht die Herstellung der Verbindung zwischen Amazon MQ-Brokern und privaten Brokern oder lokalen Brokern, die mit Nicht-Amazon MQ-CA-Zertifikaten betrieben werden. Um eine Verbindung zu privaten oder lokalen Brokern herzustellen, müssen Sie die Peer-Verifizierung auf dem nachgeschalteten Amazon MQ-Broker deaktivieren.

SSL-Konfigurationsschlüssel für den AMQP-Client

Konfiguration	Konfigurationsschlüssel	Unterstützte Werte
SSL-Peer-Überprüfung für AMQP-Clients	`amqp_client.ssl_options.verify`	`verify_none` , `verify_peer`

Wie überschreibt man die SSL-Peer-Verifizierung für den AMQP-Client

Sie können die SSL-Peer-Verifizierung von AMQP-Clients mithilfe der Amazon MQ MQ-API und der Amazon MQ MQ-Konsole auf RabbitMQ 4-Brokern außer Kraft setzen.

Das folgende Beispiel zeigt, wie Sie die SSL-Peer-Verifizierung des AMQP-Clients überschreiben können, indem Sie: AWS CLI

```
aws mq update-configuration --configuration-id <config-id> --data "$(echo
 "amqp_client.ssl_options.verify=verify_none" | base64 --wrap=0)"
```

Bei einem erfolgreichen Aufruf wird eine Konfigurationsrevision erstellt. Sie müssen die Konfiguration Ihrem RabbitMQ-Broker zuordnen und den Broker neu starten, um die Überschreibung anzuwenden. Weitere Einzelheiten finden Sie unter Creating and applying broker configurations

> **Important**
>
> Bei der Verwendung `verify_none` ist die SSL-Verschlüsselung immer noch aktiv, aber die Identität des Peers wird nicht verifiziert. Verwenden Sie diese Einstellung nur bei Bedarf und stellen Sie sicher, dass Sie dem Netzwerkpfad zum Zielbroker vertrauen.

Amazon MQ für RabbitMQ — Authentifizierung und Autorisierung

Amazon MQ for RabbitMQ unterstützt die folgenden Authentifizierungs- und Autorisierungsmethoden:

Einfache Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer intern im RabbitMQ-Broker gespeichert und über die Webkonsole oder die Management-API verwaltet. Berechtigungen für Vhosts, Exchanges, Warteschlangen und Themen werden direkt in RabbitMQ konfiguriert. Dies ist die Standardmethode. Weitere Informationen finden Sie unter Einfache Authentifizierung und Autorisierung.

OAuth 2.0 Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer und ihre Berechtigungen von einem externen OAuth 2.0-Identitätsanbieter (IdP) verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen für Vhosts, Exchanges, Warteschlangen und Themen werden über das Scope-System des OAuth 2.0-Anbieters zentralisiert. Dies vereinfacht die Benutzerverwaltung und ermöglicht die Integration in bestehende Identitätssysteme. Weitere Informationen finden Sie unter Authentifizierung und Autorisierung OAuth 2.0.

IAM-Authentifizierung und -Autorisierung

Bei dieser Methode authentifizieren sich Broker-Benutzer mithilfe von AWS IAM-Anmeldeinformationen über den IAM-Outbound-Federation. IAM-Anmeldeinformationen werden verwendet, um JWT-Token vom AWS Security Token Service (STS) abzurufen, und diese JWT-Token dienen als 2.0-Token für die Authentifizierung. OAuth Diese Methode nutzt die bestehende OAuth 2.0-Unterstützung in Amazon MQ für RabbitMQ, wo sie als 2.0-Identitätsanbieter AWS fungiert. OAuth Die Benutzerauthentifizierung wird von AWS IAM abgewickelt, während die Ressourcenberechtigungen für Vhosts, Exchanges, Warteschlangen und Themen über in RabbitMQ konfigurierte IAM-Richtlinien und Bereichsaliase verwaltet werden. Weitere Informationen finden Sie unter IAM-Authentifizierung und -Autorisierung.

LDAP-Authentifizierung und -Autorisierung

Bei dieser Methode werden Broker-Benutzer und ihre Berechtigungen von einem externen LDAP-Verzeichnisdienst verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen werden über den LDAP-Server zentralisiert, sodass Benutzer mit ihren vorhandenen Verzeichnisdienstanmeldedaten auf RabbitMQ zugreifen können. Weitere Informationen finden Sie unter LDAP-Authentifizierung und -Autorisierung.

HTTP-Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer und ihre Berechtigungen von einem externen HTTP-Server verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen werden über den HTTP-Server zentralisiert, sodass Benutzer über ihren eigenen Authentifizierungs- und Autorisierungsanbieter auf RabbitMQ zugreifen können. Weitere Informationen zu dieser Methode finden Sie unter HTTP-Authentifizierung und Autorisierung.

Authentifizierung mit SSL-Zertifikaten

Amazon MQ unterstützt Mutual TLS (mTLS) für RabbitMQ-Broker. Das SSL-Authentifizierungs-Plugin verwendet Client-Zertifikate von mTLS-Verbindungen, um Benutzer zu authentifizieren. Bei dieser Methode werden Broker-Benutzer mithilfe von X.509-Clientzertifikaten anstelle von Benutzernamen und Kennwörtern authentifiziert. Das Zertifikat des Clients wird anhand einer vertrauenswürdigen Zertifizierungsstelle (CA) validiert, und der Benutzername wird aus einem Feld im Zertifikat extrahiert, z. B. dem Common Name (CN) oder dem Subject Alternative Name (SAN). Diese Methode bietet eine starke Authentifizierung, ohne dass Anmeldeinformationen über das Netzwerk übertragen werden müssen. Weitere Informationen finden Sie unter SSL-Zertifikatsauthentifizierung.

> **Note**
>
> RabbitMQ unterstützt mehrere Authentifizierungs- und Autorisierungsmethoden, die gleichzeitig verwendet werden können. Sie können beispielsweise sowohl OAuth 2.0 als auch die einfache (interne) Authentifizierung aktivieren. Weitere Informationen finden Sie im OAuth 2.0-Tutorial-Abschnitt zur Aktivierung sowohl der OAuth 2.0-Authentifizierung als auch der einfachen (internen) Authentifizierung sowie in der Dokumentation zur RabbitMQ-Zugriffskontrolle.
>
> Amazon MQ empfiehlt, beim Testen von Authentifizierungskonfigurationen einen internen Benutzer zu erstellen. Auf diese Weise kann die Zugriffskonfiguration mithilfe der RabbitMQ-Management-API validiert werden. Weitere Informationen finden Sie unter Zugriffsvalidierung.

Einfache Authentifizierung und Autorisierung

Amazon MQ für RabbitMQ-Broker-Benutzer

> **Note**
>
> In diesem Thema wird die Verwaltung von Broker-Benutzern mit dem standardmäßigen internen Authentifizierungs- und Autorisierungsmechanismus von RabbitMQ beschrieben. Informationen zu allen unterstützten Authentifizierungs- und Autorisierungsmethoden finden Sie unter Amazon MQ for RabbitMQ Authentication and Authorization.

Jeder AMQP 0-9-1-Client-Verbindung ist ein Benutzer zugeordnet. Dieser Benutzer muss authentifiziert sein. Jede Client-Verbindung zielt auch auf einen virtuellen Host (vhost) ab. Der Benutzer muss über eine Reihe von Berechtigungen für diesen Vhost verfügen. Ein Benutzer kann die Berechtigung haben, Warteschlangen und Exchanges in einem Vhost zu konfigurieren,schreiben, und zu lesen. Sie geben Benutzeranmeldedaten und den Ziel-Vhost an, wenn die Verbindung hergestellt wird.

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 Management-API oder die RabbitMQ-Webkonsole hinzufügen und verwalten. Sie können auch die RabbitMQ-Webkonsole oder die Management-API verwenden, um Benutzerberechtigungen und Tags festzulegen oder zu ändern.

> **Note**
>
> RabbitMQ-Benutzer werden nicht über Amazon MQ BenutzerAPI gespeichert oder angezeigt.

> **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“.

Um einen neuen Benutzer mit der RabbitMQ-Management-API zu erstellen, verwenden Sie den folgenden API-Endpunkt und den folgenden Anforderungstext. Ersetzen Sie *username* und *password* durch Ihre neuen Anmeldeinformationen.

```
PUT /api/users/username HTTP/1.1

    {"password":"password","tags":"administrator"}
```

Important

- Fügen Sie keine persönlich identifizierbare Informationen (PII) oder andere vertrauliche oder sensible Informationen in Broker-Benutzernamen hinzu. Broker-Benutzernamen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.
- Wenn Sie den Zugriff auf alle Administratorkonten verlieren, finden Sie unter Brokerzugriff wiederherstellen Informationen zur Verwendung der IAM-Authentifizierung für die Wiederherstellung.

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 Anforderungstext verwenden. Ersetzen Sie *vhost* und *username* durch Ihre Daten. 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-Management-API finden Sie unter HTTP-basierte RabbitMQ-Management-API.

OAuth 2.0 Authentifizierung und Autorisierung für Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ unterstützt mehrere Authentifizierungs- und Autorisierungsmethoden. Informationen zu allen unterstützten Methoden finden Sie unter Authentifizierung und Autorisierung für Amazon MQ für RabbitMQ-Broker.

Bei der OAuth 2.0-Authentifizierung und -Autorisierung werden Broker-Benutzer und ihre Berechtigungen von einem externen OAuth 2.0-Identitätsanbieter (IdP) verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen für Vhosts, Exchanges, Warteschlangen und Themen werden über das Scope-System des OAuth 2.0-Anbieters zentralisiert. Dies vereinfacht die Benutzerverwaltung und ermöglicht die Integration in bestehende Identitätssysteme.

⚠ Wichtige Überlegungen

- OAuth 2.0-Integration wird auf Amazon MQ für ActiveMQ-Broker nicht unterstützt.
- Amazon MQ for RabbitMQ unterstützt kein Serverzertifikat, das von einer privaten Zertifizierungsstelle ausgestellt wurde.
- Das RabbitMQ OAuth 2.0-Plugin unterstützt keine Token-Introspektion-Endpunkte und undurchsichtige Zugriffstoken. Es führt auch keine Token-Widerrufsprüfungen durch.
- Sie müssen die IAM-Berechtigung angeben`mq:UpdateBrokerAccessConfiguration`, um OAuth 2.0 auf vorhandenen Brokern zu aktivieren.
- Amazon MQ erstellt automatisch einen Systembenutzer `monitoring-AWS-OWNED-DO-NOT-DELETE` mit nur Überwachungsberechtigungen. Dieser Benutzer verwendet das interne Authentifizierungssystem von RabbitMQ auch auf OAuth 2.0-fähigen Brokern und ist auf den Zugriff auf die Loopback-Schnittstelle beschränkt.

Informationen zur Konfiguration von OAuth 2.0 für Ihre Amazon MQ for RabbitMQ-Broker finden Sie unter. Verwendung der 2.0-Authentifizierung OAuth und -Autorisierung

Auf dieser Seite

- Zusätzliche Validierungen für die 2.0-Authentifizierung OAuth

Unterstützte 2.0-Konfigurationen OAuth

Amazon MQ for RabbitMQ unterstützt alle konfigurierbaren Variablen im RabbitMQ OAuth 2.0-Plugin, mit den folgenden Ausnahmen:

- `auth_oauth2.https.cacertfile`
- `auth_oauth2.oauth_providers.{id/index}.https.cacertfile`
- `management.oauth_client_secret`

  Da Amazon MQ diesen Schlüssel nicht unterstützt, unterstützen wir UAA nicht als IdP.
- `management.oauth_resource_servers.{id/index}.oauth_client_secret`
- `auth_oauth2.signing_keys.{id/index}`

Zusätzliche Validierungen für die 2.0-Authentifizierung OAuth

Amazon MQ erzwingt außerdem die folgenden zusätzlichen Validierungen für die 2.0-Authentifizierung: OAuth

- Alles URLs muss damit beginnen. `https://`
- Unterstützte Signaturalgorithmen: `Ed25519` `Ed25519ph` `Ed448Ed448ph`,`EdDSA`,`ES256K`,`ES256`,,`ES384`,`ES512`,`HS256`,`HS384`,`HS512`,`PS256`,`PS384`,`PS512`,`RS256` und`RS512`.

IAM-Authentifizierung und Autorisierung für Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ unterstützt mehrere Authentifizierungs- und Autorisierungsmethoden. Informationen zu allen unterstützten Methoden finden Sie unter Authentifizierung und Autorisierung für Amazon MQ für RabbitMQ-Broker.

Die IAM-Authentifizierung und -Autorisierung ermöglicht Broker-Benutzern die Authentifizierung mithilfe von IAM-Anmeldeinformationen über den AWS IAM-Outbound-Federation. Bei dieser Methode werden IAM-Anmeldeinformationen verwendet, um JWT-Token vom AWS Security Token Service (STS) abzurufen. Diese JWT-Token dienen als OAuth 2.0-Token für die Authentifizierung und nutzen die bestehende OAuth 2.0-Unterstützung in Amazon MQ für RabbitMQ, wo sie als 2.0-

Identitätsanbieter AWS fungiert. OAuth AWS IAM kümmert sich um die Benutzerauthentifizierung, während die Ressourcenberechtigungen für virtuelle Hosts, Exchanges, Warteschlangen und Themen über IAM-Richtlinien und Bereichsaliase verwaltet werden, die in RabbitMQ konfiguriert sind.

⚠ Wichtige Überlegungen

- Die IAM-Authentifizierung wird auf den RabbitMQ-Versionen 3.13, 4.2 und höher unterstützt. Es wird auf Amazon MQ für ActiveMQ-Broker nicht unterstützt.
- Für die IAM-Authentifizierung muss der ausgehende IAM-Verbund konfiguriert sein und in Ihrem Konto verfügbar sein. AWS
- Diese Methode baut auf der vorhandenen OAuth 2.0-Infrastruktur in Amazon MQ für RabbitMQ auf und AWS dient als 2.0-Identitätsanbieter. OAuth
- Amazon MQ erstellt automatisch einen Systembenutzer `monitoring-AWS-OWNED-DO-NOT-DELETE` mit nur Überwachungsberechtigungen. Dieser Benutzer verwendet das interne Authentifizierungssystem von RabbitMQ auch bei IAM-fähigen Brokern und ist auf den Zugriff auf die Loopback-Schnittstelle beschränkt.

Auf dieser Seite

Wie funktioniert die IAM-Authentifizierung

Die IAM-Authentifizierung für Amazon MQ for RabbitMQ verwendet den IAM-Outbound-Verbund, um IAM-Anmeldeinformationen für die Authentifizierung bei RabbitMQ-Brokern zu aktivieren AWS . IAM-Anmeldeinformationen werden verwendet, um JWT-Token vom AWS Security Token Service (STS) zu erhalten, und diese JWT-Token dienen als 2.0-Token für die Authentifizierung beim RabbitMQ-Broker. OAuth

Einschränkungen

Die IAM-Authentifizierung für Amazon MQ for RabbitMQ hat die folgende Einschränkung:

- Konfiguration von Bereichsansprüchen — Sie können einen Bereichsanspruch nicht direkt verwenden, da das JWT-Token von STS verschachtelt ist. Der Schlüssel ist`sts.amazonaws.com`,

dass die Verwendung von Bereichsaliasen in der RabbitMQ-Konfiguration erforderlich ist, um IAM-Rollen RabbitMQ-Berechtigungen zuzuordnen. Diese Einschränkung verhindert auch, dass IAM-Richtlinien für die Autorisierung vollständig verwendet werden, sodass stattdessen eine RabbitMQ-Konfiguration für die Autorisierung erforderlich ist.

Informationen zur Konfiguration der IAM-Authentifizierung und -Autorisierung für Ihre Amazon MQ for RabbitMQ-Broker finden Sie unter. Verwendung der IAM-Authentifizierung und -Autorisierung

HTTP-Authentifizierung und Autorisierung für Amazon MQ für RabbitMQ

Amazon MQ for RabbitMQ unterstützt die Authentifizierung und Autorisierung von Broker-Benutzern über einen externen HTTP-Server. Weitere unterstützte Methoden finden Sie unter Authentifizierung und Autorisierung für Amazon MQ für RabbitMQ-Broker.

> **Note**
>
> Das HTTP-Authentifizierungs-Plugin ist nur für Amazon MQ für RabbitMQ Version 4 und höher verfügbar.

> **Wichtige Überlegungen**
>
> - Der HTTP-Server muss über das öffentliche Internet zugänglich sein. Amazon MQ for RabbitMQ kann so konfiguriert werden, dass es sich beim HTTP-Server mithilfe von Mutual TLS authentifiziert.
> - Amazon MQ for RabbitMQ erzwingt die Verwendung von AWS ARNs für Einstellungen, die Zugriff auf das lokale Dateisystem erfordern. Weitere Informationen finden Sie unter ARN-Unterstützung in der RabbitMQ-Konfiguration.
> - Sie müssen die IAM-Berechtigung angeben`mq:UpdateBrokerAccessConfiguration`, um die HTTP-Authentifizierung auf vorhandenen Brokern zu aktivieren.
> - Amazon MQ erstellt automatisch einen Systembenutzer `monitoring-AWS-OWNED-DO-NOT-DELETE` mit nur Überwachungsberechtigungen. Dieser Benutzer verwendet das interne Authentifizierungssystem von RabbitMQ auch bei HTTP-fähigen Brokern und ist auf den Zugriff auf die Loopback-Schnittstelle beschränkt. Amazon MQ verhindert das Löschen dieses Benutzers, indem es das geschützte Benutzer-Tag hinzufügt.

Informationen zur Konfiguration der HTTP-Authentifizierung für Ihre Amazon MQ for RabbitMQ-Broker finden Sie unter. Verwendung der HTTP-Authentifizierung und -Autorisierung

Auf dieser Seite

Unterstützte HTTP-Konfigurationen

Amazon MQ for RabbitMQ unterstützt alle konfigurierbaren Variablen im RabbitMQ HTTP-Authentifizierungs-Plugin, mit den folgenden Ausnahmen, die erforderlich sind. AWS ARNs Einzelheiten zur ARN-Unterstützung finden Sie unter ARN-Unterstützung in der RabbitMQ-Konfiguration.

Konfigurationen, die Folgendes erfordern ARNs

`auth_http.ssl_options.cacertfile`

Stattdessen `aws.arns.auth_http.ssl_options.cacertfile` verwenden

`auth_http.ssl_options.certfile`

Stattdessen `aws.arns.auth_http.ssl_options.certfile` verwenden

`auth_http.ssl_options.keyfile`

Stattdessen `aws.arns.auth_http.ssl_options.keyfile` verwenden

Nicht unterstützte SSL-Optionen

Die folgenden SSL-Konfigurationsoptionen werden ebenfalls nicht unterstützt:

Vollständige Liste anzeigen

- `auth_http.ssl_options.cert`
- `auth_http.ssl_options.client_renegotiation`
- `auth_http.ssl_options.dh`
- `auth_http.ssl_options.dhfile`
- `auth_http.ssl_options.honor_cipher_order`

- `auth_http.ssl_options.honor_ecc_order`
- `auth_http.ssl_options.key.RSAPrivateKey`
- `auth_http.ssl_options.key.DSAPrivateKey`
- `auth_http.ssl_options.key.PrivateKeyInfo`
- `auth_http.ssl_options.log_alert`
- `auth_http.ssl_options.password`
- `auth_http.ssl_options.psk_identity`
- `auth_http.ssl_options.reuse_sessions`
- `auth_http.ssl_options.secure_renegotiate`
- `auth_http.ssl_options.versions.$version`
- `auth_http.ssl_options.sni`
- `auth_http.ssl_options.crl_check`

Zusätzliche Validierungen für HTTP-Konfigurationen in Amazon MQ

Amazon MQ erzwingt außerdem die folgenden zusätzlichen Validierungen für die HTTP-Authentifizierung und -Autorisierung:

- `auth_http.http_method`muss entweder oder sein `get` `post`
- Die folgenden Pfadkonfigurationen müssen HTTPS verwenden URLs:
  - `auth_http.user_path`
  - `auth_http.vhost_path`
  - `auth_http.resource_path`
  - `auth_http.topic_path`
- Falls eine Einstellung die Verwendung eines AWS ARN erfordert, `aws.arns.assume_role_arn` muss dieser angegeben werden.

SSL-Zertifikatsauthentifizierung für Amazon MQ für RabbitMQ

Amazon MQ for RabbitMQ unterstützt die Authentifizierung von Broker-Benutzern mithilfe von X.509-Client-Zertifikaten. Weitere unterstützte Methoden finden Sie unter Authentifizierung und Autorisierung für Amazon MQ für RabbitMQ-Broker.

> **Note**
>
> Das SSL-Zertifikat-Authentifizierungs-Plugin ist nur für Amazon MQ für RabbitMQ Version 4 und höher verfügbar.

> **Wichtige Überlegungen**
>
> - Client-Zertifikate müssen von einer vertrauenswürdigen Zertifizierungsstelle (CA) signiert werden. Amazon MQ for RabbitMQ validiert die Zertifikatskette während der Authentifizierung.
> - Amazon MQ for RabbitMQ erzwingt die Verwendung von AWS ARNs für zertifikatsbezogene Einstellungen wie CA-Zertifikate und für Einstellungen, die Zugriff auf das lokale Dateisystem erfordern. Weitere Informationen finden Sie unter ARN-Unterstützung in der RabbitMQ-Konfiguration.
> - Amazon MQ erstellt automatisch einen Systembenutzer `monitoring-AWS-OWNED-DO-NOT-DELETE` mit nur Überwachungsberechtigungen. Dieser Benutzer verwendet das interne Authentifizierungssystem von RabbitMQ auch bei Brokern, für die SSL-Zertifikate aktiviert sind, und ist auf den Zugriff auf die Loopback-Schnittstelle beschränkt. Amazon MQ verhindert das Löschen dieses Benutzers, indem es das geschützte Benutzer-Tag hinzufügt.

Informationen zur Konfiguration der SSL-Zertifikatsauthentifizierung für Ihre Amazon MQ for RabbitMQ Broker finden Sie unter. Verwendung der SSL-Zertifikatsauthentifizierung

Auf dieser Seite

Unterstützte SSL-Konfigurationen

Amazon MQ für RabbitMQ unterstützt die SSL/TLS Konfiguration von Client-Verbindungen. Einzelheiten zur ARN-Unterstützung finden Sie unter ARN-Unterstützung in der RabbitMQ-Konfiguration.

Konfigurationen, die Folgendes erfordern ARNs

`ssl_options.cacertfile`

Stattdessen `aws.arns.ssl_options.cacertfile` verwenden

Anmeldekonfigurationen für SSL-Zertifikate

Die folgenden Konfigurationen steuern, wie Benutzernamen aus Client-Zertifikaten extrahiert werden:

`ssl_cert_login_from`

Gibt an, welches Zertifikatsfeld für die Benutzernamenextraktion verwendet werden soll. Unterstützte Werte:

- `distinguished_name`- Verwenden Sie den vollständigen Distinguished Name
- `common_name`- Verwenden Sie das Feld Common Name (CN)
- `subject_alternative_name`oder `subject_alt_name` — Verwenden Sie den alternativen Namen des Betreffs

`ssl_cert_login_san_type`

Gibt bei Verwendung des alternativen Antragstellers den SAN-Typ an. Unterstützte Werte: `dnsip`,`email`,`uri`, `other_name`

`ssl_cert_login_san_index`

Gibt bei Verwendung von Subject Alternative Name den Index des zu verwendenden SAN-Eintrags an (auf Null basierend). Muss eine nicht negative Ganzzahl sein.

SSL-Optionen für Client-Verbindungen

Die folgenden SSL-Optionen gelten für Client-Verbindungen:

`ssl_options.verify`

Modus „Peer-Verifizierung“. Unterstützte Werte:`verify_none`, `verify_peer`

`ssl_options.fail_if_no_peer_cert`

Ob Verbindungen abgelehnt werden sollen, wenn der Client kein Zertifikat bereitstellt. Boolescher Wert:

`ssl_options.depth`

Maximale Tiefe der Zertifikatskette für die Überprüfung.

`ssl_options.hostname_verification`

Modus zur Überprüfung des Hostnamens. Unterstützte Werte:`wildcard`, `none`

Nicht unterstützte SSL-Optionen

Die folgenden SSL-Konfigurationsoptionen werden nicht unterstützt:

Vollständige Liste anzeigen

- `ssl_options.cert`
- `ssl_options.client_renegotiation`
- `ssl_options.dh`
- `ssl_options.dhfile`
- `ssl_options.honor_cipher_order`
- `ssl_options.honor_ecc_order`
- `ssl_options.key.RSAPrivateKey`
- `ssl_options.key.DSAPrivateKey`
- `ssl_options.key.PrivateKeyInfo`
- `ssl_options.log_alert`
- `ssl_options.password`
- `ssl_options.psk_identity`
- `ssl_options.reuse_sessions`
- `ssl_options.secure_renegotiate`
- `ssl_options.versions.$version`
- `ssl_options.sni`
- `ssl_options.crl_check`

Zusätzliche Validierungen für SSL-Konfigurationen in Amazon MQ

Amazon MQ erzwingt außerdem die folgenden zusätzlichen Validierungen für die SSL-Zertifikatsauthentifizierung:

- Falls eine Einstellung die Verwendung eines AWS ARN erfordert, `aws.arns.assume_role_arn` muss dieser angegeben werden.

LDAP-Authentifizierung und Autorisierung für Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ unterstützt die Authentifizierung und Autorisierung von Broker-Benutzern mithilfe eines externen LDAP-Servers. Weitere unterstützte Methoden finden Sie unter Authentifizierung und Autorisierung für Amazon MQ für RabbitMQ-Broker.

⚠ Wichtige Überlegungen

- Der LDAP-Server muss über das öffentliche Internet zugänglich sein. Amazon MQ for RabbitMQ kann so konfiguriert werden, dass es sich mithilfe von Mutual TLS beim LDAP-Server authentifiziert.
- Amazon MQ for RabbitMQ erzwingt die Verwendung von AWS ARNs für sensible LDAP-Einstellungen wie Passwörter und für Einstellungen, die Zugriff auf das lokale Dateisystem erfordern. Weitere Informationen finden Sie unter ARN-Unterstützung in der RabbitMQ-Konfiguration.
- Sie müssen die IAM-Berechtigung angeben, um LDAP auf `mq:UpdateBrokerAccessConfiguration` vorhandenen Brokern zu aktivieren.
- Amazon MQ erstellt automatisch einen Systembenutzer `monitoring-AWS-OWNED-DO-NOT-DELETE` mit nur Überwachungsberechtigungen. Dieser Benutzer verwendet das interne Authentifizierungssystem von RabbitMQ auch bei LDAP-fähigen Brokern und ist auf den Zugriff auf die Loopback-Schnittstelle beschränkt. Amazon MQ verhindert das Löschen dieses Benutzers, indem es das geschützte Benutzer-Tag hinzufügt.

Informationen zur Konfiguration von LDAP für Ihre Amazon MQ for RabbitMQ-Broker finden Sie unter. Verwendung der LDAP-Authentifizierung und -Autorisierung

Auf dieser Seite

Unterstützte LDAP-Konfigurationen

Amazon MQ for RabbitMQ unterstützt alle konfigurierbaren Variablen im RabbitMQ LDAP-Plugin, mit den folgenden Ausnahmen, die erforderlich sind. AWS ARNs Einzelheiten zur ARN-Unterstützung finden Sie unter ARN-Unterstützung in der RabbitMQ-Konfiguration.

Konfigurationen, die Folgendes erfordern ARNs

`auth_ldap.dn_lookup_bind.password`

Stattdessen `aws.arns.auth_ldap.dn_lookup_bind.password` verwenden

`auth_ldap.other_bind.password`

Stattdessen `aws.arns.auth_ldap.other_bind.password` verwenden

`auth_ldap.ssl_options.cacertfile`

Stattdessen `aws.arns.auth_ldap.ssl_options.cacertfile` verwenden

`auth_ldap.ssl_options.certfile`

Stattdessen `aws.arns.auth_ldap.ssl_options.certfile` verwenden

`auth_ldap.ssl_options.keyfile`

Stattdessen `aws.arns.auth_ldap.ssl_options.keyfile` verwenden

SSL-Optionen werden nicht unterstützt

Die folgenden SSL-Konfigurationsoptionen werden ebenfalls nicht unterstützt:

Vollständige Liste anzeigen

- `auth_ldap.ssl_options.cert`
- `auth_ldap.ssl_options.client_renegotiation`
- `auth_ldap.ssl_options.dh`
- `auth_ldap.ssl_options.dhfile`
- `auth_ldap.ssl_options.honor_cipher_order`
- `auth_ldap.ssl_options.honor_ecc_order`
- `auth_ldap.ssl_options.key.RSAPrivateKey`

- `auth_ldap.ssl_options.key.DSAPrivateKey`
- `auth_ldap.ssl_options.key.PrivateKeyInfo`
- `auth_ldap.ssl_options.log_alert`
- `auth_ldap.ssl_options.password`
- `auth_ldap.ssl_options.psk_identity`
- `auth_ldap.ssl_options.reuse_sessions`
- `auth_ldap.ssl_options.secure_renegotiate`
- `auth_ldap.ssl_options.versions.$version`
- `auth_ldap.ssl_options.sni`

Zusätzliche Validierungen für LDAP-Konfigurationen in Amazon MQ

Amazon MQ erzwingt außerdem die folgenden zusätzlichen Validierungen für die LDAP-Authentifizierung und -Autorisierung:

- `auth_ldap.log`kann nicht gesetzt werden auf `network_unsafe`
- Der LDAP-Server muss LDAPS verwenden. Entweder `auth_ldap.use_ssl` oder `auth_ldap.use_starttls` muss explizit aktiviert werden
- Falls eine Einstellung die Verwendung eines AWS ARN erfordert, `aws.arns.assume_role_arn` muss dieser angegeben werden.
- `auth_ldap.servers`muss eine gültige IP-Adresse oder ein gültiger FQDN sein
- Bei den folgenden Schlüsseln muss es sich um einen gültigen LDAP-Distinguished Name handeln:
  - `auth_ldap.dn_lookup_base`
  - `auth_ldap.dn_lookup_bind.user_dn`
  - `auth_ldap.other_bind.user_dn`
  - `auth_ldap.group_lookup_base`

Plugins

Amazon MQ für RabbitMQ unterstützt auch die folgenden Plugins.

