



Entwicklerhandbuch

Amazon MQ



Amazon MQ: Entwicklerhandbuch

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

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon MQ?	1
Inwiefern unterscheidet sich Amazon MQ von Amazon SQS oder Amazon SNS?	1
Wie sehen meine ersten Schritte mit Amazon MQ aus?	1
Bitte geben Sie uns Feedback	2
Einrichtung	3
Schritt 1: Voraussetzungen	3
So melden Sie sich für ein AWS-Konto an	3
Einen Administratorbenutzer erstellen	4
Erstellen eines Benutzers und Abrufen Ihrer AWS-Anmeldeinformationen	5
Schritt 3: Vorbereiten der Verwendung des Beispiel-Codes	6
Nächste Schritte	7
Erste Schritte	8
Voraussetzungen	8
Erstellen und Herstellen einer Verbindung zu einem ActiveMQ-Broker	8
Schritt 1: Erstellen Sie einen ActiveMQ-Broker	9
Schritt 2: Verbinden einer Java-Anwendung mit Ihrem Broker	11
Schritt 3: (Optional) zu einer AWS Lambda Funktion verbinden	17
Schritt 4: Löschen Ihres Brokers	19
Nächste Schritte	20
Erstellen und Verbinden mit einem RabbitMQ-Broker	20
Schritt 1: Erstellen eines RabbitMQ-Brokers	21
Schritt 2: Verbinden einer JVM-Anwendung mit Ihrem Broker	23
Schritt 3: (Optional) Herstellen einer Verbindung mit einer - AWS Lambda Funktion	27
Schritt 4: Löschen Ihres Brokers	30
Nächste Schritte	31
Verwalten eines Brokers	32
Wartung eines Brokers	32
Einstellen des Broker-Wartungsfensters	33
Upgrade der Engine-Version	36
Manuelles Upgraden der Engine-Version	37
Automatisches Upgraden der Engine-Unterversion	40
Broker-Status	41
Auflisten von Brokern und Anzeigen von Broker-Details	42
So listen Sie Broker auf und zeigen Broker-Details an	43

Zugreifen auf die Webkonsole eines Brokers ohne öffentliche Zugänglichkeit	46
Voraussetzungen	46
Zugreifen auf die Webkonsole eines Brokers ohne öffentliche Zugänglichkeit	47
Neustarten eines Brokers	47
So starten Sie einen Amazon MQ-Broker neu	48
Löschen eines Brokers	48
Löschen eines Amazon MQ-Brokers	49
Broker-Konfigurationen	49
Broker-Konfigurationslebenszyklus	49
Instance-Typen	50
Amazon MQ für ActiveMQ Instance-Typen	51
Instance-Typen von Amazon MQ für RabbitMQ	58
Markieren von Ressourcen	59
Markieren für die Kostenzuordnung	60
Verwalten von Tags in der Amazon MQ-Konsole	61
Verwalten mit Amazon MQ-API-Aktionen	62
Amazon MQ für ActiveMQ	63
ActiveMQ Engine	63
Grundelemente	64
Broker-Architektur	77
Broker-Konfigurationen	92
Versionsverwaltung	128
Funktionierende Java-Beispiele	131
ActiveMQ Tutorials	142
Erstellen und Konfigurieren eines Brokers	142
Erstellen und Konfigurieren eines Netzwerks von Brokern	150
Verbinden einer Java-Anwendung mit Ihrem Broker	156
Integration von ActiveMQ Brokern in LDAP	161
Erstellen und Verwalten von Broker-Benutzern	177
Best Practices für Amazon MQ für ActiveMQ	180
Herstellen einer Verbindung mit Amazon MQ	180
Sicherstellen einer effektiven Amazon MQ-Leistung	184
Vermeiden von langsamen Neustarts durch Wiederherstellung vorbereiteter XA- Transaktionen	187
Regionsübergreifende Replikation	188
Primär- und Replikat-Broker	189

Erstellen/Löschen eines CRDR-Brokers	190
Initiieren eines Switchovers/Failovers	194
Metriken	197
Kontingente	199
Broker	200
Konfigurationen	201
Benutzer	201
Datenspeicherung	202
API-Drosselung	203
Amazon MQ	204
RabbitMQ-Engine	204
Grundelemente	204
Broker-Architektur	224
Broker-Konfigurationen	227
Versionsverwaltung	232
RabbitMQ-Tutorials	236
Bearbeiten von Broker-Einstellungen	236
Verwenden von Python Pika mit Amazon MQ for RabbitMQ	237
Beheben der angehaltenen Warteschlangen-Synchronisierung	244
Best Practices für Amazon MQ for RabbitMQ	251
Lazy-Warteschlangen aktivieren	252
Verwenden Sie persistente und dauerhafte Warteschlangen	253
Warteschlangen kurz halten	253
Bestätigung und Bestätigung konfigurieren	254
Konfigurieren des Vorabrufs	255
Konfigurieren von Celery	257
Automatische Wiederherstellung nach Netzwerkausfällen	257
Aktivieren von Classic Queue v2 für Ihren RabbitMQ-Broker	258
Kontingente	259
Broker	259
Datenspeicherung	260
API-Drosselung	261
Sicherheit	262
Datenschutz	263
Verschlüsselung	264
Verschlüsselung im Ruhezustand	264

Verschlüsselung während der Übertragung	273
Identity and Access Management	275
Zielgruppe	276
Authentifizierung mit Identitäten	276
Verwalten des Zugriffs mit Richtlinien	280
Funktionsweise von Amazon MQ mit IAM	283
Beispiele für identitätsbasierte Richtlinien	289
API-Authentifizierung und -Autorisierung	292
Von AWS verwaltete Richtlinien	297
Verwenden von servicegebundenen Rollen	298
Fehlerbehebung	304
Compliance-Validierung	306
Ausfallsicherheit	308
Sicherheit der Infrastruktur	308
Bewährte Methoden für die Gewährleistung der Sicherheit	309
Broker ohne öffentlichen Zugriff bevorzugen	309
Immer eine Autorisierungszuordnung konfigurieren	309
Blockieren unnötiger Protokolle	310
Protokollierung und Überwachung	311
Zugreifen auf CloudWatch-Metriken	311
AWS Management Console	312
AWS Command Line Interface	314
Amazon CloudWatch API	314
Überwachen von Brokern mithilfe von CloudWatch	314
Protokollieren und Überwachen von Amazon MQ für ActiveMQ Broker	315
Protokollierung und Überwachung von Amazon MQ für RabbitMQ Broker	325
Protokollieren von API-Aufrufen mit CloudTrail	333
Amazon MQ-Informationen in CloudTrail	334
Beispiel für einen Amazon MQ-Protokolldateieintrag	336
Konfigurieren von Amazon MQ zur Veröffentlichung von Protokollen in CloudWatch Logs	338
Konfigurieren von Amazon MQ für ActiveMQ-Protokolle	339
Konfigurieren von Amazon MQ für RabbitMQ-Protokolle	344
Kontingente	345
Broker	345
Konfigurationen	346
Benutzer	347

Datenspeicherung	348
API-Drosselung	349
Fehlerbehebung	351
Allgemeine Problembehebung	352
Ich kann keine Verbindung zu meiner Broker-Webkonsole oder -Endpunkten herstellen.	352
SSL-Ausnahmen	358
Ich habe einen Broker erstellt, aber die Brokererstellung ist fehlgeschlagen.	359
Mein Broker wurde neu gestartet und ich bin mir nicht sicher, warum.	359
Fehlerbehebung: Amazon MQ für ActiveMQ	360
Abrufen von CloudWatch Logs	361
Herstellen einer Verbindung zum Broker nach einem Neustart	361
Einige Clients können keine Verbindung herstellen	362
JSP-Ausnahme auf der Webkonsole	363
Fehlerbehebung: Amazon MQ für RabbitMQ	363
In CloudWatch können keine Metriken für meine Warteschlangen oder virtuellen Hosts angezeigt werden.	364
Wie aktiviere ich Plugins in Amazon MQ für RabbitMQ?	364
Ich kann die Amazon-VPC-Konfiguration für den Broker nicht ändern.	364
Fehlerbehebung: Erforderliche Codes für die Amazon-MQ-Aktion	364
RABBITMQ_MEMORY_ALARM	365
RABBITMQ_INVALID_KMS_KEY	374
BROKER_ENI_DELETED	375
BROKER_OOM	376
RABBITMQ_DISK_ALARM	378
Zugehörige Ressourcen	380
Amazon MQ-Ressourcen	380
Amazon MQ für ActiveMQ-Ressourcen	381
Amazon MQ für RabbitMQ-Ressourcen	381
Versionshinweise	383
Dokumentverlauf	415
AWS-Glossar	430
.....	cdxxxi

Was ist Amazon MQ?

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

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

Themen

- [Inwiefern unterscheidet sich Amazon MQ von Amazon SQS oder Amazon SNS?](#)
- [Wie sehen meine ersten Schritte mit Amazon MQ aus?](#)
- [Bitte geben Sie uns Feedback](#)

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

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

[Amazon SQS](#) und [Amazon SNS](#) sind Warteschlangen- und Themen-Services, die hochgradig skalierbar und benutzerfreundlich sind und keine Einrichtung von Message Brokern erfordern. Wir empfehlen diese Services für neue Anwendungen, die von praktisch unbegrenzter Skalierbarkeit und einfacher APIs profitieren können.

Wie sehen meine ersten Schritte mit Amazon MQ aus?

- Weitere Informationen zum Erstellen Ihres ersten Brokers mit Amazon MQ finden Sie unter [Getting Started with Amazon MQ](#).

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

Bitte geben Sie uns Feedback

Wir freuen uns über Ihr Feedback. Um Kontakt mit uns aufzunehmen, besuchen Sie das [Amazon MQ-Diskussionsforum](#).

Einrichten von Amazon MQ

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

Themen

- [Schritt 1: Voraussetzungen](#)
- [Schritt 2: Erstellen eines Benutzers und Abrufen Ihrer AWS-Anmeldeinformationen](#)
- [Schritt 3: Vorbereiten der Verwendung des Beispiel-Codes](#)
- [Nächste Schritte](#)

Schritt 1: Voraussetzungen

So melden Sie sich für ein AWS-Konto an

Wenn Sie kein AWS-Konto haben, führen Sie die folgenden Schritte zum Erstellen durch.

Anmeldung für ein AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/signup>.
2. Folgen Sie den Online-Anweisungen.

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

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

AWS sendet Ihnen eine Bestätigungs-E-Mail, sobald die Anmeldung abgeschlossen ist. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

Einen Administratorbenutzer erstellen

Wenn Sie sich für AWS-Konto registriert haben, sichern Sie Root-Benutzer des AWS-Kontos, aktivieren Sie AWS IAM Identity Center erstellen Sie einen Administratorbenutzer, damit Sie nicht den Root-Benutzer für alltägliche Aufgaben verwenden.

Schützen Ihres Root-Benutzer des AWS-Kontos

1. Melden Sie sich bei [AWS Management Console](#) als Kontobesitzer an, indem Sie Stammbenutzer 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-AnmeldungBenutzerhandbuch zu .

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen dazu finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für den Root-Benutzer Ihres AWS-Konto \(Konsole\)](#) im IAM-Benutzerhandbuch.

Erstellen eines Administratorbenutzers

1. IAM Identity Center aktivieren.

Eine genaue Anleitung finden Sie unter [Aktivierung von AWS IAM Identity Center](#) im AWS IAM Identity Center-Benutzerhandbuch.

2. Gewähren Sie im IAM Identity Center einem Administratorbenutzer Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter [Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis](#) im AWS IAM Identity CenterBenutzerhandbuch.

Als Administratorbenutzer anmelden

- 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 unter [Anmelden beim AWS-Zugangsportale](#) im AWS-Anmeldung Benutzerhandbuch zu.

Schritt 2: Erstellen eines Benutzers und Abrufen Ihrer AWS-Anmeldeinformationen

Benutzer benötigen programmgesteuerten Zugriff, wenn sie außerhalb der AWS Management Console mit AWS interagieren möchten. Die Vorgehensweise, um programmgesteuerten Zugriff zu gewähren, hängt davon ab, welcher Benutzertyp auf zugreift AWS.

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

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
Mitarbeiteridentität (Benutzer, die in IAM Identity Center verwaltet werden)	Verwenden Sie temporäre Anmeldeinformationen, um programmgesteuerte Anforderungen an die AWS CLI, AWS-SDKs oder AWS-APIs zu signieren.	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. <ul style="list-style-type: none"> • Informationen zur AWS CLI finden Sie unter Konfigurieren der AWS CLI für die Verwendung von 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 zu AWS-SDKs und Tools.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmgesteuerte Anforderungen an die AWS CLI, AWS-SDKs oder AWS-APIs zu signieren.	Folgen Sie den Anweisungen unter Verwenden temporärer Anmeldeinformationen mit AWS-Ressourcen im IAM-Benutzerhandbuch.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	<p>(Nicht empfohlen)</p> <p>Verwenden Sie langfristige Anmeldeinformationen, um programmgesteuerte Anforderungen an die AWS CLI, AWS-SDKs oder AWS-APIs zu signieren.</p>	<p>Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten.</p> <ul style="list-style-type: none"> • Informationen zur AWS CLI finden Sie unter Authentifizierung mit IAM-Benutzer-Anmeldeinformationen im AWS Command Line Interface-Benutzerhandbuch. • Informationen zu AWS-SDKs und Tools finden Sie unter Authentifizierung mit langfristigen Anmeldeinformationen im Referenzhandbuch zu AWS-SDKs und Tools. • Informationen zu AWS-APIs finden Sie unter Verwalten von Zugriffsschlüsseln für IAM-Benutzer im IAM-Benutzerhandbuch.

Schritt 3: Vorbereiten der Verwendung des Beispiel-Codes

Die folgenden Tutorials zeigen, wie Sie mit Amazon MQ-Brokern arbeiten können, indem Sie die AWS Management Consolesowie wie Sie eine Verbindung zu Ihrem Amazon MQ für ActiveMQ und Amazon MQ für RabbitMQ Broker programmatisch herstellen. Wenn Sie den Beispiel-Code verwenden möchten, müssen Sie das [Java Standard Edition Development Kit](#) installieren und einige Änderungen am Code vornehmen.

Sie können Broker auch programmgesteuert mit der Amazon MQ [REST-API](#) und den AWS SDKs erstellen und verwalten.

Nächste Schritte

Sie sind nun bereit für die ersten Schritte mit Amazon MQ und können [einen Broker erstellen](#). Abhängig von Ihrem Broker-Engine-Typ können Sie dann [Verbinden einer Java-Anwendung mit Ihrem Amazon MQ for ActiveMQ -Broker](#) oder verwenden Sie die RabbitMQ Java-Client-Bibliothek, um [Verbinden Sie eine JVM-basierte Anwendung mit Ihrem Amazon MQ for RabbitMQ Broker](#).

Erste Schritte mit Amazon MQ

Dieser Abschnitt hilft Ihnen dabei, sich mit Amazon MQ vertraut zu machen, indem er Ihnen zeigt, wie Sie einen Amazon MQ for ActiveMQ- oder RabbitMQ-Broker erstellen und Ihre Anwendung mit diesem verbinden.

Das Erstellen und ihre Verbindung zu einer Broker-Instance funktioniert für jede Broker-Engine etwas anders. Wählen Sie einen der folgenden Motortypen, die Sie verwenden möchten, um detaillierte Informationen zur Erstellung und Verbindung mit einem Broker zu erhalten. Nachdem Sie eine Broker erstellt und eine Verbindung mit ihr hergestellt haben, finden Sie Anweisungen, mit denen Sie ihn löschen können.

Themen

- [Voraussetzungen](#)
- [Erstellen und Herstellen einer Verbindung zu einem ActiveMQ-Broker](#)
- [Erstellen und Verbinden mit einem RabbitMQ-Broker](#)

Voraussetzungen

Bevor Sie beginnen, führen Sie die Schritte in [Setting Up Amazon MQ](#) aus.

Erstellen und Herstellen einer Verbindung zu einem ActiveMQ-Broker

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Die kombinierte Beschreibung der Broker-Instanceclass(m5,t3) undsize(large,micro) ist einBroker-Instance-Typ(zum Beispielmq.m5.large). Weitere Informationen finden Sie unter [Broker](#).

Themen


- [Schritt 1: Erstellen Sie einen ActiveMQ-Broker](#)
- [Schritt 2: Verbinden einer Java-Anwendung mit Ihrem Broker](#)
- [Schritt 3: \(Optional\) zu einer AWS Lambda Funktion verbinden](#)
- [Schritt 4: Löschen Ihres Brokers](#)

- [Nächste Schritte](#)

Schritt 1: Erstellen Sie einen ActiveMQ-Broker

Die erste und häufigste Amazon MQ-Aufgabe ist das Erstellen eines Brokers. Das folgende Beispiel zeigt, wie Sie AWS Management Console verwenden können, 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 [Broker Architecture](#).
 - 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 [Amazon MQ Single-Instance-Broker](#).
 - 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 [Aktive/Standby Amazon MQ-Broker für hohe Verfügbarkeit](#).
 - Weitere Informationen zu den Beispiel-Blueprints für ein Netzwerk von Brokern finden Sie unter [Beispiel-Vorlagen](#).
 - 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. Brokernamen sind für andere AWS-Services zugänglich, einschließlich CloudWatch Logs. Brokernamen sind nicht für private oder sensible Daten gedacht.

- b. Wählen Sie den Broker-Instance-Typ (z. B. mq.m5.large). Weitere Informationen finden Sie unter [Broker instance types](#).
5. Geben Sie im Abschnitt Zugriff auf ActiveMQ-Webkonsole einen Benutzernamen und ein Passwort an. Die folgenden Einschränkungen gelten in Bezug auf Benutzernamen und Passwörter des Brokers:
 - Ihr Benutzername darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten.
 - Ihr 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-Services 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 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.

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

7. Klicken Sie auf **MyBroker**.

Auf der Seite **MyBroker**, in dem Abschnitt Connect (Verbinden), notieren Sie die URL der [ActiveMQ-Webkonsole](#) für Ihren Broker, zum Beispiel:

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

Beachten Sie auch die [Wire-Level-Protokoll-Endpunkte](#). Das folgende Beispiel zeigt einen OpenWire-Endpunkt.

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

Schritt 2: Verbinden einer Java-Anwendung mit Ihrem Broker

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

Sie können unter Verwendung [verschiedener ActiveMQ-Clients](#) eine Verbindung zu ActiveMQ-Brokern einrichten. Wir empfehlen die Verwendung des [ActiveMQ-Clients](#).