- RabbitMQ-Verwaltungs-Plugin
- Schaufel-Plugin

- Föderations-Plugin
- Konsistentes Hash-Austausch-Plugin
- OAuth Plug-in 2
- LDAP-Plugin
- HTTP-Plugin
- SSL-Zertifikats-Plugin
- aws-Plugin
- JMS Topic Exchange-Plugin

RabbitMQ-Verwaltungs-Plugin

Amazon MQ for RabbitMQ unterstützt das RabbitMQ-Verwaltungs-Plugin, das eine HTTP-basierte Verwaltungs-API zusammen mit einer browserbasierten Benutzeroberfläche für die RabbitMQ-Webkonsole bereitstellt. Sie können die Webkonsole und die Management-API zum Erstellen und Verwalten von Broker-Benutzern und -Richtlinien verwenden.

Shovel Plugin

Amazon MQ for RabbitMQ unterstützt das RabbitMQ Shovel-Plugin, mit dem Sie Nachrichten von Warteschlangen und Börsen auf einem Broker auf einen anderen verschieben können. Sie können Shovel verwenden, um lose gekoppelte Broker zu verbinden und Nachrichten von Knoten mit schwereren Nachrichtenladungen zu verteilen.

> **Important**
>
> Sie können die Shovel zwischen Warteschlangen oder Exchanges nicht konfigurieren, wenn das Shovel-Ziel ein privater Broker ist.
> Amazon MQ unterstützt die Verwendung statischer Shoveln nicht.

Es werden nur dynamische Schaufeln unterstützt. Dynamische Schaufeln werden mithilfe von Laufzeitparametern konfiguriert und können jederzeit programmgesteuert über eine Client-Verbindung gestartet und gestoppt werden. Mithilfe der RabbitMQ-Management-API können Sie beispielsweise eine PUT-Anfrage an den folgenden API-Endpunkt erstellen, um eine dynamische Schaufel zu konfigurieren. In diesem Beispiel kann {vhost} durch den Namen des Vhosts des Brokers und {name} durch den Namen der neuen dynamischen Schaufel ersetzt werden.

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

Im Anforderungstext müssen Sie entweder eine Warteschlange oder einen Exchange angeben, aber nicht beides. Im folgenden Beispiel wird eine dynamische Schaufel zwischen einer in src-queue angegebenen lokalen Warteschlange und einer in dest-queue definierten Remote-Warteschlange konfiguriert. In ähnlicher Weise können Sie die Parameter src-exchange und dest-exchange verwenden, um einen 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-
west2.amazonaws.com:5671",
"dest-queue": "destination-queue-name"
}
}
```

Federation Plugin

Amazon MQ unterstützt föderierte Börsen und Warteschlangen mithilfe des RabbitMQ-Verbund-Plug-ins. 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 Peers 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 Downstream-Brokern mithilfe der RabbitMQ-Webkonsole oder der Management-API aktivieren.

> **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.

Sie können z. B. mithilfe der Management-API den Verbund konfigurieren, indem Sie Folgendes tun:

- Konfigurieren Sie einen oder mehrere Upstreams, die Verbundverbindungen zu anderen Knoten definieren. Sie können Verbundverbindungen mithilfe der RabbitMQ-Webkonsole oder der Management-API definieren. Mithilfe der Verwaltungs-API können Sie eine POST-Anfrage an/api/parameters/federation-upstream/%2f/myupstreammit 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 Management-API konfigurieren. Mithilfe der Verwaltungs-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 Hauptteil der Anfrage geht davon aus, dass die Namen der Exchanges auf dem Server mit amq beginnen. Durch die Verwendung des regulären Ausdrucks ^amq\. wird sichergestellt, dass der Verbund für alle Exchanges aktiviert ist, deren Namen mit „amq“ beginnen. Die Exchanges auf Ihrem RabbitMQ-Server können unterschiedlich benannt werden.

Consistent Hash Exchange Plugin

Amazon MQ für RabbitMQ unterstützt das RabbitMQ Consistent Hash Exchange Type-Plugin. 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.

OAuth 2.0-Plug-In

Amazon MQ für RabbitMQ unterstützt das OAuth 2-Authentifizierungs-Backend-Plugin. Dieses Plugin ist abhängig von Ihrer Broker-Konfiguration bedingt aktiviert. Wenn dieses Plugin aktiviert ist, bietet es OAuth 2.0-Authentifizierung und -Autorisierung mit Integration in externe OAuth 2.0-Identitätsanbieter für eine zentrale Benutzerverwaltung und Zugriffskontrolle. Weitere Informationen zur OAuth 2.0-Authentifizierung finden Sie unterOAuth 2.0 Authentifizierung und Autorisierung.

LDAP-Plugin

Amazon MQ for RabbitMQ unterstützt das LDAP-Authentifizierungs-Backend-Plugin. Dieses Plugin ist abhängig von Ihrer Broker-Konfiguration bedingt aktiviert. Wenn dieses Plugin aktiviert ist, bietet es LDAP-Authentifizierung und -Autorisierung mit Integration in externe LDAP-Verzeichnisdienste für eine zentrale Benutzerauthentifizierung und -autorisierung. Weitere Informationen zur LDAP-Authentifizierung finden Sie unter. LDAP-Authentifizierung und -Autorisierung

HTTP-Plugin

Amazon MQ for RabbitMQ unterstützt das HTTP-Authentifizierungs-Backend-Plugin. Dieses Plugin ist abhängig von Ihrer Broker-Konfiguration bedingt aktiviert. Wenn dieses Plugin aktiviert ist, bietet es HTTP-Authentifizierung und -Autorisierung mit Integration in externe HTTP-Server für eine zentralisierte Benutzerauthentifizierung und -autorisierung. Weitere Hinweise zur HTTP-Authentifizierung finden Sie unterHTTP-Authentifizierung und Autorisierung.

> **Note**
>
> Das HTTP-Authentifizierungs-Plugin ist nur für Amazon MQ für RabbitMQ Version 4 und höher verfügbar.

SSL-Zertifikats-Plugin

Amazon MQ unterstützt Mutual TLS (mTLS) für RabbitMQ-Broker. Das SSL-Authentifizierungs-Plugin verwendet Client-Zertifikate von mTLS-Verbindungen, um Benutzer zu authentifizieren. Dieses Plugin ist abhängig von Ihrer Broker-Konfiguration bedingt aktiviert. Wenn es aktiviert ist, ermöglicht es eine zertifikatsbasierte Authentifizierung mithilfe von X.509-Client-Zertifikaten für eine starke Authentifizierung, ohne dass Anmeldeinformationen über das Netzwerk übertragen werden müssen. Weitere Informationen zur SSL-Zertifikatsauthentifizierung finden Sie unter. Authentifizierung mit SSL-Zertifikaten

> **Note**
>
> Das SSL-Zertifikat-Authentifizierungs-Plugin ist nur für Amazon MQ für RabbitMQ Version 4 und höher verfügbar.

aws-Plugin

Das aws-Plugin wird von Amazon MQ für RabbitMQ basierend auf Ihrer Broker-Konfiguration bedingt aktiviert. Dieses von Amazon MQ entwickelte und verwaltete Community-Plugin ermöglicht den sicheren Abruf von Anmeldeinformationen und Zertifikaten von AWS Diensten, die die AWS ARNs RabbitMQ-Konfigurationseinstellungen verwenden. Weitere Informationen zur ARN-Unterstützung finden Sie unterARN support in RabbitMQ configuration.

JMS Topic Exchange-Plugin

Das JMS Topic Exchange Plugin wird immer von Amazon MQ für RabbitMQ aktiviert. Es arbeitet mit dem RabbitMQ JMS-Client zusammen, sodass neue und bestehende JMS-Anwendungen eine Verbindung zu Amazon MQ für RabbitMQ herstellen können.

> **Note**
>
> Das JMS Topic Exchange-Plugin ist nur für Amazon MQ für RabbitMQ Version 4 und höher verfügbar. Es ist standardmäßig aktiviert, wird aber nur aktiviert, wenn der RabbitMQ JMS-Client zur Ausführung von JMS-Workloads verwendet wird.

Unterstützte Protokolle

Sie können auf Ihre RabbitMQ-Broker zugreifen, indem Sie jede Programmiersprache verwenden, die RabbitMQ unterstützt, und indem Sie TLS für eine der folgenden Protokollspezifikationen aktivieren:

- AMQP (0-9-1)
- AMQP 1.0
- JMS 1.1
- JMS 2.0
- JMS 3.1

Unterstützung für Amazon MQ für RabbitMQ JMS

Sie können jetzt JMS 1.1-, 2.0- und 3.1-Workloads auf Amazon MQ für RabbitMQ 4 mit dem RabbitMQ JMS-Client ausführen.

RabbitMQ JMS-Client

Der RabbitMQ JMS-Client ist eine Open-Source-JMS-Client-Bibliothek, die Sie benötigen, um Ihre JMS-Anwendung mit Amazon MQ RabbitMQ-Brokern zu verbinden. Weitere Informationen GitHub finden Sie im offiziellen Repository.

Unterstützt JMS 1.1, 2.0 und 3.1 APIs

Ab Amazon MQ für RabbitMQ 4 ist das Plugin immer aktiviert. `jms-topic-exchange` Daher können Sie Amazon MQ für RabbitMQ 4 und den RabbitMQ JMS-Client für Ihre JMS-Arbeitslast verwenden. Alle in JMS 1.1 definierten APIs JMS werden unterstützt, mit Ausnahme von:

- Serversitzungen APIs werden nicht unterstützt.
- XA-Transaktionen APIs werden nicht unterstützt.
- Der JMS-Selektor für das JMS-Warteschlangenziel wird nicht unterstützt.
- Das `NoLocal` JMS-Abonnementattribut wird nicht unterstützt.

Alle neu APIs in JMS 2.0 und JMS 3.1 hinzugefügten Dateien werden unterstützt, mit Ausnahme von:

- `JMSProducer.setDeliveryDelay`API wird nicht unterstützt.

Weitere Informationen zum Verbinden Ihrer JMS-Anwendung mit Amazon MQ für RabbitMQ Broker finden Sie im Tutorial zum Verbinden Ihrer JMS-Anwendung mit Amazon MQ für RabbitMQ Broker

Authentifizierung und Autorisierung

Alle in diesem Abschnitt aufgeführten Authentifizierungs- und Autorisierungsmechanismen werden unterstützt. Die Anmeldeinformationen, die für die Verbindung mit dem Broker über den JMS-Client verwendet werden, sind dieselben, als ob Sie mit einem AMQP-Java-Client eine Verbindung zum RabbitMQ-Broker herstellen würden.

Interoperabilität mit AMQP-Warteschlangen auf RabbitMQ

Sie können den RabbitMQ JMS-Client verwenden, um JMS-Nachrichten an einen AMQP-Exchange zu senden und Nachrichten aus einer AMQP-Warteschlange zu verarbeiten (diese Funktion unterstützt keine JMS-Themen). Auf diese Weise können Sie zusammenarbeiten oder bestimmte JMS-Workloads zu AMQP-Workloads migrieren. Weitere Informationen finden Sie in der offiziellen Client-Dokumentation.

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

In den Engine-Versionen 3.13 und niedriger von Amazon MQ für RabbitMQ lautet die aktuelle Standard-Betreiberrichtlinie:

```
vhost name pattern apply-to definition priority/
 default_operator_policy_AWS_managed .* classic_queues {"ha-mode":"all","ha-
sync-mode":"automatic","queue-version":2} 0
```

In den Versionen 4.0 und höher wurde die standardmäßige Betreiberrichtlinie wie folgt geändert:

```
vhost name pattern apply-to definition priority/
 default_operator_policy_AWS_managed .* classic_queues {"queue-version":2} 0
```

Diese Änderung ist erforderlich, da die klassische Warteschlangenspiegelung und HA-Richtlinieneinstellungen in RabbitMQ 4 nicht unterstützt werden.
Sie können keine Richtlinie erstellen, die sowohl für klassische gespiegelte Warteschlangen als auch für Quorumwarteschlangen 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` einer Quorumwarteschlangenrichtlinie erstellen.

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 (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.

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. Klicken Sie auf der Seite mit den Broker-Details in -VerbindungenWählen Sie im Abschnitt die OptionRabbitMQ Webkonsole-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`*** *`number-of-messages`*Ersetzen Sie ihn durch den von Amazon MQ empfohlenen Wert entsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers, z. B. **`8000000`** für einen `mq.m7g.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.

So wenden Sie Standardrichtlinien und virtuelle Host-Limits mithilfe der RabbitMQ-Verwaltungs-API 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-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 Standardeinstellung/vhost, der als`%2F` encodiert ist. Um die Richtlinie auf einen anderen Vhost anzuwenden, ersetzen Sie`%2F`durch den Vhost-Namen.

   > **Note**
   >
   > Ersetzen Sie *`username`* und *`password`* durch Ihre Administrator-Anmeldedaten. *`number-of-messages`*Ersetzen Sie ihn durch den von Amazon MQ empfohlenen Wert entsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers. *`policy-name`*Ersetzen Sie es durch einen Namen für Ihre Richtlinie. *`broker-endpoint`*Ersetzen Sie es durch die URL, die Sie sich 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 Sie *username* und *password* durch Ihre Administrator-Anmeldedaten. *max-connections*Ersetzen Sie ihn durch den von Amazon MQ empfohlenen Wert entsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers. 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 '{"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
```

Quorum-Warteschlangen für RabbitMQ auf Amazon MQ

Quorumwarteschlangen 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:

- vorübergehende Warteschlangen verwenden
- haben lange Warteschlangenrückstände
- niedrige Latenz priorisieren

Um eine Quorum-Warteschlange 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 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 Dokumentation zu RabbitMQ-Quorumwarteschlangen. 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 Quorum-Warteschlangen 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 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 Quorum-Warteschlange 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. Konfiguration von Betreiberrichtlinien

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 mithilfe des Arguments `delivery-limit` policy ein Limit für die Nachrichtenzustellung festlegen, 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.

Best Practices für Amazon MQ for RabbitMQ

Folgen Sie diesen Richtlinien zur Produktionsreife, um die Broker-Leistung zu maximieren und die Effizienz des Nachrichtendurchsatzes zu optimieren, wenn Sie mit Amazon MQ for RabbitMQ Brokern arbeiten.

> **Important**
>
> Derzeit unterstützt Amazon MQ keine Streams oder die Verwendung der strukturierten Protokollierung in JSON, die in RabbitMQ 3.9.x eingeführt wurde.

Themen

Bewährte Methoden für die Brokereinrichtung und das Verbindungsmanagement in Amazon MQ für RabbitMQ

Broker-Setup und Verbindungsmanagement sind der erste Schritt, um Probleme mit dem Broker-Nachrichtendurchsatz, der Ressourcennutzung und der Fähigkeit, Produktionsworkloads zu bewältigen, zu vermeiden. Beachten Sie bei der Erstellung und Konfiguration eines Amazon MQ for RabbitMQ-Brokers die folgenden bewährten Methoden zur Auswahl geeigneter Instance-Typen, zur effizienten Verwaltung von Verbindungen und zur Konfiguration des Vorabrufs von Nachrichten, um die Leistung Ihres Brokers zu maximieren.

> **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“.

Schritt 1: Verwenden Sie Cluster-Bereitstellungen

Für Produktionsworkloads empfehlen wir die Verwendung von Clusterbereitstellungen anstelle von Einzelinstanz-Brokern, um eine hohe Verfügbarkeit und Nachrichtenstabilität zu gewährleisten. Clusterbereitstellungen beseitigen einzelne Fehlerquellen und bieten eine bessere Fehlertoleranz.

Clusterbereitstellungen bestehen aus drei RabbitMQ-Broker-Knoten, die auf drei Availability Zones verteilt sind. Sie bieten einen automatischen Failover und stellen sicher, dass der Betrieb auch dann fortgesetzt wird, wenn eine gesamte Availability Zone nicht verfügbar ist. Amazon MQ repliziert Nachrichten automatisch auf allen Knoten, um die Verfügbarkeit bei Knotenausfällen oder Wartungsarbeiten sicherzustellen.

Cluster-Bereitstellungen sind für Produktionsumgebungen unerlässlich und werden durch das Amazon MQ Service Level Agreement unterstützt.

Weitere Informationen finden Sie unter Cluster-Bereitstellung in Amazon MQ für RabbitMQ.

Schritt 2: Wählen Sie den richtigen Broker-Instance-Typ

Der Nachrichtendurchsatz eines Broker-Instance-Typs hängt von Ihrem Anwendungsfall ab. `M7g.medium`sollte nur zum Testen der Anwendungsleistung verwendet werden. Wenn Sie diese kleinere Instanz verwenden, bevor Sie größere Instanzen in der Produktion verwenden, kann dies die Anwendungsleistung verbessern. Bei Instance-Typen `m7g.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 in den Größenrichtlinien von Amazon MQ für RabbitMQ.

Schritt 3: Verwenden Sie Quorum-Warteschlangen

Quorum-Warteschlangen sollten bei Cluster-Bereitstellung die Standardwahl für replizierte Warteschlangentypen in Produktionsumgebungen für RabbitMQ-Broker auf Version 3.13 und höher sein. Quorum-Warteschlangen sind moderne replizierte Warteschlangenarten, die eine hohe Zuverlässigkeit, einen hohen Durchsatz und eine stabile Latenz bieten.

Quorum-Warteschlangen verwenden den Raft-Konsensusalgorithmus, um eine bessere Fehlertoleranz zu gewährleisten. Wenn der Leader-Knoten nicht mehr verfügbar ist, wählen die Quorum-Warteschlangen mit Stimmenmehrheit automatisch einen neuen Leiter, sodass die Nachrichtenzustellung mit minimaler Unterbrechung fortgesetzt wird. Da sich jeder Knoten in einer anderen Availability Zone befindet, bleibt Ihr Messaging-System auch dann verfügbar, wenn eine gesamte Availability Zone vorübergehend nicht verfügbar ist.

Um eine Quorum-Warteschlange zu deklarieren, setzen Sie den Header `x-queue-type` `quorum` beim Erstellen Ihrer Warteschlangen auf.

Weitere Informationen zu Quorum-Warteschlangen, einschließlich Migrationsstrategien und Best Practices, finden Sie unter Quorum-Warteschlangen in Amazon MQ für RabbitMQ.

Schritt 4: 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, für jeden Prozess eine Verbindung und anschließend für jeden Thread einen Kanal zu verwenden. Vermeiden Sie eine übermäßige Kanalnutzung, um Kanallecks zu vermeiden.

Bewährte Methoden für die Beständigkeit und Zuverlässigkeit von Nachrichten in Amazon MQ für RabbitMQ

Bevor Sie Ihre Anwendung in die Produktionsumgebung überführen, sollten Sie sich an die folgenden bewährten Methoden halten, um Nachrichtenverlust und Ressourcenüberlastung zu verhindern.

Schritt 1: Verwenden Sie persistente Nachrichten und dauerhafte Warteschlangen

Dauerhafte Nachrichten können dazu beitragen, die Datenbeständigkeit in Situationen zu schützen, in denen ein Broker abstürzt oder neu startet. 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 kennzeichnen, indem Sie den Übermittlungsmodus „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());
```

Schritt 2: Konfigurieren Sie die Bestätigung durch den Herausgeber und die Empfangsbestätigung für Endverbraucher

Der Vorgang der Bestätigung, dass eine Nachricht an den Broker gesendet wurde, wird als Bestätigung durch den Herausgeber bezeichnet. Durch Bestätigungen des Herausgebers wird Ihre Anwendung darüber informiert, wann Nachrichten zuverlässig gespeichert wurden. Mithilfe von Bestätigungen durch den Herausgeber können Sie auch kontrollieren, wie viele Nachrichten auf dem Broker gespeichert werden. Ohne Bestätigung durch den Herausgeber gibt es keine Bestätigung dafür, dass eine Nachricht erfolgreich verarbeitet wurde, und Ihr Broker kann Nachrichten löschen, die er nicht verarbeiten kann.

Wenn eine Kundenanwendung eine Bestätigung über die Zustellung und den Empfang von Nachrichten an den Broker zurücksendet, wird dies auch als Empfangsbestätigung für Verbraucher bezeichnet. Sowohl Bestätigung als auch Bestätigung sind für die Gewährleistung der Datensicherheit bei der Zusammenarbeit mit RabbitMQ-Brokern unerlässlich.

Die Bestätigung der Verbraucherzustellung wird in der Regel in der Clientanwendung konfiguriert. Bei der Arbeit mit AMQP 0-9-1 kann die Bestätigung durch Konfiguration der Methode aktiviert werden. `basic.consume` AMQP 0-9-1-Clients können auch Herausgeberbestätigungen 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.

Sie können die Konfiguration so konfigurieren`consumer_timeout`, dass erkannt wird, wenn Verbraucher Lieferungen nicht bestätigen. Wenn der Kunde innerhalb des Timeout-Werts keine Empfangsbestätigung sendet, wird der Kanal geschlossen und Sie erhalten eine. `PRECONDITION_FAILED` Um den Fehler zu diagnostizieren, verwenden Sie die UpdateConfigurationAPI, um den Wert zu erhöhen. `consumer_timeout`

Schritt 3: Halten Sie die Warteschlangen kurz

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 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 Speichernutzung überprüfen, indem Sie die Metriken der Knoten `RabbitMemUsed` und des `RabbitMqMemLimit` Brokerknotens 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 Richtlinien für automatisches Löschen und TTL definieren, um den Ressourcenverbrauch proaktiv zu reduzieren und die Anzahl der Benutzer auf ein NACKs Minimum zu reduzieren. Das Warteschleifen von Nachrichten auf dem Broker ist CPU-intensiv, sodass eine hohe Anzahl von Nachrichten die Leistung des Brokers beeinträchtigen kann. NACKs

Bewährte Methoden für Leistungsoptimierung und Effizienz in Amazon MQ für RabbitMQ

Sie können die Leistung Ihres Amazon MQ for RabbitMQ-Brokers optimieren, indem Sie den Durchsatz maximieren, die Latenz minimieren und eine effiziente Ressourcennutzung sicherstellen. Gehen Sie wie folgt vor, um die Leistung Ihrer Anwendung zu optimieren.

Schritt 1: Halten Sie die Nachrichtengröße unter 1 MB

Wir empfehlen, Nachrichten unter 1 Megabyte (MB) zu halten, um optimale Leistung und Zuverlässigkeit zu gewährleisten.

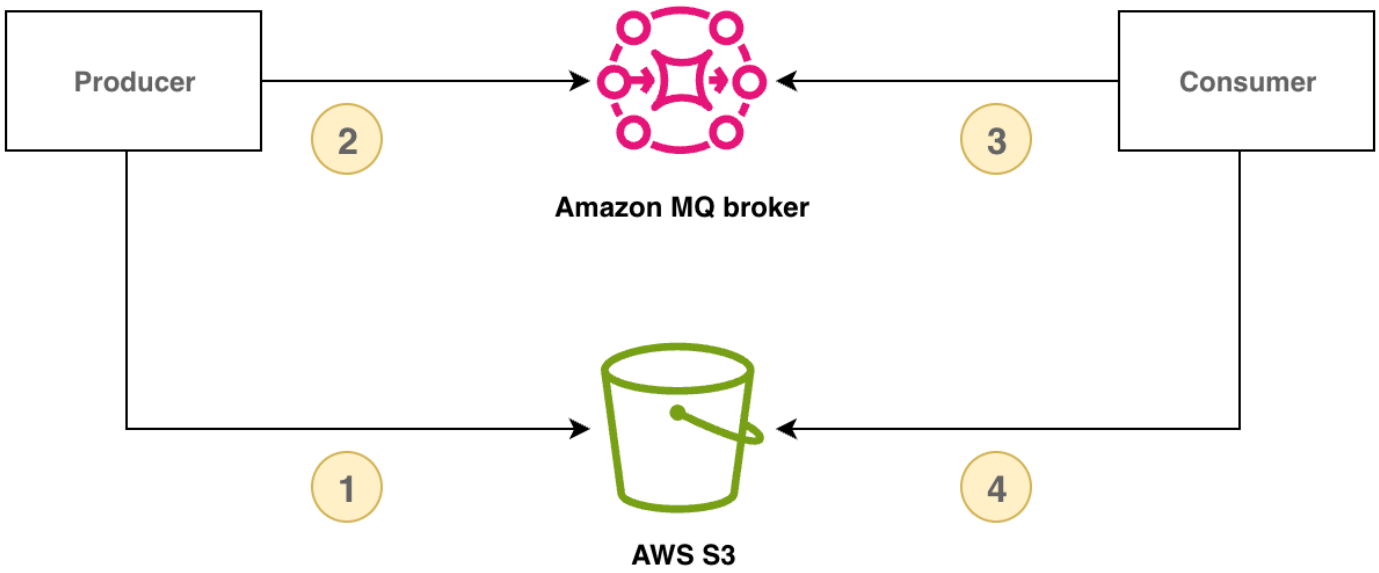
RabbitMQ 3.13 unterstützt standardmäßig Nachrichtengrößen von bis zu 128 MB, aber große Nachrichten können unvorhersehbare Speicheralarme auslösen, die die Veröffentlichung blockieren und bei der knotenübergreifenden Replikation von Nachrichten potenziell zu hohem Speicherdruck führen. Zu große Nachrichten können sich auch auf die Neustart- und Wiederherstellungsprozesse des Brokers auswirken, was das Risiko für die Servicekontinuität erhöht und zu Leistungseinbußen führen kann.

Speichern und Abrufen großer Payloads mithilfe des Schemas für die Schadensüberprüfung

Um große Nachrichten zu verwalten, können Sie das Muster der Anspruchsprüfung implementieren, indem Sie die Nachrichtennutzdaten in einem externen Speicher speichern und nur die Nutzlast-

Referenz-ID über RabbitMQ senden. Der Verbraucher verwendet die Nutzlast-Referenz-ID, um die umfangreiche Nachricht abzurufen und zu verarbeiten.

Das folgende Diagramm zeigt, wie Amazon MQ für RabbitMQ und Amazon S3 zur Implementierung des Antragsprüfungsmusters verwendet wird.



Das folgende Beispiel demonstriert dieses Muster mit Amazon MQ, dem AWS SDK for Java 2.x und Amazon S3:

1. Definieren Sie zunächst eine Message-Klasse, die die Amazon S3 S3-Referenz-ID enthalten soll.

```
class Message {
    // Other data fields of the message...

    public String s3Key;
    public String s3Bucket;
}
```

2. Erstellen Sie eine Publisher-Methode, die die Payload in Amazon S3 speichert und eine Referenznachricht über RabbitMQ sendet.

```
public void publishPayload() {
    // Store the payload in S3.
    String payload = PAYLOAD;
    String prefix = S3_KEY_PREFIX;
    String s3Key = prefix + "/" + UUID.randomUUID();
    s3Client.putObject(PutObjectRequest.builder()
```

```
            .bucket(S3_BUCKET).key(s3Key).build(),
            RequestBody.fromString(payload));

    // Send the reference through RabbitMQ.
    Message message = new Message();
    message.s3Key = s3Key;
    message.s3Bucket = S3_BUCKET;
    // Assign values to other fields in your message instance.

    publishMessage(message);
}
```

3. Implementieren Sie eine Consumer-Methode, die die Payload von Amazon S3 abruft, die Payload verarbeitet und das Amazon S3 S3-Objekt löscht.