Voraussetzungen

Aktivieren der VPC-Attribute

Note

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

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 in der Broker-Liste den Namen Ihres Brokers (z. B. MyBroker) enthalten.
3. Notieren Sie auf der Seite **MyBroker** im Abschnitt Verbindungen die Adressen und Ports der Webkonsolen-URL des Brokers und die Protokolle auf Leitungsebene.
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>5.15.8</version>
  </dependency>
  <dependency>
    <groupId>org.apache.activemq</groupId>
    <artifactId>activemq-pool</artifactId>
    <version>5.15.8</version>
  </dependency>
</dependencies>
```

Weitere Informationen über `activemq-client.jar` finden Sie unter [Ursprüngliche Konfiguration](#) in der Apache ActiveMQ-Dokumentation.

Important

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

Erstellen eines Nachrichtenproduzenten und Senden einer Nachricht

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 Aktiv/Standby-Broker stellt Amazon MQ zwei URLs der ActiveMQ Web Console bereit, jedoch 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 [Broker Architecture](#).

Für Drahtebene Protokollendpunkte können Sie zulassen, dass Ihre Anwendung eine Verbindung zu einem beliebigen Endpunkt herstellen kann, indem Sie die [Failover-Transport](#) verwenden.


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

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

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

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

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

 Note

Nachrichtenproduzenten sollten immer die `PooledConnectionFactory`-Klasse. Weitere Informationen finden Sie unter [Verwenden Sie immer Verbindungspools](#).

- 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);
```

- 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.");
```

- Bereinigen Sie den Produzenten.

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

Erstellen eines Nachrichtenkonsumenten und Empfangen der Nachricht

- 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](#).

- 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);
```

- 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-Dienste (wie Amazon SQS), ist der Verbraucher ständig mit dem Broker verbunden.

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

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

Schritt 3: (Optional) zu einer AWS Lambda Funktion verbinden

AWS Lambda kann eine Verbindung zu Ihrem Amazon MQ -Broker herstellen und diese nutzen. Wenn Sie einen Broker mit Lambda verbinden, erstellen Sie eine [Ereignisquellen-Zuweisung](#), der Nachrichten aus einer Warteschlange liest und die Funktions[synchron](#). 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](#)
 - [logs:CreateLogStream](#)
 - [logs:PutLogEvents](#)
 - [secretsmanager: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 unter [Internet- und Servicezugriff für VPC-verbundene Funktionen](#) im AWS Lambda Entwicklerhandbuch.
 - Erstellen Sie mithilfe eines VPC-Endpunkts eine Verbindung zwischen Ihrer Amazon Virtual Private Cloud (Amazon VPC) und Lambda. Ihre Amazon VPC muss sich auch mit AWS Security Token Service (AWS STS) und Secrets-Manager-Endpunkten verbinden. Weitere Informationen finden Sie unter [Konfigurieren von Schnittstellen-VPC-Endpunkten für Lambda](#) im AWS Lambda Entwicklerhandbuch.
3. [Konfigurieren Sie Ihren Broker als Ereignisquelle](#) Verwendung für eine Lambda -Funktion unter Verwendung der AWS Management Console. Sie können auch den [create-event-source-mapping](#) AWS Command Line Interface Befehl verwenden.
 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": "QUJD0kFBQUE=",
        "connectionId": "myJMScoID",
        "redelivered": false,
        "destination": {
          "physicalname": "testQueue"
        },
        "timestamp": 1598827811958,
```

```
    "brokerInTime": 1598827811958,
    "brokerOutTime": 1598827811959
  },
  {
    "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
    "messageType": "jms/bytes-message",
    "data": "3DT00W7crj51prgVLQaGQ82S48k=",
    "connectionId": "myJMScoID1",
    "persistent": false,
    "destination": {
      "physicalName": "testQueue"
    },
    "timestamp": 1598827811958,
    "brokerInTime": 1598827811958,
    "brokerOutTime": 1598827811959
  }
]
}
```

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.

Schritt 4: Löschen Ihres Brokers

Wenn Sie keinen Amazon MQ-Broker verwenden (und in naher Zukunft nicht voraussehen), ist es eine bewährte Vorgehensweise, ihn aus Amazon MQ zu löschen, um Ihre AWS-Kosten nachzuerfolgen.

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

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 anschließend auf Löschen.
3. Geben Sie im Dialogfenster *MyBroker* löschen? den Text `delete` ein und wählen Sie dann Löschen aus.

Das Löschen eines Brokers dauert ca. 15 Minuten.

Nächste Schritte

Nachdem Sie nun einen Broker erstellt, eine Anwendung mit diesem verbunden und eine Nachricht gesendet und empfangen haben, können Sie Folgendes ausprobieren:

- [Creating and configuring a broker](#) (Erweiterte Einstellungen)
- [Tutorial: Bearbeiten der Broker-Engine-Version, des Instance-Typs, von CloudWatch Logs und von Wartungseinstellungen](#)
- [Creating and applying broker configurations](#)
- [Listing brokers and viewing broker details](#)
- [Erstellen und Verwalten von ActiveMQ-Broker-Benutzern](#)
- [Rebooting a Broker](#)
- [Zugreifen auf CloudWatch Metriken für Amazon MQ](#)

Sie können auch damit beginnen, sich eingehender mit [Bewährten Methoden für Amazon MQ](#) und [Amazon-MQ-REST-APIs](#) zu beschäftigen, und dann [eine Migration auf Amazon MQ](#) planen.

Erstellen und Verbinden mit einem RabbitMQ-Broker

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Die kombinierte Beschreibung der Broker-Instanceclass(m5,t3) undsize(large,micro) ist einBroker-Instance-Typ(z. B.mq.m5.large).

Themen

- [Schritt 1: Erstellen eines RabbitMQ-Brokers](#)
- [Schritt 2: Verbinden einer JVM-Anwendung mit Ihrem Broker](#)
- [Schritt 3: \(Optional\) Herstellen einer Verbindung mit einer - AWS Lambda Funktion](#)
- [Schritt 4: Löschen Ihres Brokers](#)
- [Nächste Schritte](#)

Schritt 1: Erstellen eines RabbitMQ-Brokers

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

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Klicken Sie auf derWählen Sie Broker-EngineWählen Sie auf der SeiteRabbitMQKlicken Sie auf und danach auf Weiter.
3. Klicken Sie auf derAuswählen des BereitstellungsmodusWählen Sie auf der SeiteBereitstellungsmodusZum BeispielCluster-BereitstellungKlicken Sie auf und danach auf Weiter.
 - Ein Single-Instance-Broker besteht aus einem Broker in einer Availability Zone hinter einem Network Load Balancer (NLB). Der Broker kommuniziert mit Ihrer Anwendung und mit einem Amazon EBS-Speicher-Volume. Weitere Informationen finden Sie unter [Single-Instance Broker](#).
 - 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 [Cluster-Bereitstellung für hohe Verfügbarkeit](#).
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 - AWS Services zugänglich, einschließlich - CloudWatch Protokollen. Brokernamen sind nicht für private oder sensible Daten gedacht.

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

 Note

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

5. Klicken Sie auf **Konfigurieren der Einstellungen**. Klicken Sie auf der Seite **Zugriff auf RabbitMQ**-Abschnitt eine **Benutzername:** und **Passwort**. 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 - AWS Services zugänglich, einschließlich - CloudWatch Protokollen. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

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

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

Die Erstellung eines Brokers dauert etwa 15 Minuten.

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

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

9. Wählen Sie **MyBroker**.

Notieren Sie sich auf der **MyBroker** Seite im Abschnitt Verbinden die URL der [RabbitMQ-Webkonsole](#) Ihres Brokers, zum Beispiel:

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

Beachten Sie auch die [Sicherer AMQP-Endpunkt](#). Es folgt ein Beispiel für ein amqpsEndpunkt Zuweisen auf Listener-Port5671.

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

Schritt 2: Verbinden einer JVM-Anwendung mit Ihrem Broker

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

Voraussetzungen

Note

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

Aktivieren 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 in 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 die 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 Source (ursprünglich Benutzerdefiniert) 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-Endpoint zu konfigurieren, den Sie zuvor notiert haben. Für AMQP-SWire-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 `AMQPBuilder`-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 zum Arbeiten mit der RabbitMQ-Java-Client-Bibliothek finden Sie im [RabbitMQ Java-Client-API-Handbuch](#).


Schritt 3: (Optional) Herstellen einer Verbindung mit einer - AWS Lambda Funktion

AWS Lambda kann eine Verbindung zu Ihrem Amazon MQ-Broker herstellen und Nachrichten von diesem verwenden. Wenn Sie einen Broker mit Lambda verbinden, erstellen Sie eine

[Ereignisquellen-Zuweisung](#), der Nachrichten aus einer Warteschlange liest und die Funktion [synchron](#). 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](#)
 - [-Protokolle:CreateLogGroup](#)
 - [-Protokolle:CreateLogStream](#)
 - [-Protokolle:PutLogEvents](#)
 - [secretsmanager: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 unter [Internet- und Servicezugriff für VPC-verbundene Funktionen](#) im AWS 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-Endpunkten herstellen. Weitere Informationen finden Sie unter [Konfigurieren von Schnittstellen-VPC-Endpunkten für Lambda](#) im AWS Lambda Entwicklerhandbuch.

3. [Konfigurieren Sie Ihren Broker als Ereignisquelle](#) Verwendung für eine Lambda -Funktion unter Verwendung der AWS Management Console. Sie können auch den [create-event-source-mapping](#) AWS Command Line Interface Befehl verwenden.
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,
                118,
                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": "eyJ0aW1lb3V0IjowLCJkYXRhIjo1Q1pybWYwR3c4T3Y0YnFMUXhENEUifQ=="
}
]
}
}
```

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.

Schritt 4: Löschen Ihres Brokers

Wenn Sie keinen Amazon MQ-Broker verwenden (und in naher Zukunft nicht voraussehen), ist es eine bewährte Methode, ihn aus Amazon MQ zu löschen, um Ihre AWS Kosten zu senken.

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

1. Melden Sie sich bei der [Amazon-MQ-Konsole](#) an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und wählen Sie dann Löschen aus.
3. Geben Sie im Dialogfeld Delete **MyBroker?** ein delete und wählen Sie dann Delete aus.

Das Löschen eines Brokers dauert ca. 15 Minuten.

Nächste Schritte

Nachdem Sie nun einen Broker erstellt, eine Anwendung mit diesem verbunden und eine Nachricht gesendet und empfangen haben, können Sie Folgendes ausprobieren:

- [Tutorial: Bearbeiten der Broker-Engine-Version, des Instance-Typs, von CloudWatch Logs und von Wartungseinstellungen](#)
- [Listing brokers and viewing broker details](#)
- [Erstellen und Verwalten von ActiveMQ-Broker-Benutzern](#)
- [Rebooting a Broker](#)
- [Zugreifen auf CloudWatch Metriken für Amazon MQ](#)

Sie können auch beginnen, sich eingehend mit [Bewährte Methoden für Amazon MQ](#) und [Amazon-MQ-REST-APIs](#) zu beschäftigen, bevor Sie die Migration zu Amazon MQ planen.

Verwalten eines Amazon MQ -Brokers

In den folgenden Abschnitten finden Sie Anleitungen zur Verwaltung und Wartung Ihrer Amazon MQ-Broker.

Themen

- [Verwalten eines Amazon MQ-Brokers](#)
- [Aktualisieren einer Amazon MQ-Broker-Engine-Version](#)
- [Broker-Status](#)
- [Auflisten von Amazon MQ-Brokern und Anzeigen von Broker-Details](#)
- [Zugreifen auf die Webkonsole eines Brokers ohne öffentliche Zugänglichkeit](#)
- [Neustart eines Amazon MQ-Brokers](#)
- [Löschen eines Amazon MQ-Brokers](#)
- [Verwaltung von Amazon-MQ-Brokerkonfigurationen](#)
- [Instance-Typen](#)
- [Markieren von Ressourcen](#)

Verwalten eines Amazon MQ-Brokers

In regelmäßigen Abständen führt Amazon MQ Wartungsarbeiten an der Hardware, dem Betriebssystem oder der Engine-Software ein Nachrichtenbroker durch. Die Dauer der Wartung variiert, kann jedoch bis zu zwei Stunden dauern, abhängig von den Vorgängen, die für den Nachrichtenbroker geplant sind. Wenn Sie beispielsweise [Upgrades für automatische Engines](#) aktiviert, oder den Typ der Broker-Instance geändert haben, übernimmt Amazon MQ Ihre Änderungen während des nächsten geplanten Wartungsfensters.

Um Ausfallzeiten während eines Wartungsfensters zu minimieren, empfehlen wir, einen Broker-Bereitstellungsmodus mit hoher Verfügbarkeit über mehrere Availability Zones (AZ) hinweg auszuwählen. Je nach Typ Ihres Broker-Engines stellt Amazon MQ die folgenden Multi-AZ-Bereitstellungsmodi zur Verfügung.

- Amazon MQ for ActiveMQ— Amazon MQ für ActiveMQ bietet [aktive/Standby-Modus-Bereitstellungen](#) für Hochverfügbarkeit. Im Aktiv-/Standby-Modus führt Amazon MQ Wartungsvorgänge jeweils eine Instance durch und stellt sicher, dass mindestens eine Instance

verfügbar bleibt. Darüber hinaus können Sie ein [Netzwerk von Brokern](#) mit Wartungsfenstern über die Woche verstreut konfigurieren.

- Amazon MQ für RabbitMQ— Amazon MQ für RabbitMQ bietet die [Cluster](#)-Bereitstellungen für Hochverfügbarkeit. In Cluster-Bereitstellungen führt Amazon MQ Wartungsvorgänge für jeweils einen Knoten durch, wobei immer mindestens zwei Knoten in Betrieb bleiben.

Weitere Informationen zu den empfohlenen Best Practices von Amazon MQ, um sicherzustellen, dass Ihre Broker während und nach einem Wartungsfenster effektiv arbeiten, finden Sie in der folgenden Dokumentation zu Ihrem Broker-Engine-Typ.

- [the section called “Best Practices für Amazon MQ für ActiveMQ”](#)
- [the section called “Best Practices für Amazon MQ for RabbitMQ”](#)

Sie können planen, dass die Wartung einmal pro Woche zu einem bestimmten Zeitpunkt, der bis zu zwei Stunden dauert, durchgeführt wird. Dadurch wird das Fenster für Wartungsaktionen von Amazon MQ festgelegt, die geplant und gestartet werden sollen.

Sie können das Wartungsfenster planen, wenn Sie Ihren Broker zum ersten Mal erstellen oder indem Sie Ihre Brokereinstellungen aktualisieren. Im folgenden Thema wird das Anpassen des Broker-Wartungsfensters mithilfe der und AWS Management Console AWS CLI der Amazon MQ-API beschrieben.

Themen

- [Einstellen des Broker-Wartungsfensters](#)

Einstellen des Broker-Wartungsfensters

Während des ausgewählten Wartungsfensters führt Amazon MQ alle ausstehenden Änderungen durch, z. B. automatische Nebenversions-Upgrades. Um das Broker-Wartungsfenster anzupassen, können Sie die AWS CLI, AWS Management Console die oder die Amazon MQ-API verwenden.

Important

Sie können das Wartungsfenster eines Brokers bis zu vier Mal vor dem nächsten geplanten Wartungsfenster anpassen. Amazon MQ wendet eine Begrenzung von vier Wartungsfenster-Anpassungen an, um sicherzustellen, dass kritische Software- und Sicherheitspatches sowie wichtige Hardware-Upgrades nicht auf unbestimmte Zeit verschoben werden.

Sobald ein Broker-Wartungsfenster abgeschlossen ist, setzt Amazon MQ das Limit zurück, sodass Sie den Zeitplan anpassen können, bevor das nächste Wartungsfenster eintritt. Die Broker-Verfügbarkeit ist bei der Anpassung des Broker-Wartungsfensters nicht betroffen.

AWS Management Console

So passen Sie das Broker-Wartungsfenster mithilfe der an AWS Management Console

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

Note

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

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

AWS CLI

So passen Sie das Broker-Wartungsfenster mithilfe der an 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-3i45j6k17819, wäre die Broker-ID b-1234a5b6-78cd-901e-2fgh-3i45j6k17819.

- `--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-Broker](#) CLI-Befehl, um zu überprüfen, ob das Wartungsfenster erfolgreich aktualisiert wurde.

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

Amazon 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 den verfügbaren Amazon-MQ-Endpunkten finden Sie unter [Amazon-MQ-Endpunkte und -Kontingente](#) in der Allgemeine AWS-Referenz

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

Verwenden Sie den `maintenanceWindowStartTime`-Parameter und den [WeeklyStartTime](#)-Ressourcentyp in der Anforderungsnutzlast.

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

- (Optional) Verwenden Sie die [DescribeBroker](#) -API-Operation, um zu überprüfen, ob das Wartungsfenster erfolgreich aktualisiert wurde. `broker-id` wird als Pfadparameter angegeben.

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

Aktualisieren einer Amazon MQ-Broker-Engine-Version

Amazon MQ bietet neue Broker-Engine-Versionen für alle unterstützten Broker-Engine-Typen. Neue Engine-Versionen können Sicherheits-Patches, Fehlerbehebungen und andere Verbesserungen der Broker-Engine enthalten. Wenn von Amazon MQ eine neue Engine-Version unterstützt wird, können Sie festlegen, wie und wann ein Upgrade für den Broker durchgeführt wird.

Broker-Engine-Versionen sind als `X.Y.Z` organisiert. In der Amazon MQ Implementierung jedes Modultyps, gilt `X.Y` als Hauptversion und `Z` als Unterversion. Es gibt zwei Typen von Aktualisierungen:

- Hauptversions-Upgrade: Tritt auf, wenn sich die Versionsnummern der Haupt-Engine ändern. Zum Beispiel gilt ein Upgrade von Version 1.0 zu Version 1.1 als Hauptversions-Upgrade.
- Unterversion-Upgrade: Tritt auf, wenn sich nur die Versionsnummer der Neben-Engine ändert. Beispielsweise wird ein Upgrade von Version 1.1.0 auf Version 1.1.1 als Nebenversion Upgrade betrachtet.

Weitere Informationen zur Haupt- und Nebenversions-Verwaltung für jeden bestimmten Broker-Engine-Typ finden Sie in den folgenden Themen.

- [the section called "Versionsverwaltung."](#)

- [the section called “Versionsverwaltung.”](#)

Wenn Sie die [Automatische Unterversion-Upgrades](#) aktiviert haben, aktualisiert Amazon MQ Ihren Broker auf neue Nebenversionen, sobald diese verfügbar sind. Automatische Upgrades von Unterversionen werden nur durchgeführt, wenn der Broker eine Unterversion ausgeführt wird, die niedriger ist als die neue empfohlene Unterversion. Bei größeren Upgrades müssen Sie die Engine-Version manuell aktualisieren.

Sowohl manuelle als auch automatische Versions-Upgrades erfolgen während des geplanten Wartungsfensters oder nachdem Sie [So starten Sie Ihren Broker neu](#).

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

Themen

- [Manuelles Upgraden der Engine-Version](#)
- [Automatisches Upgraden der Engine-Unterversion](#)

Manuelles Upgraden der Engine-Version

So upgraden Sie die Engine-Version eines Brokers manuell auf eine neue Haupt- oder Nebenversion: AWS Management Console, die AWS CLI oder die Amazon MQ-API.

AWS Management Console

So upgraden Sie die Engine-Version eines Brokers mit der AWS Management Console

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Wählen Sie im linken Navigationsbereich die Option Broker, und dann in der Liste den Broker aus, den Sie upgraden möchten.
3. Klicken Sie auf der Seite Broker-Details auf Edit.
4. **UNDERTechnische Daten, für-Broker-Engine-Version** Wählen Sie in der Dropdown-Liste die neue Versionsnummer .
5. Scrollen Sie ans Seitenende und wählen Sie Änderungen im Zeitplan.
6. Klicken Sie auf der Änderungen für Broker-Seite, für Wann Änderungen angewendet werden Wählen Sie eine der folgenden Optionen .

- Wählen Sie Nach dem nächsten Neustart, wenn Sie möchten, dass Amazon MQ das Versions-Upgrade während des nächsten geplanten Wartungsfensters abschliesst.
- Wählen Sie Sofort, wenn Sie den Broker neu starten und die Engine-Version sofort aktualisieren möchten.

 **Important**

Ihr Broker ist offline, während er neu gestartet wird.

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

AWS CLI

So upgraden Sie die Engine-Version eines Brokers mit der 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-3i45j6k17819`, wäre die Broker-ID `b-1234a5b6-78cd-901e-2fgh-3i45j6k17819`.
 - `--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 [Neustart-Broker](#)-CLI-Befehl, um Ihren Broker neu zu starten, wenn Sie die Engine-Version sofort aktualisieren möchten.

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

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

⚠ Important

Ihr Broker ist offline, während er neu gestartet wird.

Amazon MQ-API

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

1. Verwenden Sie die [UpdateBroker](#) API-Operation. 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 [RebootBroker](#) API-Vorgang, um Ihren Broker neu zu starten, wenn Sie die Engine-Version sofort aktualisieren möchten. `broker-id` wird als Pfadparameter angegeben.

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

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

 **Important**

Ihr Broker ist offline, während er neu gestartet wird.


Automatisches Upgraden der Engine-Unterversion

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

AWS Management Console

So aktivieren Sie automatische Upgrades von Unterversionen mithilfe der AWS Management Console

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Wählen Sie im linken Navigationsbereich die Option Broker, um dann in der Liste den Broker auszuwählen, den Sie upgraden möchten.
3. Klicken Sie auf der Seite Broker-Details auf Edit.
4. **UNTERWARTUNG**, wählen Sie **Aktivieren von automatischen Upgrades von Unterversionen**.

 **Note**

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

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

AWS CLI

Um die automatische Aktualisierung von Unterversionen mithilfe der AWS CLI zu aktivieren, verwenden Sie den CLI-Befehl [Update-Broker](#) und geben Sie die folgenden Parameter an.

- `--broker-id` - Die eindeutige ID, die Amazon MQ für die Broker-Instance generiert. Sie können die ID von Ihrem Broker ARN analysieren. Beispielsweise angesichts der folgenden ARN, `arn:aws:mq:us-`

east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819, wäre die Broker-ID b-1234a5b6-78cd-901e-2fgh-3i45j6k17819.

- `--auto-minor-version-upgrade`— Aktiviert die Option für automatische Unterversions-Upgrades.

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

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

Amazon MQ-API

Um die automatische Aktualisierung von Unterversionen mithilfe der Amazon-MQ-API zu aktivieren, verwenden Sie API-Operation [UpdateBroker](#). Geben Sie `broker-id` an als Pfadparameter. Im folgenden Beispiel wird von einem Broker in der `us-west-2` Region ausgegangen. Weitere Informationen zu den verfügbaren Amazon-MQ-Endpunkten finden Sie unter [Amazon-MQ-Endpunkte und -Kontingente](#) in der Allgemeine AWS-Referenz

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

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

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

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

Broker-Status

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

Konsole	API	Beschreibung
Fehler beim Erstellen	CREATION_FAILED	Der Broker konnte nicht erstellt werden.
Wird erstellt	CREATION_IN_PROGRESS	Der Broker wird derzeit erstellt.
Wird gelöscht	DELETION_IN_PROGRESS	Der Broker wird derzeit gelöscht.
Laufender Neustart	REBOOT_IN_PROGRESS	Die Broker wird derzeit neu gestartet.
In Ausführung	RUNNING	Die Broker ist betriebsbereit.
Kritische Aktion erforderlich	CRITICAL_ACTION_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 erforderlichen Code für die Aktion aus der Liste the section called “Fehlerbehebung: Erforderliche Codes für die Amazon-MQ-Aktion” auswählen.

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

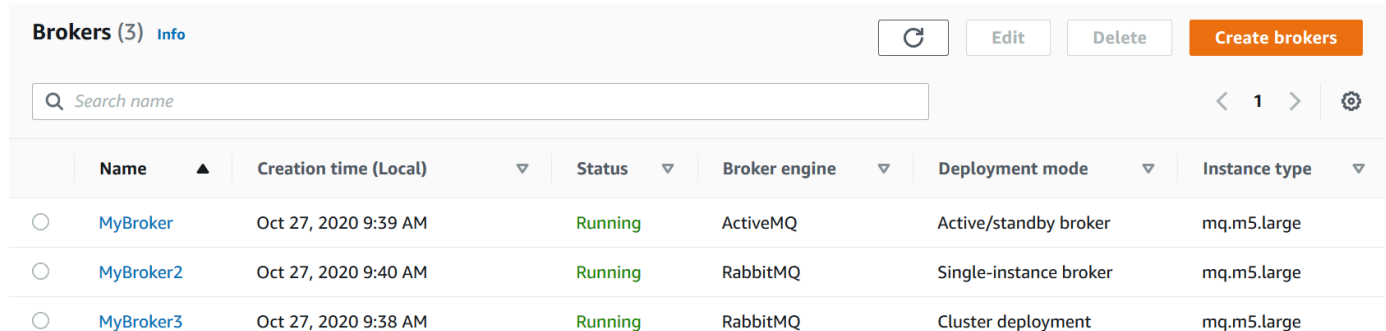
Wenn Sie von Amazon MQ anfordern, einen Broker zu erstellen, kann der Vorgang ca. 10 Minuten dauern.

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

So listen Sie Broker auf und zeigen Broker-Details an

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.

Ihre Broker in der aktuellen Region werden aufgelistet.



The screenshot shows the Amazon MQ console interface. At the top, there is a header 'Brokers (3) Info' with a refresh button, 'Edit', 'Delete', and a prominent orange 'Create brokers' button. Below the header is a search bar with the placeholder text 'Search name'. The main content is a table with the following columns: Name, Creation time (Local), Status, Broker engine, Deployment mode, and Instance type. There are three rows of data representing different brokers.

	Name ▲	Creation time (Local) ▼	Status ▼	Broker engine ▼	Deployment mode ▼	Instance type ▼
<input type="radio"/>	MyBroker	Oct 27, 2020 9:39 AM	Running	ActiveMQ	Active/standby broker	mq.m5.large
<input type="radio"/>	MyBroker2	Oct 27, 2020 9:40 AM	Running	RabbitMQ	Single-instance broker	mq.m5.large
<input type="radio"/>	MyBroker3	Oct 27, 2020 9:38 AM	Running	RabbitMQ	Cluster deployment	mq.m5.large

Für jeden Broker werden die folgenden Informationen angezeigt:

- Name
- Erstellungsdatum
- [Status](#)
- Bereitstellungsmodus
- [Instance-Typ](#)

2. Wählen Sie den Namen Ihres Brokers.

Auf der Seite **MyBroker** werden die [konfigurierten](#) Details für Ihre Broker angezeigt:

Details

ARN [Info](#)

arn:aws:mq:us-west-2:123878009876:broker:MyBroker:b-2f91ed40-de60-40b2-9141-ddce16cb0a0f

Specifications

Broker status

Running

Broker name

MyBroker

Broker instance type [Info](#)

mq.m5.large

Deployment mode [Info](#)

Active/standby broker

Storage type [Info](#)

Amazon Elastic File System

Broker engine [Info](#)

ActiveMQ

Broker engine version

5.15.12

Configuration

Configuration name

MyBroker-configuration

Configuration revision

Revision 1 - Auto-generated default for MyBroker-configuration on ActiveMQ 5.15.12

CloudWatch Logs

General

Disabled - [Logs](#)

Audit

Disabled - [Logs](#)

Security and network

VPC [Info](#)

vpc-286cba5b [↗](#)

Subnet(s) [Info](#)

subnet-4388bb98 [↗](#)

subnet-7942b82g [↗](#)

Security group(s) [Info](#)

sg-1abc5867 [↗](#)

Public accessibility [Info](#)

Yes

IP Addresses

53.208.204.167

46.290.203.267

Maintenance

Automatic minor version upgrade

Yes


Maintenance window

Saturday 19:00 - 21:00 UTC

Für Amazon MQ for RabbitMQ Broker können Sie Ihre ausgewählten Einstellungen auf der Seite **MyBroker2** unter dem Details-Abschnitt ansehen.

Details

ARN [Info](#)

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

Broker name

MyBroker2

Broker status

Running

Creation time

Oct 27, 2020 9:40 AM

Broker engine [Info](#)

RabbitMQ

Deployment mode [Info](#)

Single-instance broker

Broker instance type [Info](#)

mq.m5.large

Broker engine version

3.8.6

CloudWatch Logs

Disabled - [Logs](#)

Maintenance

Automatic minor version upgrade

Yes

Maintenance window

Tuesday 18:00 - 20:00 UTC

Security and network

VPC [Info](#)

vpc-111cca5b [↗](#)

Subnet(s) [Info](#)

subnet-8vr11jn8 [↗](#)

Public accessibility [Info](#)

Yes

Unter dem Abschnitt Details werden die folgenden Informationen angezeigt:

- Im Abschnitt Verbindungen die URL für Amazon MQ for ActiveMQ Web-Konsole und die Wire-Level-Protokoll-Endpunkte

Connections

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

**Enable connections to your broker**

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

**ActiveMQ Web Console**

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

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

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

Endpoints

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

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

Im Abschnitt Verbindungen für Amazon MQ für RabbitMQ Broker, die URL der Webkonsole und den sicheren AMQP-Endpunkt.

Connections

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

RabbitMQ web console

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

Endpoints

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

- Für Amazon MQ für ActiveMQ Broker finden Sie im Benutzer-Abschnitt im [Benutzer](#) im Zusammenhang mit dem Broker

**Important**

Verwalten von Benutzern über die AWS Management Console und die Amazon MQ-API wird für Amazon MQ für RabbitMQ-Broker nicht unterstützt.

Zugreifen auf die Webkonsole eines Brokers ohne öffentliche Zugänglichkeit

Wenn Sie die öffentliche Zugänglichkeit für Ihren Broker deaktivieren, müssen Sie die folgenden Schritte ausführen, um Zugriff auf die ActiveMQ-Webkonsole Ihres Brokers zu erhalten.

Note

Die Namen der VPCs und der Sicherheitsgruppen sind für das folgende Beispiel spezifisch.

Voraussetzungen

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

- VPCs
 - Die VPC ohne Internet-Gateway, der der Amazon MQ-Broker zugewiesen ist, mit dem Namen `private-vpc`.
 - Eine zweite VPC, mit einem Internet-Gateway, namens `public-vpc`.
 - Beide VPCs müssen verbunden sein (beispielsweise mit [VPC-Peering](#)), so dass die Amazon EC2-Instances in der öffentlichen VPC mit den EC2-Instances in der privaten VPC kommunizieren können.
 - Wenn Sie VPC-Peering verwenden, müssen die Routing-Tabellen für beide VPCs für die Peering-Verbindung konfiguriert werden.
- Sicherheitsgruppen
 - Die Sicherheitsgruppe, die für die Erstellung des Amazon MQ-Brokers verwendet wird, namens `private-sg`.
 - Eine zweite Sicherheitsgruppe, die für die EC2-Instance in der VPC `public-vpc` verwendet wird, namens `public-sg`
 - `private-sg` muss eingehende Verbindungen zulassen von `public-sg`. Wir empfehlen, diese Sicherheitsgruppe auf Port 8162 für ActiveMQ und auf Port 443 für RabbitMQ zu beschränken.
 - `public-sg` muss eingehende Verbindungen von Ihrem Computer auf Port 22 zulassen.

Zugreifen auf die Webkonsole eines Brokers ohne öffentliche Zugänglichkeit

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 wie [FoxyProxy](#) auf 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.

Neustart eines Amazon MQ-Brokers

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

Note

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

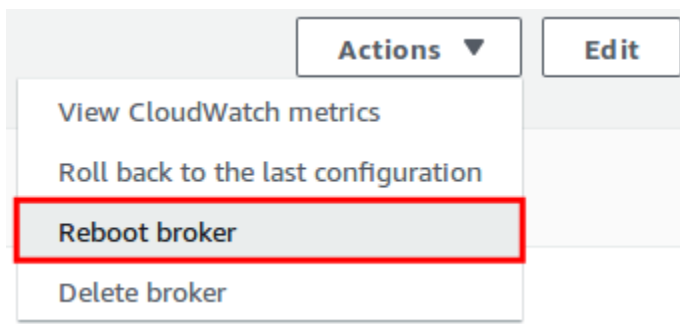
Das folgende Beispiel zeigt, wie Sie einen Amazon MQ-Broker mithilfe der AWS Management Console neu starten.

So starten Sie einen Amazon MQ-Broker neu

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Wählen Sie in der Broker-Liste den Namen Ihres Brokers (z. B. MyBroker).
3. Wählen Sie auf der Seite **MyBroker** den Eintrag 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 in naher Zukunft nicht voraussehen), ist es eine bewährte Methode, ihn aus Amazon MQ zu löschen, um Ihre AWS-Kosten nachzuverfolgen.

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

Löschen eines Amazon MQ-Brokers

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und klicken Sie anschließend auf Löschen.
3. Geben Sie im Dialogfenster *MyBroker* löschen? den Text delete ein und wählen Sie dann Löschen aus.

Das Löschen eines Brokers dauert ca. 15 Minuten.

Verwaltung von Amazon-MQ-Brokerkonfigurationen

Eine Konfiguration enthält alle Einstellungen für Ihren Broker. Sie können eine Konfiguration erstellen, bevor Sie Broker erstellen. Sie können die Konfiguration dann auf mindestens einen Broker anwenden.

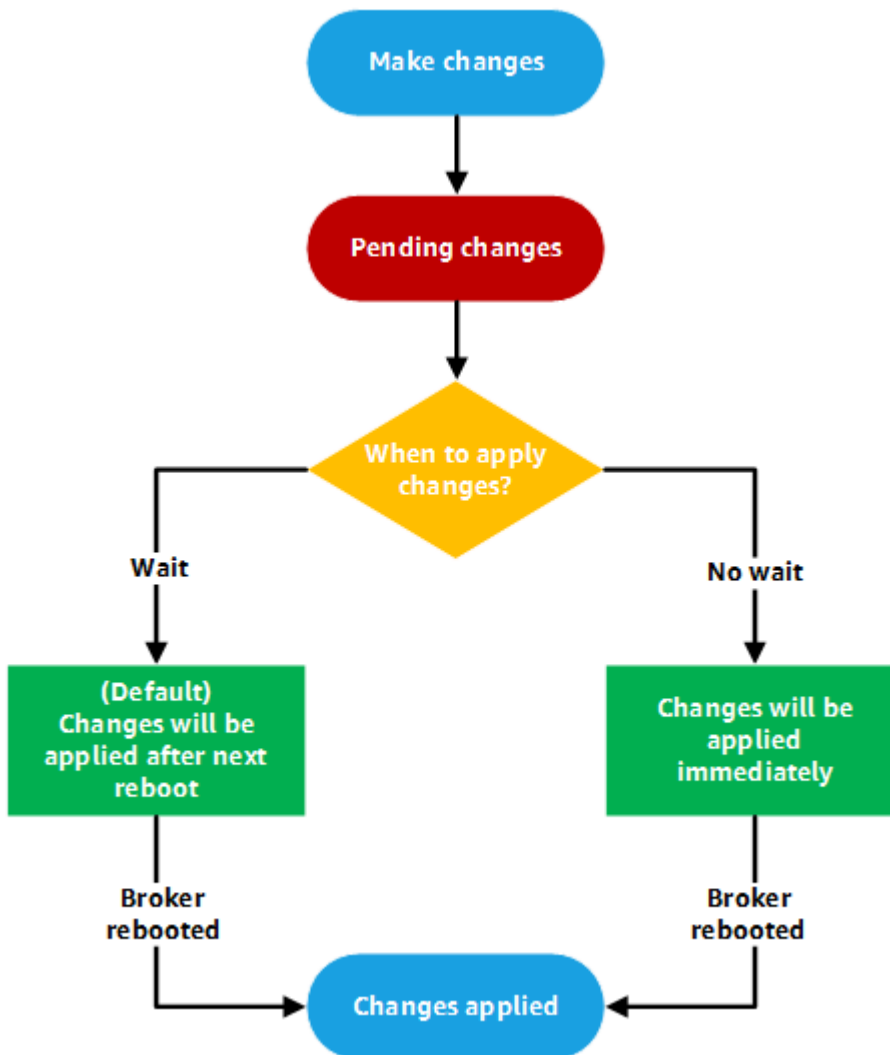
Amazon-MQ-Broker-Konfigurationslebenszyklus

Änderungen an einer Konfigurationsrevision oder an einem ActiveMQ Benutzer nicht, um die Änderungen sofort anzuwenden. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder [den Broker neu starten](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).

Das folgende Diagramm verdeutlicht den Konfigurationslebenszyklus.