```
public void consumeMessage(Message message) {
    // Retrieve the payload from S3.
    String payload = s3Client.getObjectAsBytes(GetObjectRequest.builder()
        .bucket(message.s3Bucket).key(message.s3Key).build())
        .asUtf8String();

    // Process the complete message.
    processPayload(message, payload);

    // Delete the S3 object.
    s3Client.deleteObject(DeleteObjectRequest.builder()
        .bucket(message.s3Bucket).key(message.s3Key).build());
}
```

Schritt 2: Nutzung und langlebige Verbraucher **`basic.consume`**

Die Verwendung `basic.consume` bei einem langlebigen Verbraucher ist effizienter als die Abfrage einzelner Nachrichten mithilfe von. `basic.get` Weitere Informationen finden Sie unter Abfragen einzelner Nachrichten.

Schritt 3: Konfigurieren Sie den Vorabruf

Sie können den RabbitMQ-Prefetch-Wert verwenden, um zu optimieren, wie Ihre Verbraucher Nachrichten konsumieren. RabbitMQ implementiert den Channel-Prefetch-Mechanismus, der von AMQP 0-9-1 bereitgestellt wird, indem die Prefetch-Anzahl auf Verbraucher im Gegensatz zu Kanälen 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.

Schritt 4: Verwenden Sie Celery 5.5 oder höher mit Quorum-Warteschlangen

Python Celery, ein verteiltes Aufgabenwarteschlangensystem, kann bei hoher Aufgabenlast viele unkritische Meldungen generieren. Diese zusätzliche Broker-Aktivität kann die Nichtverfügbarkeit des Brokers auslösen the section called “ `RABBITMQ_MEMORY_ALARM` ” und dazu führen. Gehen Sie wie folgt vor, um die Wahrscheinlichkeit zu verringern, dass ein Speicheralarm ausgelöst wird:

Für alle Celery-Versionen

1. Schalten Sie `task_create_missing_queues`es aus, um die Abwanderung in der Warteschlange zu verringern.
2. Schalten Sie es dann aus, `worker_enable_remote_control` um die dynamische Erstellung von `celery@...pidbox` Warteschlangen zu beenden. Dadurch wird die Abwanderung von Warteschlangen auf dem Broker reduziert.

   ```
   worker_enable_remote_control = false
   ```

3. Um die Aktivität unkritischer Nachrichten weiter zu reduzieren, schalten Sie Celery aus, worker-send-task-eventsindem Sie beim Starten Ihrer Celery-Anwendung Celery entweder nicht einschließen `-E` oder `--task-events` kennzeichnen.
4. Starten Sie Ihre Cellery-Anwendung mit den folgenden Parametern:

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

Für Celery Versionen 5.5 und höher

1. Führen Sie ein Upgrade auf Celery Version 5.5 durch, die Mindestversion, die Quorum-Warteschlangen unterstützt, oder auf eine neuere Version. Um zu überprüfen, welche Version von Celery Sie verwenden, verwenden Sie. `celery --version` Weitere Informationen zu Quorumwarteschlangen finden Sie unter. the section called “Quorum-Warteschlangen”
2. Nach dem Upgrade auf Celery 5.5 oder höher konfigurieren Sie die Konfiguration `task_default_queue_type` auf „Quorum“.
3. Anschließend müssen Sie in den Broker-Transportoptionen auch „Bestätigungen veröffentlichen“ aktivieren:

```
broker_transport_options = {"confirm_publish": True}
```

Bewährte Methoden für Netzwerkstabilität und Überwachung in Amazon MQ für RabbitMQ

Netzwerkstabilität und Überwachung von Broker-Metriken sind für die Aufrechterhaltung zuverlässiger Messaging-Anwendungen unerlässlich. Führen Sie die folgenden bewährten Methoden durch, um automatische Wiederherstellungsmechanismen und Strategien zur Ressourcenüberwachung zu implementieren.

Schritt 1: 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 unbehandelte Ausnahme in der I/O Verbindungsschleife ausgelöst wird, wenn ein Timeout für den Socket-Lesevorgang erkannt wird oder wenn der Server einen Heartbeat 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.

Schritt 2: Überwachen Sie die Metriken und Alarme von Brokern

Wir empfehlen, die CloudWatch Kennzahlen und Alarme für Ihren Amazon MQ for RabbitMQ-Broker regelmäßig zu überwachen, um potenzielle Probleme zu identifizieren und zu beheben, bevor sie sich auf Ihre Messaging-Anwendung auswirken. Eine proaktive Überwachung ist für die Aufrechterhaltung einer stabilen Messaging-Anwendung und die Sicherstellung einer optimalen Leistung unerlässlich.

Amazon MQ for RabbitMQ veröffentlicht dazu Kennzahlen, die Einblicke in CloudWatch die Leistung des Brokers, die Ressourcennutzung und den Nachrichtenfluss bieten. Zu den wichtigsten zu überwachenden Kennzahlen gehören die Speicherauslastung und die Festplattennutzung. Sie können CloudWatch Alarme einrichten, wenn Ihr Broker an Ressourcengrenzen stößt oder Leistungseinbußen auftreten.

Überwachen Sie die folgenden wichtigen Kennzahlen:

`RabbitMQMemUsed` und **`RabbitMQMemLimit`**

Überwachen Sie die Speichernutzung, um Speicheralarme zu verhindern, die die Nachrichtenveröffentlichung blockieren könnten.

`RabbitMQDiskFree` und **`RabbitMQDiskFreeLimit`**

Überwachen Sie die Festplattennutzung, um Speicherplatzprobleme zu vermeiden, die zu Brokerausfällen führen können.

Überwachen Sie bei Clusterbereitstellungen auch knotenspezifische Messwerte, um knotenspezifische Probleme zu identifizieren.

> **Note**
>
> Weitere Informationen dazu, wie Sie einen Alarm bei hohem Speicherverbrauch verhindern können, finden Sie unter Alarme bei hohem Speicherbedarf beheben und verhindern.

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
- Reduzierung der Anzahl der Verbindungen und Kanäle
- Schritt 2: Connect eine JVM-basierte Anwendung mit Ihrem Broker
- Schritt 3: (Optional) Connect zu einer AWS Lambda Funktion herstellen
- Verwenden der OAuth 2.0-Authentifizierung und -Autorisierung für Amazon MQ für RabbitMQ
- Verwenden der IAM-Authentifizierung und -Autorisierung für Amazon MQ für RabbitMQ
- Verwenden der LDAP-Authentifizierung und -Autorisierung für Amazon MQ für RabbitMQ
- Verwendung der HTTP-Authentifizierung und -Autorisierung für Amazon MQ für RabbitMQ
- Verwendung der SSL-Zertifikatsauthentifizierung für Amazon MQ für RabbitMQ
- Verwendung von mTLS für AMQP- und Management-Endpunkte
- Ihre JMS-Anwendung verbinden

Bearbeiten von Broker-Einstellungen

Sie können Ihre Broker-Einstellungen bearbeiten, z. B. die Aktivierung oder Deaktivierung CloudWatch von Protokollen mithilfe von. AWS-Managementkonsole

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 klicken Sie dann auf Bearbeiten.
3. Wählen Sie auf der *MyBroker* Seite Bearbeiten im Abschnitt Spezifikationen eine Broker-Engine-Version oder einen Broker-Instance-Typ aus.
4. Klicken Sie im Abschnitt CloudWatch Protokolle 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 zu veröffentlichen. CloudWatch 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 Maintenance (Wartung) den Wartungszeitplan für Ihren Broker:

   Um den Broker auf neue Versionen zu aktualisieren, sobald diese AWS veröffentlicht werden, wählen Sie Automatische Upgrades für Nebenversionen aktivieren. Automatische Upgrades werden während der-Wartungsfensterdefiniert durch den Wochentag, die Tageszeit (im 24-Stunden-Format) und die Zeitzone (standardmäßig UTC).
6. Wählen Sie Schedule modifications (Änderungen planen).

   > **Note**
   >
   > Wenn Sie nur Enable automatic minor versions (Automatische kleinere Aktualisierungen aktivieren) wählen, wechselt die Schaltfläche zu Save (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 zum Senden von Nachrichten an die Standard-Exchange des Brokers und das Einrichten eines Verbrauchers für den Empfang von Nachrichten aus der Warteschlange.

Themen

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. In der folgenden Tabelle

werden die erforderlichen Mindestberechtigungen als Muster für reguläre Ausdrücke (Regexp) beschrieben.

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>**— Die AWS Region, in der Sie Ihren Amazon MQ for RabbitMQ Broker erstellt haben. 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 empfängt.

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 der Herausgeber, den Sie im vorherigen Schritt erstellt haben, `BasicMessageReceiver` erbt `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 `BasicMessageReceiver`-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 Speichernutzung überprüfen, indem Sie die Node-Metriken `RabbitMemUsed` und die `RabbitMqMemLimit` Broker-Node-Metriken unter vergleichen 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`*. Wählen Sie Zahl aus der Drop-down-Liste aus.

        > **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 *`username`* und *`password`* durch die Anmeldeinformationen Ihres Broker-Administrators. Möglicherweise müssen Sie den Wert von `ha-sync-batch-size` (*`100`*) anhand der Anzahl und Größe der nicht synchronisierten Nachrichten in Ihren Warteschlangen anpassen und kalibrieren. 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 Zeile Richtlinie sollte in Ihrer Warteschlange der Name der neuen Richtlinie aufgeführt sein, die Sie erstellt haben (z. B.). `batch-size-policy`
3. Um den Synchronisierungsvorgang mit einer reduzierten Batchgröße neu zu starten, brechen Sie zunächst die Warteschlangensynchronisierung ab. Starten Sie dann die Warteschlangensynchronisierung neu.

> **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 anhand der Amazon-Metrik überwachen, wie viel Speicher Ihre RabbitMQ-Knoten verwenden. CloudWatch `RabbitMQMemUsed` Sie können auch die `RabbitMQMemLimit`-Metrik, um das Speicherlimit eines Knotens zu überwachen. Weitere Informationen erhalten 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 zur Bearbeitung von Broker-Einstellungen finden Sie unter. Bearbeiten 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 unterhttps://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html.

Zugehörige Ressourcen

- UpdateBrokerInput— Verwenden Sie diese Broker-Eigenschaft, um einen Broker-Instance-Typ mithilfe der Amazon MQ 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.

Reduzierung der Anzahl der Verbindungen und Kanäle

Verbindungen zu Ihrem RabbitMQ on Amazon MQ Broker können entweder durch Ihre Client-Anwendungen oder durch manuelles Schließen über die RabbitMQ-Webkonsole geschlossen werden. Gehen Sie wie folgt vor, um eine Verbindung über die RabbitMQ-Webkonsole zu schließen:

1. Melden Sie sich bei der AWS-Managementkonsole RabbitMQ-Webkonsole Ihres Brokers an 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. RabbitMQ auf Amazon MQ 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.

Für Broker ohne öffentlichen Zugriff können Sie Verbindungen vorübergehend blockieren, indem Sie eingehenden Datenverkehr auf dem entsprechenden Nachrichtenprotokoll-Port verweigern, z. B. Port `5671` für AMQP-Verbindungen. 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 Hinzufügen von Regeln zu einer Sicherheitsgruppe im Amazon-VPC-Benutzerhandbuch.

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-Clientbibliotheken finden Sie unter RabbitMQ-Clientbibliotheken und Entwicklertools.

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 der VPC-Attribute

Um sicherzustellen, dass Ihr Broker innerhalb Ihrer VPC zugänglich ist, müssen Sie die`enableDnsHostnames`und`enableDnsSupport`VPC Attribute Weitere Informationen finden Sie unter DNS-Support in Ihrer VPC im Amazon-VPC-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. Notieren Sie sich auf der ***`MyBroker`***Seite im Abschnitt Verbindungen die Adressen und Ports der Webkonsolen-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 Security Groups (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. In dem Dialogfeld Edit inbound rules (Bearbeiten von Regeln für eingehenden Datenverkehr), fügen Sie eine Regel für jede URL oder jeden Endpunkt hinzu, auf den Sie öffentlich zugreifen möchten (im folgenden Beispiel wird gezeigt, wie Sie dies für eine Broker-Webkonsole tun).

   a. Klicken Sie auf Add Rule (Regel hinzufügen).
   b. Wählen Sie für Type (Typ) Custom TCP (Benutzerdefiniertes TCP).
   c. FürSourceverlassenBenutzerdefiniertund geben Sie dann die IP-Adresse des Systems ein, auf das auf die Webkonsole zugegriffen werden soll (z. B.`192.0.2.1`).
   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 das 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 Top-Level-Paket mit `Connection` und `Channel` API-Klassen, die eine AMQP-0-9-1-Verbindung bzw. einen Kanal darstellen. 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 verwendet, um das AMQP`Builder`-Klasse zum Erstellen eines Nachrichteneigenschaftenobjekts mit dem Inhaltstyp`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 API Guide.

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 Ereignisquellen-Zuweisung, die Sie erstellen, liest Nachrichten von Ihrem Broker in Batches und wandelt sie in eine Lambda -Payload in Form eines JSON-Objekts um.

So verbinden Sie Ihren Broker mit einer Lambda Funktion

1. Fügen Sie die folgenden IAM-Rollenberechtigungen zu der Ausführungsrolle Ihrer Lambda-Funktion hinzu.

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

- Verwalter von Geheimnissen: GetSecretValue

> **Note**
>
> Ohne die erforderlichen IAM-Berechtigungen ist Ihre Funktion nicht in der Lage, Datensätze aus Amazon MQ Ressourcen erfolgreich zu 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 unterInternet- und Servicezugriff für VPC-verbundene FunktionenimAWS Lambda Entwicklerhandbuch.
   - Erstellen Sie mithilfe eines VPC-Endpunkts eine Verbindung zwischen Ihrer Amazon Virtual Private Cloud (Amazon VPC) und Lambda. Ihre Amazon VPC muss auch eine Verbindung zu AWS -Security-Token-Service (AWS STS) und Secrets Manager Manager-Endpunkten herstellen. Weitere Informationen finden Sie unterKonfigurieren von Schnittstellen-VPC-Endpunkten für LambdaimAWS Lambda Entwicklerhandbuch.

3. Konfigurieren Sie Ihren Broker als EreignisquelleVerwendung für eine Lambda -Funktion unter Verwendung der AWS-Managementkonsole. 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 zur Verbindung von Amazon MQ mit Lambda, zu den Optionen, die Lambda für eine Amazon MQ-Ereignisquelle unterstützt, und zu Fehlern bei der Zuordnung von Ereignisquellen finden Sie unter Using Lambda with Amazon MQ im Developer Guide.AWS Lambda

Verwenden der OAuth 2.0-Authentifizierung und -Autorisierung für Amazon MQ für RabbitMQ

In diesem Tutorial wird beschrieben, wie Sie die OAuth 2.0-Authentifizierung für Ihre Amazon MQ for RabbitMQ-Broker mit Amazon Cognito als 2.0-Anbieter konfigurieren. OAuth

> **Note**
>
> Amazon Cognito ist in China (Peking) und China (Ningxia) nicht verfügbar.

> **Important**
>
> Dieses Tutorial ist spezifisch für Amazon Cognito, Sie können jedoch auch andere Identitätsanbieter (IdPs) verwenden. Weitere Informationen finden Sie unter OAuth 2.0-Authentifizierungsbeispiele.

Auf dieser Seite

Voraussetzungen für die Konfiguration der OAuth 2.0-Authentifizierung

Sie können die in diesem Tutorial benötigten Amazon Cognito-Ressourcen festlegen, indem Sie das AWS CDK Stack-Plug-In Amazon Cognito Stack für OAuth RabbitMQ 2 bereitstellen. Wenn Sie Amazon Cognito manuell einrichten, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen, bevor Sie OAuth 2.0 auf Ihrem Amazon MQ für RabbitMQ-Broker konfigurieren:

Voraussetzungen für die Einrichtung von Amazon Cognito

- Richten Sie einen Amazon Cognito Cognito-Endpunkt ein, indem Sie einen Benutzerpool erstellen. Lesen Sie dazu den Blog mit dem Titel How to use OAuth 2.0 in Amazon Cognito: Learn about the different OAuth 2.0-Stipendien.
- Erstellen Sie einen Ressourcenserver, der `rabbitmq` im Benutzerpool aufgerufen wird und für den die folgenden Bereiche definiert sind:`read:all`, `write:allconfigure:all`, und. `tag:administrator` Diese Bereiche werden mit RabbitMQ-Berechtigungen verknüpft.

  Informationen zum Erstellen eines Ressourcenservers finden Sie unter Definieren eines Ressourcenservers für Ihren Benutzerpool (AWS-Managementkonsole) im Amazon Cognito Developer Guide.
- Erstellen Sie die folgenden Anwendungsclients:
  - Anwendungsclient für den Benutzerpool des Typs`Machine-to-Machine application`. Dies ist ein vertraulicher Client mit einem geheimen Client-Schlüssel, der für RabbitMQ AMQP-Clients verwendet wird. Weitere Informationen zu Anwendungsclients und deren Erstellung finden Sie unter App-Clienttypen und Einen App-Client erstellen.
  - Anwendungsclient für den Benutzerpool des Typs`Single-page application`. Dies ist ein öffentlicher Client, der verwendet wird, um Benutzer an der RabbitMQ-Managementkonsole anzumelden. Sie müssen diesen Anwendungsclient so aktualisieren, dass er den Endpunkt des Amazon MQ for RabbitMQ-Brokers enthält, den Sie im folgenden Verfahren als zulässige Callback-URL erstellen. Weitere Informationen finden Sie unter Verwaltete Anmeldung mit der Amazon Cognito Cognito-Konsole einrichten.

Voraussetzung für die Einrichtung von Amazon MQ

- Eine funktionierende Docker-Installation zur Ausführung eines Bash-Skripts, das überprüft, ob das OAuth 2.0-Setup erfolgreich ist oder nicht.
- AWS CLI Version `>=2.28.23`, um das Hinzufügen eines Benutzernamens und eines Passworts bei der Broker-Erstellung optional zu machen.

Konfiguration der OAuth 2.0-Authentifizierung mit Amazon Cognito mithilfe AWS CLI

Das folgende Verfahren zeigt, wie Sie die OAuth 2.0-Authentifizierung für Ihre Amazon MQ for RabbitMQ-Broker einrichten, die Amazon Cognito als IdP verwenden. Dieses Verfahren dient dazu, die erforderlichen Ressourcen AWS CLI zu erstellen und zu konfigurieren.

Stellen Sie im folgenden Verfahren sicher, dass Sie die Platzhalterwerte, wie ConfigurationID und Revision, durch ihre *<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>* tatsächlichen *<2>* Werte ersetzen.

1. Erstellen Sie mit dem AWS CLI Befehl create-configuration eine neue Konfiguration, wie im folgenden Beispiel gezeigt.

```
aws mq create-configuration \
  --name "rabbitmq-oauth2-config" \
  --engine-type "RABBITMQ" \
  --engine-version "3.13"
```

   Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
    "Created": "2025-07-17T16:03:01.759000+00:00",
    "Description": "Auto-generated default for rabbitmq-oauth2-config on RabbitMQ
 3.13",
    "Revision": 1
    },
    "Name": "rabbitmq-oauth2-config"
}
```

2. Erstellen Sie eine Konfigurationsdatei, die aufgerufen wird**`rabbitmq.conf`**, um OAuth 2.0 als Authentifizierungs- und Autorisierungsmethode zu verwenden, wie im folgenden Beispiel gezeigt.

```
auth_backends.1 = oauth2
```

```
# FIXME: Update this value with the token signing key URL of your Amazon Cognito
 user pool.
# If you used the AWS CDK stack to deploy Amazon Cognito, this is one of the stack
 outputs.
auth_oauth2.jwks_url = ${RabbitMqOAuth2TestStack.JwksUri}
auth_oauth2.resource_server_id = rabbitmq
# Amazon Cognito does not include an audience field in access tokens
auth_oauth2.verify_aud = false

# Amazon Cognito does not allow * in its custom scopes. Use aliases to translate
 between Amazon Cognito and RabbitMQ.
auth_oauth2.scope_prefix = rabbitmq/
auth_oauth2.scope_aliases.1.alias = rabbitmq/read:all
auth_oauth2.scope_aliases.1.scope = rabbitmq/read:*/*
auth_oauth2.scope_aliases.2.alias = rabbitmq/write:all
auth_oauth2.scope_aliases.2.scope = rabbitmq/write:*/*
auth_oauth2.scope_aliases.3.alias = rabbitmq/configure:all
auth_oauth2.scope_aliases.3.scope = rabbitmq/configure:*/*

# Allow OAuth 2.0 login for RabbitMQ management console
management.oauth_enabled = true
# FIXME: Update this value with the client ID of your public application client
management.oauth_client_id
 = ${RabbitMqOAuth2TestStack.ManagementConsoleAppClientId}
# FIXME: Update this value with the base JWKS URI (without /.well-known/jwks.json)
auth_oauth2.issuer = ${RabbitMqOAuth2TestStack.Issuer}
management.oauth_scopes = rabbitmq/tag:administrator
```

Diese Konfiguration verwendet Bereichsaliase, um die in Amazon Cognito definierten Bereiche RabbitMQ-kompatiblen Bereichen zuzuordnen.

3. Aktualisieren Sie die Konfiguration mithilfe des Befehls update-configuration, wie im folgenden Beispiel gezeigt AWS CLI . Fügen Sie in diesem Befehl die Konfigurations-ID hinzu, die Sie als Antwort auf Schritt 1 dieses Verfahrens erhalten haben. Beispiel, **`c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`**.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-b600ac8e-8183-4f74-
a713-983e59f30e3d",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-b600ac8e-8183-4f74-a713-983e59f30e3d",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-oauth2-config",
    "Warnings": []
}
```

4. Erstellen Sie einen Broker mit der OAuth 2.0-Konfiguration, die Sie in Schritt 2 dieses Verfahrens erstellt haben. Verwenden Sie dazu den AWS CLI Befehl create-broker, wie im folgenden Beispiel gezeigt. Geben Sie in diesem Befehl die Konfigurations-ID und die Revisionsnummer an, die Sie in den Antworten von Schritt 1 bzw. 2 erhalten haben. Beispiel: **`c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`** und **`2`**.

```
aws mq create-broker \
 --broker-name "rabbitmq-oauth2-broker" \
 --engine-type "RABBITMQ" \
 --engine-version "3.13" \
 --host-instance-type "mq.m7g.large" \
 --deployment-mode "CLUSTER_MULTI_AZ" \
 --logs '{"General": true}' \
 --publicly-accessible \
 --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-
eb15b38b22ca>","Revision": <2>}' \
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-oauth2-
broker:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Vergewissern Sie sich, dass der Status des Brokers von `CREATION_IN_PROGRESS` zu wechselt`RUNNING`, indem Sie den AWS CLI Befehl describe-broker verwenden, wie im folgenden

Beispiel gezeigt. Geben Sie in diesem Befehl die Broker-ID ein, die Sie im Ergebnis des vorherigen Schritts erhalten haben, z. B. **b-2a1b5133-a10c-49d2-879b-8c176c34cf73**

```
aws mq describe-broker \
 --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt. Die folgende Antwort ist eine abgekürzte Version der vollständigen Ausgabe, die der `describe-broker` Befehl zurückgibt. Diese Antwort zeigt den Brokerstatus und die Authentifizierungsstrategie, die zur Sicherung des Brokers verwendet wurden. In diesem Fall weist die `config_managed` Authentifizierungsstrategie darauf hin, dass der Broker OAuth zwei Authentifizierungsmethoden verwendet.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

Um sich mit der RabbitMQ Management Console anzumelden OAuth2, muss der Broker-Endpunkt als gültige Callback-URL im entsprechenden Amazon Cognito-App-Client hinzugefügt werden. Weitere Informationen finden Sie in Schritt 5 bei der Einrichtung unseres Amazon Cognito CDK-Beispielstapels.

6. Überprüfen Sie die OAuth 2.0-Authentifizierung und Autorisierung mit dem folgenden `perf-test.sh` Skript.

   Verwenden Sie dieses Bash-Skript, um die Konnektivität zu Ihrem Amazon MQ for RabbitMQ Broker zu testen. Dieses Skript ruft ein Token von Amazon Cognito ab und überprüft, ob die Verbindung ordnungsgemäß konfiguriert wurde. Wenn es erfolgreich konfiguriert wurde, werden Sie sehen, wie Ihr Broker Nachrichten veröffentlicht und verarbeitet.

   Wenn Sie eine `ACCESS_REFUSED` Fehlermeldung erhalten, können Sie mithilfe der CloudWatch Protokolle Ihres Brokers Fehler in Ihren Konfigurationseinstellungen beheben. Sie finden den Link für die CloudWatch Protokollgruppe für Ihren Broker in der Amazon MQ MQ-Konsole.

   In diesem Skript müssen Sie die folgenden Werte angeben:

- `CLIENT_IDundCLIENT_SECRET`: Sie finden diese Werte auf der App-Client-Seite der Amazon Cognito Cognito-Konsole.
- Cognito-Domain: Sie finden diese Domain auf der Amazon Cognito Cognito-Konsole. Wählen Sie unter Branding die Option Domain aus. Auf der Domain-Seite finden Sie diesen Wert im Abschnitt Resource Servers.
- Amazon MQ-Broker-Endpunkt: Sie finden diesen Wert unter Verbindungen auf der Broker-Detailseite der Amazon MQ MQ-Konsole.

```
#! /bin/bash
set -e

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
 application client
CLIENT_ID=${RabbitMqOAuth2TestStack.AmqpAppClientId}
CLIENT_SECRET=${RabbitMqOAuth2TestStack.AmqpAppClientSecret}

# FIXME: Update this value with the domain of your Amazon Cognito user pool
RESPONSE=$(curl -X POST ${RabbitMqOAuth2TestStack.TokenEndpoint} \
                -H "Content-Type: application/x-www-form-urlencoded" \
                -d
 "grant_type=client_credentials&client_id=${CLIENT_ID}&client_secret=${CLIENT_SECRET}&scope
configure:all rabbitmq/read:all rabbitmq/tag:administrator rabbitmq/write:all")


# Extract the access_token from the response.
# This token will be passed in the password field when connecting to the broker.
# Note that the username is left blank, the field is ignored by the plugin.
BROKER_PASSWORD=$(echo ${RESPONSE} | jq -r '.access_token')

# FIXME: Update this value with the endpoint of your broker. For
 example, b-89424106-7e0e-4abe-8e98-8de0dada7630.mq.us-east-1.on.aws.
BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://:${BROKER_PASSWORD}@${BROKER_DNS}:5671

# Produce/consume messages using the above connection string
QUEUES_COUNT=1
PRODUCERS_COUNT=1
CONSUMERS_COUNT=1
PRODUCER_RATE=1
```

```
docker run -it --rm --ulimit nofile=40960:40960 pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --flag persistent --rate $PRODUCER_RATE
```

Konfiguration OAuth 2.0 und einfache Authentifizierung mit Amazon Cognito

Wenn Sie einen Broker mit OAuth 2.0-Authentifizierung erstellen, können Sie eine der folgenden Authentifizierungsmethoden angeben:

- OAuth Nur 2.0: Um diese Methode zu verwenden, geben Sie bei der Erstellung des Brokers keinen Benutzernamen und kein Passwort an. Das vorherige Verfahren zeigt, wie nur die OAuth 2.0-Authentifizierungsmethode verwendet wird.
- Sowohl OAuth 2.0 als auch einfache Authentifizierung: Um diese Methode zu verwenden, geben Sie bei der Erstellung des Brokers einen Benutzernamen und ein Passwort ein. Fügen Sie `auth_backends.2 = internal` außerdem Ihre Broker-Konfiguration hinzu, wie im folgenden Verfahren gezeigt.

Stellen Sie im folgenden Verfahren sicher, dass Sie die Platzhalterwerte, wie z. B. *<ConfigurationId>* und*<Revision>*, durch ihre tatsächlichen Werte ersetzen.

1. Um beide Authentifizierungsmethoden zu verwenden, erstellen Sie Ihre Broker-Konfiguration, wie im folgenden Beispiel gezeigt.

```
auth_backends.1 = oauth2
auth_backends.2 = internal

# FIXME: Update this value with the token signing key URL of your Amazon Cognito
 user pool
auth_oauth2.jwks_url = ${RabbitMqOAuth2TestStack.JwksUri}
auth_oauth2.resource_server_id = rabbitmq
auth_oauth2.verify_aud = false

auth_oauth2.scope_prefix = rabbitmq/
```

```
auth_oauth2.scope_aliases.1.alias = rabbitmq/read:all
auth_oauth2.scope_aliases.1.scope = rabbitmq/read:*/*
auth_oauth2.scope_aliases.2.alias = rabbitmq/write:all
auth_oauth2.scope_aliases.2.scope = rabbitmq/write:*/*
auth_oauth2.scope_aliases.3.alias = rabbitmq/configure:all
auth_oauth2.scope_aliases.3.scope = rabbitmq/configure:*/*
```

Diese Konfiguration verwendet Bereichsaliase, um die in Amazon Cognito definierten Bereiche RabbitMQ-kompatiblen Bereichen zuzuordnen.

2. Erstellen Sie einen Broker, der beide Authentifizierungsmethoden verwendet, wie im folgenden Beispiel gezeigt.

```
aws mq create-broker \
 --broker-name "rabbitmq-oauth2-broker-with-internal-user" \
 --engine-type "RABBITMQ" \
 --engine-version "3.13" \
 --host-instance-type "mq.m7g.large" \
 --deployment-mode "CLUSTER_MULTI_AZ" \
 --logs '{"General": true}' \
 --publicly-accessible \
 --configuration '{"Id": "<ConfigurationId>","Revision": <Revision>}' \
 --users '[{"Username":"<myUser>","Password":"<myPassword11>"}]'
```

3. Stellen Sie sicher, dass der Brokerstatus und die Konfiguration für die Einrichtung der Authentifizierungsmethode erfolgreich waren, wie in den Schritten 5 und 6 des Konfiguration der OAuth 2.0-Authentifizierung mit Amazon Cognito Verfahrens beschrieben.

Verwenden der IAM-Authentifizierung und -Autorisierung für Amazon MQ für RabbitMQ

Das folgende Verfahren zeigt, wie Sie die AWS IAM-Authentifizierung und -Autorisierung für einen Amazon MQ for RabbitMQ-Broker aktivieren. Nach der Aktivierung von IAM können sich Benutzer mithilfe von AWS IAM-Anmeldeinformationen authentifizieren, um auf die RabbitMQ Management API zuzugreifen und sich über AMQP zu verbinden. Einzelheiten zur Funktionsweise der IAM-Authentifizierung mit Amazon MQ für RabbitMQ finden Sie unter. the section called “IAM-Authentifizierung und -Autorisierung”

Voraussetzungen

- AWS Administratoranmeldedaten für das AWS Konto, dem der Amazon MQ for RabbitMQ-Broker gehört
- Eine mit diesen Administratoranmeldedaten konfigurierte Shell-Umgebung (mithilfe von AWS CLI-Profilen oder Umgebungsvariablen)
- AWS CLI installiert und konfiguriert
- `jq`Befehlszeilen-JSON-Prozessor installiert
- `curl`Befehlszeilentool installiert

Konfiguration der IAM-Authentifizierung und -Autorisierung mit AWS CLI

1. Legen Sie Umgebungsvariablen fest

   Stellen Sie die erforderlichen Umgebungsvariablen für Ihren Broker ein:

   ```
   export AWS_DEFAULT_REGION=<region>
   export BROKER_ID=<broker-id>
   ```

2. Aktivieren Sie ausgehende JWT-Token

   Aktivieren Sie den ausgehenden Web-Identitätsverbund für Ihr Konto: AWS

   ```
   ISSUER_IDENTIFIER=$(aws iam enable-outbound-web-identity-federation --query
    'IssuerIdentifier' --output text)
   echo $ISSUER_IDENTIFIER
   ```

   In der Ausgabe wird eine eindeutige URL zur Aussteller-ID für Ihr Konto im folgenden Format angezeigt. `https://<id>.tokens.sts.global.api.aws`

3. Erstellen Sie das IAM-Richtliniendokument

   Erstellen Sie ein Richtliniendokument, das Berechtigungen zum Abrufen von Web-Identitätstoken gewährt:

```
cat > policy.json << 'EOF'
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "sts:GetWebIdentityToken",
                "sts:TagGetWebIdentityToken"
            ],
            "Resource": "*"
        }
    ]
}
EOF
```

4. Erstellen Sie die Vertrauensrichtlinie

   Rufen Sie Ihre Anruferidentität ab und erstellen Sie ein Dokument zur Vertrauensrichtlinie:

```
CALLER_ARN=$(aws sts get-caller-identity --query Arn --output text)
cat > trust-policy.json << EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "$CALLER_ARN"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
```

5. Erstellen Sie die IAM-Rolle

Erstellen Sie die IAM-Rolle und fügen Sie die Richtlinie hinzu:

```
aws iam create-role --role-name RabbitMqAdminRole --assume-role-policy-document
 file://trust-policy.json
aws iam put-role-policy --role-name RabbitMqAdminRole --policy-name
 RabbitMqAdminRolePolicy --policy-document file://policy.json
```

6. Konfigurieren Sie die OAuth2 RabbitMQ-Einstellungen

   Erstellen Sie eine RabbitMQ-Konfigurationsdatei mit Authentifizierungs- und Autorisierungseinstellungen: OAuth2

```
cat > rabbitmq.conf << EOF
auth_backends.1 = oauth2
auth_backends.2 = internal

auth_oauth2.jwks_url = ${ISSUER_IDENTIFIER}/.well-known/jwks.json
auth_oauth2.resource_server_id = rabbitmq
auth_oauth2.scope_prefix = rabbitmq/

auth_oauth2.additional_scopes_key = sub
auth_oauth2.scope_aliases.1.alias = arn:aws:iam::$(aws sts get-caller-identity --
query Account --output text):role/RabbitMqAdminRole
auth_oauth2.scope_aliases.1.scope = rabbitmq/tag:administrator rabbitmq/read:*/*
 rabbitmq/write:*/* rabbitmq/configure:*/*
auth_oauth2.https.hostname_verification = wildcard

management.oauth_enabled = true
EOF
```

7. Aktualisieren Sie die Broker-Konfiguration

   Wenden Sie die neue Konfiguration auf Ihren Broker an:

```
# Retrieve the configuration ID
CONFIG_ID=$(aws mq describe-broker --broker-id $BROKER_ID --query
 'Configurations[0].Id' --output text)
```

```
# Create a new configuration revision
REVISION=$(aws mq update-configuration --configuration-id $CONFIG_ID --data "$(cat
 rabbitmq.conf | base64 --wrap=0)" --query 'LatestRevision.Revision' --output text)

# Apply the configuration to the broker
aws mq update-broker --broker-id $BROKER_ID --configuration Id=$CONFIG_ID,Revision=
$REVISION

# Reboot the broker to apply changes
aws mq reboot-broker --broker-id $BROKER_ID
```

Warten Sie, bis der Broker-Status wieder erreicht ist, `RUNNING` bevor Sie mit dem nächsten Schritt fortfahren.

8. Besorgen Sie sich ein JWT-Token

   Nehmen Sie die IAM-Rolle an und holen Sie sich ein Web-Identitätstoken:

```
# Assume the RabbitMqAdminRole
ROLE_CREDS=$(aws sts assume-role --role-arn arn:aws:iam::$(aws sts get-caller-
identity --query Account --output text):role/RabbitMqAdminRole --role-session-name
 rabbitmq-session)

# Configure the session with temporary credentials
export AWS_ACCESS_KEY_ID=$(echo "$ROLE_CREDS" | jq -r '.Credentials.AccessKeyId')
export AWS_SECRET_ACCESS_KEY=$(echo "$ROLE_CREDS" | jq -r
 '.Credentials.SecretAccessKey')
export AWS_SESSION_TOKEN=$(echo "$ROLE_CREDS" | jq -r '.Credentials.SessionToken')

# Obtain the web identity token
TOKEN_RESPONSE=$(aws sts get-web-identity-token \
    --audience "rabbitmq" \
    --signing-algorithm ES384 \
    --duration-seconds 300 \
    --tags Key=scope,Value="rabbitmq/tag:administrator")

# Extract the token
TOKEN=$(echo "$TOKEN_RESPONSE" | jq -r '.WebIdentityToken')
```

9. Greifen Sie auf die RabbitMQ Management API zu

   Verwenden Sie das JWT-Token, um auf die RabbitMQ Management API zuzugreifen:

   ```
   BROKER_URL=<broker-id>.mq.<region>.on.aws

   curl -u ":$TOKEN" \
       -X GET https://${BROKER_URL}/api/overview \
       -H "Content-Type: application/json"
   ```

   Eine erfolgreiche Antwort bestätigt, dass die IAM-Authentifizierung korrekt funktioniert. Die Antwort enthält Informationen zur Brokerübersicht im JSON-Format.

10. Stellen Sie mithilfe des JWT-Tokens eine Connect über AMQP her

    Testen Sie die AMQP-Konnektivität mithilfe des JWT-Tokens mit dem Perf-Test-Tool:

    ```
    BROKER_DNS=<broker-endpoint>
    CONNECTION_STRING=amqps://:${TOKEN}@${BROKER_DNS}:5671

    docker run -it --rm --ulimit nofile=40960:40960 pivotalrabbitmq/perf-test:latest \
        --queue-pattern 'test-queue-%d' --queue-pattern-from 1 --queue-pattern-to 1 \
        --producers 1 --consumers 1 \
        --uri ${CONNECTION_STRING} \
        --flag persistent --rate 1
    ```

    Wenn Sie eine `ACCESS_REFUSED` Fehlermeldung erhalten, können Sie mithilfe der Protokolle für Ihren Broker Fehler in Ihren Konfigurationseinstellungen beheben. CloudWatch Sie finden den Link für die Protokollgruppe CloudWatch Logs für Ihren Broker in der Amazon MQ MQ-Konsole.

Verwenden der LDAP-Authentifizierung und -Autorisierung für Amazon MQ für RabbitMQ

In diesem Tutorial wird beschrieben, wie Sie die LDAP-Authentifizierung und -Autorisierung für Ihre Amazon MQ for RabbitMQ-Broker mithilfe von konfigurieren. AWS Managed Microsoft AD

Auf dieser Seite

Voraussetzungen für die Konfiguration der LDAP-Authentifizierung und -Autorisierung

Sie können die in diesem Tutorial erforderlichen AWS Ressourcen einrichten, indem Sie den AWS CDK-Stack für Amazon MQ für die RabbitMQ LDAP-Integration mit bereitstellen. AWS Managed Microsoft AD

Dieser CDK-Stack erstellt automatisch alle erforderlichen AWS Ressourcen AWS Managed Microsoft AD, einschließlich LDAP-Benutzer und -Gruppen, Network Load Balancer, Zertifikate und IAM-Rollen. Eine vollständige Liste der vom Stack erstellten Ressourcen finden Sie in der README-Datei des Pakets.

Wenn Sie die Ressourcen manuell einrichten, anstatt den CDK-Stack zu verwenden, stellen Sie sicher, dass Sie über die entsprechende Infrastruktur verfügen, bevor Sie LDAP auf Ihrem Amazon MQ für RabbitMQ-Broker konfigurieren.

Voraussetzung für die Einrichtung von Amazon MQ

AWS CLI-Version >= 2.28.23, um das Hinzufügen eines Benutzernamens und Kennworts bei der Brokererstellung optional zu machen.

Konfiguration von LDAP in RabbitMQ mit CLI AWS

Dieses Verfahren verwendet AWS CLI, um die erforderlichen Ressourcen zu erstellen und zu konfigurieren. Stellen Sie im folgenden Verfahren sicher, dass Sie die Platzhalterwerte, wie ConfigurationID und Revision, durch ihre `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` tatsächlichen <2> Werte ersetzen.

1. Erstellen Sie mit dem `create-configuration` AWS CLI-Befehl eine neue Konfiguration, wie im folgenden Beispiel gezeigt.

```
aws mq create-configuration \
  --name "rabbitmq-ldap-config" \
  --engine-type "RABBITMQ" \
  --engine-version "3.13"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
"Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-ae0c-
eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
"Created": "2025-07-17T16:03:01.759000+00:00",
    "Description": "Auto-generated default for rabbitmq-ldap-config on RabbitMQ
 3.13",
    "Revision": 1
    },
    "Name": "rabbitmq-ldap-config"
}
```

2. Erstellen Sie eine Konfigurationsdatei`rabbitmq.conf`, die aufgerufen wird, um LDAP als Authentifizierungs- und Autorisierungsmethode zu verwenden, wie im folgenden Beispiel gezeigt. Ersetzen Sie alle Platzhalterwerte in der Vorlage (gekennzeichnet mit`${RabbitMqLdapTestStack.*}`) durch tatsächliche Werte aus Ihren bereitgestellten Stack-Ausgaben oder AWS CDK einer gleichwertigen Infrastruktur.

```
auth_backends.1 = ldap

# LDAP authentication settings - For more information,
# see https://www.rabbitmq.com/docs/ldap#basic

# FIXME: Replace the ${RabbitMqLdapTestStack.*} placeholders with actual values
# from your deployed prerequisite CDK stack outputs.
auth_ldap.servers.1 = ${RabbitMqLdapTestStack.NlbDnsName}
auth_ldap.dn_lookup_bind.user_dn = ${RabbitMqLdapTestStack.DnLookupUserDn}
auth_ldap.dn_lookup_base = ${RabbitMqLdapTestStack.DnLookupBase}
auth_ldap.dn_lookup_attribute = ${RabbitMqLdapTestStack.DnLookupAttribute}
auth_ldap.port = 636
auth_ldap.use_ssl = true
auth_ldap.ssl_options.verify = verify_peer
auth_ldap.log = network
```

```
# AWS integration for secure credential retrieval
# - see: https://github.com/amazon-mq/rabbitmq-aws
# The aws plugin allows RabbitMQ to securely retrieve credentials and certificates
# from AWS services.

# Replace the ${RabbitMqLdapTestStack.*} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.auth_ldap.ssl_options.cacertfile = ${RabbitMqLdapTestStack.CaCertArn}
aws.arns.auth_ldap.dn_lookup_bind.password =
 ${RabbitMqLdapTestStack.DnLookupUserPasswordArn}
aws.arns.assume_role_arn = ${RabbitMqLdapTestStack.AmazonMqAssumeRoleArn}

# LDAP authorization queries - For more information,
# see: https://www.rabbitmq.com/docs/ldap#authorisation

# FIXME: Replace the ${RabbitMqLdapTestStack.*} placeholders with actual group DN
# values from your deployed prerequisite CDK stack outputs
# Uses Active Directory groups created by the prerequisite CDK stack
auth_ldap.queries.tags = '''
[{administrator, {in_group,
 "${RabbitMqLdapTestStack.RabbitMqAdministratorsGroupDn}"}},
{management,    {in_group,
 "${RabbitMqLdapTestStack.RabbitMqMonitoringUsersGroupDn}"}}]
'''

# FIXME: This provides all authenticated users access to all vhosts
# - update to restrict access as required
auth_ldap.queries.vhost_access = '''
{constant, true}
'''

# FIXME: This provides all authenticated users full access to all
# queues and exchanges - update to restrict access as required
auth_ldap.queries.resource_access = '''
{for, [    {permission, configure, {constant, true}},
     {permission, write,
      {for, [{resource, queue,    {constant, true}},
             {resource, exchange, {constant, true}}]}},
     {permission, read,
      {for, [{resource, exchange, {constant, true}},
             {resource, queue,    {constant, true}}]}}
    ]
}
'''
```

```
# FIXME: This provides all authenticated users access to all topics
# - update to restrict access as required
auth_ldap.queries.topic_access = '''
{for, [{permission, write, {constant, true}},
     {permission, read,  {constant, true}}
    ]
}
'''
```

3. Aktualisieren Sie die Konfiguration mit dem `update-configuration` AWS CLI-Befehl, wie im folgenden Beispiel gezeigt. Fügen Sie in diesem Befehl die Konfigurations-ID hinzu, die Sie als Antwort auf Schritt 1 dieses Verfahrens erhalten haben. Beispiel, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-b600ac8e-8183-4f74-
a713-983e59f30e3d",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-b600ac8e-8183-4f74-a713-983e59f30e3d",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-ldap-config",
    "Warnings": []
}
```

4. Erstellen Sie einen Broker mit der LDAP-Konfiguration, die Sie in Schritt 2 dieses Verfahrens erstellt haben. Verwenden Sie dazu den `create-broker` AWS CLI-Befehl, wie im folgenden Beispiel gezeigt. Geben Sie in diesem Befehl die Konfigurations-ID und die

Revisionsnummer an, die Sie in den Antworten von Schritt 1 bzw. 2 erhalten haben. Beispiel: `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` und 2.

```
aws mq create-broker \
 --broker-name "rabbitmq-ldap-test-1" \
 --engine-type "RABBITMQ" \
 --engine-version "3.13" \
 --host-instance-type "mq.m7g.large" \
 --deployment-mode "CLUSTER_MULTI_AZ" \
 --logs '{"General": true}' \
 --publicly-accessible \
 --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}'
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-ldap-
broker:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

> **Beschränkung der Benennung von Brokern**
>
> Die IAM-Rolle, die durch den vorausgesetzten CDK-Stack erstellt wurde, schränkt Broker-Namen zunächst ein. `rabbitmq-ldap-test` Stellen Sie sicher, dass Ihr Brokername diesem Muster folgt, da die IAM-Rolle sonst nicht berechtigt ist, die Rolle für die ARN-Auflösung zu übernehmen.

5. Stellen Sie mithilfe des `describe-broker` AWS CLI-Befehls sicher`RUNNING`, dass der Status des Brokers von `CREATION_IN_PROGRESS` zu wechselt, wie im folgenden Beispiel gezeigt. Geben Sie in diesem Befehl die Broker-ID ein, die Sie im Ergebnis des vorherigen Schritts erhalten haben, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73` z. B.

```
aws mq describe-broker \
```

```
 --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt. Die folgende Antwort ist eine abgekürzte Version der vollständigen Ausgabe, die der `describe-broker` Befehl zurückgibt. Diese Antwort zeigt den Brokerstatus und die Authentifizierungsstrategie, mit der der Broker gesichert wurde. In diesem Fall weist die `config_managed` Authentifizierungsstrategie darauf hin, dass der Broker die LDAP-Authentifizierungsmethode verwendet.

```
{
"AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Überprüfen Sie den RabbitMQ-Zugriff mit einem der Testbenutzer, die mit dem CDK-Stack als Voraussetzung erstellt wurden

```
# FIXME: Replace ${RabbitMqLdapTestStack.ConsoleUserPasswordArn} with the actual
 ARN from your deployed prerequisite CDK stack outputs
CONSOLE_PASSWORD=$(aws secretsmanager get-secret-value \
  --secret-id ${RabbitMqLdapTestStack.ConsoleUserPasswordArn} \
  --query 'SecretString' --output text)

# FIXME: Replace BrokerConsoleURL with the actual ConsoleURL retrieved by
# calling describe-broker for the broker created above
# Call management API /api/overview (should succeed)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  https://${BrokerConsoleURL}/api/overview

# Try to create a user (should fail - console user only has monitoring permissions)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  -X PUT https://${BrokerConsoleURL}/api/users/testuser \
  -H "Content-Type: application/json" \
  -d '{"password":"testpass","tags":"management"}'
```

Verwendung der HTTP-Authentifizierung und -Autorisierung für Amazon MQ für RabbitMQ

In diesem Tutorial wird beschrieben, wie Sie die HTTP-Authentifizierung und -Autorisierung für Ihre Amazon MQ for RabbitMQ-Broker mithilfe eines externen HTTP-Servers konfigurieren.

> **Note**
>
> Das HTTP-Authentifizierungs-Plugin ist nur für Amazon MQ für RabbitMQ Version 4 und höher verfügbar.

Auf dieser Seite

Voraussetzungen für die Konfiguration der HTTP-Authentifizierung und -Autorisierung

Sie können die in diesem Tutorial erforderlichen AWS Ressourcen einrichten, indem Sie den AWS CDK-Stack für Amazon MQ für die HTTP-Authentifizierungsintegration von RabbitMQ bereitstellen.

Dieser CDK-Stack erstellt automatisch alle erforderlichen AWS Ressourcen, einschließlich des HTTP-Authentifizierungsservers, der Zertifikate und der IAM-Rollen. Eine vollständige Liste der vom Stack erstellten Ressourcen finden Sie in der README-Datei des Pakets.

Wenn Sie die Ressourcen manuell einrichten, anstatt den CDK-Stack zu verwenden, stellen Sie sicher, dass Sie über die entsprechende Infrastruktur verfügen, bevor Sie die HTTP-Authentifizierung auf Ihrem Amazon MQ für RabbitMQ-Broker konfigurieren.

Voraussetzung für die Einrichtung von Amazon MQ

AWS CLI-Version >= 2.28.23, um das Hinzufügen eines Benutzernamens und Kennworts bei der Brokererstellung optional zu machen.

Konfiguration der HTTP-Authentifizierung in RabbitMQ mit CLI AWS

Dieses Verfahren verwendet AWS CLI, um die erforderlichen Ressourcen zu erstellen und zu konfigurieren. Stellen Sie im folgenden Verfahren sicher, dass Sie die Platzhalterwerte durch ihre tatsächlichen Werte ersetzen.

1. Erstellen Sie mit dem `create-configuration` AWS CLI-Befehl eine neue Konfiguration, wie im folgenden Beispiel gezeigt.

```
aws mq create-configuration \
  --name "rabbitmq-http-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:03:01.759000+00:00",
        "Description": "Auto-generated default for rabbitmq-http-config on RabbitMQ
 4.2",
        "Revision": 1
    },
    "Name": "rabbitmq-http-config"
}
```

2. Erstellen Sie eine Konfigurationsdatei, die aufgerufen wird`rabbitmq.conf`, um HTTP als Authentifizierungs- und Autorisierungsmethode zu verwenden, wie im folgenden Beispiel gezeigt. Ersetzen Sie alle Platzhalterwerte in der Vorlage (gekennzeichnet mit`${...}`) durch tatsächliche Werte aus Ihren bereitgestellten AWS CDK Stack-Ausgaben oder einer gleichwertigen Infrastruktur.

```
auth_backends.1 = cache
auth_backends.2 = http
auth_cache.cached_backend = http

# HTTP authentication settings
```

```
# For more information, see https://github.com/rabbitmq/rabbitmq-auth-backend-http

# FIXME: Replace the ${...} placeholders with actual values
# from your deployed prerequisite CDK stack outputs.
auth_http.http_method = post
auth_http.user_path = ${HttpServerUserPath}
auth_http.vhost_path = ${HttpServerVhostPath}
auth_http.resource_path = ${HttpServerResourcePath}
auth_http.topic_path = ${HttpServerTopicPath}

# TLS/HTTPS configuration
auth_http.ssl_options.verify = verify_peer
auth_http.ssl_options.sni = test.amazonaws.com

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.auth_http.ssl_options.cacertfile = ${CaCertArn}
```

3. Aktualisieren Sie die Konfiguration mit dem `update-configuration` AWS CLI-Befehl. Verwenden Sie die Konfigurations-ID aus Schritt 3.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
```

```
    },
    "Name": "rabbitmq-http-config",
    "Warnings": []
}
```

4. Erstellen Sie einen Broker mit der HTTP-Konfiguration. Verwenden Sie die Konfigurations-ID und die Revisionsnummer aus den vorherigen Schritten.

```
aws mq create-broker \
  --broker-name "rabbitmq-http-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}'
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-http-
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Stellen Sie mithilfe des `describe-broker` AWS CLI-Befehls sicher`RUNNING`, dass der Status des Brokers von `CREATION_IN_PROGRESS` zu wechselt.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt. Die `config_managed` Authentifizierungsstrategie gibt an, dass der Broker die HTTP-Authentifizierungsmethode verwendet.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Überprüfen Sie den RabbitMQ-Zugriff mit einem der Testbenutzer, die mit dem erforderlichen CDK-Stack erstellt wurden

```
# FIXME: Replace ${RabbitMqHttpAuthElbStack.ConsoleUserPasswordArn} with the actual
 ARN from your deployed prerequisite CDK stack outputs
CONSOLE_PASSWORD=$(aws secretsmanager get-secret-value \
  --secret-id ${RabbitMqHttpAuthElbStack.ConsoleUserPasswordArn} \
  --query 'SecretString' --output text)

# FIXME: Replace BrokerConsoleURL with the actual ConsoleURL retrieved by
# calling describe-broker for the broker created above
# Call management API /api/overview (should succeed)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  https://${BrokerConsoleURL}/api/overview

# Try to create a vhost (should fail - console user only has management
 permissions)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  -X PUT https://${BrokerConsoleURL}/api/vhosts/test-vhost \
  -H "Content-Type: application/json" \
  -d '{}'
```

Verwendung der SSL-Zertifikatsauthentifizierung für Amazon MQ für RabbitMQ

In diesem Tutorial wird beschrieben, wie Sie die SSL-Zertifikatsauthentifizierung für Ihre Amazon MQ for RabbitMQ-Broker mithilfe einer privaten Zertifizierungsstelle konfigurieren.

> **Note**
>
> Das SSL-Zertifikat-Authentifizierungs-Plugin ist nur für Amazon MQ für RabbitMQ Version 4 und höher verfügbar.

Auf dieser Seite

Voraussetzungen für die Konfiguration der SSL-Zertifikatsauthentifizierung

Die SSL-Zertifikatsauthentifizierung verwendet Mutual TLS (mTLS), um Clients mit X.509-Zertifikaten zu authentifizieren. Sie können die in diesem Tutorial erforderlichen AWS Ressourcen einrichten, indem Sie den AWS CDK-Stack für Amazon MQ für die MTLS-Integration von RabbitMQ bereitstellen.

Dieser CDK-Stack erstellt automatisch alle erforderlichen AWS Ressourcen, einschließlich Zertifizierungsstelle, Client-Zertifikate und IAM-Rollen. Eine vollständige Liste der vom Stack erstellten Ressourcen finden Sie in der README-Datei des Pakets.

> **Note**
>
> Bevor Sie den CDK-Stack bereitstellen, legen Sie die `RABBITMQ_TEST_USER_NAME` Umgebungsvariable fest. Dieser Wert wird als Common Name (CN) im Client-Zertifikat verwendet und muss mit dem Benutzernamen übereinstimmen, den Sie in den Schritten der Anleitung verwenden. Beispiel: `export RABBITMQ_TEST_USER_NAME="myuser"`

Wenn Sie die Ressourcen manuell einrichten, anstatt den CDK-Stack zu verwenden, stellen Sie sicher, dass Sie über die entsprechende Infrastruktur verfügen, bevor Sie die SSL-Zertifikatsauthentifizierung auf Ihrem Amazon MQ für RabbitMQ-Broker konfigurieren.

Voraussetzung für die Einrichtung von Amazon MQ

AWS CLI-Version >= 2.28.23, um das Hinzufügen eines Benutzernamens und Kennworts bei der Brokererstellung optional zu machen.

Konfiguration der SSL-Zertifikatsauthentifizierung in RabbitMQ mit CLI AWS

Dieses Verfahren verwendet AWS CLI, um die erforderlichen Ressourcen zu erstellen und zu konfigurieren. Stellen Sie im folgenden Verfahren sicher, dass Sie die Platzhalterwerte, wie ConfigurationID und Revision, durch ihre `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` tatsächlichen <2> Werte ersetzen.

1. Erstellen Sie mit dem `create-configuration` AWS CLI-Befehl eine neue Konfiguration, wie im folgenden Beispiel gezeigt.

```
aws mq create-configuration \
  --name "rabbitmq-ssl-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

   Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:03:01.759000+00:00",
        "Description": "Auto-generated default for rabbitmq-ssl-config on RabbitMQ
 4.2",
        "Revision": 1
    },
```

```
    "Name": "rabbitmq-ssl-config"
}
```

2. Erstellen Sie eine Konfigurationsdatei`rabbitmq.conf`, die aufgerufen wird, um die SSL-Zertifikatsauthentifizierung zu verwenden, wie im folgenden Beispiel gezeigt. Ersetzen Sie alle Platzhalterwerte in der Vorlage (gekennzeichnet mit`${...}`) durch tatsächliche Werte aus Ihren bereitgestellten Stack-Ausgaben für die AWS CDK erforderlichen Komponenten oder einer gleichwertigen Infrastruktur.

```
auth_mechanisms.1 = EXTERNAL
ssl_cert_login_from = common_name

auth_backends.1 = internal

# Reject if no client cert
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# FIXME: Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.ssl_options.cacertfile = ${CaCertArn}
```

3. Aktualisieren Sie die Konfiguration mit dem `update-configuration` AWS CLI-Befehl, wie im folgenden Beispiel gezeigt. Fügen Sie in diesem Befehl die Konfigurations-ID hinzu, die Sie als Antwort auf Schritt 1 dieses Verfahrens erhalten haben. Beispiel, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-ssl-config",
    "Warnings": []
}
```

4. Erstellen Sie einen Broker mit der Konfiguration für die SSL-Zertifikatauthentifizierung, die Sie in Schritt 2 dieses Verfahrens erstellt haben. Verwenden Sie dazu den `create-broker` AWS CLI-Befehl, wie im folgenden Beispiel gezeigt. Geben Sie in diesem Befehl die Konfigurations-ID und die Revisionsnummer an, die Sie in den Antworten von Schritt 1 bzw. 2 erhalten haben. Beispiel: `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` und 2.

```
aws mq create-broker \
  --broker-name "rabbitmq-ssl-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}' \
  --users '[{"Username":"testuser","Password":"testpassword"}]'
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-ssl-
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
```

```
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Stellen Sie sicher, dass der Status des Brokers von `CREATION_IN_PROGRESS` zu wechselt`RUNNING`, indem Sie den `describe-broker` AWS CLI-Befehl verwenden, wie im folgenden Beispiel gezeigt. Geben Sie in diesem Befehl die Broker-ID ein, die Sie im Ergebnis des vorherigen Schritts erhalten haben. Beispiel, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt. Die folgende Antwort ist eine abgekürzte Version der vollständigen Ausgabe, die der `describe-broker` Befehl zurückgibt. Diese Antwort zeigt den Brokerstatus und die Authentifizierungsstrategie, die zur Sicherung des Brokers verwendet wurden. In diesem Fall weist die `config_managed` Authentifizierungsstrategie darauf hin, dass der Broker die SSL-Zertifikatsauthentifizierungsmethode verwendet.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Überprüfen Sie die SSL-Zertifikatsauthentifizierung mit dem folgenden `ssl.sh` Skript.

Verwenden Sie dieses Bash-Skript, um die Konnektivität zu Ihrem Amazon MQ for RabbitMQ Broker zu testen. Dieses Skript verwendet Ihr Client-Zertifikat zur Authentifizierung und überprüft, ob die Verbindung ordnungsgemäß konfiguriert wurde. Wenn es erfolgreich konfiguriert wurde, werden Sie sehen, wie Ihr Broker Nachrichten veröffentlicht und verarbeitet.

Wenn Sie eine `ACCESS_REFUSED` Fehlermeldung erhalten, können Sie mithilfe der CloudWatch Protokolle Ihres Brokers Fehler in Ihren Konfigurationseinstellungen beheben. Sie finden den Link für die CloudWatch Protokollgruppe für Ihren Broker in der Amazon MQ MQ-Konsole.

In diesem Skript müssen Sie die folgenden Werte angeben:

- `USERNAME`: Der allgemeine Name (CN) aus Ihrem Client-Zertifikat.
- `CLIENT_KEYSTORE`: Pfad zu Ihrer Client-Keystore-Datei (PKCS12 Format). Wenn Sie den erforderlichen CDK-Stack verwendet haben, lautet der Standardpfad. `$(pwd)/certs/client-keystore.p12`
- `KEYSTORE_PASSWORD`: Passwort für Ihren Client-Keystore. Wenn Sie den vorausgesetzten CDK-Stack verwendet haben, lautet das Standardkennwort. `changeit`
- `BROKER_DNS`: Sie finden diesen Wert unter Verbindungen auf der Seite mit den Broker-Details der Amazon MQ MQ-Konsole.