Important

Das nächste geplante Wartungsfenster löst einen Neustart . Wenn der Broker vor dem nächsten geplanten Wartungsfenster neu gestartet wird, werden die Änderungen nach dem Neustart übernommen.



Bei ActiveMQ enthält eine Konfiguration alle Einstellungen für Ihren Broker im XML-Format (ähnlich wie die `activemq.xml`-Datei von ActiveMQ). Weitere Informationen zum Erstellen, Anwenden und Bearbeiten von ActiveMQ-Brokerkonfigurationen finden Sie unter [Creating and applying broker configurations](#).

Bei RabbitMQ enthält eine Konfiguration alle Einstellungen für Ihren RabbitMQ-Broker im Cuttlefish-Format. Weitere Informationen zum Erstellen, Anwenden und Bearbeiten von RabbitMQ-Brokerkonfigurationen finden Sie unter [Creating and applying broker configurations](#).

Instance-Typen

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


Themen

- [Amazon MQ für ActiveMQ Instance-Typen](#)
- [Instance-Typen von Amazon MQ für RabbitMQ](#)

Amazon MQ für ActiveMQ Instance-Typen

Important

Sie können Amazon EBS nur mit dem `mq.m5Broker`-Instance-Typ. Weitere Informationen finden Sie unter [Storage](#).


Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
<code>mq.t2.micro</code>	1	1	Niedrig	<p>Verwenden Sie den <code>mq.t2.micro</code>-Instance-Typ für eine grundlegende Auswertung von Amazon MQ. Dieser Instance-Typ (nur bei Single-Instance-Brokern) ist für das AWS kostenlose Kontingent qualifiziert.</p> <div data-bbox="1258 1654 1510 1879" style="border: 1px solid #add8e6; border-radius: 10px; padding: 5px;"> <p> Note</p> <p>Verwendung des <code>mq.t2.micro</code></p> </div>

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				<p>IO - Instance-Typ hängt von CPU-Guthaben und Basisleistung ab – mit der Fähigkeit, über der Baseline-Ebene zu bersten (Weitere Informationen finden Sie unter CpuCreditBalance - Metrik). Wenn Ihre Anwendung Feste Leistung erfordert, erwägen Sie,</p>

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				<p>einen mq.m5.large -Instance-Typ zu verwenden .</p>
mq.t3.micro	2	1	Niedrig	<p>Verwenden Sie den mq.t3.micro -Instance-Typ für eine grundlegende Auswertung von Amazon MQ. Dieser Instance-Typ (nur bei Single-Instance-Brokern) ist für das AWSkostenlose Kontingent</p>

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
mq.m4.large	2	8	Mittel	Mit dem Instance-Typ mq.m4.large können Sie die Kompatibilität mit vorhandenen Broker-Bereitstellungen sichern. Wir raten zur Verwendung einer mq.m5.*-Instance für neue Broker.
mq.m5.large	2	8	Hoch	Verwenden Sie die mq.m5.large-Instance für reguläre Entwicklungs-, Test- und Produktions-Workloads.

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
mq.m5.xlarge	4	16	Hoch	Verwenden Sie Instances vom Typ mq.m5.xlarge , mq.m5.2xlarge und mq.m5.4xlarge für reguläre Entwicklungs-, Test- und Produktions-Workloads, die einen höheren Durchsatz erfordern.
mq.m5.2xlarge	8	32	Hoch	
mq.m5.4xlarge	16	64	Hoch	

 **Note**

Wenn Ihr System persistente Nachrichten verwendet , hängt sein Durchsatz davon ab, wie schnell Nachrichten

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				<p>abgerufen werden. Wenn Nachrichten nicht sofort abgerufen werden, lässt sich durch Verwenden größerer Instance-Typen bei persistenten Nachrichten möglicherweise nicht der Systemdurchsatz verbessern. In diesem Fall empfehlen wir, für das concurrentStoreAndDispatchQ</p>


Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				<p>ueues - Attribut false festzuleg en. Weitere Informati onen finden Sie unter Gleichzei tige Speicheru ng und Bereitste llung für Warteschl angen mit langsamen Konsument en deaktivie ren.</p>

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

Instance-Typen von Amazon MQ für RabbitMQ

Important

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

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
<code>mq.t3.micro</code>	2	1	Niedrig	<p>Verwenden Sie den <code>mq.t3.micro</code>-Instance-Typ für eine grundlegende Auswertung von Amazon MQ. Dieser Instance-Typ (nur bei Single-Instance-Brokern) ist für das AWSkostenlose Kontingent</p> <div data-bbox="1258 1344 1510 1816" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>Der <code>mq.t3.micro</code>-Instance-Typ unterstützt nicht Cluster-B</p> </div>

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				Vorbereitung.
mq.m5.large	2	8	Hoch	Verwenden Sie die mq.m5.large -Instance für reguläre Entwicklungs-, Test- und Produktions-Workloads.
mq.m5.xlarge	4	16	Hoch	Verwenden Sie Instances vom Typ mq.m5.xlarge ,
mq.m5.2xlarge	8	32	Hoch	mq.m5.2xlarge und
mq.m5.4xlarge	16	64	Hoch	mq.m5.4xlarge für reguläre Entwicklungs-, Test- und Produktions-Workloads, die einen höheren Durchsatz erfordern.

Markieren von Ressourcen

Amazon MQ unterstützt das Markieren von Ressourcen, um Ihre Kostenzuordnung nachzuverfolgen. Sie können Ressourcen beim Erstellen oder durch Anzeigen der Details dieser Ressource markieren.

Themen

- [Markieren für die Kostenzuordnung](#)
- [Verwalten von Tags in der Amazon MQ-Konsole](#)
- [Verwalten mit Amazon MQ-API-Aktionen](#)

Markieren für die Kostenzuordnung

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. Organisieren Sie Ihre AWS-Rechnung mit Kostenzuordnungs-Tags, damit Sie Ihre eigene Kostenstruktur wiedergeben können. Melden Sie sich an, um Ihre AWS-Kontorechnung mit Tag-Schlüsseln und Werten zu erhalten. Weitere Informationen finden Sie unter [Einrichten Ihres monatlichen Kostenzuordnungsberichts](#) im AWS BillingBenutzerhandbuch.

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

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

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

Verwalten von Tags in der Amazon MQ-Konsole

Hinzufügen von Tags zu neuen Ressourcen

Mit Amazon MQ können Sie Tags zu Ressourcen hinzufügen, während sie erstellt werden. Sie können schnell Tags zu den Ressourcen hinzufügen, die Sie in der Amazon MQ-Konsole erstellen.

Um Tags hinzuzufügen, während Sie einen neuen Broker erstellen:

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.

Tags - optional

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

Key

Value - optional

4. (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 unter Tags auf der Seite Erstellen einer Konfiguration Tag hinzufügen aus.
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.

Anzeigen und Verwalten von Tags für vorhandene Ressourcen

Mit Amazon MQ können Sie Tags für Ihre Ressourcen in der Amazon MQ Konsole anzeigen und verwalten. Sie können Tags für eine einzelne Ressource verwalten, indem Sie die Tags auf der Detailseite dieser Ressource bearbeiten. So bearbeiten Sie Tags für Amazon MQ-Ressourcen:

1. Wählen Sie entweder **Broker** oder **Konfigurationen** in der Amazon MQ-Konsole.

Überprüfen Sie im Abschnitt **Tags** die vorhandenen Tags für diese Ressource.
2. Um neue oder bestehende Tags hinzuzufügen oder zu verwalten, wählen Sie **Edit (Bearbeiten)** (oder **Create tag (Tag erstellen)**, wenn es keine vorhandenen Tags gibt) aus.
3. Aktualisieren der Tags für Ihre Ressource:
 - Bearbeiten Sie zum Ändern bestehender Tags die Werte **Key (Schlüssel)** und **Value (Wert)**.
 - Um vorhandene Tags zu entfernen, wählen Sie **Remove (Entfernen)** aus.
 - Um ein neues Tag hinzuzufügen, wählen Sie **Add tag (Tag hinzufügen)** aus und geben Sie einen **Key (Schlüssel)** und **Value (Wert)** ein.
4. Wählen Sie **Save**.

Verwalten mit Amazon MQ-API-Aktionen

Mit Amazon MQ können die Tags Ihrer Ressourcen über die REST-API anzeigen und verwalten.

Weitere Informationen finden Sie in der [Amazon MQ REST API-Referenz](#).

Arbeiten mit Amazon MQ für ActiveMQ

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 AWS Management Console, Amazon MQ-REST-API oder der AWS Command Line Interface erstellen, verwalten oder löschen.

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 API-Referenz](#).

Themen

- [ActiveMQ Engine](#)
- [ActiveMQ Tutorials](#)
- [Best Practices für Amazon MQ für ActiveMQ](#)
- [Regionsübergreifende Datenreplikation für Amazon MQ für ActiveMQ](#)
- [Kontingente in Amazon MQ für ActiveMQ](#)

ActiveMQ Engine

Dieser Abschnitt beschreibt die grundlegenden Elemente eines ActiveMQ-Brokers, gibt einen Überblick über die Optionen der ActiveMQ-Broker-Architektur, erläutert die Konfigurationsparameter des Brokers und bietet ein Arbeitsbeispiel mit Java Message Service (JMS).

Themen

- [Grundelemente](#)
- [Broker-Architektur](#)
- [Konfigurationen für Amazon MQ für ActiveMQ Broker](#)
- [Verwalten von Amazon MQ für ActiveMQ Engine-Versionen](#)
- [Funktionierende Beispiele für die Verwendung von Java Message Service \(JMS\) mit ActiveMQ](#)

Grundelemente

Dieser Abschnitt bietet eine Einführung in wichtige Konzepte zum Verständnis von Amazon MQ.

Themen

- [Broker](#)
- [Broker-Instance-Typen](#)
- [Konfiguration](#)
- [Benutzer](#)
- [Storage](#)

Broker

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Die kombinierte Beschreibung der Broker-Instanceclass(m5,t3) undsize(large,micro) ist einBroker-Instance-Typ(zum Beispielmq.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-Volumen.
- 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 [Broker Architecture](#).

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:

- [Creating and configuring a broker](#)
- [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 über WebSocket

Attribute

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

- Einen Namen (MyBroker)
- Eine ID (b-1234a5b6-78cd-901e-2fgh-3i45j6k17819)
- Einen Amazon-Ressourcennamen (ARN) (arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819)
- Eine ActiveMQ-Webkonsolen-URL (https://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.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 `diactivemq-webconsole` können Sie die ActiveMQ Webkonsole nicht verwenden, da die Gruppe nicht berechtigt ist, Nachrichten an den Amazon MQ -Broker zu senden oder von ihm Nachrichten zu empfangen.

- Wire-Level-Protokoll-Endpunkte:


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

 Note

Dies ist ein 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 Aktiv/Standby-Broker stellt Amazon MQ zwei URLs der ActiveMQ Web Console bereit, jedoch 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-Instance-Typen

Important

Sie können Amazon EBS nur mit dem `mq.m5` Broker-Instance-Typ. Weitere Informationen finden Sie unter [Storage](#).


Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
<code>mq.t2.micro</code>	1	1	Niedrig	<p>Verwenden Sie den <code>mq.t2.micro</code>-Instance-Typ für eine grundlegende Auswertung von Amazon MQ. Dieser Instance-Typ (nur bei Single-Instance-Brokern) ist für das AWS kostenlose Kontingent qualifiziert.</p> <div data-bbox="1258 1417 1510 1879" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Verwendung des <code>mq.t2.micro</code>-Instance-Typ hängt von</p> </div>

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				<p>CPU-Guthaben und Basisleistung ab – mit der Fähigkeit, über der Baseline-Ebene zu bersten (Weitere Informationen finden Sie unter CpuCreditBalance - Metrik). Wenn Ihre Anwendung Feste Leistung erfordert, erwägen Sie, einen <code>m5.large</code> - Instance-Typ zu</p>

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				verwenden
mq.t3.micro	2	1	Niedrig	Verwenden Sie den mq.t3.micro -Instance-Typ für eine grundlegende Auswertung von Amazon MQ. Dieser Instance-Typ (nur bei Single-Instance-Brokern) ist für das AWSkostenlose Kontingent
mq.m4.large	2	8	Mittel	Mit dem Instance-Typ mq.m4.large können Sie die Kompatibilität mit vorhandenen Broker-Bereitstellungen sichern. Wir raten zur Verwendung einer mq.m5.*-Instance für neue Broker.

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
mq.m5.large	2	8	Hoch	Verwenden Sie die mq.m5.large -Instance für reguläre Entwicklungs-, Test- und Produktions-Workloads.

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
mq.m5.xlarge	4	16	Hoch	Verwenden Sie Instances vom Typ mq.m5.xlarge , mq.m5.2xlarge und mq.m5.4xlarge für reguläre Entwicklungs-, Test- und Produktions-Workloads, die einen höheren Durchsatz erfordern.
mq.m5.2xlarge	8	32	Hoch	
mq.m5.4xlarge	16	64	Hoch	

 **Note**

Wenn Ihr System persistente Nachrichten verwendet , hängt sein Durchsatz davon ab, wie schnell Nachrichten

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				<p>abgerufen werden. Wenn Nachrichten nicht sofort abgerufen werden, lässt sich durch Verwenden größerer Instance-Typen bei persistenten Nachrichten möglicherweise nicht der Systemdurchsatz verbessern. In diesem Fall empfehlen wir, für das concurrentStoreAndDispatchQ</p>

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				<p>ueues - Attribut false festzuleg en. Weitere Informati onen finden Sie unter Gleichzei tige Speicheru ng und Bereitste llung für Warteschl angen mit langsamen Konsument en deaktivie ren.</p>

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

Konfiguration

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.

⚠ Important

Das Vornehmen von Änderungen an einer Konfiguration nichtwenden 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](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).

Derzeit ist es nicht möglich, eine Konfiguration zu löschen.

Weitere Informationen zum Erstellen, Bearbeiten und Verwalten von Konfigurationen finden Sie unter:

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

Um die Änderungen an Ihren Konfigurationen nachzuverfolgen, können Sie Konfigurationsrevisionen erstellen. Weitere Informationen finden Sie unter [Creating and applying broker configurations](#).

Attribute

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

- Einen Namen (MyConfiguration)
- Eine ID (c-1234a5b6-78cd-901e-2fgh-3i45j6k17819)
- Einen Amazon-Ressourcennamen (ARN) (arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k17819)

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](#)

Benutzer

Ein ActiveMQBenutzer ist 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 die [ActiveMQ-Webkonsole](#) zuzugreifen.

Eine Gruppe ist 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 finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).

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:

- [Erstellen und Verwalten von ActiveMQ-Broker-Benutzern](#)
- [Benutzer](#)

Attribute

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](#)

Storage

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 demmq .m5Broker-Instance-Typ.
- Obwohl Sie den Broker-Instance-Typ ändern können, können Sie den Speichertyp des Brokers nicht mehr 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	<ul style="list-style-type: none"> • Aktienkurse • Aktualisierungen von Standortdaten 	5,000	Keine

Speichertyp	Persistenz	Beispiela nwendungsfall	Ungefähre maximale Anzahl von Nachricht en, die pro Produzent pro Sekunde (1-KB- Nachricht) in die Warteschlange gestellt werden	Replikation
		<ul style="list-style-type: none"> • Häufig geänderte Daten 		
Amazon EBS	Persistent	<ul style="list-style-type: none"> • Umfangreiche Textmengen • Antragsbearbeitung 	500	Mehrere Kopien innerhalb einer einzigen Availability Zone (AZ)
Amazon EFS	Persistent	Finanztransaktionen	80	Mehrere Kopien über mehrere AZs hinweg

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 für Langlebigkeit und Replizierbarkeit über mehrere AZs hinweg entworfen, um Datenverlust aufgrund des Ausfalls einer einzelnen Komponente oder eines Problems, das die Verfügbarkeit einer AZ betrifft, zu verhindern. ist für den Durchsatz optimiert und wird über mehrere Server innerhalb einer einzigen AZ repliziert. Amazon EBS ist für den Durchsatz optimiert und über mehrere Server innerhalb einer einzelnen AZ repliziert.

Broker-Architektur

Amazon-MQ-for-ActiveMQ-Broker können als Single-Instance-Broker oder aktiv/standby-Broker erstellt werden. Für beide Bereitstellungsmodi bietet Amazon MQ eine hohe Haltbarkeit, indem seine Daten redundant gespeichert werden.

Note

Amazon MQ verwendet [Apache KahaDB](#) als Datenspeicher. Andere Datenspeicher, wie JDBC und LevelDB, werden nicht unterstützt.