```
#! /bin/bash
set -e

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
 application client
USERNAME=<client_cert_common_name>
CLIENT_KEYSTORE=$(pwd)/certs/client-keystore.p12
KEYSTORE_PASSWORD=changeit

BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://${BROKER_DNS}:5671

# Produce/consume messages using the above connection string
QUEUES_COUNT=1
PRODUCERS_COUNT=1
CONSUMERS_COUNT=1
PRODUCER_RATE=1

finch run --rm --ulimit nofile=40960:40960 \
    -v ${CLIENT_KEYSTORE}:/certs/client-keystore.p12:ro \
    -e JAVA_TOOL_OPTIONS="-Djavax.net.ssl.keyStore=/certs/client-
keystore.p12 -Djavax.net.ssl.keyStorePassword=${KEYSTORE_PASSWORD} -
Djavax.net.ssl.keyStoreType=PKCS12" \
```

```
    pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-cert-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "cert-test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --sasl-external \
    --use-default-ssl-context \
    --flag persistent --rate $PRODUCER_RATE
```

Verwendung von mTLS für AMQP- und Management-Endpunkte

In diesem Tutorial wird beschrieben, wie Mutual TLS (mTLS) für AMQP-Client-Verbindungen und die RabbitMQ-Verwaltungsschnittstelle mithilfe einer privaten Zertifizierungsstelle konfiguriert wird.

> **Note**
>
> Die Verwendung privater Zertifizierungsstellen für mTLS ist nur für Amazon MQ for RabbitMQ Version 4 und höher verfügbar.

Auf dieser Seite

Voraussetzungen für die Konfiguration von mTLS

Sie können die in diesem Tutorial erforderlichen AWS Ressourcen einrichten, indem Sie den AWS CDK-Stack für Amazon MQ für die MTLS-Integration mit RabbitMQ bereitstellen.

Dieser CDK-Stack erstellt automatisch alle erforderlichen AWS Ressourcen, einschließlich Zertifizierungsstelle, Client-Zertifikate und IAM-Rollen. Eine vollständige Liste der vom Stack erstellten Ressourcen finden Sie in der README-Datei des Pakets.

Wenn Sie die Ressourcen manuell einrichten, anstatt den CDK-Stack zu verwenden, stellen Sie sicher, dass Sie über die entsprechende Infrastruktur verfügen, bevor Sie mTLS auf Ihrem Amazon MQ für RabbitMQ-Broker konfigurieren.

Voraussetzung für die Einrichtung von Amazon MQ

AWS CLI-Version >= 2.28.23, um das Hinzufügen eines Benutzernamens und Kennworts bei der Brokererstellung optional zu machen.

Konfiguration von mTLS in RabbitMQ mit CLI AWS

Dieses Verfahren verwendet AWS CLI, um die erforderlichen Ressourcen zu erstellen und zu konfigurieren. Stellen Sie im folgenden Verfahren sicher, dass Sie die Platzhalterwerte, wie ConfigurationID und Revision, durch ihre `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` tatsächlichen <2> Werte ersetzen.

1. Erstellen Sie mit dem `create-configuration` AWS CLI-Befehl eine neue Konfiguration, wie im folgenden Beispiel gezeigt.

```
aws mq create-configuration \
  --name "rabbitmq-mtls-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

   Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:03:01.759000+00:00",
        "Description": "Auto-generated default for rabbitmq-mtls-config on RabbitMQ
 4.2",
        "Revision": 1
    },
    "Name": "rabbitmq-mtls-config"
}
```

2. Erstellen Sie eine Konfigurationsdatei`rabbitmq.conf`, die aufgerufen wird, um mTLS für AMQP und Verwaltungsendpunkte zu konfigurieren, wie im folgenden Beispiel gezeigt. Ersetzen Sie alle Platzhalterwerte in der Vorlage (gekennzeichnet mit`${...}`) durch tatsächliche Werte aus Ihren bereitgestellten Stack-Ausgaben oder einer AWS CDK gleichwertigen Infrastruktur.

```
auth_backends.1 = internal

# TLS configuration
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true
management.ssl.verify = verify_peer

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# FIXME: Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.ssl_options.cacertfile = ${CaCertArn}
aws.arns.management.ssl.cacertfile = ${CaCertArn}
```

3. Aktualisieren Sie die Konfiguration mit dem `update-configuration` AWS CLI-Befehl, wie im folgenden Beispiel gezeigt. Fügen Sie in diesem Befehl die Konfigurations-ID hinzu, die Sie als Antwort auf Schritt 1 dieses Verfahrens erhalten haben. Beispiel, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "Created": "2025-07-17T16:57:04.520931+00:00",
```

```
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-mtls-config",
    "Warnings": []
}
```

4. Erstellen Sie einen Broker mit der mTLS-Konfiguration, die Sie in Schritt 2 dieses Verfahrens erstellt haben. Verwenden Sie dazu den `create-broker` AWS CLI-Befehl, wie im folgenden Beispiel gezeigt. Geben Sie in diesem Befehl die Konfigurations-ID und die Revisionsnummer an, die Sie in den Antworten von Schritt 1 bzw. 2 erhalten haben. Beispiel: `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` und 2.

```
aws mq create-broker \
  --broker-name "rabbitmq-mtls-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}' \
  --users '[{"Username":"testuser","Password":"testpassword"}]'
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-mtls-
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Stellen Sie mithilfe des `describe-broker` AWS CLI-Befehls sicher`RUNNING`, dass der Status des Brokers von `CREATION_IN_PROGRESS` zu wechselt, wie im folgenden Beispiel gezeigt.

Geben Sie in diesem Befehl die Broker-ID ein, die Sie im Ergebnis des vorherigen Schritts erhalten haben. Beispiel, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Dieser Befehl gibt eine Antwort zurück, die dem folgenden Beispiel ähnelt. Die folgende Antwort ist eine abgekürzte Version der vollständigen Ausgabe, die der `describe-broker` Befehl zurückgibt.

```
{
    "AuthenticationStrategy": "simple",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Überprüfen Sie die mTLS-Authentifizierung mit dem folgenden `mtls.sh` Skript.

   Verwenden Sie dieses Bash-Skript, um die Konnektivität zu Ihrem Amazon MQ for RabbitMQ Broker zu testen. Dieses Skript verwendet Ihr Client-Zertifikat zur Authentifizierung und überprüft, ob die Verbindung ordnungsgemäß konfiguriert wurde. Wenn es erfolgreich konfiguriert wurde, werden Sie sehen, wie Ihr Broker Nachrichten veröffentlicht und verarbeitet.

   Wenn Sie eine `ACCESS_REFUSED` Fehlermeldung erhalten, können Sie mithilfe der CloudWatch Protokolle Ihres Brokers Fehler in Ihren Konfigurationseinstellungen beheben. Sie finden den Link für die CloudWatch Protokollgruppe für Ihren Broker in der Amazon MQ MQ-Konsole.

   In diesem Skript müssen Sie die folgenden Werte angeben:

   - `USERNAME`und`PASSWORD`: Die RabbitMQ-Benutzeranmeldeinformationen, die Sie mit dem Broker erstellt haben.
   - `CLIENT_KEYSTORE`: Pfad zu Ihrer Client-Keystore-Datei (Format). PKCS12 Wenn Sie den erforderlichen CDK-Stack verwendet haben, lautet der Standardpfad. `$(pwd)/certs/client-keystore.p12`

- KEYSTORE_PASSWORD: Passwort für Ihren Client-Keystore. Wenn Sie den erforderlichen CDK-Stack verwendet haben, lautet das Standardkennwort. `changeit`
- BROKER_DNS: Sie finden diesen Wert unter Verbindungen auf der Seite mit den Broker-Details der Amazon MQ MQ-Konsole.

```
#! /bin/bash
set -e

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
 application client
USERNAME=<testuser>
PASSWORD=<testpassword>
CLIENT_KEYSTORE=$(pwd)/certs/client-keystore.p12
KEYSTORE_PASSWORD=changeit

BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://${USERNAME}:${PASSWORD}@${BROKER_DNS}:5671

# Produce/consume messages using the above connection string
QUEUES_COUNT=1
PRODUCERS_COUNT=1
CONSUMERS_COUNT=1
PRODUCER_RATE=1

finch run --rm --ulimit nofile=40960:40960 \
    -v ${CLIENT_KEYSTORE}:/certs/client-keystore.p12:ro \
    -e JAVA_TOOL_OPTIONS="-Djavax.net.ssl.keyStore=/certs/client-
keystore.p12 -Djavax.net.ssl.keyStorePassword=${KEYSTORE_PASSWORD} -
Djavax.net.ssl.keyStoreType=PKCS12" \
    pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-cert-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "cert-test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --use-default-ssl-context \
    --flag persistent --rate $PRODUCER_RATE
```

Ihre JMS-Anwendung verbinden

Dieses Tutorial zeigt Ihnen, wie Sie Ihre JMS-Anwendung mithilfe des RabbitMQ JMS-Clients mit Amazon MQ for RabbitMQ Broker verbinden. Sie erfahren, wie Sie einen Producer zum Senden von Nachrichten und einen Consumer zum Empfangen von Nachrichten aus RabbitMQ-Warteschlangen einrichten.

Bevor Sie beginnen, fügen Sie Ihrem Maven-Projekt die entsprechende RabbitMQ-JMS-Abhängigkeit hinzu:

Für JMS 1.1 und 2.0:

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>2.12.0</version>
  </dependency>

</dependencies>
```

Für JMS 3.1:

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>3.5.0</version>
  </dependency>

</dependencies>
```

Erstellen Sie einen Produzenten

Das folgende Codebeispiel zeigt, wie man mit JMS in eine RabbitMQ-Warteschlange schreibt:

```
import jakarta.jms.*;
import com.rabbitmq.jms.admin.*;
```

```
// Setting the connection factory
RMQConnectionFactory factory = new RMQConnectionFactory();
factory.setHost(envProps.getProperty("RABBITMQ_HOST", "localhost"));
factory.setPort(Integer.parseInt(envProps.getProperty("RABBITMQ_PORT", "5672")));
factory.setUsername(envProps.getProperty("RABBITMQ_USERNAME", "guest"));
factory.setPassword(envProps.getProperty("RABBITMQ_PASSWORD", "guest"));
factory.setVirtualHost(envProps.getProperty("RABBITMQ_VIRTUAL_HOST", "/"));
factory.useSslProtocol();

connection = factory.createConnection();
connection.start();

String queueName = "test-queue-jms";
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

RMQDestination destination = new RMQDestination(queueName, true, false);

// Send the message to the queue
MessageProducer producer = session.createProducer(destination);
producer.setDeliveryMode(DeliveryMode.PERSISTENT);

String msg_content = "Hello World!!";
TextMessage textMessage = session.createTextMessage(msg_content);
producer.send(textMessage);

System.out.printf("Published to AMQP queue '%s': %s", queueName, msg_content);
```

Erstellen Sie einen Verbraucher

Das folgende Codebeispiel zeigt, wie mit JMS aus einer RabbitMQ-Warteschlange gelesen wird:

```
import jakarta.jms.*;
import com.rabbitmq.jms.admin.*;

// Setting the connection factory
RMQConnectionFactory factory = new RMQConnectionFactory();
factory.setHost(envProps.getProperty("RABBITMQ_HOST", "localhost"));
factory.setPort(Integer.parseInt(envProps.getProperty("RABBITMQ_PORT", "5672")));
factory.setUsername(envProps.getProperty("RABBITMQ_USERNAME", "guest"));
factory.setPassword(envProps.getProperty("RABBITMQ_PASSWORD", "guest"));
factory.setVirtualHost(envProps.getProperty("RABBITMQ_VIRTUAL_HOST", "/"));
factory.useSslProtocol();
```

```
// Establish the connection and session
jakarta.jms.Connection connection = factory.createConnection();

String queueName = "test-queue-jms";
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

RMQDestination destination = new RMQDestination();
destination.setDestinationName(queueName);
destination.setAmqp(true);
destination.setAmqpQueueName(queueName);

// Initialize consumer
MessageConsumer consumer = session.createConsumer(destination);
consumer.setMessageListener(message -> {
    try {
        if (message instanceof TextMessage) {
            TextMessage textMessage = (TextMessage) message;
            System.out.printf("Message: %s%n", textMessage.getText());
        } else if (message instanceof BytesMessage) {
            BytesMessage bytesMessage = (BytesMessage) message;
            byte[] bytes = new byte[(int) bytesMessage.getBodyLength()];
            bytesMessage.readBytes(bytes);
            String content = new String(bytes);
            System.out.printf("Message: %s%n", content);
        } else {
            System.out.printf("Message: [%s]%n", message.getClass().getSimpleName());
        }
    } catch (JMSException e) {
        System.err.printf("Error processing message: %s%n", e.getMessage());
    }
});

connection.start();
```

Sicherheit in Amazon MQ

Cloud-Sicherheit hat AWS höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für Amazon MQ gelten, finden Sie unter AWS Services in Scope by Compliance Program AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. 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 erfahren auch, wie Sie andere AWS Dienste nutzen können, die Ihnen helfen, Ihre Amazon MQ MQ-Ressourcen zu überwachen und zu sichern.

Topics

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 unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie 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 die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- 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 für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

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 anderen AWS-Services über die Konsole AWS CLI, API oder AWS SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen

Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Für sowohl Amazon MQ für ActiveMQ als auch für Amazon MQ für RabbitMQ verwenden Broker keine persönlich identifizierbare Informationen (PII) oder andere vertrauliche oder sensible Informationen für die Brokernamen oder Benutzernamen, wenn Sie Ressourcen über die Broker-Webkonsole oder die Amazon-MQ-API erstellen. Broker-Namen und Benutzernamen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

> **Important**
>
> TLS 1.3 ist für RabbitMQ-Broker nicht verfügbar.

Verschlüsselung

Die in Amazon MQ gespeicherten Benutzerdaten werden im Ruhezustand verschlüsselt. Die Amazon MQ-Verschlüsselung im Ruhezustand bietet eine erhöhte Sicherheit, indem Ihre Daten mit Hilfe von Verschlüsselungsschlüsseln verschlüsselt werden, die im 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) zur Verschlüsselung während der Übertragung.

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 KMS-Schlüssel. 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 owned KMS key (default) (Amazon-MQ-eigener KMS-Schlüssel (Standard)) – Der Schlüssel ist Eigentum von Amazon MQ und wird von diesem verwaltet. Er 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.
- Select existing customer managed KMS key (Vorhandenen, 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. Löschen Sie den Broker, um die Zugriffsrechte zu widerrufen.
- Für Amazon MQ for ActiveMQ-Broker, die Amazon Elastic File System (EFS) zum Speichern von Nachrichtendaten verwenden, kann es mehrere Stunden dauern, bis die Berechtigungen zur Verwendung der KMS-Schlüssel in Ihrem Konto widerrufen werden, nachdem Sie die erforderlichen Maßnahmen ergriffen haben.
- Bei Brokern für Amazon MQ for RabbitMQ und Amazon MQ for ActiveMQ, die EBS zum Speichern von Nachrichtendaten verwenden, gilt: wenn Sie Amazon EBS die Berechtigung zum Verwenden der KMS-Schlüssel in Ihrem Konto entziehen, kann Amazon MQ Ihren Broker nicht mehr verwalten und er wechselt möglicherweise in einen degradierten Zustand.
- 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.
- Es kann mehrere Stunden dauern, bis ein Schlüssel deaktiviert oder eine Gewährung zurückgezogen wird, nachdem Sie die erforderlichen Maßnahmen ergriffen haben.

- Für die Verschlüsselung oder Entschlüsselung von CloudWatch Protokollen können Sie nicht konfigurieren, was Amazon MQ für Ihren Verschlüsselungsschlüssel verwendet. CloudWatch logs schützt Daten im Ruhezustand durch Verschlüsselung, und Protokollgruppen werden verschlüsselt. Der CloudWatch Protokolldienst verwaltet die serverseitige Verschlüsselung standardmäßig. Weitere Informationen zur Verschlüsselung von Protokollgruppen finden Sie im Amazon CloudWatch Logs-Benutzerhandbuch.

Wenn Sie einen Single-Instance-Broker mit einem KMS-Schlüssel für RabbitMQ erstellen, werden zwei `CreateGrant`-Ereignisse in AWS CloudTrail protokolliert. Das erste Ereignis ist das Erstellen einer Erteilung für den KMS-Schlüssel durch Amazon MQ. Das zweite Ereignis ist das Erstellen einer Erteilung zur Nutzung durch EBS.

CreateGrant AWS CloudTrail Protokolleintrag: Einzelinstanz-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

Sie sehen ein Ereignis für das Erstellen der EBS-Erteilung.


```
                                    {
    "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 Cluster-Bereitstellung mit einem KMS-Schlüssel für RabbitMQ erstellen, werden fünf `CreateGrant`-Ereignisse in AWS CloudTrail protokolliert. Bei den ersten beiden Ereignissen handelt es sich um das Erstellen von Erteilungen für Amazon MQ. Bei den anderen drei Ereignissen handelt es sich um Erteilungen, die von EBS zur eigenen Nutzung 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

Sie sehen drei Ereignisse für das Erstellen der EBS-Erteilung.

```
                                    {
    "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 zum Verwenden von CMK-Schlüssel finden AWS KMS keysSie im 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 während der Übertragung zwischen den Brokern der Amazon-MQ-Bereitstellung. Alle Daten, die zwischen Amazon MQ-Brokern übertragen werden, werden mittels starker Transport Layer Security (TLS) verschlüsselt. Dies gilt für alle verfügbaren Protokolle.

Amazon MQ for RabbitMQ: Amazon MQ für RabbitMQ erfordert eine starke Verschlüsselung mit Transport Layer Security (TLS) für alle Client-Verbindungen. Der RabbitMQ-Cluster-Replikationsverkehr durchläuft nur die VPC Ihres Brokers, und der gesamte Netzwerkverkehr zwischen AWS Rechenzentren wird auf der physischen Ebene transparent verschlüsselt. Die geclusterten Broker von Amazon MQ für RabbitMQ unterstützen derzeit keine knotenübergreifende Verschlüsselung für die Cluster-Replikation. Weitere Informationen data-in-transit dazu finden Sie unter Verschlüsseln und in-Transit. Data-at-Rest

Amazon MQ für ActiveMQ Protokolle

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

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

Unterstützte TLS-Cipher-Suites für ActiveMQ

ActiveMQ auf Amazon MQ unterstützt die folgenden Verschlüsselungs-Suiten:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_ SHA384

- TLS_ECDHE_RSA_WITH_AES_256_CBC_ SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_MIT_AES_256_GCM_ SHA384
- TLS_DHE_RSA_MIT_AES_256_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_MIT_AES_256_GCM_ SHA384
- TLS_RSA_MIT_AES_256_CBC_ SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_GCM_ SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_ SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_MIT_AES_128_GCM_ SHA256
- TLS_DHE_RSA_MIT_AES_128_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_MIT_AES_128_GCM_ SHA256
- TLS_RSA_MIT_AES_128_CBC_ SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA

Amazon MQ für RabbitMQ-Protokolle

Sie können auf Ihre RabbitMQ-Broker zugreifen, indem Sie die folgenden Protokolle mit aktiviertem TLS verwenden:

- AMQP (0-9-1)

Unterstützte TLS-Cipher-Suites für RabbitMQ

RabbitMQ auf Amazon MQ unterstützt die folgenden Verschlüsselungs-Suiten:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_ SHA384
- TLS_ECDHE_RSA_WITH_AES_128_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. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) ist, Amazon MQ-Ressourcen zu nutzen. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe Fehlerbehebung für Amazon MQ-Identität und -Zugriff).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe Funktionsweise von Amazon MQ mit IAM).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe 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 oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter So melden Sie sich bei Ihrem AWS-Konto an im Benutzerhandbuch für AWS-Anmeldung .

AWS Bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter AWS Signature Version 4 for API requests im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter Tasks that require root user credentials im IAM-Benutzerhandbuch.

Benutzer und Gruppen

Ein IAM-Benutzer ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Erfordern, dass menschliche Benutzer für den Zugriff AWS mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine IAM-Gruppe spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

IAM-Rollen

Eine IAM-Rolle ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie von einer Benutzer-

zu einer IAM-Rolle (Konsole) wechseln AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter Methoden, um eine Rolle zu übernehmen im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter Übersicht über JSON-Richtlinien im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcenund unter welchen Bedingungendurchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter Choose between managed policies and inline policies im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM-Entitäten im -IAM-Benutzerhandbuch.
- Richtlinien zur Dienstkontrolle (SCPs) — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter Service-Kontrollrichtlinien im AWS Organizations -Benutzerhandbuch.
- Richtlinien zur Ressourcenkontrolle (RCPs) — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter Richtlinien zur Ressourcenkontrolle (RCPs).
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter Sitzungsrichtlinien im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die sich daraus ergebenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter Bewertungslogik für Richtlinien im IAM-Benutzerhandbuch.

Funktionsweise von Amazon MQ mit IAM

Bevor Sie mit IAM den Zugriff auf Amazon MQ verwalten können, sollten Sie sich darüber informieren, welche IAM-Funktionen Sie mit Amazon MQ verwenden können. Einen allgemeinen Überblick darüber, wie Amazon MQ und andere AWS Services mit IAM zusammenarbeiten, finden Sie unter AWS Services That Work with IAM im IAM-Benutzerhandbuch.

Amazon MQ verwendet IAM für Amazon MQ MQ-API-Operationen, um Broker zu erstellen, zu aktualisieren, zu löschen und aufzulisten. Für den Broker-Zugriff zum Veröffentlichen und Abonnieren von Nachrichten unterstützt Amazon MQ für ActiveMQ die native ActiveMQ-Authentifizierung und LDAP, während Amazon MQ für RabbitMQ die IAM-Authentifizierung und andere Methoden unterstützt. Weitere Informationen finden Sie unter the section called “Authentifizierung und Autorisierung von Brokern”.

Themen

- Identitätsbasierte Amazon MQ-Richtlinien
- Ressourcenbasierte Amazon MQ -Richtlinien
- Autorisierung auf der Basis von Amazon MQ-Tags
- Amazon MQ IAM-Rollen

Identitätsbasierte Amazon MQ-Richtlinien

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Amazon MQ unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Aktionen

Administratoren können mithilfe von JSON-Richtlinien angeben, wer Zugriff auf was hat. AWS Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Richtlinienaktionen in Amazon MQ verwenden das folgende Präfix vor der Aktion: `mq:`. Um einem Benutzer beispielsweise die Berechtigung zum Ausführen einer Amazon MQ-Instance mit der Amazon MQ`CreateBroker`-API-Operation zu erteilen, fügen Sie die Aktion `mq:CreateBroker` in seine Richtlinie ein. 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 unterVon Amazon MQ definierte AktionenimIAM-Benutzerhandbuch.

Ressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen Amazon-Ressourcennamen (ARN) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, 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.

Ressourcentypen	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 Broker namens `MyBroker` mit brokerId `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` in Ihrer Anweisung anzugeben, verwenden Sie den folgenden 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": "*"
```

Die API-Aktion `CreateTags` erfordert sowohl einen Broker als auch eine Konfiguration. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie sie durch Kommas. ARNs

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

Eine Liste der Amazon MQ-Ressourcentypen und ihrer Eigenschaften ARNs finden Sie unter Von Amazon MQ definierte Ressourcen im IAM-Benutzerhandbuch. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter Von Amazon MQ definierte Aktionen.

Bedingungsschlüssel

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. 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. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontextschlüssel für AWS globale Bedingungen im IAM-Benutzerhandbuch.

Amazon MQ stellt keine servicespezifischen Bedingungsschlüssel bereit, unterstützt jedoch die Verwendung einiger globaler Bedingungsschlüssel. Die Liste der Amazon MQ-Bedingungsschlüssel für Amazon MQ finden Sie in der folgenden Tabelle oderBedingungsschlüssel für Amazon MQimIAM-Benutzerhandbuch. 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.	Zeichenfolge
as: ResourceTag /$ {} TagKey	Filtert Aktionen basierend auf den Tags, die der Ressource zugeordnet sind.	Zeichenfolge
war: TagKeys	Filtert Aktionen basierend auf den Tag-Schlüsseln, die in der Anforderung übergeben werden.	Zeichenfolge

Beispiele

Beispiele für identitätsbasierte Amazon MQ-Richtlinien finden Sie unter .

Ressourcenbasierte Amazon MQ -Richtlinien

Derzeit unterstützt Amazon MQ keine IAM-Authentifizierung unter Verwendung ressourcenbasierter Berechtigungen oder ressourcenbasierter 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:

JSON

```
{
    "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
- Zugriffssteuerung mit IAM-Tags

Amazon MQ IAM-Rollen

Eine IAM-Rolle 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 über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende 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 zum Konto. Dies 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 mit der AWS-Managementkonsole AWS CLI, oder AWS API ausführen. 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

- Best Practices für Richtlinien
- Verwenden der Amazon MQ-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Best Practices 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. Wenn Sie identitätsbasierte Richtlinien erstellen oder bearbeiten, befolgen Sie diese Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen

finden Sie unter Von AWS verwaltete Richtlinien oder Von 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 Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. 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 mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, 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 Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Best Practices 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 es Ihnen ermöglichen, Details

zu den Amazon MQ MQ-Ressourcen in Ihrem AWS Konto aufzulisten und einzusehen. 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 weiterhin die Amazon MQ MQ-Konsole verwenden können, fügen Sie den Entitäten auch die folgende AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch:

```
AmazonMQReadOnlyAccess
```

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. 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 umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
```

```
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

API-Authentifizierung und Amazon MQ-Autorisierung für

Amazon MQ verwendet die standardmäßige Signierung von AWS Anfragen 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 Ihre IAM-Richtlinienberechtigungen bearbeiten.