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 über WebSocket

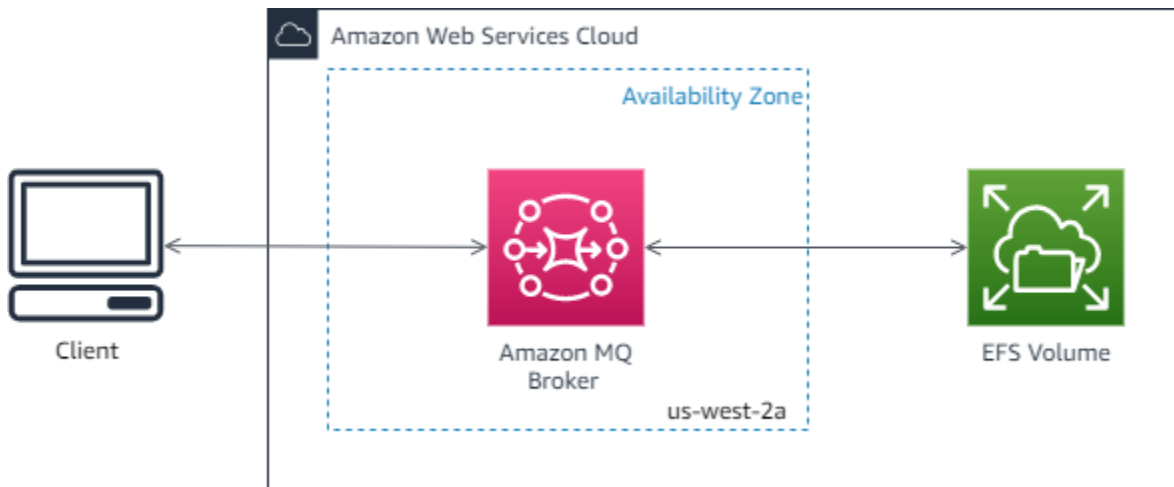
Themen

- [Amazon MQ Single-Instance-Broker](#)
- [Aktive/Standby Amazon MQ-Broker für hohe Verfügbarkeit](#)
- [Amazon MQ Netzwerk von Brokern](#)

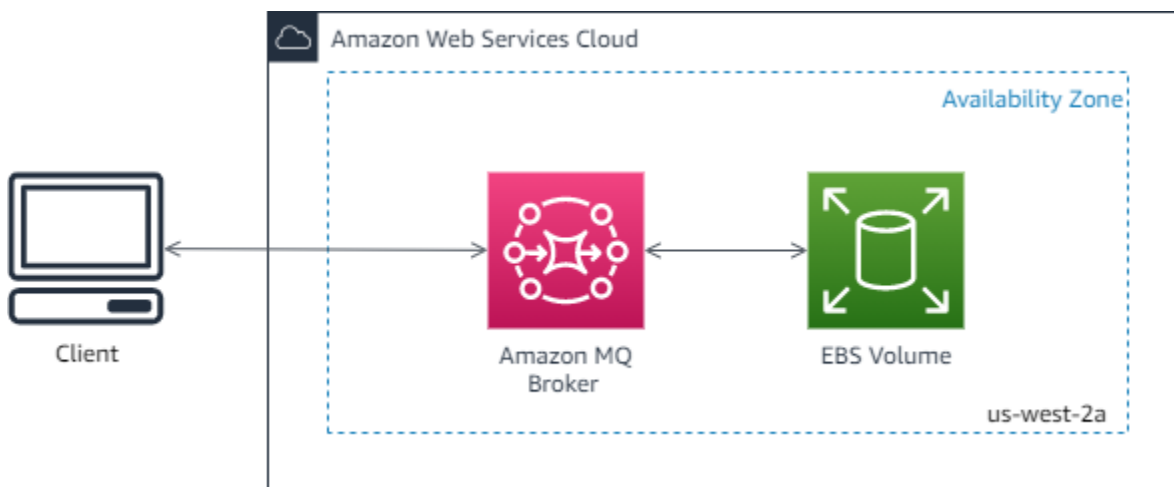
Amazon MQ Single-Instance-Broker

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-Volumen. Amazon EFS Speicher-Volumen bieten ein Höchstmaß an Beständigkeit und Verfügbarkeit, indem Daten redundant über mehrere Availability Zones (AZs) gespeichert werden. Amazon EBS bietet Speicher auf Blockebene, der für niedrige Latenz und hohen Durchsatz optimiert ist. Weitere Informationen zu Speicheroptionen finden Sie unter [Storage](#).

Das folgende Diagramm veranschaulicht einen Single-Instance-Broker mit Amazon EFS-Speicher, der über mehrere Availability Zones (AZs) repliziert wird.



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



Aktive/Standby Amazon MQ-Broker für hohe Verfügbarkeit

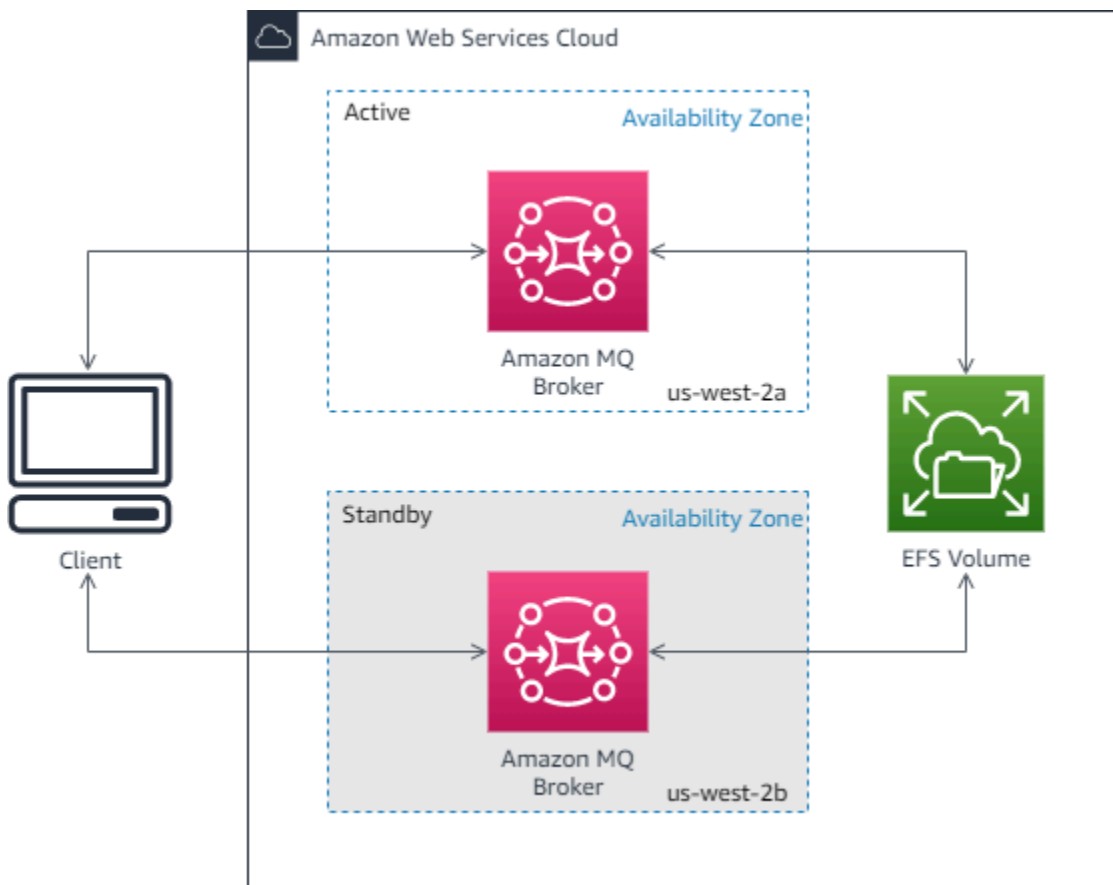
Ein aktiver/Standby-Broker besteht aus zwei Brokern in zwei verschiedenen Availability Zones, die in einem redundanten Paar. Diese Broker kommunizieren synchron mit Ihrer Anwendung und mit Amazon EFS. Amazon EFS Speicher-Volumes bieten ein Höchstmaß an Haltbarkeit und Verfügbarkeit, indem Daten redundant über mehrere Availability Zones (AZs) gespeichert werden. Weitere Informationen finden Sie unter [Storage](#).

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

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

Für einen Aktiv/Standby-Broker stellt Amazon MQ zwei URLs der ActiveMQ Web Console bereit, jedoch 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. Für Drahtebene Protokollendpunkte können Sie zulassen, dass Ihre Anwendung eine Verbindung zu einem beliebigen Endpunkt herstellen kann, indem Sie die [Failover-Transport](#) verwenden.

Das folgende Diagramm zeigt einen aktiv/standby-Broker mit Amazon-EFS-Speicher.



Amazon MQ Netzwerk von Brokern

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

Ein -Netzwerk von Brokern besteht aus mehreren gleichzeitig aktiven [Single-Instance-Broknern](#) oder [aktiven/Standby-Broknern](#). Sie können Netzwerke von Brokern in einer Vielzahl von [Topologien](#) konfigurieren (zum Beispiel concentrator, hub-and-spokes, tree, oder mesh) abhängig von den Anforderungen Ihrer Anwendung, wie Hochverfügbarkeit und Skalierbarkeit.

Zum Beispiel kann ein [Hub-und-Spoke](#)-Netzwerk von Brokern die Ausfallsicherheit erhöhen und Nachrichten erhalten, wenn ein Broker nicht erreichbar ist. Ein Netzwerk von Brokern mit einer [Konzentrator](#) Topologie kann Nachrichten von einer größeren Anzahl von Brokern sammeln, die eingehende Nachrichten akzeptieren, und sie auf zentralere Broker konzentrieren, um die Belastung vieler eingehender Nachrichten besser zu bewältigen.

Ein Tutorial und detaillierte Konfigurationsinformationen finden Sie im Folgenden:

- [Creating and Configuring a Network of Brokers](#)
- [Korrekte Konfiguration Ihres Netzwerk von Brokern](#)
- [networkConnector](#)
- [networkConnectionStartAsync](#)
- [Netzwerke von Brokern](#) in der ActiveMQ-Dokumentation

Vorteile aus der Nutzung eines Netzwerks von Brokern:

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

Themen

- [Wie funktioniert ein Netzwerk von Brokern?](#)
- [Wie geht ein Netzwerk von Brokern mit Anmeldeinformationen um?](#)
- [Beispiel-Vorlagen](#)
- [Topologien für Netzwerke von Brokern](#)
- [Regionsübergreifend](#)
- [Dynamisches Failover mit Transport Connectors](#)

Wie funktioniert ein Netzwerk von Brokern?

Amazon MQ unterstützt das ActiveMQ-Netzwerk von Brokern auf verschiedene Weise. Erstens können Sie die Parameter innerhalb der Konfiguration jedes Brokers bearbeiten, um ein Netzwerk von Brokern zu erstellen, genau wie bei nativem ActiveMQ. Zweitens hat Amazon MQ Beispiel-Vorlagen, die AWS CloudFormation verwenden, um die Erstellung eines Netzwerks von Brokern zu automatisieren. Sie können diese Beispiel-Vorlagen direkt aus der Amazon MQ-Konsole verwenden, oder Sie können die zugehörigen AWS CloudFormation-Vorlagen bearbeiten, um eigene Topologien und Konfigurationen zu erstellen.

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

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

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

Wie geht ein Netzwerk von Brokern mit Anmeldeinformationen um?

Damit sich Broker A mit Broker B in einem Netzwerk verbinden kann, muss Broker A gültige Anmeldeinformationen verwenden, wie jeder andere Produzent oder Verbraucher. Anstatt

ein Passwort in der `<networkConnector>`-Konfiguration von Broker A anzugeben, müssen Sie zunächst einen Benutzer auf dem Broker A mit den gleichen Werten wie ein anderer Benutzer auf dem Broker B anlegen (dies sind separate, einzigartige Benutzer, die die gleichen Werte für Benutzername und Passwort verwenden). Wenn Sie das Attribut `username` in der `<networkConnector>`-Konfiguration angeben, fügt Amazon MQ das Passwort zur Laufzeit automatisch hinzu.

⚠ Important

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

Broker müssen sich im gleichen VPC oder in per Peering verbundenen VPCs befinden. Weitere Informationen finden Sie unter [Voraussetzungen](#) im Tutorial [Creating and Configuring a Network of Brokers](#).

Beispiel-Vorlagen

Um mit der Nutzung eines Netzwerks von Brokern zu beginnen, bietet Amazon MQ Beispiel-Vorlagen. Diese Beispiel-Vorlagen erstellen ein Netzwerk von Brokern und alle verwandten Ressourcen mit AWS CloudFormation. Die beiden verfügbaren Beispiel-Vorlagen sind:

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

Sample blueprints for a network of brokers

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

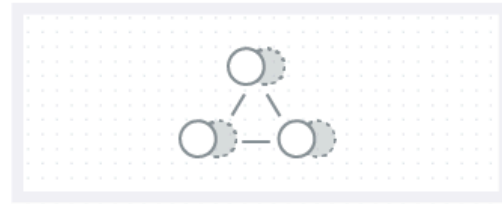
Mesh network of single-instance brokers

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



Mesh network of active/standby brokers

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



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

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

Topologien für Netzwerke von Brokern

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



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

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



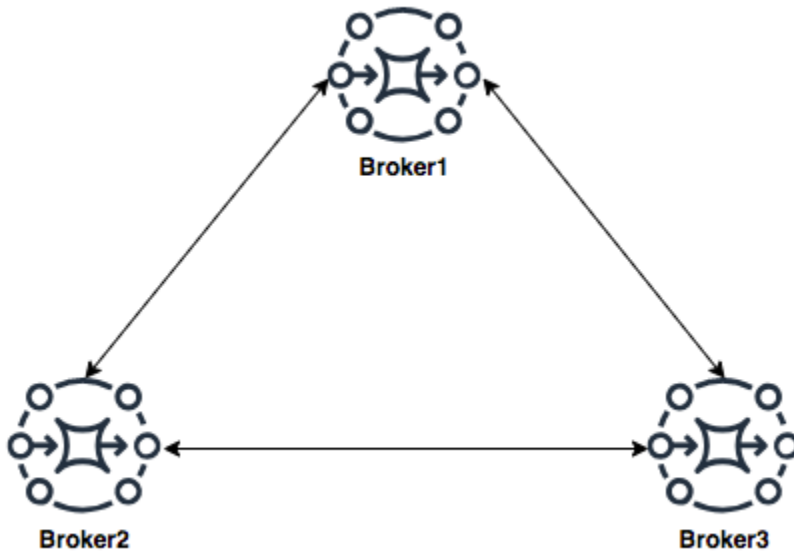
Durch die Verwendung von Duplex- und Nicht-Duplex-Netzwerk-Connectors ist es möglich, ein Netzwerk von Brokern in einer beliebigen Anzahl von Netzwerktopologien aufzubauen.

Note

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

Mesh-Topologie

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



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

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

Netzwerk-Connectors für Broker1:

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

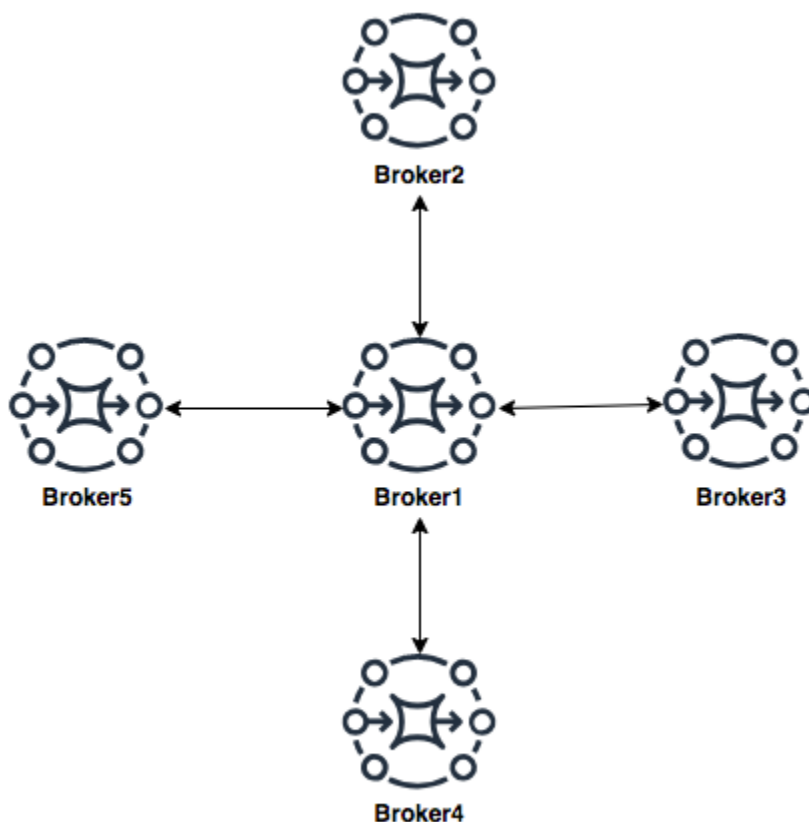
Netzwerk-Connectors für Broker2:

```
<networkConnectors>
  <networkConnector name="connector_2_to_3" userName="myCommonUser" duplex="true"
    uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

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

Hub-and-Spoke-Topologie

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

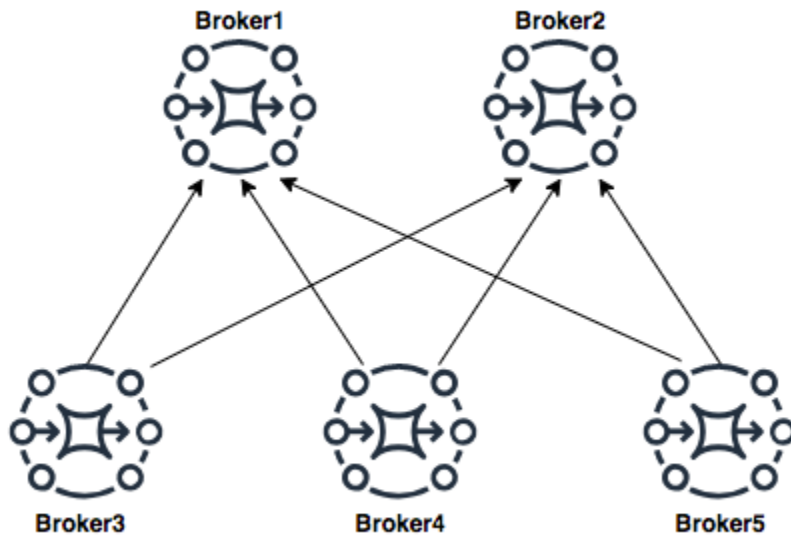


Um in diesem Beispiel das Hub-and-Spoke-Netzwerk von Brokern zu konfigurieren, können Sie einen `networkConnector` an jeden der Broker auf den Spokes in der Konfiguration von Broker1 hinzufügen.

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

Konzentrator-Topologie

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



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

Netzwerk-Connectors für Broker3:

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

Netzwerk-Connectors für Broker4:

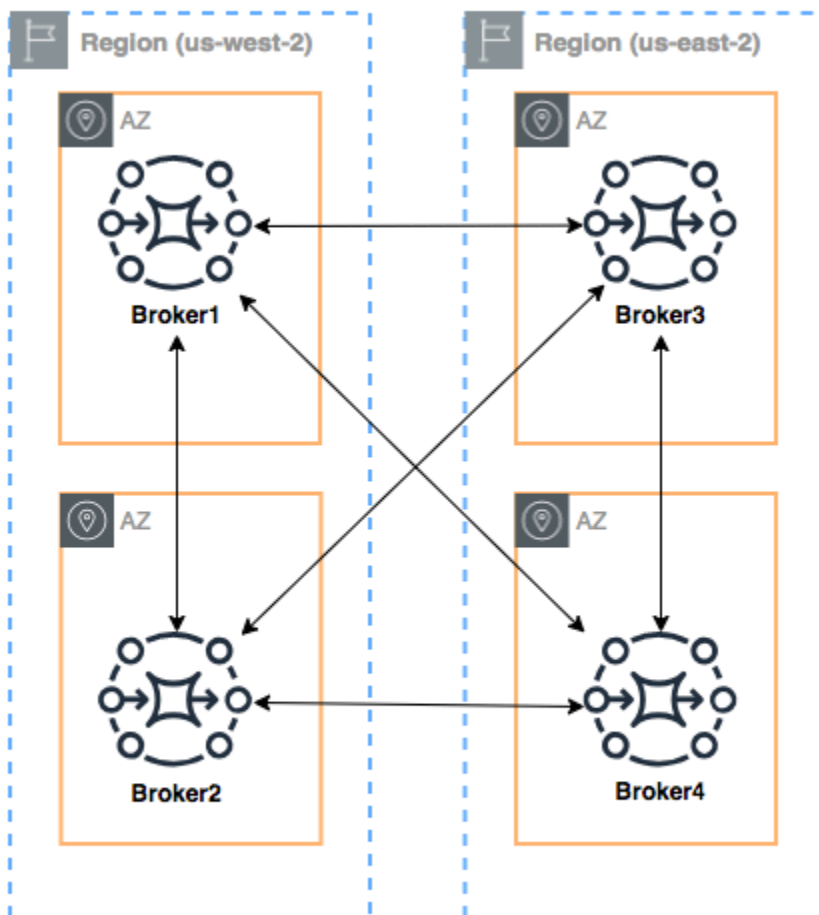
```
<networkConnectors>
  <networkConnector name="4_to_1" userName="myCommonUser" duplex="false"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)"/>
  <networkConnector name="4_to_2" userName="myCommonUser" duplex="false"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-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

Netzwerk-Connectors für Broker5:


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

Regionsübergreifend

Zur Konfiguration eines Netzwerk von Brokern, das sich über mehrere AWS-Regionen erstreckt, stellen Sie Broker in diesen Regionen bereit und konfigurieren Sie Netzwerk-Connectors zu den Endpunkten 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-987615k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
east-2.amazonaws.com:61617)"/>
  <networkConnector name="1_to_3" userName="myCommonUser" duplex="true"
    uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
  <networkConnector name="1_to_4" userName="myCommonUser" duplex="true"
    uri="static:(ssl://b-62a7fb31-d51c-466a-a873-905cd660b553-4.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

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

Dynamisches Failover mit Transport Connectors

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

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

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

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

Verfügbare Optionen:

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

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

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

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

Weitere Informationen zum Failover finden Sie unter [Brokerseitige Failover-Optionen](#) in der Active MQ-Dokumentation.