Themen

- Erforderliche IAM-Berechtigungen zum Erstellen eines Amazon MQ-Brokers
- Amazon MQ REST API-Berechtigungen–Referenz
- Referenz für zusätzliche Amazon MQ MQ-Berechtigungen
- 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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Action": [
            "mq:*",
            "ec2:CreateNetworkInterface",
            "ec2:DeleteNetworkInterface",
            "ec2:DetachNetworkInterface",
            "ec2:DescribeInternetGateways",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeRouteTables",
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcs"
        ],
        "Effect": "Allow",
        "Resource": "*"
    },{
        "Action": [
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterfacePermission",
            "ec2:DescribeNetworkInterfacePermissions"
```

```
        ],
        "Effect": "Allow",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "ec2:AuthorizedService": "mq.amazonaws.com"
            }
        }
    }]
}
```

Weitere Informationen erhalten 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

In der folgenden Tabelle sind Amazon MQ REST APIs und die entsprechenden IAM-Berechtigungen aufgeführt.

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
DescribeConfigurationRevision	mq:DescribeConfigurationRevision

Amazon MQ REST APIs	Erforderliche Berechtigungen
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

Referenz für zusätzliche Amazon MQ MQ-Berechtigungen

In der folgenden Tabelle sind die Amazon MQ MQ-API und die zusätzlichen IAM-Berechtigungen aufgeführt, die für bestimmte Funktionen wie die OAuth 2.0-Authentifizierung erforderlich sind.

Amazon MQ REST-API	Berechtigung	Description
UpdateBroker	mq:UpdateBrokerAccessConfiguration	Sie benötigen diese Berechtigung, um die Authentifizierungs- und Autorisierungsoptionen in der zugehörigen Broker-Konfiguration zu aktualisieren. Weitere Informationen finden Sie unter OAuth 2.0 Authentifizierung und Autorisierung für Amazon MQ for RabbitMQ.

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 MQ-API-Aktionen beschrieben, die derzeit Berechtigungen auf ARNs Ressourcenebene unterstützen, sowie die unterstützten Ressourcen, Ressourcen- 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*

API-Aktion	Ressourcentypen (*erforderlich)
DescribeUser	Broker*
ListConfigurationRevisions	Konfigurationen*
ListConfigurationRevisions	Konfigurationen*
ListTags	Broker, Konfigurationen
ListUsers	Broker*
RebootBroker	Broker*
UpdateBroker	Broker*
UpdateConfiguration	Konfigurationen*
UpdateUser	Broker*

Authentifizierung und Autorisierung von Brokern

Amazon MQ bietet je nach Broker-Engine-Typ unterschiedliche Authentifizierungs- und Autorisierungsmethoden.

Authentifizierung und Autorisierung für Amazon MQ for ActiveMQ

Amazon MQ for ActiveMQ unterstützt die folgenden Authentifizierungs- und Autorisierungsmethoden:

Einfache Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer über die Amazon MQ MQ-Konsole oder API erstellt und verwaltet. Benutzern können spezifische Berechtigungen für den Zugriff auf Warteschlangen, Themen und die ActiveMQ Web Console zugewiesen werden. Weitere Informationen zu dieser Methode finden Sie unter ActiveMQ-Broker-Benutzer erstellen.

LDAP-Authentifizierung und -Autorisierung

Bei dieser Methode authentifizieren sich Broker-Benutzer anhand der auf Ihrem LDAP-Server gespeicherten Anmeldeinformationen. Über den LDAP-Server können Sie Benutzer hinzufügen, löschen und ändern sowie Themen und Warteschlangen Berechtigungen zuweisen, sodass eine zentrale Authentifizierung und Autorisierung gewährleistet ist. Weitere Informationen zu dieser Methode finden Sie unter ActiveMQ-Broker mit LDAP integrieren.

Authentifizierung und Autorisierung für Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ unterstützt die folgenden Authentifizierungs- und Autorisierungsmethoden:

Einfache Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer intern im RabbitMQ-Broker gespeichert und über die Webkonsole oder die Management-API verwaltet. Berechtigungen für Vhosts, Exchanges, Warteschlangen und Themen werden direkt in RabbitMQ konfiguriert. Dies ist die Standardmethode. Weitere Informationen finden Sie unter Einfache Authentifizierung und Autorisierung.

OAuth 2.0 Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer und ihre Berechtigungen von einem externen OAuth 2.0-Identitätsanbieter (IdP) verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen für Vhosts, Exchanges, Warteschlangen und Themen werden über das Scope-System des OAuth 2.0-Anbieters zentralisiert. Dies vereinfacht die Benutzerverwaltung und ermöglicht die Integration in bestehende Identitätssysteme. Weitere Informationen finden Sie unter Authentifizierung und Autorisierung OAuth 2.0.

IAM-Authentifizierung und -Autorisierung

Bei dieser Methode authentifizieren sich Broker-Benutzer mithilfe von AWS IAM-Anmeldeinformationen über den IAM-Outbound-Federation. IAM-Anmeldeinformationen werden verwendet, um JWT-Token vom AWS Security Token Service (STS) abzurufen, und diese JWT-Token dienen als 2.0-Token für die Authentifizierung. OAuth Diese Methode nutzt die bestehende OAuth 2.0-Unterstützung in Amazon MQ für RabbitMQ, wo sie als 2.0-Identitätsanbieter AWS fungiert. OAuth Die Benutzerauthentifizierung wird von AWS IAM abgewickelt, während die Ressourcenberechtigungen für Vhosts, Exchanges, Warteschlangen und Themen über in RabbitMQ konfigurierte IAM-Richtlinien und Bereichsaliase verwaltet werden. Weitere Informationen finden Sie unter IAM-Authentifizierung und -Autorisierung.

LDAP-Authentifizierung und -Autorisierung

Bei dieser Methode werden Broker-Benutzer und ihre Berechtigungen von einem externen LDAP-Verzeichnisdienst verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen werden über den LDAP-Server zentralisiert, sodass Benutzer mit ihren vorhandenen Verzeichnisdienstanmeldedaten auf RabbitMQ zugreifen können. Weitere Informationen finden Sie unter LDAP-Authentifizierung und -Autorisierung.

HTTP-Authentifizierung und Autorisierung

Bei dieser Methode werden Broker-Benutzer und ihre Berechtigungen von einem externen HTTP-Server verwaltet. Benutzerauthentifizierung und Ressourcenberechtigungen werden über den HTTP-Server zentralisiert, sodass Benutzer über ihren eigenen Authentifizierungs- und Autorisierungsanbieter auf RabbitMQ zugreifen können. Weitere Informationen zu dieser Methode finden Sie unter HTTP-Authentifizierung und Autorisierung.

Authentifizierung mit SSL-Zertifikaten

Amazon MQ unterstützt Mutual TLS (mTLS) für RabbitMQ-Broker. Das SSL-Authentifizierungs-Plugin verwendet Client-Zertifikate von mTLS-Verbindungen, um Benutzer zu authentifizieren. Bei dieser Methode werden Broker-Benutzer mithilfe von X.509-Clientzertifikaten anstelle von Benutzernamen und Kennwörtern authentifiziert. Das Zertifikat des Clients wird anhand einer vertrauenswürdigen Zertifizierungsstelle (CA) validiert, und der Benutzername wird aus einem Feld im Zertifikat extrahiert, z. B. dem Common Name (CN) oder dem Subject Alternative Name (SAN). Diese Methode bietet eine starke Authentifizierung, ohne dass Anmeldeinformationen über das Netzwerk übertragen werden müssen. Weitere Informationen finden Sie unter SSL-Zertifikatsauthentifizierung.

> **Note**
>
> RabbitMQ unterstützt mehrere Authentifizierungs- und Autorisierungsmethoden, die gleichzeitig verwendet werden können. Sie können beispielsweise sowohl OAuth 2.0 als auch die einfache (interne) Authentifizierung aktivieren. Weitere Informationen finden Sie im OAuth 2.0-Tutorial-Abschnitt zur Aktivierung sowohl der OAuth 2.0-Authentifizierung als auch der einfachen (internen) Authentifizierung sowie in der Dokumentation zur RabbitMQ-Zugriffskontrolle.
>
> Amazon MQ empfiehlt, beim Testen von Authentifizierungskonfigurationen einen internen Benutzer zu erstellen. Auf diese Weise kann die Zugriffskonfiguration mithilfe der RabbitMQ-Management-API validiert werden. Weitere Informationen finden Sie unter Zugriffsvalidierung.

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 vom Kunden verwaltete 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 unter Von AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

Amazon MQ unterstützt die folgenden AWS verwalteten Richtlinien:

- AmazonMQApiFullAccess
- AmazonMQApiReadOnlyAccess
- MQFullZugriff auf Amazon
- AmazonMQReadOnlyAccess
- AmazonMQServiceRolePolicy

AWS verwaltete Richtlinie: Amazon MQService RolePolicy

Sie können nicht anhängen`AmazonMQServiceRolePolicy`an Ihre IAM-Entitäten. 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 Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf der Amazon MQ-Dokumentverlauf-Seite.

Änderungen	Beschreibung	Date
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. Servicebezogene Rollen sind von Amazon MQ vordefiniert und beinhalten alle Berechtigungen, die der Service benötigt, um andere AWS Services in Ihrem Namen aufzurufen.

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 serviceverknüpften Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Amazon MQ

Amazon MQ verwendet die serviceverknüpfte Rolle namens AWSServiceRoleForAmazonMQ — Amazon MQ verwendet diese servicebezogene Rolle, um Services in Ihrem Namen aufzurufen AWS .

Die dienstverknüpfte AWSService RoleForAmazon MQ-Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `mq.amazonaws.com`

Amazon MQ verwendet die Berechtigungsrichtlinie `AmazonMQServiceRolePolicy`, die der mit dem AWSService RoleForAmazon MQ-Dienst verknüpften Rolle zugeordnet ist, 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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:vpc/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:CreateVpcEndpoint"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:vpc-endpoint/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AMQManaged": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateVpcEndpoint"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteVpcEndpoints"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AMQManaged": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup"
            ],
```

```
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/amazonmq/*"
            ]
        }
    ]
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte 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 serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie zum ersten Mal einen Broker erstellen, erstellt Amazon MQ eine servicebezogene Rolle, um AWS Services in Ihrem Namen anzurufen. 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 Features 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 serviceverknüpfte Rolle mit dem `mq.amazonaws.com` Servicenamen. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch. Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

> **Important**
>
> Service Linked Roles werden nur für Amazon MQ for RabbitMQ erstellt.

Bearbeiten einer serviceverknüpften Rolle für Amazon MQ

Amazon MQ erlaubt es Ihnen nicht, die mit dem AWSService RoleForAmazon MQ-Dienst verknüpfte Rolle 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 ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften 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 serviceverknüpften 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 dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um vom MQ verwendete Amazon MQ-Ressourcen zu löschen AWSService RoleForAmazon

- Löschen Sie Ihre Amazon MQ-Broker mithilfe der AWS-Managementkonsole Amazon MQ CLI oder der Amazon MQ MQ-API. Weitere Informationen zum Löschen von Brokern finden Sie unter ???.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die mit dem AWSService RoleForAmazon MQ-Dienst verknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften 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
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

Diagnostizieren und beheben Sie mithilfe der folgenden Informationen gängige Probleme, die bei der Verwendung von Amazon MQ und IAM auftreten können.

Themen

Ich bin nicht autorisiert, eine Aktion in Amazon MQ auszuführen

Wenn Ihnen AWS-Managementkonsole mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion auszuführen, 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 anzuzeigen, *`widget`* aber nicht über die `mq:`*`GetWidget`* entsprechenden Berechtigungen verfügt.

```
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 auszuführen: 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, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen 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 Funktionsweise von Amazon MQ mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen in AWS-Konten Ihrem Besitz gewähren können, finden Sie im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto , dem Sie gehören.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter Gewähren von Zugriff für extern authentifizierte Benutzer (Identitätsverbund) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Compliance-Validierung für Amazon MQ

Externe Prüfer bewerten die Sicherheit und Konformität von Amazon MQ im Rahmen mehrerer AWS Compliance-Programme. Zu diesen Programmen gehören SOC, PCI, HIPAA und andere.

Informationen darüber, ob AWS-Service ein in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter AWS-Services 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. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der AWS Sicherheitsdokumentation.

Ausfallsicherheit bei Amazon MQ

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander 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 verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Infrastruktursicherheit in Amazon MQ

Als verwalteter Service ist Amazon MQ 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, um über das Netzwerk auf Amazon MQ zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Best Practices für die Sicherheit in Amazon MQ

Die folgenden Entwurfsmuster können die Sicherheit Ihres Amazon MQ-Broker verbessern.

Themen

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. Dadurch wird die Anfälligkeit Ihres Brokers für Distributed Denial of Service (DDoS) -Angriffe aus dem öffentlichen Internet erheblich reduziert. Weitere Informationen finden Sie im Sicherheitsblog unter So bereiten Sie sich auf DDo S-Angriffe vor, indem Sie Ihre Angriffsfläche reduzieren. AWS

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 für private Broker zu verbessern, sollten Sie die Verbindungen unnötiger Protokolle und Ports einschränken, indem Sie Ihre Amazon VPC-Sicherheitsgruppe ordnungsgemäß konfigurieren. Um beispielsweise den Zugriff auf die meisten Protokolle einzuschränken OpenWire und gleichzeitig den Zugriff auf die Webkonsole zu ermöglichen, könnten Sie nur den Zugriff auf 61617 und 8162 zulassen. Auf diese Weise wird Ihr Risiko begrenzt, indem Protokolle blockiert werden, die Sie nicht verwenden, während gleichzeitig die normale OpenWire Funktion der Webkonsole gewährleistet wird.

Erlauben Sie nur die Protokoll-Ports, die Sie verwenden.

- AMQP: 5671
- MQTT: 8883
- OpenWire: 61617
- STOMP: 61614
- WebSocket: 61619

Weitere Informationen finden Sie unter:

- 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 damit Kennzahlen für Ihren Amazon MQ-Broker anzeigen und 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 abgerufen 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 der AWS-Managementkonsole AWS CLI, und API auf CloudWatch Metriken zugreifen.

Möglicherweise möchten Sie auf CloudWatch Metriken zugreifen, ohne die zu verwenden AWS-Managementkonsole.

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 der CloudWatch API auf Amazon MQ-Metriken zuzugreifen. Weitere Informationen finden Sie unter Get Statistics for a Metric im CloudWatch Amazon-Benutzerhandbuch.

Zugreifen auf CloudWatch Metriken mit dem AWS-Managementkonsole

Das folgende Beispiel zeigt Ihnen, wie Sie mit dem auf CloudWatch Metriken für Amazon MQ zugreifen können. AWS-Managementkonsole 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 CloudWatch-Konsole an.
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	Description
`AmqpMaximumConnections`	Anzahl	Die maximale Anzahl von Clients, die Sie über AMQP mit Ihrem Broker verbinden

Metrik	Einheit	Description
		können. Weitere Informationen zu Verbindungskontingenten finden Sie unter Quotas in Amazon MQ.
`BurstBalance`	Prozent	Der Anteil der verbleibenden Burst-Gutschriften auf dem Amazon EBS-Volumen, das zum Persistieren von Nachrichtendaten für durchsatzoptimierte Broker verwendet wird. Wenn dieses Guthaben Null erreicht, sinkt die vom Amazon EBS Volumen bereitgestellten IOPS, bis das Burst-Guthaben wieder aufgefüllt wird. Weitere Informationen zur Funktionsweise von Burst-Gutschriften in Amazon EBS finden Sie unter I/O-Guthaben und Burst-Performance.

Metrik	Einheit	Description
`CpuCreditBalance`	Guthaben (vCPU-Minuten)	**Important** Diese Metrik ist nur für den Broker-Instance-Typ `mq.t2.micro` verfügbar. Die Metriken für CPU-Guthaben sind nur mit fünfminütigen Intervallen verfügbar. Die Anzahl verdienter CPU-Guthaben, die eine Instance angesammelt hat, seit sie gestartet wurde (einschließlich der Anzahl der Start-Guthaben). Das Guthabenkonto ist für die Broker-Instance zur Verwendung für über die Basis-CPU-Nutzung hinausgehende Bursts verfügbar. 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	Description
`CpuUtilization`	Prozent	Der Prozentsatz der zugewiesenen Amazon EC2-Recheneinheiten, die zurzeit vom Broker genutzt werden.
`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 prozentuale Anteil am ActiveMQ JVM-Speicherlimit, der vom Broker derzeit genutzt wird.
`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.

Metrik	Einheit	Description
`JournalFilesForFullRecovery`	Anzahl	Die Anzahl der Journaldateien, die nach einem unsauberen Shutdown erneut abgespielt werden.
`MqttMaximumConnections`	Anzahl	Die maximale Anzahl von Clients, die Sie über MQTT mit Ihrem Broker verbinden können. 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.

Metrik	Einheit	Description
`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.
`StompMaximumConnections`	Anzahl	Die maximale Anzahl von Clients, die Sie über STOMP mit Ihrem Broker verbinden können. 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.

Metrik	Einheit	Description
`TotalProducerCount`	Anzahl	Die Gesamtzahl der Nachricht enproduzenten, die auf Zielen auf dem aktuellen Broker aktiv sind.
`VolumeReadOps`	Anzahl	Die Anzahl der auf dem Amazon EBS-Volumen ausgeführten Lesevorgänge.
`VolumeWriteOps`	Anzahl	Die Anzahl der auf dem Amazon EBS-Volumen ausgeführten Schreibvo rgänge.
`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	Description
`Broker`	Der Name des Brokers **Note** Ein Broker mit einer einzigen Instance hat das Suffix -1. Ein active/standby Broker für Hochverfügbarkeit hat die

Dimension	Description
	Suffixe -1 und -2 für sein redundantes Paar.

ActiveMQ-Ziel-Metriken (Warteschlange und Thema)

Important

Die folgenden Metriken 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	Description
`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

Metrik	Einheit	Description
		Nachricht bei einem Broker und ihrer Zustellung an einen Verbraucher. **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.

Metrik	Einheit	Description
`DispatchCount`	Anzahl	Die Anzahl der Nachrichten pro Minute, die an Verbrauch er gesendet wurden.
`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.

Metrik	Einheit	Description
`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	Description
`Broker`	Der Name des Brokers. **Note** Ein Broker mit einer einzigen Instance hat das Suffix `-1`. Ein active/standby Broker für Hochverfügbarkeit 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	Description
`ExchangeCount`	Anzahl	Die Gesamtzahl der auf dem Broker konfigurierten Börsen.
`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 Nachricht en in der Warteschlange. **Note** Die produzierte Anzahl ist die Gesamtsumme der bereitgestellten und unerkannten Nachrichten auf dem Broker.

Metrik	Einheit	Description
`MessageReadyCount`	Anzahl	Die Gesamtzahl der bereitgestellten Nachrichten in den Warteschlangen.
`MessageUnacknowledgedCount`	Anzahl	Die Gesamtzahl der nicht bestätigten Nachrichten in den Warteschlangen.
`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 veröffentlichte 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.

Metrik	Einheit	Description
AckRate	Anzahl	Die Rate, mit der Nachrichten von den Verbrauchern anerkannt werden. Die erzeugte Zahl stellt die Anzahl der Nachrichten pro Sekunde zum Zeitpunkt der Probenahme dar.
SystemCpuUtilization	Prozent	Der Prozentsatz der zugewiesenen Amazon EC2-Recheneinheiten, die zurzeit vom Broker genutzt werden. 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 Volume des RAM, das von einem RabbitMQ-Broker verwendet wird. Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar.

Metrik	Einheit	Description
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.
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.
RabbitMQFdUsed	Anzahl	Anzahl der verwendeten Datei-Deskriptoren Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar.

Metrik	Einheit	Description
`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	Description
`Broker`	Der Name des Brokers.

RabbitMQ-Knoten-Metriken

Metrik	Einheit	Description
`SystemCpuUtilization`	Prozent	Der Prozentsatz der zugewiesenen Amazon EC2-Recheneinheiten, die zurzeit vom Broker genutzt werden.
`RabbitMQMemLimit`	Bytes	Das RAM-Limit für einen RabbitMQ-Knoten.
`RabbitMQMemUsed`	Bytes	Das Volumen des RAM, das von einem RabbitMQ-Knoten verwendet wird. Wenn der Speicherverbrauch über das

Metrik	Einheit	Description
		Limit hinausgeht, blockiert der Cluster alle Herstellerverbindungen.
`RabbitMQDiskFreeLimit`	Bytes	Das Festplattenlimit für einen RabbitMQ-Knoten. Diese Metrik unterscheidet sich je nach Instance-Größe.
`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	Description
`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`

Dimension	Description
	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	Description
`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.
`MessageCount`	Anzahl	Die Gesamtzahl für `MessageReadyCount` und `MessageUnacknowledgedCount` (auch als Warteschlangentiefe bezeichnet).

Dimensionen für RabbitMQ-Queue-Metriken

> **Note**
>
> Amazon MQ für RabbitMQ veröffentlicht keine Metriken für virtuelle Hosts und Warteschlangen mit Namen, die Leerzeichen, Registerkarten oder andere Nicht-ASCII-Zeichen enthalten.
> Weitere Informationen zu Dimensionsnamen finden Sie unter Dimension in der Amazon CloudWatch API-Referenz.

Dimension	Description
`Queue`	Der Name der Warteschlange.
`VirtualHost`	Der Name des virtuellen Hosts.
`Broker`	Der Name des Brokers.

RabbitMQ-Netzwerkmetriken

Metrik	Einheit	Description
`NetworkOut`	Bytes	Anzahl der von der Instance auf allen Netzwerkschnittstellen gesendeten Byte. Diese Metrik gibt das an eine einzelne Instance ausgehende Netzwerkdatenvolumen an. Der ermittelte Wert ist die Anzahl der während des Zeitraums gesendeten Bytes. Wenn Sie eine grundlegene Überwachung (fünf Minuten) verwenden und die Statistik eine Summe ist, können Sie diese Zahl durch 300 teilen, um Bytes/Sekunde zu ermitteln. Wenn Sie eine detaillierte (einminütige) Überwachung haben und die Statistik eine Summe ist, teilen Sie sie durch 60. Sie können auch die mathematische Funktion CloudWatch Metrik verwenden`DIFF_TIME` , um die Byte pro Sekunde zu ermitteln. Wenn Sie beispielsweise CloudWatch als grafisch dargestellt NetworkOut haben`m1`,

Metrik	Einheit	Description
		gibt die mathematische Formel die Metrik in Byte/Sekunde `m1/(DIFF_TIME(m1))` zurück. Weitere Informationen zu `DIFF_TIME` und anderen metrischen mathematischen Funktionen finden Sie unter Metrische Mathematik verwenden. Aussagekräftige Statistiken: Summe, Durchschnitt, Minimum, Maximum
`NetworkIn`	Bytes	Anzahl der von der Instance auf allen Netzwerkschnittstellen empfangenen Byte. Diese Metrik gibt das an eine einzelne Instance eingehende Netzwerkdatenvolumen an. Der ermittelte Wert ist die Anzahl der während des Zeitraums empfangenen Byte. Wenn Sie eine grundlegene Überwachung (fünf Minuten) verwenden und die Statistik eine Summe ist, können Sie diese Zahl durch 300 teilen, um Bytes/Sekunde zu ermitteln. Wenn Sie eine detaillierte (einminütige) Überwachung haben und die Statistik eine Summe ist, teilen Sie sie durch 60. Sie können auch die mathematische Funktion CloudWatch Metrik verwenden`DIFF_TIME` , um die Byte pro Sekunde zu ermitteln. Wenn Sie beispielsweise CloudWatch als grafisch dargestellt NetworkIn haben`m1`, gibt die mathematische Formel die Metrik in Byte/Sekunde `m1/(DIFF_TIME(m1))` zurück. Weitere Informationen zu `DIFF_TIME` und anderen metrischen mathematischen Funktionen finden Sie unter Metrische Mathematik verwenden. Aussagekräftige Statistiken: Summe, Durchschnitt, Minimum, Maximum

Abmessungen für RabbitMQ-Broker

Dimension	Description
`BrokerId`	ID des Maklers

Konfigurieren von Amazon MQ für RabbitMQ-Protokolle

Wenn Sie die CloudWatch Protokollierung für Ihre RabbitMQ-Broker aktivieren, verwendet Amazon MQ eine servicebezogene 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 MQ-API-Aufrufen 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-Aufrufe 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 eine Amazon MQ MQ-API, 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-Managementkonsole, 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 Amazon SNS SNS-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 Anforderungsparameter `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 CloudTrailUserIdentity 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 Anforderung von einer beliebigen Quelle dar und enthält Informationen über die Anforderung an eine Amazon MQ-API, die IP-Adresse des Anforderers, die Identität des Anforderers, das Datum und die Uhrzeit der Anforderung und so weiter.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag für einen `CreateBroker`API-Aufruf.

> **Note**
>
> Da es CloudTrail sich bei Protokolldateien nicht um einen geordneten öffentlichen Stack-Trace handelt APIs, 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
- Hinzufügen der CreateLogGroup-Berechtigung zu Ihrem Amazon-MQ-Benutzer
- Konfigurieren einer ressourcenbasierten Richtlinie für Amazon MQ.
- Serviceübergreifende Confused-Deputy-Prävention

Grundlegendes zur Struktur der Protokollierung von Protokollen CloudWatch

Sie können die allgemeine Protokollierung und die Audit-Protokollierung aktivieren, wenn Sie erweiterte Broker-Einstellungen konfigurieren, wenn Sie einen Broker erstellen oder wenn Sie einen Broker bearbeiten.

Die allgemeine Protokollierung aktiviert die `INFO` Standardprotokollierungsebene (die `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 der ActiveMQ Web Console durchgeführt wurden, und veröffentlicht `audit.log` sie in einer Protokollgruppe in Ihrem Konto. CloudWatch 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.

Das folgende BeispielIAM-basierte RichtlinieErteilen der Berechtigung für`logs:CreateLogGroup`für Benutzer, denen diese Richtlinie angehängt ist.

JSON

```
{
                    "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 zum Einrichten von Benutzern und zum Konfigurieren von IAM-Richtlinien finden Sie unter Identitätsverwaltung im Überblick im IAM-Benutzerhandbuch.

Weitere Informationen finden Sie `CreateLogGroup` in der Amazon CloudWatch Logs API-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 CloudWatch Logs-API-Aktionen zu gewähren:

- `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

JSON

```
{
                        "Version":"2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": { "Service":
 "mq.amazonaws.com" },
                                "Action": [ "logs:CreateLogStream",
 "logs:PutLogEvents" ],
                                "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*"
                            }
                        ]
                    }
```

Diese ressourcenbasierte Richtlinie muss mithilfe des, AWS CLI wie im folgenden Befehl gezeigt, konfiguriert werden. 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.

JSON

```
{
                        "Version":"2012-10-17",
                        "Statement": [
                            {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": "mq.amazonaws.com"
                            },
                            "Action": [
                                "logs:CreateLogStream",
                                "logs:PutLogEvents"
                            ],
                            "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*",
                            "Condition": {
                                "StringEquals": {
                                "aws:SourceAccount": "123456789012",
                                "aws:SourceArn": "arn:aws:mq:us-
west-1:123456789012:broker:my-broker:123456789012"
                                }
                            }
                            }
                        ]
                        }
```

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.

JSON

```
{
                            "Version":"2012-10-17",
                            "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                "Service": [
                                    "mq.amazonaws.com"
```

```
                                ]
                                },
                                "Action": [
                                "logs:CreateLogStream",
                                "logs:PutLogEvents"
                                ],
                                "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*",
                                "Condition": {
                                "ArnLike": {
                                    "aws:SourceArn":
 "arn:aws:mq:*:123456789012:broker:*"
                                },
                                "StringEquals": {
                                    "aws:SourceAccount": "123456789012"
                                }
                                }
                            }
                            ]
                        }
```

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. • Es darf nur Zeichen aus den darstellbaren ASCII-Zeichen enthalten. • 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.

Limit	Beschreibung
Konfigurationsname	• Er muss 1–150 Zeichen umfassen. •

Limit	Beschreibung
	Es darf nur Zeichen aus den darstellbaren ASCII-Zeichen enthalten. • 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.

Limit	Beschreibung
Username	• Er muss 1–100 Zeichen umfassen. • Es darf nur Zeichen aus den darstellbaren ASCII-Zeichen enthalten. • Er darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten. • Er darf keine Kommas enthalten. (,).
Passwort	• Es muss 12–250 Zeichen umfassen. • Es darf nur Zeichen aus den darstellbaren ASCII-Zeichen enthalten. • Es muss mindestens 4 eindeutige Zeichen enthalten.

Limit	Beschreibung
	• 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.m5.*` Instance-Typ-Broker. Weitere Informationen zu Amazon MQ Instance-Typen finden Sie unter Broker instance types.
Job-Scheduler-Nutzungslimit pro Broker gestützt von Amazon EBS	**Important** Gilt nicht für RabbitMQ-Broker. 50 GB. Weitere Informationen zur Verwendung der Job-Planer finden Sie unter `JobSchedulerUsage` imApache ActiveMQ -API-Dokumentation.

Limit	Beschreibung
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.m5.*` Instance-Typ-Broker.

API-Drosselung

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 REST API-Referenz.

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.

API-Aufrust-Limit	API-Ratenlimits
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. Allgemeine Informationen zur Fehlerbehebung finden Sie unter. the section called “Fehlerbehebung: Allgemeines Amazon MQ” Informationen zur Fehlerbehebung für Ihre spezifische Engine-Version finden Sie in den folgenden Abschnitten.

Fehlerbehebung bei ActiveMQ auf Amazon MQ

Thema Fehlerbehebung	Description
Allgemeine Problembehebung	Verwenden Sie die Informationen in diesem Abschnitt, um häufig auftretende Probleme zu diagnostizieren und zu lösen, die bei der Arbeit mit ActiveMQ auf Amazon MQ-Brokern auftreten können.
BROKER_ENI_DELETED	ActiveMQ auf Amazon MQ `BROKER_ENI_DELETED` löst einen Alarm aus, wenn Sie das Elastic Network Interface (ENI) eines Brokers löschen.
BROKER_OOM	ActiveMQ auf Amazon MQ löst einen BROKER_OOM-Alarm aus, wenn der Broker aufgrund unzureichender Speicherkapazität eine Neustartschleife durchläuft.

Fehlerbehebung bei RabbitMQ auf Amazon MQ

Thema Fehlerbehebung	Description
Allgemeine Problembehebung	Diagnostizieren Sie häufig auftretende Probleme, auf die Sie bei der Arbeit mit

Thema Fehlerbehebung	Description
	RabbitMQ-Brokern stoßen könnten.
RABBITMQ_MEMORY_ALARM	RabbitMQ löst einen hohen Speicheralarm aus, wenn die Speicherauslastung des Brokers, die anhand der CloudWatch Metrik `RabbitMQMemUsed` identifiziert wird, das Speicherlimit überschreitet, das durch identifiziert wurde. `RabbitMQMemLimit`
RABBITMQ_INVALID_KMS_KEY	RabbitMQ auf Amazon MQ gibt den Code INVALID_KMS_KEY aus, der für eine kritische Aktion erforderlich ist, wenn ein Broker, der mit einem vom Kunden verwalteten AWS KMS key(CMK) erstellt wurde, feststellt, dass der (KMS) -Schlüssel deaktiviert ist. AWS Key Management Service
RABBITMQ_INVALID_ASSUMEROLE	RabbitMQ auf Amazon MQ löst den Code INVALID_ASSUMEROLE aus, der für eine kritische Aktion erforderlich ist, wenn der in angegebene IAM-Rollen-ARN nicht von Amazon MQ übernommen werden kann. `aws.arns.assume_role_arn`

Thema Fehlerbehebung	Description
RABBITMQ_INVALID_ARN_LDAP	RabbitMQ auf Amazon MQ gibt den Code INVALID_ARN_LDAP für eine kritische Aktion erforderlich aus, wenn der ARN für das Passwort des LDAP-Dienstkontos ungültig oder nicht zugänglich ist.
RABBITMQ_INVALID_ARN_HTTP	RabbitMQ auf Amazon MQ gibt den Code INVALID_ARN_HTTP für eine kritische Aktion erforderlich aus, wenn eines oder mehrere ARNs SSL-Zertifikate oder die Schlüsseldatei für HTTP auth_backend ungültig oder nicht zugänglich sind.
RABBITMQ_INVALID_ARN_SSL	RabbitMQ auf Amazon MQ löst den Code INVALID_ARN_SSL aus, der für kritische Aktionen erforderlich ist, wenn ein oder mehrere CA-Zertifikats-Truststore für EXTERNAL ARNs auth_mechanism ungültig oder nicht zugänglich sind.

Thema Fehlerbehebung	Description
RABBITMQ_INVALID_ARN	RabbitMQ auf Amazon MQ gibt den Code INVALID_ARN aus, der für eine kritische Aktion erforderlich ist, wenn einer oder mehrere Punkte ARNs in der Broker-Konfiguration ungültig oder nicht zugänglich sind.
RABBITMQ_DISK_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.

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, eine Verbindung zu Ihrem Broker über einen FIPS-Endpunkt zu erstellen. Amazon MQ unterstützt FIPS-Endpunkte nur bei Verwendung von API-Operationen und nicht für Verbindungen auf Wire-Level mit der Broker-Instance selbst.
3. Überprüfen Sie, ob die Option Public Accessibility (öffentliche Zugänglichkeit) für Ihren Broker auf Yes (Ja) gestellt ist. Wenn dies auf Nein gestellt ist, überprüfen Sie das Netzwerk Ihres SubnetzesZugriffskontrolllisten (ACL)-Regeln. Wenn Sie ein benutzerdefiniertes Netzwerk erstellt haben ACLs, müssen Sie möglicherweise die Netzwerk-ACL-Regeln ändern, um Zugriff auf Ihren Broker zu gewähren. Weitere Informationen zum Amazon VPC-Netzwerk finden Sie unter Aktivieren des Internetzugangs im Amazon VPC-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 Engine-Typen gruppiert, da ActiveMQ auf Amazon MQ und RabbitMQ auf Amazon MQ unterschiedliche Ports für Verbindungen verwenden.

   ActiveMQ auf Amazon MQ

   - Webkonsole — Port8162
   - OpenWire — Hafen 61617
   - AMQP — Port5671
   - STOMP – Portierung von 61614
   - MQT-Port8883
   - WSS — Port61619

RabbitMQ auf Amazon MQ

- Webkonsole und Verwaltungs-API — Port `443`und`15671`
- AMQP — Port`5671`

5. Führen Sie die folgenden Netzwerkkonnektivitätstests für Ihren Broker-Engine-Typ .

> **Note**
>
> Führen Sie für Broker ohne öffentliche Zugänglichkeit die Tests von einer Amazon EC2 Instance innerhalb derselben Amazon VPC aus wie Ihr Amazon MQ-Broker aus und bewerten Sie die Antworten.

ActiveMQ on Amazon MQ

Um die Netzwerkkonnektivität Ihres ActiveMQ on Amazon MQ Brokers zu testen

1. Öffnen Sie ein Terminal-Fenster oder eine Eingabeaufforderung.
2. Führen Sie Folgendes aus:`nslookup`, um Ihren Broker-DNS-Eintrag abzufragen. Füraktive/Standby-Funktion-Bereitstellungen verwenden, testen Sie sowohl die aktiven als auch die Standby-Endpunkte. Die active/standby Endpunkte werden mit einem Suffix identifiziert `-1` oder der eindeutigen `-2` 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 bedeutet, dass der Domänenname 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. *`port`*Ersetzen Sie diese durch die Portnummer `8162` für die Webkonsole oder durch andere Anschlüsse auf Kabelebene, um bei Bedarf weitere Protokolle zu testen.

   > **Note**
   >
   > Bei active/standby Bereitstellungen erhalten Sie eine `Connect failed` Fehlermeldung, wenn Sie den Standby-Endpunkt `telnet` verwenden. Dies wird erwartet, da die Standby-Instance selbst ausgeführt wird, der ActiveMQ Prozess jedoch nicht ausgeführt wird und keinen Zugriff auf das Amazon EFS Speichervolume 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 eines Amazon VPC Flow-Protokollsfür 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 VerbindungspaketeACCEPTED, einschließlich eines Rücksendepakets. Weitere Informationen und ein Beispiel für ein Flow Log finden Sie unterBeispiele für Flow-Protokolldatensätzeim Entwicklerhandbuch für Amazon VPC.

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
```

Wird der Befehl erfolgreich ausgeführt, sollte es sich bei der Ausgabe um ein HTML-Dokument handeln, 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>
        ...
```

RabbitMQ on Amazon MQ

Um die Netzwerkkonnektivität Ihres RabbitMQ auf dem Amazon MQ-Broker zu testen

1. Öffnen Sie ein Terminal-Fenster oder eine Eingabeaufforderung.
2. Führen Sie Folgendes aus:`nslookup`, um Ihren Broker-DNS-Eintrag 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 ihn durch einen Port 443 für die Webkonsole *`port`* ersetzen und die AMQP-Verbindung 5671 auf Wire-Level testen.