Konfigurationen für Amazon MQ für ActiveMQ Broker

Eine Konfiguration enthält alle Einstellungen für Ihren ActiveMQ-Broker im XML-Format (ähnlich 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.

Themen

- [Arbeiten mit Spring XML-Konfigurationsdateien](#)
- [Erstellen, Bearbeiten und Anwenden von ActiveMQ-Broker-Konfigurationen](#)
- [In Amazon MQ-Konfigurationen zulässige Elemente](#)
- [In Amazon MQ-Konfigurationen zulässige Elemente und ihre Attribute](#)
- [In Amazon MQ-Konfigurationen zulässige Elemente, untergeordnete Sammlungselemente und deren untergeordnete Attribute](#)

Arbeiten mit 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-5.17.3.xsd.zip](#)
- [amazon-mq-active-mq-5.17.2.xsd.zip](#)
- [amazon-mq-active-mq-5.17.1.xsd.zip](#)
- [amazon-mq-active-mq-5.16.5.xsd.zip](#)
- [amazon-mq-active-mq-5.16.4.xsd.zip](#)
- [amazon-mq-active-mq-5.16.3.xsd.zip](#)
- [amazon-mq-active-mq-5.16.2.xsd.zip](#)
- [amazon-mq-active-mq-5.15.15.xsd.zip](#)
- [amazon-mq-active-mq-5.15.14.xsd.zip](#)
- [amazon-mq-active-mq-5.15.13.xsd.zip](#)
- [amazon-mq-active-mq-5.15.12.xsd.zip](#)
- [amazon-mq-active-mq-5.15.10.xsd.zip](#)
- [amazon-mq-active-mq-5.15.9.xsd.zip](#)
- [amazon-mq-active-mq-5.15.8.xsd.zip](#)

- [amazon-mq-active-mq-5.15.6.xsd.zip](#)
- [amazon-mq-active-mq-5.15.0.xsd.zip](#)

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.

Erstellen, Bearbeiten und Anwenden von ActiveMQ-Broker-Konfigurationen

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. Sie können eine Konfiguration unmittelbar oder während eines Wartungsfensters übernehmen.

Weitere Informationen finden Sie hier:

- [Konfiguration](#)
- [Amazon-MQ-Broker-Konfigurationslebenszyklus](#)
- [Amazon MQ Broker Configuration Parameters](#)

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

Themen

- [Eine neue Konfiguration erstellen](#)
- [Erstellen einer neuen Konfigurationsversion](#)
- [Eine Konfigurationsrevision auf Ihren Broker anwenden](#)
- [Eine Konfigurationsüberarbeitung bearbeiten](#)

Eine neue 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.

Amazon MQ ×

Brokers

Configurations

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 unter [the section called "Versionsverwaltung."](#)

5. Wählen Sie Create configuration (Konfiguration erstellen).

Erstellen einer neuen Konfigurationsversion


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

Note

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

Auf der Seite **MyConfiguration** werden der Broker-Engine-Typ und die Version angezeigt, die von Ihrer neuen Konfigurationsversion verwendet werden (z. B. Apache ActiveMQ 5.15.8).

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.

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

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

```

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


```

3. Klicken Sie auf Bearbeiten der Konfiguration. Nehmen Sie Änderungen an der XML-Konfiguration vor.
4. Wählen Sie Save (Speichern).

Die Speichern der Revision wird angezeigt.

5. (Optional) Geben Sie A description of the changes in this revision ein.
6. Wählen Sie Save (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 unter [Amazon MQ Broker Configuration Parameters](#).

Das Vornehmen von Änderungen an einer Konfiguration nicht wenden 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](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).
Derzeit ist es nicht möglich, eine Konfiguration zu löschen.

Eine Konfigurationsrevision auf Ihren Broker anwenden

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

Amazon MQ ×

Brokers

Configurations

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

 Important

Ihr Broker ist offline, während er neu gestartet wird.


5. Wählen Sie Apply (Anwenden) aus.

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

Eine Konfigurationsüberarbeitung bearbeiten

1. Melden Sie sich bei der [Amazon-MQ-Konsole](#) an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker). Klicken Sie auf und danach auf Bearbeiten.
3. Wählen Sie auf der Seite **MyBroker** Edit (Bearbeiten).


- Wählen Sie auf der Seite Edit **MyBroker** im Bereich Configuration eine Configuration und eine Revision aus und wählen Sie Edit.

 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 Seite **MyBroker** werden der Broker-Engine-Typ und die Version angezeigt, die von der Konfiguration verwendet wird (z. B. Apache ActiveMQ 5.15.0).

- 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.

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

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

```

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

```

- Klicken Sie auf Bearbeiten der Konfiguration Nehmen Sie Änderungen an der XML-Konfiguration vor.
- Wählen Sie Save (Speichern).

Die Speichern der Revision wird angezeigt.

- (Optional) Geben Sie A description of the changes in this revision ein.

9. Wählen Sie Save (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 unter [Amazon MQ Broker Configuration Parameters](#).

Das Vornehmen von Änderungen an einer Konfiguration nichtwenden 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](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#). Derzeit ist es nicht möglich, eine Konfiguration zu löschen.

In Amazon 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 [\(Attribute\)](#) | [\(untergeordnete Sammlungselemente\)](#)

-Element

compositeTopic ([Attribute](#) | [untergeordnete Sammlungselemente](#))

constantPendingMessageLimitStrategy ([Attribute](#))

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](#))

-Element

prefetchRatePendingMessageLimitStrategy [\(Attribute\)](#)

priorityDispatchPolicy

priorityNetworkDispatchPolicy

queryBasedSubscriptionRecoveryPolicy [\(Attribute\)](#)

queue [\(Attribute\)](#)

redeliveryPlugin [\(Attribute | untergeordnete Sammlungselemente\)](#)

redeliveryPolicy [\(Attribute\)](#)

redeliveryPolicyMap [\(untergeordnete Sammlungselemente\)](#)

retainedMessageSubscriptionRecoveryPolicy [\(untergeordnete Sammlungs
elemente\)](#)

roundRobinDispatchPolicy

sharedDeadLetterStrategy [\(Attribute | untergeordnete Sammlungselemente\)](#)

simpleDispatchPolicy

simpleMessageGroupMapFactory

statisticsBrokerPlugin

storeCursor

storeDurableSubscriberCursor [\(Attribute\)](#)

strictOrderDispatchPolicy

tempDestinationAuthorizationEntry [\(Attribute\)](#)

tempQueue [\(Attribute\)](#)

tempTopic [\(Attribute\)](#)

-Element

timedSubscriptionRecoveryPolicy [\(Attribute\)](#)

timeStampingBrokerPlugin [\(Attribute\)](#)

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

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

-Element	Attribut
	dedicatedTaskRunner
	deleteAllMessagesOnStartup
	keepDurableSubsActive
	enableMessageExpirationOnActiveDurableSubs
	maxPurgedDestinationsPerSweep
	maxSchedulerRepeatAllowed
	monitorConnectionSplits
	networkConnectorStartAsync
	offlineDurableSubscriberTaskSchedule
	offlineDurableSubscriberTimeout
	persistenceThreadPriority
	persistent
	populateJMSXUserID
	producerSystemUsagePortion
	rejectDurableConsumers
	rollbackOnlyOnAsyncException
	schedulePeriodForDestinationPurge
	schedulerSupport

-Element	Attribut
	splitSystemUsageForProducersConsumers
	taskRunnerPriority
	timeBeforePurgeTempDestinations
	useAuthenticatedPrincipalForJMSXUserID
	useMirroredQueues
	useTempMirroredQueues
	useVirtualDestSubs
	useVirtualDestSubsOnCreation
	useVirtualTopics
cachedMessageGroupMapFactory	cacheSize
compositeQueue	concurrentSend
	copyMessage
	forwardOnly
	name
	sendWhenNotMatched
compositeTopic	concurrentSend
	copyMessage
	forwardOnly
	name


-Element	Attribut
	sendWhenNotMatched
BedingungNetworkBridgeFilterFactory	rateDuration
	rateLimit
	replayDelay
	replayWhenNoConsumers
	selectorAware
constantPendingMessageLimit Strategy	limit
discarding	deadLetterQueue
	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
discardingDLQBrokerPlugin	dropAll
	dropOnly
	dropTemporaryQueues

 Unterstützt in
Apache ActiveMQ 5.16.x

-Element	Attribut
	dropTemporaryTopics
	reportInterval
filteredDestination	queue
	selector
	topic
fixedCountSubscriptionRecoveryPolicy	maximumSize
fixedSizedSubscriptionRecoveryPolicy	maximumSize
	useSharedBuffer
forcePersistencyModeBrokerPlugin	persistenceFlag
individualDeadLetterStrategy	destinationPerDurableSubscriber
	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
	queuePrefix
	queueSuffix
	topicPrefix
	topicSuffix

-Element	Attribut
	useQueueForQueueMessages
	useQueueForTopicMessages
messageGroupHashBucketFactory	bucketCount
	cacheSize
mirroredQueue	copyMessage
	postfix
	prefix
oldestMessageEvictionStrategy	evictExpiredMessagesHighWatermark
oldestMessageWithLowestPriorityEvictionStrategy	evictExpiredMessagesHighWatermark
policyEntry	advisoryForConsumed
	advisoryForDelivery
	advisoryForDiscardingMessages
	advisoryForFastProducers
	advisoryForSlowConsumers
	advisoryWhenFull
	allConsumersExclusiveByDefault
	alwaysRetroactive
	blockedProducerWarningInterval
	consumersBeforeDispatchStarts

-Element	Attribut
	cursorMemoryHighWaterMark
	doOptimizeMessageStorage
	durableTopicPrefetch
	enableAudit
	expireMessagesPeriod
	gcInactiveDestinations
	gcWithNetworkConsumers
	inactiveTimeoutBeforeGC
	inactiveTimeoutBeforeGC
	includeBodyForAdvisory
	lazyDispatch
	maxAuditDepth
	maxBrowsePageSize
	maxDestinations
	maxExpirePageSize
	maxPageSize
	maxProducersToAudit
	maxQueueAuditDepth
	memoryLimit
	messageGroupMapFactoryType

-Element	Attribut
	minimumMessageSize
	optimizedDispatch
	optimizeMessageStoreInFlightLimit
	persistJMSRedelivered
	prioritizedMessages
	producerFlowControl
	queue
	queueBrowserPrefetch
	queuePrefetch
	reduceMemoryFootprint
	sendAdvisoryIfNoConsumers
	sendFailIfNoSpace
	sendFailIfNoSpaceAfterTimeout
	<div data-bbox="829 1325 1507 1499"> Unterstützt in Apache ActiveMQ 5.16.4 und höher</div>
	sendDuplicateFromStoreToDLQ
	storeUsageHighWaterMark
	strictOrderDispatch
	tempQueue

-Element	Attribut
	tempTopic
	timeBeforeDispatchStarts
	topic
	topicPrefetch
	useCache
	useConsumerPriority
usePrefetchExtension	
prefetchRatePendingMessageLimitStrategy	multiplier
queryBasedSubscriptionRecoveryPolicy	query
queue	DLQ
	physicalName
redeliveryPlugin	fallbackToDeadLetter
	sendToDlqIfMaxRetriesExceeded
redeliveryPolicy	backOffMultiplier
	collisionAvoidancePercent
	initialRedeliveryDelay
	maximumRedeliveries
	maximumRedeliveryDelay
	preDispatchCheck

-Element	Attribut
	queue redeliveryDelay tempQueue tempTopic topic useCollisionAvoidance useExponentialBackOff
sharedDeadLetterStrategy	enableAudit expiration maxAuditDepth maxProducersToAudit processExpired processNonPersistent
storeDurableSubscriberCursor	immediatePriorityDispatch useCache
tempDestinationAuthorizationEntry	admin queue read tempQueue tempTopic

-Element	Attribut
	topic
	write
tempQueue	DLQ
	physicalName
tempTopic	DLQ
	physicalName
timedSubscriptionRecoveryPolicy	zeroExpirationOverride
timeStampingBrokerPlugin	recoverDuration
	futureOnly
	processNetworkMessages
	tTlCeiling
topic	DLQ
	physicalName
transportConnector	<ul style="list-style-type: none"> name updateClusterClients rebalanceClusterClients updateClusterClientsOnRemove
uniquePropertyMessageEvictionStrategy	evictExpiredMessagesHighWatermark

-Element	Attribut
	propertyName
virtualTopic	concurrentSend
	local
	dropOnResourceLimit
	name
	postfix
	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

networkConnectionStartAsync

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.

Standard: `false`

Beispielkonfiguration

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

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

Es folgt eine detaillierte Auflistung der in Ihnen erhaltenen 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 Sammlungs element	Untergeordnetes Element
authorizationMap	authorizationEntries	authorizationEntry
	defaultEntry	tempDestinationAuthorizationEntry
		authorizationEntry
	tempDestinationAuthorizationEntry	tempDestinationAuthorizationEntry
authorizationPlugin	map	authorizationMap
broker	destinationInterceptors	mirroredQueue
	destinationPolicy	virtualDestinationInterceptor
		policyMap

-Element	Untergeordnetes Sammlungs element	Untergeordnetes Element
	destinations	queue
		tempQueue
		tempTopic
		topic
	networkConnectors	networkConnector
	persistenceAdapter	kahaDB
	plugins	authorizationPlugin
		discardingDLQBrokerPlugin
		forcePersistencyModeBrokerPlugin
		redeliveryPlugin
		statisticsBrokerPlugin
		timeStampingBrokerPlugin
	systemUsage	systemUsage
	transportConnector	name
		updateClusterClients
		rebalanceClusterClients

-Element	Untergeordnetes Sammlungs element	Untergeordnetes Element
		updateClusterClientsOnRemove
compositeQueue	forwardTo	queue tempQueue tempTopic topic filteredDestination
compositeTopic	forwardTo	queue tempQueue tempTopic topic filteredDestination
policyEntry	deadLetterStrategy	discarding individualDeadLetterStrategy sharedDeadLetterStrategy queue tempQueue tempTopic topic

-Element	Untergeordnetes Sammlungs element	Untergeordnetes Element
	dispatchPolicy	priorityDispatchPo licy
		priorityNetworkDis patchPolicy
		roundRobinDispatch Policy
		simpleDispatchPolicy
		strictOrderDispatc hPolicy
		clientIdFilterDisp atchPolicy
	messageEvictionStr ategy	oldestMessageEvict ionStrategy
		oldestMessageWithL owestPriorityEvict ionStrategy
		uniquePropertyMess ageEvictionStrategy
	messageGroupMapFac tory	cachedMessageGroup MapFactory
		messageGroupHashBu cketFactory
		simpleMessageGroup MapFactory

-Element	Untergeordnetes Sammlungs element	Untergeordnetes Element
	pendingDurableSubscriberPolicy	fileDurableSubscriberCursor storeDurableSubscriberCursor vmDurableCursor
	pendingMessageLimitStrategy	constantPendingMessageLimitStrategy prefetchRatePendingMessageLimitStrategy
	pendingQueuePolicy	fileQueueCursor storeCursor vmQueueCursor
	pendingSubscriberPolicy	fileCursor vmCursor
	slowConsumerStrategy	abortSlowAckConsumerStrategy abortSlowConsumerStrategy
	subscriptionRecoveryPolicy	fixedCountSubscriptionRecoveryPolicy fixedSizedSubscriptionRecoveryPolicy

-Element	Untergeordnetes Sammlungs element	Untergeordnetes Element
		lastImageSubscript ionRecoveryPolicy
		noSubscriptionReco veryPolicy
		queryBasedSubscrip tionRecoveryPolicy
		retainedMessageSub scriptionRecoveryP olicy
timedSubscriptionR ecoveryPolicy		
policyMap	defaultEntry	policyEntry
	policyEntries	policyEntry
redeliveryPlugin	redeliveryPolicyMap	redeliveryPolicyMap
redeliveryPolicyMap	defaultEntry	redeliveryPolicy
	redeliveryPolicyEn tries	redeliveryPolicy
retainedMessageSub scriptionRecoveryP olicy	wrapped	fixedCountSubscrip tionRecoveryPolicy
		fixedSizedSubscrip tionRecoveryPolicy
		lastImageSubscript ionRecoveryPolicy

-Element	Untergeordnetes Sammlungs element	Untergeordnetes Element
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
		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 `dieactivemq-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.