```
$ 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 eines Amazon VPC Flow-Protokollsfür jede Netzwerkschnittstelle und überprüfen Sie die generierten Flussprotokolle. Suchen Sie nach den privaten IP-Adressen des Brokers, wenn der`telnet`-Befehl aufgerufen wurde, und bestätigen Sie, dass die Verbindungspakete`ACCEPTED`, einschließlich eines Rücksendepakets. Weitere Informationen und ein Beispiel für ein Flow Log finden Sie unterBeispiele für Flow-ProtokolldatensätzeimEntwicklerhandbuch für Amazon VPC.

> **Note**
>
> Dieser Schritt gilt nicht für öffentlich zugängliche RabbitMQ on Amazon MQ-Broker.

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
```

Wird der Befehl erfolgreich ausgeführt, sollte es sich bei der Ausgabe um ein HTML-Dokument handeln, 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 **`telnet`** bestätigen, 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.

- Überprüfen Sie Ihre IAM-Berechtigungen. Um einen Broker zu erstellen, müssen Sie entweder die AWS verwaltete IAM-Richtlinie verwenden `AmazonMQFullAccess` oder über die richtigen Amazon EC2 EC2-Berechtigungen in Ihrer benutzerdefinierten IAM-Richtlinie verfügen. Weitere Informationen zu den erforderlichen Amazon EC2 Berechtigungen finden Sie unterIAM-Berechtigungen erforderlich, um einen Amazon MQ-Broker zu erstellenaus.
- Überprüfen Sie, ob sich das Subnetz, das Sie für Ihren Broker auswählen, in einer freigegebenen Amazon Virtual Private Cloud (VPC) befindet. Um einen Amazon MQ-Broker in einer freigegebenen Amazon VPC zu erstellen, müssen Sie ihn in dem Konto erstellen, das Eigentümer der Amazon VPC ist.

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 zu den Wartungsfenstern für Broker 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.t3.micro` kann dazu führen, dass dem Broker keine Ressourcen mehr zur Verfügung stehen. Eine hohe CPU-Auslastung oder eine hohe Broker-Speicherauslastung kann dazu führen, dass ein Broker unerwartet neu gestartet wird. Verwenden Sie die folgenden CloudWatch Messwerte für Ihren Engine-Typ, um zu sehen, wie viel CPU und Arbeitsspeicher von Ihrem Broker genutzt werden.
  - ActiveMQ auf Amazon MQ — Prüfen Sie `CpuUtilization` den Prozentsatz der zugewiesenen Amazon EC2 EC2-Recheneinheiten, die der Broker derzeit verwendet. Überprüfen `HeapUsage`Sie den Prozentsatz des ActiveMQ JVM-Speicherlimits, den der Broker derzeit verwendet.
  - RabbitMQ auf Amazon MQ — Prüfen Sie `SystemCpuUtilization` den Prozentsatz der zugewiesenen Amazon EC2 EC2-Recheneinheiten, die der Broker derzeit verwendet. Check`RabbitMQMemUsed`für das Volumen des in Bytes verwendeten RAM und dividieren durch`RabbitMQMemLimit`für den Prozentsatz des Speichers, der vom RabbitMQ-Knoten verwendet wird.

  Weitere Informationen zu Broker-Instance-Typen und zur Auswahl des richtigen Instance-Typs für Ihren Workload finden Sie unter. Broker instance types

Fehlerbehebung bei ActiveMQ auf Amazon MQ

Verwenden Sie die Informationen in diesem Abschnitt, um häufig auftretende Probleme zu diagnostizieren und zu lösen, die bei der Arbeit mit ActiveMQ auf Amazon MQ-Brokern auftreten können.

Inhalt

- Ich sehe beim Ausführen von Operationen die Ausnahme org.apache.jasper.JasperException: An exception occurred processing JSP page auf der ActiveMQ-Konsole.

Ich kann in Logs keine allgemeinen Logs oder Audit-Logs für meinen Broker sehen, obwohl ich die CloudWatch 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 CloudWatch Logs-API-Aktionen zu gewähren:
   - `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 ActiveMQ auf Amazon MQ für die 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 Broker-Neustart höchstwahrscheinlich durch eine ungewöhnlich große Anzahl von Nachrichten verursacht, die im Amazon-EFS- oder Amazon-EBS-Speichervolumen Ihres Brokers bestehen. 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-Logs für Ihren ActiveMQ on Amazon MQ-Broker unter Logs. CloudWatch Wenn das Limit erreicht wurde, sehen Sie `Reached Maximum Connections` in den Broker-Protokollen. Weitere Informationen zu CloudWatch Protokollen für ActiveMQ bei Amazon MQ-Brokern 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 ist deutlicher, wenn Clients sich über einen Network Load Balancer (NLB) mit dem Broker verbinden, die häufig die Verbindung zum Broker trennen und sich wieder mit dem Broker verbinden. 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 Verbindungssocket 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 eine einfache Authentifizierung und `AuthorizationPlugin` für die Autorisierung von Warteschlangen und Themen verwenden, stellen Sie sicher, dass Sie das `AuthorizationEntries`-Element in Ihrer XML-Konfigurationsdatei verwenden, und erlauben Sie der `activemq-webconsole`Gruppenberechtigung für alle Warteschlangen und Themen. 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>
```

Stellen Sie bei der Integration Ihres Brokers in LDAP sicher, dass Sie die Erlaubnis für die `amazonmq-console-admins`-Gruppe erteilen. Weitere Informationen zur LDAP-Integration finden Sie unter. the section called “Funktionsweise der LDAP-Integration”

Fehlerbehebung: RabbitMQ auf Amazon MQ

Verwenden Sie die Informationen in diesem Abschnitt, um häufig auftretende Probleme zu diagnostizieren und zu lösen, die bei der Arbeit mit RabbitMQ auf Amazon MQ-Brokern auftreten können.

Inhalt

Ich kann keine Metriken für meine Warteschlangen oder virtuellen Hosts in sehen. CloudWatch

Wenn Sie keine Metriken für Ihre Warteschlangen oder virtuellen Hosts anzeigen können CloudWatch, überprüfen Sie, ob Ihre Warteschlangen- oder virtuellen Hostnamen Leerzeichen, Tabulatoren oder andere Nicht-ASCII-Zeichen enthalten.

Amazon MQ kann keine Metriken für virtuelle Hosts und Warteschlangen mit Namen veröffentlichen, die Leerzeichen, Registerkarten oder andere Nicht-ASCII-Zeichen enthalten.

Weitere Informationen zu Dimensionsnamen finden Sie unter Dimension in der Amazon CloudWatch API-Referenz.

Wie aktiviere ich Plugins in RabbitMQ auf Amazon MQ?

RabbitMQ auf Amazon MQ unterstützt derzeit nur die RabbitMQ Management-, Shovel-, Federation- und Consistent-Hash-Exchange-Plug-ins, die standardmäßig aktiviert sind. Weitere Informationen zur Verwendung unterstützter Plugins finden Sie unter the section called “Plugins”.

Ich kann die Amazon-VPC-Konfiguration für den Broker nicht ändern.

Amazon MQ unterstützt das Ändern der Amazon-VPC-Konfiguration nicht, nachdem Ihr Broker erstellt wurde. Bitte beachten Sie, dass Sie einen neuen Broker mit der neuen Amazon-VPC-Konfiguration erstellen und die Client-Verbindungs-URL mit der neuen Broker-Verbindungs-URL aktualisieren müssen.

Clusterbereitstellungen haben meine Warteschlangensynchronisationen angehalten.

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”.

Mein Einzelinstanz-Broker Amazon MQ für RabbitMQ befindet sich in einer Neustartschleife.

Ein Einzelinstanz-Broker von Amazon MQ für RabbitMQ, der einen hohen Speicheralarm auslöst, läuft Gefahr, nicht mehr verfügbar zu werden, wenn er neu gestartet wird und nicht genügend Arbeitsspeicher für den Start zur Verfügung steht. 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 Best Practices nicht anwenden, um den Alarm bei hohem Speicherbedarf zu beheben.

Um Ihren Broker wiederherzustellen, empfehlen wir, auf einen größeren Instance-Typ mit mehr Arbeitsspeicher zu aktualisieren. Im Gegensatz zu Cluster-Bereitstellungen können Sie einen Single-Instance-Broker aktualisieren, wenn bei ihm ein Alarm wegen zu hohem Speicherbedarf auftritt, da bei einem Neustart keine Warteschlangensynchronisationen zwischen den Knoten durchgeführt werden müssen.

Ich habe den Zugriff auf alle Administratorkonten auf meinem Broker verloren.

Sie können den Zugriff mithilfe der IAM-Authentifizierung wiederherstellen. Aktivieren Sie den ausgehenden Web-Identitätsverbund für Ihr AWS Konto, erstellen Sie eine IAM-Rolle mit Berechtigungen zum Abrufen von Web-Identitätstoken, konfigurieren Sie Ihren Broker so, dass er die IAM-Authentifizierung über OAuth 2.0 akzeptiert, und verwenden Sie dann IAM-

Anmeldeinformationen, um ein JWT-Token abzurufen und einen neuen Administratorbenutzer zu erstellen. Detaillierte Anweisungen finden Sie unter the section called “Verwendung der IAM-Authentifizierung und -Autorisierung”.

ActiveMQ auf Amazon MQ: Elastic Network Interface-Alarm gelöscht

ActiveMQ auf Amazon MQ löst einen BROKER_ENI_DELETED-Alarm aus, wenn Sie das Elastic Network Interface (ENI) eines Brokers löschen. Wenn Sie zum ersten Mal einen Amazon MQ-Broker erstellen, richtet Amazon MQ eine elastische Netzwerkschnittstelle in der Virtual Private Cloud (VPC) unter Ihrem Konto ein 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 permanenten Verlust der Verbindung zwischen Ihrer VPC und Ihrem Broker führen. Wenn Sie die Netzwerkschnittstelle löschen möchten, müssen Sie zuerst den Broker löschen.

ActiveMQ auf Amazon MQ: Alarm wegen Speichermangel beim Broker

ActiveMQ auf Amazon MQ 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 CloudWatch API verwenden. Die folgenden Metriken sind beim Diagnostizieren des ActiveMQ-BROKER_OOM-Alarms nützlich:

Amazon MQ-Metrik CloudWatch	Grund für eine hohe Speicherauslastung	
`TotalMessageCount`	Nachrichten werden im Speicher gespeichert,	

Amazon MQ-Metrik CloudWatch	Grund für eine hohe Speicherauslastung	
	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 prozentuale Anteil am ActiveMQ JVM-Speicherlimit, der vom Broker derzeit genutzt wird. Ein höherer Prozentsatz weist darauf hin, dass der Broker erhebliche Ressourcen verbraucht. Das 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 zurzeit vom Broker genutzt werden.	

Amazon MQ-Metrik CloudWatch	Grund für eine hohe Speicherauslastung	
`TotalConsumerCount`	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 einen hohen Speicherverbrauch verursachen und zu einem Alarm über hohe Speicherauslastung führen.	

Stellen Sie sicher, dass die Nachrichten schnell verbraucht werden, um Neustartschleifen und einen BROKER_OOM-Alarm zu vermeiden. 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 auf den Best Practices von ActiveMQ on Amazon MQ.

Amazon MQ für RabbitMQ: Alarm über hohe Speicherauslastung

Amazon MQ for RabbitMQ löst einen hohen Speicheralarm aus, wenn die durch die CloudWatch Metrik identifizierte Speichernutzung des Brokers das Speicherlimit überschreitet`RabbitMQMemUsed`, das durch identifiziert wurde. `RabbitMQMemLimit`

Ein RabbitMQ-Broker, der einen hohen Speicheralarm ausgelöst hat, blockiert alle Clients, die Nachrichten veröffentlichen. Es kann sein, dass Ihr Broker in eine Neustartschleife gerät, die Warteschlangensynchronisierung unterbrochen wird oder es treten andere Probleme auf, die die Diagnose und Lösung des Alarms erschweren.

Um einen Alarm bei hohem Speicherbedarf zu diagnostizieren und zu beheben, befolgen Sie zunächst alle bewährten Methoden für RabbitMQ und führen Sie dann die folgenden Schritte aus.

> **Important**
>
> - `RabbitMQMemLimit`wird von Amazon MQ festgelegt und speziell unter Berücksichtigung des für jeden Host-Instance-Typ verfügbaren Speichers optimiert.
> - Amazon MQ startet einen Broker nicht neu, der einen Alarm über hohe Speicherauslastung hat, und gibt eine Ausnahme für `RebootBroker`-API-Operationen zurück, solange der Broker weiterhin den Alarm auslöst.

Schritt 1: Diagnose eines Alarms bei hohem Speicherbedarf

Es gibt zwei Möglichkeiten, Alarme mit hohem Speicherbedarf auf Ihrem Amazon MQ for RabbitMQ Broker zu diagnostizieren. Wir empfehlen Ihnen, sowohl die RabbitMQ-Webkonsole als auch die Amazon MQ-Metriken zu überprüfen. CloudWatch

Diagnostizieren Sie mithilfe der RabbitMQ-Webkonsole einen Alarm bei hohem Speicherbedarf

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-Managementkonsole 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 zur Speichernutzung, die über die RabbitMQ-Webkonsole verfügbar sind, findest du unter Überlegungen zur Speichernutzung auf der RabbitMQ Server-Dokumentationswebsite.

Diagnostizieren Sie einen Alarm bei hohem Speicherbedarf 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 CloudWatch API verwenden. 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.	
`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	

Amazon MQ-Metrik CloudWatch	Grund für eine hohe Speicherauslastung	
	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.	
`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.	

Schritt 2: Alarme bei hohem Speicherbedarf beheben und verhindern

> **Note**
>
> Nachdem Sie die erforderlichen Maßnahmen ergriffen haben, kann mehrere Stunden dauern, bis der Status RABBITMQ_MEMORY_ALARM gelöscht wird.

Befolge alle bewährten Methoden für RabbitMQ als allgemeine Präventionsmethode. Für jeden einzelnen Mitwirkenden, den Sie identifizieren, empfehlen wir die folgenden Maßnahmen, um RabbitMQ-Alarme mit hohem Speicherbedarf zu beheben und zu verhindern.

Quelle für hohen Speicherverbrauch	Amazon MQ MQ-Empfehlung für die Adressierung	Amazon MQ MQ-Empfehlung zur Vorbeugung
Anzahl der Nachrichten	Verbrauchen Sie die in den Warteschlangen veröffentlichten Nachrichten, löschen Sie Nachrichten aus den Warteschlangen oder löschen Sie die Warteschlangen aus Ihrem Broker.	Aktivieren Sie verzögerte Warteschlangen und legen Sie das Limit für die Warteschlangentiefe fest oder reduzieren Sie es.
Anzahl der Warteschlangen	Reduzieren Sie die Anzahl der Warteschlangen.	Legen Sie das Limit für die Anzahl der Warteschlangen fest oder reduzieren Sie es.
Anzahl der Verbindungen	Reduzieren Sie die Anzahl der Verbindungen.	Legen Sie das Limit für die Anzahl der Verbindungen fest oder reduzieren Sie es.
Anzahl der Kanäle	Reduzieren Sie die Anzahl der Kanäle.	Legen Sie eine maximale Anzahl von Kanälen pro Verbindung für Clientanwendungen fest.

Quelle für hohen Speicherverbrauch	Amazon MQ MQ-Empfehlung für die Adressierung	Amazon MQ MQ-Empfehlung zur Vorbeugung
Anzahl der Konsumenten	Reduzieren Sie die Gesamtzahl der Verbraucher, die mit dem Broker verbunden sind.	Legen Sie einen geringen Vorabrufgrenzwert für Verbraucher fest.
Rate der Veröffentlichung von Nachrichten	Reduzieren Sie die Rate, mit der Nachrichten an den Broker veröffentlicht werden.	Schalten Sie Publisher Confirms ein.
Rate der Verbindungsversuche des Clients	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.	Verwenden Sie langlebigere Verbindungen, um die Anzahl und Häufigkeit von Verbindungsversuchen zu reduzieren.

Nachdem der Speicheralarm 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-Referenz.

> **Note**
>
> Sie können einen Broker nicht von einem Instance-Typ auf einen `mq.m5.x` Instance-Typ herabstufen. `mq.t3.micro` Um ein Downgrade durchzuführen, müssen Sie Ihren Broker löschen und einen neuen erstellen.

RabbitMQ auf Amazon MQ: Ungültiger Schlüssel AWS Key Management Service

RabbitMQ auf Amazon MQ gibt den Code INVALID_KMS_KEY aus, der für eine kritische Aktion erforderlich ist, wenn ein Broker, der mit einem vom Kunden verwalteten AWS KMS key(CMK) erstellt wurde, feststellt, dass der (KMS) -Schlüssel deaktiviert ist. AWS Key Management Service Ein RabbitMQ-Broker mit einem CMK überprüft regelmäßig, ob der KMS-Schlüssel aktiviert ist und der Broker über alle erforderlichen Erteilungen verfügt. Wenn RabbitMQ nicht überprüfen kann, ob der Schlüssel aktiviert ist, wird der Broker unter Quarantäne gestellt und RabbitMQ gibt INVALID_KMS_KEY zurück.

Ohne einen aktiven KMS-Schlüssel verfügt der Broker nicht über grundlegende Berechtigungen für vom Kunden verwaltete KMS-Schlüssel. 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 festgestellt hat, dass der KMS-Schlüssel wieder aktiv ist, wird die Quarantäne Ihres Brokers beendet. Amazon MQ startet einen Broker mit einem deaktivierten KMS-Schlüssel nicht neu und gibt eine Ausnahme für `RebootBroker`-API-Operationen zurück, solange der Broker einen ungültigen KMS-Schlüssel hat.

Diagnose und Behandlung von INVALID_KMS_KEY

Um den für die Aktion INVALID_KMS_KEY erforderlichen Code zu diagnostizieren und zu beheben, müssen Sie die AWS Befehlszeilenschnittstelle (CLI) und die Konsole verwenden. AWS Key Management Service

So aktivieren Sie Ihren KMS-Schlüssel erneut

1. Rufen Sie die `DescribeBroker`-Methode auf, um die `kmsKeyId` für Ihren CMK-Broker abzurufen.
2. Melden Sie sich bei der Konsole an. AWS Key Management Service
3. Suchen Sie auf der Seite Kundenverwaltete Schlüssel die KMS-Schlüssel-ID des problematischen Brokers und vergewissern Sie sich, dass der Status Aktiviert lautet.
4. Wenn Ihr KMS-Schlüssel deaktiviert wurde, aktivieren Sie ihn erneut, indem Sie Schlüsselaktionen und anschließend Aktivieren auswählen. Nachdem Ihr Schlüssel erneut aktiviert wurde, müssen Sie warten, bis RabbitMQ die Quarantäne des Brokers beendet.

Um zu überprüfen, ob die erforderlichen Grants 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 die KMS-Erteilung oder der Schlüssel gelöscht wurde, müssen Sie den Broker löschen und einen neuen mit allen erforderlichen Erteilungen erstellen. Schritte zum Löschen eines Brokers finden Sie unter Löschen eines Brokers.

Löschen oder deaktivieren Sie einen KMS-Schlüssel oder eine CMK-Erteilung nicht manuell, um den Code INVALID_KMS_KEY für erforderliche kritische Aktionen zu verhindern. Wenn Sie den Schlüssel löschen möchten, löschen Sie zuerst den Broker.

RabbitMQ auf Amazon MQ: Alarm beim Festplattenlimit

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 RabbitMQ on Amazon MQ-Broker, der einen Festplattenlimit-Alarm ausgelöst hat, ist für die Veröffentlichung neuer Nachrichten nicht mehr verfügbar. Wenn Sie einen Herausgeber und einen Verbraucher auf derselben Verbindung haben, kann der Verbraucher auch keine Nachrichten empfangen. 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 mit Festplattenalarm nicht neu und gibt eine Ausnahme für `RebootBroker`-API-Operationen zurück, solange der Broker den Alarm weiterhin 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 CloudWatch API verwenden. `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:

- Erstellen Sie eine neue Verbindung, um Nachrichten zu verarbeiten, 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 Status RABBITMQ_DISK_ALARM 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 zum Aktualisieren des Instance-Typs Ihres Brokers finden Sie unter `UpdateBrokerInput` in der Referenz zur Amazon-MQ-REST-API. Wir empfehlen außerdem, Ihre Herausgeber und Verbraucher auf unterschiedlichen Verbindungen zu halten.

Amazon MQ für RabbitMQ: Alarm bei Änderung des Instanztyps

`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`ist ein Hinweis darauf, dass eine angeforderte Änderung des Broker-Instance-Typs aufgrund der hohen Festplattenauslastung auf dem aktuellen RabbitMQ-Knoten nicht fortgesetzt werden kann. Amazon MQ for RabbitMQ löst diesen Alarm aus, wenn die aktuelle Festplattennutzung den Wert übersteigt, der auf dem angeforderten Instance-Typ verfügbar wäre, wie anhand der Metrik identifiziert. CloudWatch `RabbitMQDiskFree`

RabbitMQ-Broker, die in den `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` Bundesstaat wechseln, sind weiterhin für Ihre Anwendungen verfügbar, aber die angeforderte Änderung des Instance-Typs wird nicht fortgesetzt. Amazon MQ erlaubt Broker-Neustarts in diesem Status, aber Sie können den Instance-Typ nicht ändern, solange die Festplattennutzung über dem Schwellenwert für den angeforderten Instance-Typ bleibt. Der Broker gibt eine Ausnahme für `ModifyBroker` API-Operationen zurück, bei denen versucht wird, den Instance-Typ in diesem Status zu ändern.

Alarm bei Änderung des Instanztyps wird diagnostiziert und adressiert

Amazon MQ aktiviert standardmäßig Metriken für Ihren Broker. Sie können Ihre Broker-Metriken einsehen, indem Sie auf die CloudWatch Konsole zugreifen oder die CloudWatch API verwenden. `MessageCount`und `RabbitMQDiskFree` Metriken können zur Diagnose verwendet werden`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`.

Um den Quarantänestatus zu beheben und die Änderung Ihres Instance-Typs zu ermöglichen, verwenden Sie die Amazon MQ Management Console, um:

- Erstellen Sie eine neue Verbindung, um Nachrichten zu verarbeiten, 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 `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` Status gelöscht wird, nachdem Sie die erforderlichen Maßnahmen ergriffen haben.

RabbitMQ auf Amazon MQ: Ungültiges IAM übernimmt Rolle

RabbitMQ auf Amazon MQ löst den Code INVALID_ASSUMEROLE aus, der für eine kritische Aktion erforderlich ist, wenn der in angegebene ARN für die IAM-Rolle ungültig `aws.arns.assume_role_arn` ist oder von Amazon MQ nicht übernommen werden kann. Dies kann der Fall sein, wenn die Rolle nicht existiert, sich in einem anderen AWS Konto befindet als der Broker oder wenn die erforderliche Vertrauensbeziehung zu mq.amazonaws.com nicht besteht.

Ein Broker in der Quarantäne von RABBITMQ_INVALID_ASSUMEROLE kann die für die LDAP-Authentifizierung erforderlichen Anmeldeinformationen oder Zertifikate nicht abrufen, sodass die LDAP-Authentifizierung nicht verfügbar ist. Wenn LDAP die einzige konfigurierte Authentifizierungsmethode ist, können Benutzer keine Verbindung zum Broker herstellen. Die IAM-Rolle wird von Amazon MQ benötigt, um auf AWS Ressourcen zuzugreifen, auf die ARNs in der Broker-Konfiguration verwiesen wird, z. B. AWS Secrets Manager Geheimnisse oder Amazon S3 S3-Objekte, die für die LDAP-Authentifizierung verwendet werden.

Diagnose und Adressierung von RABBITMQ_INVALID_ASSUMEROLE

Um den für die Aktion RABBITMQ_INVALID_ASSUMEROLE erforderlichen Code zu diagnostizieren und zu adressieren, müssen Sie Amazon Logs und die Konsole verwenden. CloudWatch AWS Identity and Access Management

Um das Problem mit der ungültigen Übernahme der Rolle zu lösen

1. Navigieren Sie zu Amazon CloudWatch Logs Insights und führen Sie die folgende Abfrage für die Protokollgruppe Ihres Brokers aus`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Suchen Sie nach Fehlermeldungen, die den folgenden ähneln:

```
[error] <0.254.0> aws_arn_config: {handle_assume_role,{error,
{assume_role_failed,"AWS service is unavailable"}}}
```

3. Überprüfen Sie die IAM-Rollenkonfiguration und beheben Sie beispielsweise folgende Probleme:

   - Stellen Sie sicher, dass die Rolle im selben AWS Konto wie der Broker existiert
   - Stellen Sie sicher, dass die Vertrauensrichtlinie es mq.amazonaws.com ermöglicht, die Rolle zu übernehmen
   - Vergewissern Sie sich, dass die Rolle über die erforderlichen Berechtigungen für den Zugriff auf die erforderlichen Ressourcen verfügt AWS

4. Überprüfen Sie den Fix mithilfe des API-Endpunkts für die ARN-Zugriffsprüfung, bevor Sie die Broker-Konfiguration aktualisieren.
5. Aktualisieren Sie die Broker-Konfiguration und starten Sie den Broker neu.

RabbitMQ auf Amazon MQ: Ungültiger LDAP-ARN

RabbitMQ auf Amazon MQ gibt den Code INVALID_ARN_LDAP für eine kritische Aktion erforderlich aus, wenn der für das Passwort des LDAP-Servicekontos konfigurierte ARN ungültig oder nicht zugänglich ist. Dies gilt für die in oder ARNs angegebenen Daten, die auf Geheimnisse verweisen müssen, die Klartext-Passwörter enthalten. `aws.arns.auth_ldap.dn_lookup_bind.password` `aws.arns.auth_ldap.other_bind.password` AWS Secrets Manager

Ein Broker in der Quarantäne RABBITMQ_INVALID_ARN_LDAP kann sich nicht mit dem LDAP-Dienstkonto authentifizieren, sodass die LDAP-Authentifizierung nicht verfügbar ist. Wenn LDAP die einzige konfigurierte Authentifizierungsmethode ist, können Benutzer keine Verbindung zum Broker herstellen. Ungültig ARNs kann auf eine falsch formatierte ARN-Syntax, Verweise auf nicht existierende Geheimnisse, auf Geheimnisse, die sich in einer anderen AWS Region als dem Broker befinden, oder auf unzureichende secretsmanager: GetSecretValue -Berechtigungen in der IAM-Rolle zurückzuführen sein.