Standard: `null`

Beispielkonfiguration

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

networkConnector

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

Themen

- [Attribute](#)
- [Beispielkonfigurationen](#)

Attribute

conduitSubscriptions

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

Note

Durch Festlegen von `conduitSubscriptions` auf `true` können Sie den redundanten Netzwerkverkehr reduzieren. Die Verwendung dieses Attributs kann jedoch Auswirkungen auf den 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.

Standard: `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>`.

Standard: `false`

Name

Der Name der Brücke im Netzwerk von Brokern.

Standard: `bridge`

`uri`

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

Standard: `null`

`username`

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

Standard: `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 Netzwerk-Connectors lautet `connector_1_to_2`, der Benutzername, der den Brokern gemeinsam ist, ist `myCommonUser`, die Verbindung lautet `duplex` und dem OpenWire-Endpunkt-URI wird `static`: vorangestellt, was auf eine 1:1-Verbindung zwischen den Brokern hinweist.

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

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 Netzwerk-Connectors lautet `connector_1_to_2`, der gemeinsame Benutzername der Broker

ist `myCommonUser`, die Verbindung hat den Typ `duplex` und der Liste der OpenWire-Endpunkt-URIs (Werte sind durch Kommas voneinander getrennt) wird `masterslave:` vorangestellt, um eine Failover-Verbindung zwischen den Brokern anzugeben. 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-3i45j6k17819-1.mq.us-
east-2.amazonaws.com:61617,
    ssl://b-987615k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Note

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

Note

Diese XML-Konfiguration erlaubt keine Leerzeichen.

kahaDB

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


Attribute

`concurrentStoreAndDispatchQueues`

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

Standard: `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 Herunterfahrenvorgang beschleunigt wird. Die erhöhte Geschwindigkeit ist in Fällen mit großen Datenbanken oder Scheduler-Datenbanken nützlich.


Standard: `true`

journalDiskSyncInterval

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


Standard: `1000`

journalDiskSyncStrategy

 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](#).

Standard: `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](#).

Standard: `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 nachzuerfolgen, 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

percentOfJvmHeap

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

Standard: 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.

Standard: `false`

sendFailIfNoSpaceAfterTimeout

Standard: `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>
```

Verwalten von Amazon MQ für ActiveMQ Engine-Versionen

Apache ActiveMQ organisiert Versionsnummern gemäß der semantischen Versionspezifikation als `X.Y.Z`. In Amazon MQ für ActiveMQ Implementierungen, bezeichnet `X.Y` die Hauptversion und `Z` stellt die Nebenversionsnummer dar. Amazon MQ betrachtet eine Versionsänderung als Hauptversionsänderung, wenn sich die Hauptversionsnummern ändern. Beispielsweise wird ein Upgrade von Version 5.15 auf 5.16 als Upgrade einer Hauptversion betrachtet. Eine Versionsänderung gilt als unwesentlich, wenn sich nur die Unterversionsnummer ändert.

Beispielsweise wird ein Upgrade von Version 5.15.14 auf 5.15.15 als Upgrade einer Hauptversion betrachtet.

Amazon MQ für ActiveMQ unterstützt derzeit die folgenden Engine-Versionen von Apache ActiveMQ.

Hauptversionen	Nebenversionen
ActiveMQ 5.17	<ul style="list-style-type: none"> • 5.17.6 (empfohlen)
ActiveMQ 5.16	<ul style="list-style-type: none"> • 5.16.7
ActiveMQ 5.16	<ul style="list-style-type: none"> • 5.15.16

Die folgenden Nebenversionen sind veraltet.

Veraltete Nebenversionen

Hauptversionen	Nebenversionen
ActiveMQ 5.17	<ul style="list-style-type: none"> • 5.17.3 • 5.17.2 • 5.17.1
ActiveMQ 5.16	<ul style="list-style-type: none"> • 5.16,5 • 5.16.4 • 5.16.3 • 5.16.2
ActiveMQ 5.16	<ul style="list-style-type: none"> • 5.15.15 • 5.15.14 • 5.15.13 • 5.15.12 • 5.15.10 • 5.15.9 • 5.15.8 • 5.15.6

Hauptversionen

Nebenversionen

- 5.15.0

Wenn Sie einen neuen Amazon MQ für ActiveMQ Broker erstellen, können Sie jede unterstützte ActiveMQ Engine-Version angeben. Wenn Sie AWS Management Console verwenden, um einen Broker zu erstellen, verwendet Amazon MQ automatisch die neueste Engine-Versionsnummer. Wenn Sie das AWS CLI oder die Amazon MQ-API verwenden, um einen Broker zu erstellen, ist die Engine-Versionsnummer erforderlich. Wenn Sie keine Versionsnummer angeben, führt der Vorgang zu einer Ausnahme. Weitere Informationen hierzu finden Sie unter [create-broker](#) in der AWS CLI-Befehlsreferenz und [CreateBroker](#) in der Amazon-MQ-REST-API-Referenz.

Themen

- [Aktualisierungen von Haupt- und Nebenversionen](#)
- [Unterstützte Engine-Versionen auflisten](#)

Aktualisierungen von Haupt- und Nebenversionen

Mit Amazon MQ können Sie steuern, wann Ihre Broker auf neue Versionen aktualisiert werden. Wenn [Automatischer Unterversion-Upgrade](#) aktiviert ist, aktualisiert Amazon MQ Ihre Broker-Engine automatisch auf neue Nebenversionen, sobald sie von Amazon MQ veröffentlicht und unterstützt werden.

Um ein Hauptversions-Upgrade durchzuführen, müssen Sie die Modulversionsnummer Ihres Brokers manuell aktualisieren. Neben- und Hauptversions-Upgrades finden gleichzeitig mit anderen Broker-Patching-Vorgängen während Ihres geplanten [Wartungsfensters](#) statt. Wenn Sie sich gegen ein automatisches Upgrade der Nebenversion entscheiden, können Sie Ihren Broker manuell auf eine neue unterstützte Nebenversion aktualisieren, indem Sie das gleiche Verfahren wie bei einem Hauptupgrade anwenden.

Weitere Informationen zum Aktualisieren Ihrer Broker-Voreinstellungen zum Aktivieren oder Deaktivieren von Nebenversions-Upgrades und zum manuellen Aktualisieren des Brokers finden Sie unter [the section called "Upgrade der Engine-Version"](#).

Unterstützte Engine-Versionen auflisten

Sie können alle unterstützten Neben- und Hauptmodularversionen auflisten, indem Sie den [describe-broker-instance-options](#) AWS CLI-Befehl verwenden.

```
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 Instance-Typ `mq.m5.large` zu filtern, ersetzen Sie *Engine-Typ* durch `ACTIVEMQ` und *Instance-Typ* durch `mq.m5.large`.

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

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

- Mit dem OpenWire Java-Beispielcode wird eine Verbindung zu einem Broker hergestellt, eine Warteschlange erstellt und eine Nachricht gesendet und empfangen. 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

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Wählen Sie in der Broker-Liste den Namen Ihres Brokers (z. B. MyBroker) enthalten.
3. Notieren Sie auf der Seite *MyBroker* im Abschnitt Verbindungen die Adressen und Ports der Webkonsolen-URL des Brokers und die Protokolle auf Leitungsebene.

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>5.15.8</version>
  </dependency>
  <dependency>
    <groupId>org.apache.activemq</groupId>
    <artifactId>activemq-pool</artifactId>
```

```
<version>5.15.8</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.

AmazonMQExample.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 JMSEException {
        final ActiveMQConnectionFactory connectionFactory =
            createActiveMQConnectionFactory();
        final PooledConnectionFactory pooledConnectionFactory =
            createPooledConnectionFactory(connectionFactory);

        sendMessage(pooledConnectionFactory);
    }
}
```

```
        receiveMessage(connectionFactory);

        pooledConnectionFactory.stop();
    }

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

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

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

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

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

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

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

    private static void
    receiveMessage(ActiveMQConnectionFactory connectionFactory) throws JMSEException
    {
```

```
// Establish a connection for the consumer.
// Note: Consumers should not use PooledConnectionFactory.
final Connection consumerConnection = connectionFactory.createConnection();
consumerConnection.start();

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

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

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

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

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

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

private static PooledConnectionFactory
createPooledConnectionFactory(ActiveMQConnectionFactory connectionFactory) {
    // Create a pooled connection factory.
    final PooledConnectionFactory pooledConnectionFactory =
        new PooledConnectionFactory();
    pooledConnectionFactory.setConnectionFactory(connectionFactory);
    pooledConnectionFactory.setMaxConnections(10);
    return pooledConnectionFactory;
}

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



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

MQTT

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

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

public class AmazonMQExampleMqtt implements MqttCallback {

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

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

    private void run() throws MqttException, InterruptedException {
```

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

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

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

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

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

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

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

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

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

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

@Override
```

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

STOMP+WSS

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

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

import java.lang.reflect.Type;

public class AmazonMQExampleStompWss {

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

    public static void main(String[] args) throws Exception {
```

```
final AmazonMQExampleStompWss example = new AmazonMQExampleStompWss();

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

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

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

    final WebSocketHttpHeaders headers = new WebSocketHttpHeaders();

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

    final StompSessionHandler sessionHandler = new MySessionHandler();

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

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

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

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

```
}  
  
private static class MyFrameHandler implements StompFrameHandler {  
    public Type getPayloadType(final StompHeaders headers) {  
        return String.class;  
    }  
  
    public void handleFrame(final StompHeaders stompHeaders,  
                            final Object message) {  
        System.out.print("Received message from topic: " + message);  
    }  
}  
}
```

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 ActiveMQ-Brokers](#)
- [Erstellen und Konfigurieren eines Amazon MQ-Netzwerks von Brokern](#)
- [Verbinden einer Java-Anwendung mit Ihrem Amazon MQ-Broker](#)
- [Integration von ActiveMQ-Brokern in LDAP](#)
- [Erstellen und Verwalten von ActiveMQ-Broker-Benutzern](#)

Erstellen und Konfigurieren eines ActiveMQ-Brokers

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Die kombinierte Beschreibung der Broker-Instanceclass(m5,t3) undsize(large,micro) ist einBroker-Instance-Typ(zum Beispielmq.m5.large). Weitere Informationen finden Sie unter [Broker](#).


Die erste und häufigste Amazon-MQ-Aufgabe ist das Erstellen eines Brokers. Das folgende Beispiel zeigt, wie Sie einen Broker mithilfe der AWS Management Console erstellen und konfigurieren können.

Themen

- [Schritt 1: Konfigurieren grundlegender Broker-Einstellungen](#)
- [Schritt 2: \(Optional\) Konfigurieren zusätzlicher Broker-Einstellungen](#)
- [Schritt 3: Abschließen der Broker-Erstellung](#)
- [Tutorial: Bearbeiten der Broker-Engine-Version, des Instance-Typs, von CloudWatch Logs und von Wartungseinstellungen](#)


Schritt 1: Konfigurieren grundlegender Broker-Einstellungen

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 [Broker Architecture](#).
 - 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 [Amazon MQ Single-Instance-Broker](#).
 - 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 [Aktive/Standby Amazon MQ-Broker für hohe Verfügbarkeit](#).
 - Weitere Informationen zu den Beispiel-Blueprints für ein Netzwerk von Brokern finden Sie unter [Beispiel-Vorlagen](#).
 - 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. Brokernamen sind für andere AWS-Services zugänglich, einschließlich CloudWatch Logs. Brokernamen sind nicht für private oder sensible Daten gedacht.

- b. Wählen Sie den Broker-Instance-Typ (z. B. mq.m5.large). Weitere Informationen finden Sie unter [Broker instance types](#).
5. Geben Sie im Abschnitt Zugriff auf ActiveMQ-Webkonsole einen Benutzernamen und ein Passwort an. Die folgenden Einschränkungen gelten in Bezug auf Benutzernamen und Passwörter des Brokers:
 - Ihr Benutzername darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten.
 - Ihr 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-Services zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

Schritt 2: (Optional) Konfigurieren zusätzlicher Broker-Einstellungen

Important

- Subnetz(e) Für den Single-Instance-Broker ist ein Subnetz erforderlich (z. B. das Standard-Subnetz). Für einen aktiv/standby-Broker sind zwei Subnetze erforderlich.
- Sicherheitsgruppe(n) Sowohl für die Single-Instance-Broker, als auch für die Aktive/Standby-Broker mit hoher Verfügbarkeit ist mindestens eine Sicherheitsgruppe erforderlich (z. B. die Standard-Sicherheitsgruppe).
- VPC - Die Subnetz(e) und Sicherheitsgruppe(n) eines Broker müssen sich in derselben VPC befinden. EC2-Classic-Ressourcen werden nicht unterstützt. Amazon MQ unterstützt nur Standard-VPC -Tenancy und keine dedizierte VPC-Tenancy.
- Verschlüsselung – Wählen Sie den Kundenmasterschlüssel zum Verschlüsseln der Daten aus. Siehe [Verschlüsselung im Ruhezustand](#).
- Öffentlicher Zugriff - Wenn Sie den öffentlichen Zugriff deaktivieren, ist der Broker nur innerhalb Ihrer VPC erreichbar. Weitere Informationen finden Sie unter [Broker ohne öffentlichen Zugriff bevorzugen](#) und [Zugreifen auf die Webkonsole eines Brokers ohne öffentliche Zugänglichkeit](#).


1. Erweitern Sie den Abschnitt Additional settings (Erweiterte Einstellungen).
2. Wählen Sie im Abschnitt Configuration (Konfiguration) die Option Create a new configuration with default values (Neue Konfiguration mit Standardwerten erstellen) oder Select an existing configuration (Vorhandene Konfiguration auswählen) aus. Weitere Informationen finden Sie unter [Konfiguration](#) und [Amazon MQ Broker Configuration Parameters](#).
3. Wählen Sie im Abschnitt Protokolle, ob die Allgemeinen und die Audit-Protokolle in Amazon CloudWatch Logs veröffentlicht werden sollen. Weitere Informationen finden Sie unter [Configuring Amazon MQ to publish logs to Amazon CloudWatch Logs](#).

Important

Wenn Sie die [-Berechtigung nicht Ihrem CreateLogGroup Amazon MQ-Benutzer](#) hinzufügen, bevor der Benutzer den Broker erstellt oder neu startet, wird die Protokollgruppe nicht von Amazon MQ erstellt.

Wenn Sie keine [ressourcenbasierte Richtlinie für Amazon MQ konfigurieren](#), kann der Broker die Protokolle nicht in CloudWatch Logs veröffentlichen.

4. Konfigurieren Sie im Abschnitt Netzwerk und Sicherheit die Konnektivität Ihres Brokers:
 - a. Führen Sie eine der folgenden Aktionen aus:
 - Wählen Sie Standard-VPC, -Subnetz(e) und -Sicherheitsgruppe(n) verwenden.
 - Wählen Sie Vorhandene(n) VPC, Subnetz(e) und Sicherheitsgruppe(n) wählen.
 1. Wenn Sie diese Option auswählen, können Sie eine neue Virtual Private Cloud (VPC) auf der Amazon VPC-Konsole erstellen, eine vorhandene VPC oder die Standard-VPC wählen. Weitere Informationen finden Sie unter [Was ist Amazon VPC?](#) im Amazon VPC-Benutzerhandbuch.
 2. Nachdem Sie eine VPC erstellt haben, können Sie (ein) neue(s) Subnetz(e) auf der Amazon VPC-Konsole erstellen oder vorhandene auswählen. Weitere Informationen finden Sie unter [VPCs und Subnetze](#) im Amazon-VPC-Benutzerhandbuch.
 3. Nach der Erstellung oder der Auswahl der Subnetze können Sie die Sicherheitsgruppe(n) auswählen.
 - b. Wählen Sie den Kundenmasterschlüssel (Customer Master Key, CMK), der zum Verschlüsseln der Daten verwendet werden soll. Siehe [Verschlüsselung im Ruhezustand](#).
 - c. Wählen Sie die Öffentliche Zugänglichkeit für Ihren Broker.
5. Konfigurieren Sie im Abschnitt Maintenance (Wartung) den Wartungszeitplan für Ihren Broker:
 - a. Um Upgrades auf neue Versionen Ihres Brokers vorzunehmen, wenn sie von Apache veröffentlicht werden, wählen Sie Enable automatic minor version upgrades. Automatische Upgrades werden während der-Wartungsfensterdefiniert durch den Wochentag, die Tageszeit (im 24-Stunden-Format) und die Zeitzone (standardmäßig UTC).

 Note

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

- b. Führen Sie eine der folgenden Aktionen aus:

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

Schritt 3: Abschließen der Broker-Erstellung

1. Wählen Sie Deploy (Bereitstellen) aus.

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.

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

2. Klicken Sie auf **MyBroker**.

Auf der Seite **MyBroker**, in dem Abschnitt Connect (Verbinden), notieren Sie die URL der [ActiveMQ-Webkonsole](#) für Ihren Broker, zum Beispiel:

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

Beachten Sie auch die [Wire-Level-Protokoll-Endpunkte](#). Das folgende Beispiel zeigt einen Knotenendpunkt.

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

Note

Für einen Aktiv/Standby-Broker stellt Amazon MQ zwei URLs der ActiveMQ Web Console bereit, jedoch 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 [Broker Architecture](#)).

Für Drahtebene Protokollendpunkte können Sie zulassen, dass Ihre Anwendung eine Verbindung zu einem beliebigen Endpunkt herstellen kann, indem Sie die [Failover-Transport](#) verwenden.

Tutorial: Bearbeiten der Broker-Engine-Version, des Instance-Typs, von CloudWatch Logs und von Wartungseinstellungen

Zusätzlich zum [Bearbeiten von Broker-Konfigurationen und Verwalten von Konfigurationsversionen](#), können Sie Einstellungen speziell für den Broker konfigurieren.

Note

Alle Einstellungen mit Ausnahme der für die automatischen Upgrades auf Unterversionen erfordern das Planen von Änderungen. Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).

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

ActiveMQ-Broker-Optionen bearbeiten

1. Melden Sie sich bei der [Amazon-MQ-Konsole](#) an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und danach Edit (Bearbeiten).
3. Wählen Sie auf der Seite **MyBroker** bearbeiten im Abschnitt Spezifikationen eine Broker-Engine-Version oder einen Broker-Instance-Typ.
4. Wählen Sie im Abschnitt Configuration (Konfiguration) die Konfiguration und die Version für Ihren Broker. Weitere Informationen finden Sie unter [Creating and applying broker configurations](#).