Diagnose und Adressierung von RABBITMQ_INVALID_ARN_LDAP

Um den Aktionscode RABBITMQ_INVALID_ARN_LDAP zu diagnostizieren und zu adressieren, müssen Sie Amazon Logs und die Konsole verwenden. CloudWatch

Um das Problem mit dem ungültigen LDAP-ARN zu lösen

1. Navigieren Sie zu Amazon CloudWatch Logs Insights und führen Sie die folgende Abfrage für die Protokollgruppe Ihres Brokers aus`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Suchen Sie nach Fehlermeldungen, die den folgenden ähneln:

```
[error] <0.254.0> aws_arn_config: {<<"could not resolve
 ARN 'arn:aws:secretsmanager:xxx' for configuration
 'aws.arns.auth_ldap.dn_lookup_bind.password', error: \"AWS service is unavailable
\"">>,{error,"AWS service is unavailable"}}
```

3. Überprüfen Sie das Secrets Manager Manager-Geheimnis und beheben Sie alle Probleme wie:
   - Stellen Sie sicher, dass das Geheimnis in derselben AWS Region wie der Broker existiert
   - Stellen Sie sicher, dass ARN ARN-Syntax korrekt ist
   - Stellen Sie sicher, dass die IAM-Rolle über die Berechtigungen secretsmanager: verfügt GetSecretValue
4. Überprüfen Sie den Fix mithilfe des API-Endpunkts für die ARN-Zugriffsprüfung, bevor Sie die Broker-Konfiguration aktualisieren.
5. Aktualisieren Sie die Broker-Konfiguration und starten Sie den Broker neu.

RabbitMQ auf Amazon MQ: Ungültiger HTTP-ARN

RabbitMQ auf Amazon MQ gibt den Code INVALID_ARN_HTTP für eine kritische Aktion erforderlich aus, wenn eines oder mehrere ARNs SSL-Zertifikate oder die Schlüsseldatei für HTTP auth_backend ungültig oder nicht zugänglich sind. Dies gilt für in `aws.arns.auth_http.ssl_options.certfile` oder angegebene ARNS `aws.arns.auth_http.ssl_options.cacertfileaws.arns.auth_http.ssl_options.keyfile`, die auf Amazon S3 S3-Objekte und AWS Secrets Manager -Secrets verweisen müssen, die Zertifikate und private Schlüssel enthalten.

Ein Broker in der Quarantäne RABBITMQ_INVALID_ARN_HTTP kann sich nicht über den HTTP-Server authentifizieren. Wenn HTTP die einzige konfigurierte Authentifizierungsmethode ist, können Benutzer keine Verbindung zum Broker herstellen. Ungültig ARNs kann durch eine falsch formatierte ARN-Syntax, Verweise auf nicht existierende Geheimnisse, Geheimnisse, die sich in einer anderen AWS Region als der Broker befinden, oder unzureichende s3: GetObject /secretsmanager: GetSecretValue -Berechtigungen in der IAM-Rolle verursacht werden.

Diagnose und Adressierung von RABBITMQ_INVALID_ARN_HTTP

Um den für die Aktion RABBITMQ_INVALID_ARN_HTTP erforderlichen Code zu diagnostizieren und zu adressieren, müssen Sie Amazon Logs und die Konsole verwenden. CloudWatch

Um das Problem mit dem ungültigen HTTP-ARN zu lösen

1. Navigieren Sie zu Amazon CloudWatch Logs Insights und führen Sie die folgende Abfrage für die Protokollgruppe Ihres Brokers aus`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Suchen Sie nach Fehlermeldungen, die den folgenden ähneln:

```
[error] <0.209.0> aws_arn_config: {<<"could not resolve ARN 'arn:aws:s3:::xxxx' for
 configuration 'aws.arns.auth_http.ssl_options.certfile', error: \"AWS service is
 unavailable\"">>,{error,"AWS service is unavailable"}}
```

3. Überprüfen Sie das S3 Object/Secrets Manager-Geheimnis und beheben Sie alle Probleme wie:
   - Stellen Sie sicher, dass die Ressource in derselben AWS Region wie der Broker vorhanden ist
   - Stellen Sie sicher, dass ARN ARN-Syntax korrekt ist
   - Stellen Sie sicher, dass die IAM-Rolle über die Berechtigungen s3: GetObject und secretsmanager: verfügt GetSecretValue
4. Überprüfen Sie den Fix mithilfe des API-Endpunkts für die ARN-Zugriffsprüfung, bevor Sie die Broker-Konfiguration aktualisieren.
5. Aktualisieren Sie die Broker-Konfiguration und starten Sie den Broker neu.

RabbitMQ auf Amazon MQ: Ungültiger SSL-ARN

RabbitMQ auf Amazon MQ löst den Code INVALID_ARN_SSL aus, der für kritische Aktionen erforderlich ist, wenn ein oder mehrere CA-Zertifikats-Truststore

für EXTERNAL ARNs auth_mechanism ungültig oder nicht zugänglich sind.
Dies gilt für in `aws.arns.ssl_options.cacertfile` oder angegebene ARNS`aws.arns.management.ssl.cacertfile`, die auf das Amazon S3- oder ACM PCA-Objekt verweisen müssen, das das Zertifikat enthält.

Ein Broker in der Quarantäne RABBITMQ_INVALID_ARN_SSL kann Client-Zertifikate bei gegenseitigen TLS-Handshakes nicht authentifizieren, da kein gültiger Truststore konfiguriert ist. Wenn der externe Authentifizierungsmechanismus die einzige konfigurierte Authentifizierungsmethode ist, können Benutzer keine Verbindung zum Broker herstellen. Ungültig ARNs kann durch eine falsch formatierte ARN-Syntax, Verweise auf nicht existierende S3-Objekte, S3-Objekte, die sich in einer anderen AWS Region als der Broker befinden, oder unzureichende s3: GetObject /acm-pca: GetCertificateAuthorityCertificate -Berechtigungen in der IAM-Rolle verursacht werden.

Diagnose und Adressierung von RABBITMQ_INVALID_ARN_SSL

Um den für die Aktion RABBITMQ_INVALID_ARN_SSL erforderlichen Code zu diagnostizieren und zu adressieren, müssen Sie Amazon Logs und die Konsole verwenden. CloudWatch

Um das Problem mit dem ungültigen SSL-ARN zu lösen

1. Navigieren Sie zu Amazon CloudWatch Logs Insights und führen Sie die folgende Abfrage für die Protokollgruppe Ihres Brokers aus`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Suchen Sie nach Fehlermeldungen, die den folgenden ähneln:

```
[error] <0.209.0> aws_arn_config: {<<"could not resolve ARN 'arn:aws:acm-pca:xxxx'
 for configuration 'aws.arns.ssl_options.cacertfile', error: \"AWS service is
 unavailable\"">>,{error,"AWS service is unavailable"}}
```

3. Überprüfen Sie das S3/ACM-PCA-Objekt und beheben Sie beispielsweise folgende Probleme:

- Stellen Sie sicher, dass das Geheimnis in derselben Region wie der Broker existiert AWS
- Stellen Sie sicher, dass ARN ARN-Syntax korrekt ist
- Stellen Sie sicher, dass die IAM-Rolle über die Berechtigungen s3: GetObject /acm-pca: verfügt GetCertificateAuthorityCertificate

4. Überprüfen Sie den Fix mithilfe des API-Endpunkts für die ARN-Zugriffsprüfung, bevor Sie die Broker-Konfiguration aktualisieren.
5. Aktualisieren Sie die Broker-Konfiguration und starten Sie den Broker neu.

RabbitMQ auf Amazon MQ: Ungültiger ARN

RabbitMQ auf Amazon MQ gibt den Code INVALID_ARN aus, der für eine kritische Aktion erforderlich ist, wenn einer oder mehrere im Broker ARNs konfigurierte Objekte ungültig oder unzugänglich sind. Dies gilt für SSL-Zertifikate, AWS Secrets Manager Geheimnisse, Amazon S3 S3-Objekte oder andere AWS Ressourcenreferenzen, die nicht durch spezifischere Quarantänecodes wie RABBITMQ_INVALID_ARN_LDAP oder RABBITMQ_INVALID_ASSUMEROLE abgedeckt sind. ARNs

Bei einem Broker in der Quarantäne von RABBITMQ_INVALID_ARN kann es zu Funktionseinschränkungen kommen, je nachdem, welche ungültig sind. ARNs Funktionen, die von den unzugänglichen Ressourcen abhängen, sind nicht verfügbar, und der Broker protokolliert Fehler, die angeben, welcher ARN nicht behoben werden konnte. Die Auswirkungen auf die Broker-Verfügbarkeit hängen davon ab, ob der ungültige ARN für kritische Broker-Operationen erforderlich ist.

Diagnose und Adressierung von RABBITMQ_INVALID_ARN

Um den Aktionscode RABBITMQ_INVALID_ARN zu diagnostizieren und zu beheben, müssen Sie Amazon CloudWatch Logs und die entsprechende AWS Servicekonsole für die betroffene Ressource verwenden.

Um das Problem mit dem ungültigen ARN zu lösen

1. Navigieren Sie zu Amazon CloudWatch Logs Insights und führen Sie die folgende Abfrage für die Protokollgruppe Ihres Brokers aus`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Suchen Sie nach Fehlermeldungen, die den folgenden ähneln:

```
[error] <0.254.0> aws_arn_config: {<<"could not resolve ARN
 'arn:aws:s3:::bucket-name/certificate.pem' for configuration
 'aws.arns.auth_ldap.ssl_options.cacertfile', error: \"AWS service is unavailable
\"">>,{error,"AWS service is unavailable"}}
```

3. Überprüfen Sie die AWS Ressource und beheben Sie alle Probleme wie:

   - Stellen Sie sicher, dass die Ressource in derselben AWS Region wie der Broker existiert
   - Stellen Sie sicher, dass ARN ARN-Syntax korrekt ist
   - Stellen Sie sicher, dass die IAM-Rolle über die entsprechenden Berechtigungen für den Zugriff auf die Ressource verfügt

4. Überprüfen Sie den Fix mithilfe des API-Endpunkts für die ARN-Zugriffsprüfung, bevor Sie die Broker-Konfiguration aktualisieren.
5. Aktualisieren Sie die Broker-Konfiguration und starten Sie den Broker neu.

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 der AWS CLI Befehlsreferenz	Beschreibungen der AWS CLI Befehle, die Sie für die Arbeit mit Message Brokern verwenden können.
Amazon MQ im AWS CloudFormation Benutzerhandbuch	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 zum Premium-Support	Die wichtigste Webseite mit Informationen zu AWS Premium Support, einem schnell reagierenden Support-Kanal one-on-one, der Sie bei der Entwicklung und Ausführung von Anwendungen auf AWS Infrastrukturdiensten unterstützt

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-Clientbibliotheken und Entwickle rtools	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.

Date	Aktualisierung der Dokumentation
19. Februar 2026	Amazon MQ Amazon MQ unterstützt jetzt ActiveMQ 5.19, eine neue Engine-Nebenversion. Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format). • ActiveMQ 5.19-Releaseseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Aktualisieren einer Amazon MQ-Broker-Engine-Version • Verwenden von Spring XML-Konfigurationsdateien
22. Januar 2026	Amazon MQ unterstützt jetzt das JMS-Themenaustausch-Plugin für Broker auf RabbitMQ 4.2 und höher. Sie können den offiziellen RabbitMQ JMS-Client verwenden, um JMS-Workloads auf Amazon MQ for RabbitMQ Broker auszuführen. Er unterstützt JMS 1.1, 2.0 und 3.1. Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format). • Offizielle JMS 2.0-Spezifikation (abwärtskompatibel mit und erweitertem JMS 1.1) • Offizielle JMS 3.1-Spezifikation • Einschränkung des RabbitMQ JMS-Clients • Ihre JMS-Anwendung mit Amazon MQ für den RabbitMQ-Broker verbinden
8. Januar 2026	Amazon MQ unterstützt jetzt die SSL-Zertifikatsauthentifizierung für Broker auf RabbitMQ 4.2 und höher mithilfe von X.509-Client-Zertifikaten und einer Mutual TLS (mTLS) -Konfiguration. Sie können SSL-Zertifikatsauthentifizierung und mTLS über AWS-Managementkonsole, AWS CloudFormation AWS CLI, oder AWS CDK überall dort konfigurieren, AWS-Regionen wo Amazon MQ verfügbar ist.

Date	Aktualisierung der Dokumentation
	Weitere Informationen finden Sie unter SSL-Zertifikatsauthentifizierung und Konfiguration von mTLS.
6. Januar 2026	Amazon MQ unterstützt jetzt HTTP-Authentifizierung und Autorisierung für Broker auf RabbitMQ 4.2 und höher mit externen HTTP-Servern. Sie können die HTTP-Authentifizierung über AWS-Managementkonsole, AWS CloudFormation AWS CLI, oder AWS CDK überall dort konfigurieren, AWS-Regionen wo Amazon MQ verfügbar ist. Weitere Informationen finden Sie unter HTTP-Authentifizierung und Autorisierung.
20. November 2025	Amazon MQ unterstützt jetzt RabbitMQ 4.2, eine neue Hauptversion, die native Unterstützung für das AMQP 1.0-Protokoll, einen neuen Raft-basierten Metadatenspeicher Khepri, lokale Schaufeln und Nachrichtenprioritäten für Quorumwarteschlangen einführt. RabbitMQ 4.2 enthält auch verschiedene Bugfixes und Leistungsverbesserungen für den Durchsatz und die Speicherverwaltung. Diese Version führt zwar neue Funktionen ein, es gibt jedoch einige grundlegende Änderungen. Die Ausgabe des obigen Befehls sieht in etwa folgendermaßen aus (JSON format). • RabbitMQ 4 • Versionshinweise zu RabbitMQ auf Open-Source-Basis • Konfiguration von Ressourcenlimits • Unterstützte Protokolle • Amazon MQ MQ-Versionsupgrades
18. November 202	Amazon MQ unterstützt jetzt Graviton3-betriebene m7g-Instances für RabbitMQ in verschiedenen Größen von Medium bis 16xlarge in Afrika (Kapstadt). Weitere Informationen finden Sie unter Broker-Instance-Typen von Amazon MQ für RabbitMQ.

Date	Aktualisierung der Dokumentation
17. November 2025	Amazon MQ unterstützt jetzt LDAP-Authentifizierung und Autorisierung für RabbitMQ-Broker mit externen LDAP-Verzeichnisdiensten. Sie können LDAP über AWS-Managementkonsole, AWS CloudFormation, oder AWS CDK überall dort konfigurieren AWS CLI, AWS-Regionen wo Amazon MQ verfügbar ist. Weitere Informationen finden Sie unter LDAP-Authentifizierung und Autorisierung für Amazon MQ for RabbitMQ.
22. Oktober 2025	Amazon MQ ist jetzt in der Region Asien-Pazifik (Neuseeland) verfügbar. Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
3. September 2025	Amazon MQ unterstützt jetzt OAuth 2.0-Authentifizierung und Autorisierung für RabbitMQ-Broker mit öffentlichen Identitätsanbietern (). IdPs Sie können OAuth 2.0 bis AWS-Managementkonsole, AWS CloudFormation AWS CLI, oder AWS CDK überall dort konfigurieren, AWS-Regionen wo Amazon MQ verfügbar ist. Weitere Informationen finden Sie unter OAuth 2.0 Authentifizierung und Autorisierung für Amazon MQ for RabbitMQ.
22. Juli 2025	Amazon MQ unterstützt jetzt Graviton3-basierte `m7g` Instances für RabbitMQ in verschiedenen Größen von Medium bis 16xlarge. RabbitMQ-Cluster, die auf `m7g` Instances ausgeführt werden, bieten eine um bis zu 50% höhere Workload-Kapazität und einen um bis zu 85% höheren Durchsatz als vergleichbare Amazon MQ for RabbitMQ-Cluster, die auf Instances ausgeführt werden. `m5` `M7g`Instances verfügen außerdem über optimierte Festplattenvolume-Größen, die je nach Instance-Größe variieren. Weitere Informationen finden Sie unter Broker instance types. `M7g`Instances auf Amazon MQ sind heute in allen allgemein verfügbaren Regionen mit Ausnahme der Regionen Afrika (Kapstadt), Kanada West (Calgary) und Europa (Mailand) verfügbar.

Date	Aktualisierung der Dokumentation
8. Juli 2025	Amazon MQ ist jetzt in der Region Asien-Pazifik (Taipeh) verfügbar. Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
22. April 2025	Sie können jetzt Amazon MQ-Brokerkonfigurationen mithilfe der `DeleteConfiguration` API löschen. Weitere Informationen finden Sie unter Konfigurationen in der Amazon MQ API-Referenz.
16. April 2025	Amazon MQ for RabbitMQ unterstützt jetzt die Verwendung von Dual-Stack IPv4 - (und IPv6) Endpunkten für die Verbindung mit öffentlichen und privaten Brokern. Weitere Informationen erhalten Sie unter Connecting to Amazon MQ und Configuring a private Amazon MQ broker.
7. April 2025	Amazon MQ ist jetzt in den Regionen Asien-Pazifik (Thailand) und Mexiko (Zentral) verfügbar. Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
13. Februar 2025	Amazon MQ API-FIPS-Endpunkte sind jetzt in den Regionen Kanada (Zentral) und Kanada West (Calgary) verfügbar. Weitere Informationen zur Verwendung von FIPS-Endpunkten mit der Amazon MQ MQ-API finden Sie unter. Connecting to Amazon MQ Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.

Date	Aktualisierung der Dokumentation
12. Februar 2025	Amazon MQ kündigt die Termine für das Ende des Supports für den folgenden Instance-Typ an: Broker instance types • ActiveMQ`mq.t2.micro` : 12. Mai 2025 • ActiveMQ`mq.m4.large` : 12. Mai 2025 Am `mq.t2.micro` oder `mq.m4.large` nach dem 17. März 2025 können Sie keine Makler mehr erstellen.
10. Dezember 202	Amazon MQ unterstützt jetzt die Verwendung von AWS PrivateLink Verbindungen zwischen Ihren virtuellen privaten Clouds (VPCs) und der Amazon MQ MQ-API, ohne dass Ihr Datenverkehr dem öffentlichen Internet ausgesetzt wird. Weitere Informationen finden Sie unter the section called “Connect zu Amazon MQ her mit AWS PrivateLink”.
18. November 202	Amazon MQ ist jetzt in der Region Asien-Pazifik (Malaysia) verfügbar. Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
14. November 202	Amazon MQ kündigt das Ende des Supports für die folgenden Engine-Versionen an: Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • ActiveMQ 5.17:16. Juni 2025 Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen • RabbitMQ 3.11:17. Februar 2025 • RabbitMQ 3.12:17. März 2025 Weitere Informationen zum Upgrade auf die neueste Version finden Sie unter Aktualisieren einer Amazon MQ-Broker-Engine-Version

Date	Aktualisierung der Dokumentation
13. November 202	Amazon MQ unterstützt jetzt Dual-Stack-Serviceendpunkte, mit denen Sie entweder oder eine Verbindung herstellen können. IPv4 IPv6 Regionale Dual-Stack-Servicendpunkte von Amazon MQ können sowohl mit DNS-Einträgen als auch A mit DNS-Einträgen aufgelöst werden. AAAA Weitere Informationen finden Sie unter ???.
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 Spring XML-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-Warteschlangen, die den Raft-Konsensus-Algorithmus verwenden, um die Datenkonsistenz aufrechtzuerhalten. Quorumwarteschlangen ermöglich en 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

Date	Aktualisierung der Dokumentation
2. Juli 2024	Amazon MQ for RabbitMQ unterstützt jetzt RabbitMQ 3.13, eine Nebenversion. 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 Aktualisieren 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-Repository. GitHub 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.

Date	Aktualisierung der Dokumentation
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.

Date	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 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.

Date	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.

Date	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

Date	Aktualisierung der Dokumentation
23. Juni 2023	Amazon MQ unterstützt jetzt ActiveMQ 5.17.3, eine neue Engine-Unterversion. Diese Version unterstützt die neue Funktion zur regionsübergreifenden Datenreplikation (CRDR) von Amazon MQ. Weitere Informationen finden Sie hier: • Informationen zu den ersten Schritten mit CRDR finden Sie unter Regionsübergreifende 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 Spring XML-Konfigurationsdateien
21. Juni 2023	Amazon MQ for ActiveMQ bietet jetzt eine CRDR-Funktion (Cross-Region-Data Replication), 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 mit CRDR finden Sie unter Regionsübergreifende Datenreplikation für Amazon MQ für ActiveMQ im Entwicklerhandbuch.
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 -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.

Date	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.

Date	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 3.10.10, während wir daran arbeiten, eine neue Patch-Version von RabbitMQ zu unterstützen. Wir empfehlen, die Option für das Versionsupgrade zu aktivieren, um automatisch die neuesten Bugfixes, Sicherheitsupdates und Leistungsverbesserungen 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.

Date	Aktualisierung der Dokumentation
21. Februar 2023	Amazon MQ for RabbitMQ lässt sich jetzt in AWS Key Management Service (KMS) integrieren, um serverseitige Verschlüsselung anzubieten. Sie können jetzt Ihr eigenes, vom Kunden verwaltetes CMK auswählen oder einen verwalteten KMS-Schlüssel in Ihrem Konto AWS verwenden. AWS KMS Weitere Informationen finden Sie unter Verschlüsselung im Ruhezustand. Amazon MQ unterstützt die Verwendung von AWS KMS Schlüsseln auf folgende Weise. • Amazon MQ owned KMS key (default) (Amazon-MQ-eigener KMS-Schlüssel (Standard)) – Der Schlüssel ist Eigentum von Amazon MQ und wird von diesem verwaltet. Er 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. • Select existing customer managed KMS key (Vorhandenen, 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.

Date	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 die klassische Warteschlangen Version 2 () für Ihre RabbitMQ-Warteschlangen aktivieren. CQv2 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.

Date	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 Spring XML-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 Spring XML-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 Spring XML-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

Date	Aktualisierung der Dokumentation
25. April 2022	Amazon MQ: Diese Version fügt den `CRITICAL_ACTION_REQUIRED` -Broker-Status und die `ActionRequired` -API-Eigenschaft hinzu. `CRITICAL_ACTION_REQUIRED` informiert Sie, wann Ihr Broker heruntergestuft wird. `ActionRequired` stellt Ihnen einen Code zur Verfügung, mit dem Sie im Entwicklerhandbuch Anweisungen zur Behebung des Problems finden können. • Fehlerbehebung • `ActionRequired` -Dokumentation in der Amazon-MQ-API-Referenz.
20. April 2022	Amazon MQ unterstützt jetzt ActiveMQ 5.16.4, eine neue Engine-Unterversion. Weitere Informationen finden Sie hier: • ActiveMQ 5.16.4 – Versionsseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwenden von Spring XML-Konfigurationsdateien • Aktualisieren einer Amazon MQ-Broker-Engine-Version
1. März 2022	Amazon MQ ist jetzt in der Region Asien-Pazifik (Jakarta) verfügbar. Informationen 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-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.

Date	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 Protokollierung in JSON, die in 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.

Date	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` ”

Date	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 ressourcenbasierten IAM-Richtlinien, um das Problem des verwirrten Stellvertreters zu verhindern. 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.

Date	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 Spring XML-Konfigurationsdateien
8. September 2021	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.22. Diese Version enthält eine Korrektur für ein Problem mit Warteschlangen mitTime to Message TTL(Time to Live, Gültigkeitsdauer), identifiziert in der zuvor unterstützten Version, RabbitMQ 3.8.17. Wir empfehlen, Ihre vorhanden en 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 mit TTL (Per-Message) festgestellt wurde. time-to-live Wir empfehlen die Verwendung der Version 3.8.11.

Date	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 der AWS-Managementkonsole AWS CLI, oder der Amazon MQ MQ-API 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 Spring XML-Konfigurationsdateien

Date	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 im Folgenden: https://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html
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 Spring XML-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

Date	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 Spring XML-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. Informationen zum Beheben dieses Problems finden Sie unter AMQ-8104.

Date	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, ohne den Code neu schreiben zu müssen. AWS 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 auf Ihre Broker zugreifen, und mit jeder Sprache, die von den RabbitMQ-Client-Bibliotheke n unterstützt wird. Weitere Informationen zu unterstützten Protokollen 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. Um mit Amazon MQ zu beginnen, einen Broker zu erstellen und eine JVM-basierte Anwendung mit Ihrem RabbitMQ-Broker zu verbinden, sieheErste Schritte: Einen RabbitMQ-Broker erstellen und eine Verbindung zu ihm herstellen.

Date	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 Spring XML-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 Benutzer mit den Anmeldeinformationen authentif izieren, die in Ihrem Active Directory oder einem anderen LDAP-Server gespeichert sind. 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 Spring XML-Konfigurationsdateien

Date	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.

Date	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 Spring XML-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.

Date	Aktualisierung der Dokumentation
16. Dezember 201	Amazon MQ unterstützt das Erstellen von durchsatzoptimierten Brokern mithilfe von Amazon Elastic Block Store (EBS) anstelle des standardmäßigen Amazon Elastic File System (Amazon EFS) für Broker-Speicher. Verwenden Sie Amazon EFS, um die Vorteile der hohen Haltbarkeit und Replikation über mehrere Availability Zones hinweg zu nutzen. Verwenden Sie Amazon EBS, um die Vorteile der niedrigen Latenz und des hohen Durchsatzes zu nutzen. **Important** • Sie können Amazon EBS nur mit dem `mq.m5` Broker-Instance-Typ. • 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 einzelnen Availabil ity Zone und unterstützt den ActiveMQ Aktiv/Standby-Bereitst ellungsmodus nicht. Weitere Informationen finden Sie hier: • Storage • Auswählen des richtigen Broker-Speichertyps für den besten Durchsatz • Die `storageType` -Eigenschaft der `broker-instance-options` -Ressource in der Amazon-MQ-REST-API-Referenz • 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 Monitoring and logging Amazon MQ brokers

Date	Aktualisierung der Dokumentation
11. Oktober 2019	Amazon MQ unterstützt jetzt Federal Information Processing Standard 140-2 (FIPS)-konforme Endpunkte in US-Handelsregionen. Weitere Informationen finden Sie unter den folgenden Topics: • Federal Information Processing Standard (FIPS) 140-2 • 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 Ihr eigenes, vom Kunden verwaltetes CMK auswählen oder einen AWS verwalteten KMS-Schlüssel in Ihrem Konto verwenden. AWS KMS 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 vorhandenes, vom Kunden verwaltetes CMK aus — Von Kunden verwaltete CMK CMKs werden von Ihnen in (KMS) erstellt und verwaltet. AWS Key Management Service

Date	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: • Funktionsweise von Amazon MQ mit IAM • Unterstützte Berechtigungen auf Ressourcenebene für Amazon MQ-API-Aktionen
16. April 2019	Sie können jetzt über die REST-API Informationen über die Optionen der Broker-Engine und der Broker-Instance abrufen. Weitere Informationen finden Sie hier: • Broker-Instance-Optionen • Broker-Engine-Typen

Date	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 Spring XML-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.

Date	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 SOC-konformen 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 HIPAA-fähigen Service erweitert.

Date	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.

Date	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 MQ-API-Aufrufe zu protokollieren. Weitere Informationen finden Sie unter Logging Amazon MQ API calls using CloudTrail.

Date	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)

Date	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 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.

Date	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. • Wählen Sie auf der ***MyBroker***Seite im Abschnitt Verbindungen den Namen Ihrer Sicherheitsgruppe oder öffnen Sie die EC2-Konsole (anstelle der VPC-Konsole). Die EC2-Konsole ermöglicht eine intuitivere Konfiguration von Regeln für ein- und ausgehend en Datenverkehr. Weitere Informationen finden Sie im aktualisierten Abschnitt Connecting a Java application to your broker.
9. Januar 2018	• Die Berechtigung für die REST-Operations-ID `UpdateBroker` wird auf der IAM-Konsole korrekt als `mq:UpdateBroker` aufgelistet. • Die fehlerhafte `mq:DescribeEngine` -Berechtigung wurde von der IAM-Konsole entfernt.

Date	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 Verwendung des `mq.t2.micro` -Instance-Typs hängt von CPU-Guthaben und Basisleistung ab – mit der Fähigkeit, die Baseline-Ebene zu überschreiten (weitere Informationen finden Sie in der `CpuCreditBalance` -Metrik). Wenn Ihre Anwendung Fixed Performance, erwägen Sie, einem`q.m5.large` -Instance-Typ zu verwenden.
	• Sie können `mq.m4.large` - und `mq.t2.micro` -Broker erstellen.
	  Die Verwendung des `mq.t2.micro` -Instance-Typs hängt von CPU-Guthaben und Basisleistung ab – mit der Fähigkeit, die Baseline-Ebene zu überschreiten (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 mithilfe der Amazon MQ REST API und verwalten. AWS SDKs
	• Sie können auf Ihre Broker mithilfe von jeder Programmiersprache, die ActiveMQ unterstützt zugreifen, und indem Sie TLS explizit für die folgenden Protokolle aktivieren:
	  • AMQP
	  • MQTT
	  • MQTT über WebSocket
	  • OpenWire

Date	Aktualisierung der Dokumentation
	• STOMP • STOMP rüber 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.