5. Wählen Sie im Abschnitt Sicherheit und Netzwerk eine Gruppe aus der Dropdown-Liste Sicherheitsgruppe(n) aus, oder wählen Sie Neue Sicherheitsgruppe erstellen, um die Amazon VPC-Konsole zu öffnen.
6. Wählen Sie im Abschnitt CloudWatch Logs, ob die Allgemeinen und die Audit-Protokolle in Amazon CloudWatch Logs veröffentlicht werden sollen.

Weitere Informationen zum Konfigurieren von CloudWatch-Protokollen für ActiveMQ-Broker finden Sie unter [Configuring Amazon MQ to publish logs to Amazon CloudWatch Logs](#).


 **Important**

Wenn Sie nicht die [CreateLogGroup-Berechtigung Ihrem Amazon-MQ-Benutzer hinzufügen](#), bevor der Benutzer den Broker erstellt oder neu startet, wird die Protokollgruppe nicht von Amazon MQ erstellt.

Wenn Sie keine [ressourcenbasierte Richtlinie für Amazon MQ konfigurieren](#), kann der Broker die Protokolle nicht in CloudWatch Logs veröffentlichen.

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

Um Upgrades auf neue Versionen Ihres Brokers vorzunehmen, wenn sie von AWS veröffentlicht werden, wählen Sie Automatische Upgrades von Unterversionen aktivieren. Automatische Upgrades werden während der-Wartungsfensterdefiniert durch den Wochentag, die Tageszeit (im 24-Stunden-Format) und die Zeitzone (standardmäßig UTC).

 **Note**

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

8. 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.

Erstellen und Konfigurieren eines Amazon MQ-Netzwerks von Brokern

Ein -Netzwerk von Brokern besteht aus mehreren gleichzeitig aktiven [Single-Instance-Brokern](#) oder [aktiven/Standby-Brokern](#). Sie können Netzwerke von Brokern in einer Vielzahl von [Topologien](#) konfigurieren (zum Beispiel concentrator, hub-and-spokes, tree, oder mesh) abhängig von den Anforderungen Ihrer Anwendung, wie Hochverfügbarkeit und Skalierbarkeit. Zum Beispiel kann ein [Hub-und-Spoke](#)-Netzwerk von Brokern die Ausfallsicherheit erhöhen und Nachrichten erhalten, wenn ein Broker nicht erreichbar ist. Ein Netzwerk von Brokern mit einem [Konzentrator](#) Topologie kann Nachrichten von einer größeren Anzahl von Brokern sammeln, die eingehende Nachrichten akzeptieren, und sie auf zentralere Broker konzentrieren, um die Belastung vieler eingehender Nachrichten besser zu bewältigen. In diesem Tutorial erfahren Sie, wie Sie ein Zwei-Broker-Netzwerk von Brokern mit einer Source and Sink-Topologie erstellen.

Eine konzeptionelle Übersicht und detaillierte Konfigurationsinformationen finden Sie im Folgenden:

- [Amazon MQ Netzwerk von Brokern](#)
- [Korrekte Konfiguration Ihres Netzwerk von Brokern](#)
- [networkConnector](#)
- [networkConnectionStartAsync](#)
- [Netzwerke von Brokern](#) in der ActiveMQ-Dokumentation

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

Themen

- [Voraussetzungen](#)
- [Schritt 1: Zulassen von Datenverkehr zwischen Brokern](#)
- [Schritt 2: Konfigurieren von Netzwerk-Connectors für Ihren Broker](#)
- [Nächste Schritte](#)

Voraussetzungen

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

- Zwei oder mehr gleichzeitig aktive Broker (in diesem Tutorial `MyBroker1` und `MyBroker2` genannt). Weitere Informationen zum Erstellen von Brokern finden Sie unter [Creating and configuring a broker](#).
- Die beiden Broker müssen sich im gleichen VPC oder in per Peering verbundenen VPCs befinden. Weitere Informationen zu VPCs finden Sie unter [Was ist Amazon VPC?](#) im Amazon VPC User Guide 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 [Erstellen und Verwalten von ActiveMQ-Broker-Benutzern](#).


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 in der [Amazon MQ-Konsole](#) auf der Seite MyBroker2 im Abschnitt Details unter Security and network (Sicherheit und Netzwerk) den Namen Ihrer Sicherheitsgruppe oder  aus.


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 Edit inbound rules (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 bei Port Range (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 Edit` (Bearbeiten) aus.
 - b. Wählen Sie auf der Seite `Edit MyBroker1` (`MyBroker1` bearbeiten) im Abschnitt `Configuration` (Konfiguration) `View` (Anzeigen) 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 Präfix `static:` 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-Endpoint `uri` für beide Broker mit den folgenden Abfrageparametern ? `randomize=false&maxReconnectAttempts=0`. Beispiele:

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

 Note

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

- Wählen Sie Save (Speichern).
 - Geben Sie im Dialogfeld Save revision (Revision speichern) Add network of brokers connector for MyBroker2 ein.
 - Wählen Sie Save (Speichern) aus, um die neue Revision der Konfiguration zu speichern.
- Bearbeiten Sie MyBroker1, um die neueste Revision der Konfiguration so einzustellen, dass sie sofort wirksam wird.
 - Wählen Sie auf der Seite MyBroker1 Edit (Bearbeiten) aus.
 - Wählen Sie auf der Seite Edit MyBroker1 (MyBroker1 bearbeiten) im Abschnitt Configuration (Konfiguration) Schedule Modifications (Änderungen planen) aus.
 - Wählen Sie im Abschnitt Schedule broker modifications (Broker-Änderungen planen) aus, dass Änderungen Immediately (Sofort) wirksam werden sollen.
 - 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 [e eingehende Verbindungen](#) von Ihrem lokalen Computer für den Broker MyBroker1 auf Port 8162 (für die ActiveMQ Web Console) und Port 61617 (für den OpenWire-Endpunkt) aktivieren.

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-3i45j6k17819-1.mq.us-east-2.amazonaws.com:61617" \  
--user commonUser \  
--password myPassword456 \  
--destination queue://MyQueue
```

Der Verbraucher verbindet sich mit dem OpenWire-Endpunkt von MyBroker2 und beginnt, Nachrichten aus der Warteschlange MyQueue zu konsumieren.

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-987615k4-32ji-109h-8gfe-7d65c4b132a1-1.mq.us-east-2.amazonaws.com:61617" \
--user commonUser \
--password myPassword456 \
--destination queue://MyQueue \
--persistent true \
--messageSize 1000 \
--messageCount 10000
```

Der Produzent verbindet sich mit dem OpenWire-Endpunkt von MyBroker1 und beginnt, persistente Nachrichten an die Warteschlange MyQueue zu produzieren.

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 unter [Working 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 aktivieren. 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 in der Broker-Liste den Namen Ihres Brokers (z. B. MyBroker) enthalten.
3. Notieren Sie auf der Seite **MyBroker** im Abschnitt Verbindungen die Adressen und Ports der Webkonsolen-URL des Brokers und die Protokolle auf Leitungsebene.
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>5.15.8</version>
</dependency>
<dependency>
  <groupId>org.apache.activemq</groupId>
  <artifactId>activemq-pool</artifactId>
  <version>5.15.8</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:

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 Aktiv/Standby-Broker stellt Amazon MQ zwei URLs der ActiveMQ Web Console bereit, jedoch 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 [Broker Architecture](#).

Für Drahtebene Protokollendpunkte können Sie zulassen, dass Ihre Anwendung eine Verbindung zu einem beliebigen Endpunkt herstellen kann, indem Sie die [Failover-Transport](#) verwenden.

```
// Create a connection factory.
```

```
final ActiveMQConnectionFactory connectionFactory = new
    ActiveMQConnectionFactory(wireLevelEndpoint);

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

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

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

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

Note

Nachrichtenproduzenten sollten immer die `PooledConnectionFactory`-Klasse. Weitere Informationen finden Sie unter [Verwenden Sie immer Verbindungspools](#).

- 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);
```

- 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:

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-Dienste (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


Die LDAP-Integration wird für RabbitMQ-Broker nicht unterstützt.

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

- [AMQP](#)
- [MQTT](#)
- MQTT über [WebSocket](#)
- [OpenWire](#)
- [STOMP](#)
- STOMP über WebSocket

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

Um ActiveMQ-Benutzer und -Gruppen für die Arbeit mit Warteschlangen und Themen zu autorisieren, müssen Sie [die Konfiguration Ihres Brokers bearbeiten](#). Amazon MQ verwendet zum Einschränken des Lese- und Schreibzugriffs auf Ziele das [Simple Authentication Plugin](#) von ActiveMQ. Weitere Informationen und Beispiele finden Sie unter [Immer eine Autorisierungszuordnung konfigurieren](#) und [authorizationEntry](#).

 Note

Derzeit unterstützt Amazon MQ keine Clientzertifikat-Authentifizierung.

Themen

- [Integrieren von LDAP mit ActiveMQ](#)
- [Voraussetzungen](#)
- [Erste Schritte mit LDAP](#)
- [Funktionsweise der LDAP-Integration](#)

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 abrufen.

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 beispielhafte `ldif`-Datei, die diese Struktur deutlich zeigt, finden Sie unter Importieren 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 Domännennamen 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.

Authentication and Authorization

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

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

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

Fully qualified domain name

example.com

optional second server name

Service account username

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

myserviceaccount

Service account password

The password for the service account provided above.

Maximum of 128 characters

Show

LDAP login configuration

Your server configuration to search and authenticate users.

User Base

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

ou=user, dc=example, dc=com

User Search Matching

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

(uid=0)

Role Base

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

ou=user, dc=example, dc=com

Role Search Matching

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

(uid=0)

► Optional settings

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

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

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

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

Authentication and Authorization

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

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

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

Fully qualified domain name

example.com

optional second server name

Service account username

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

myserviceaccount

Service account password

The password for the service account provided above.

Maximum of 128 characters

Show

LDAP login configuration

Your server configuration to search and authenticate users.

User Base

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

ou=user, dc=example, dc=com

User Search Matching

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

(uid=0)

Role Base

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

ou=user, dc=example, dc=com

Role Search Matching

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

(uid=0)

► Optional settings

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.

Role Search Matching

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

▼ Optional settings**User Role Name**

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

Role Name

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

 User Search Subtree

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

 Role Search Subtree

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

Funktionsweise der LDAP-Integration

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

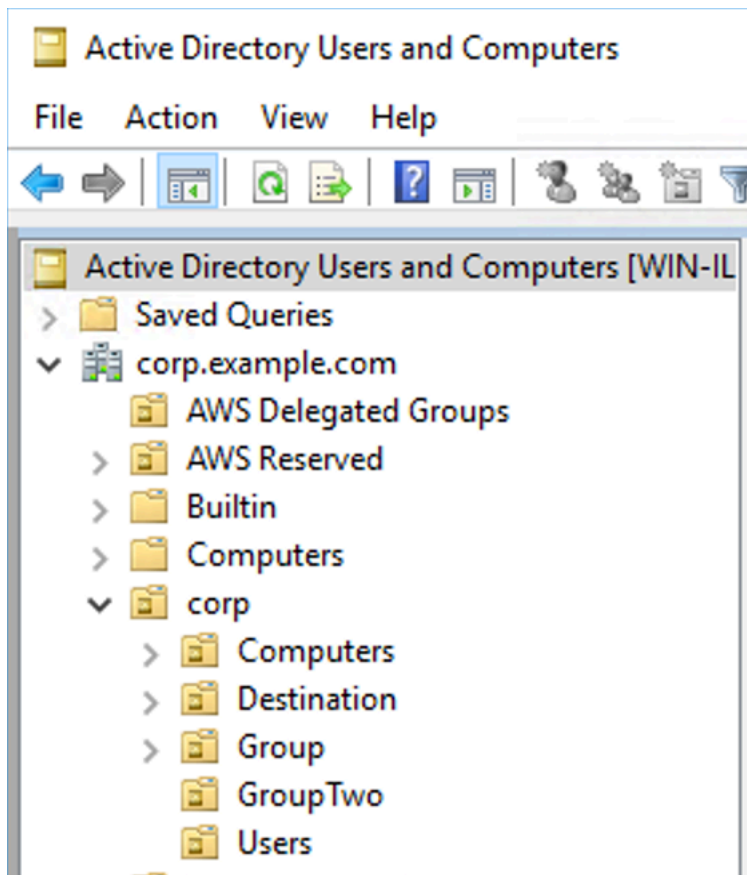
Authentifizierung

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

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

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

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



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

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

Autorisierung

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

Note

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

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

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

Important

Um zu verhindern, dass vertrauliche Informationen direkt in der Konfigurationsdatei des Brokers platziert werden, blockiert Amazon MQ die folgenden Attribute in `cachedLdapAuthorizationMap`:

- `connectionURL`
- `connectionUsername`
- `connectionPassword`

Wenn Sie einen Broker erstellen, ersetzt Amazon MQ die Werte, die Sie über die AWS Management Console, oder in der `ldapServerMetadata`-Eigenschaft Ihrer API-Anfrage für die obigen Attribute angeben.

Das folgende Beispiel illustriert die Verwendung von Verschiebungen.

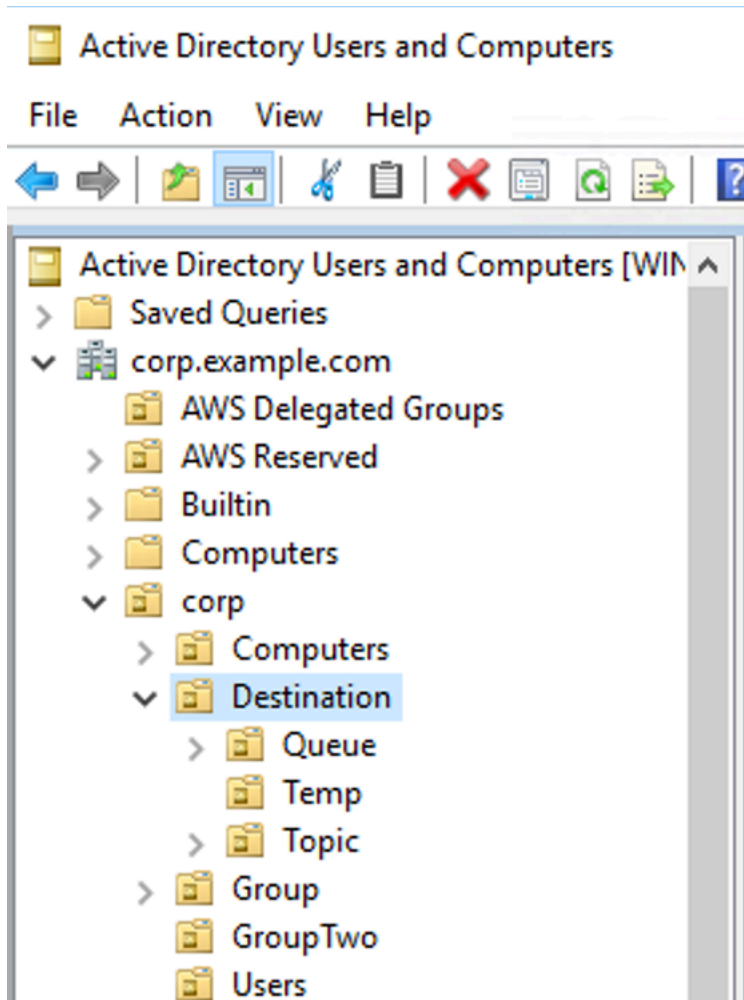
```
<authorizationPlugin>
  <map>
```

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

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

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

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



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

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

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

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

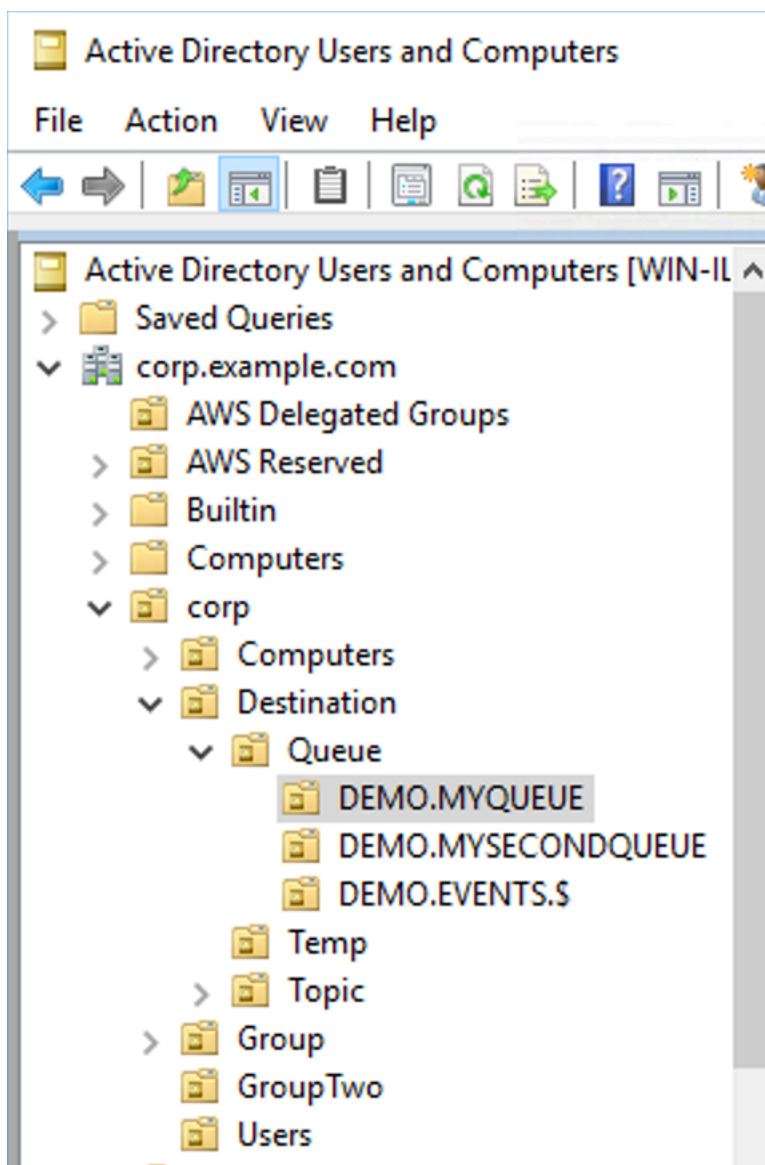
Note

Die DEMO.EVENTS.\$-Organisationseinheit befindet sich innerhalb der Queue-Organisationseinheit.

Weitere Informationen zu Platzhaltern in ActiveMQ finden Sie unter [Platzhalter](#)

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

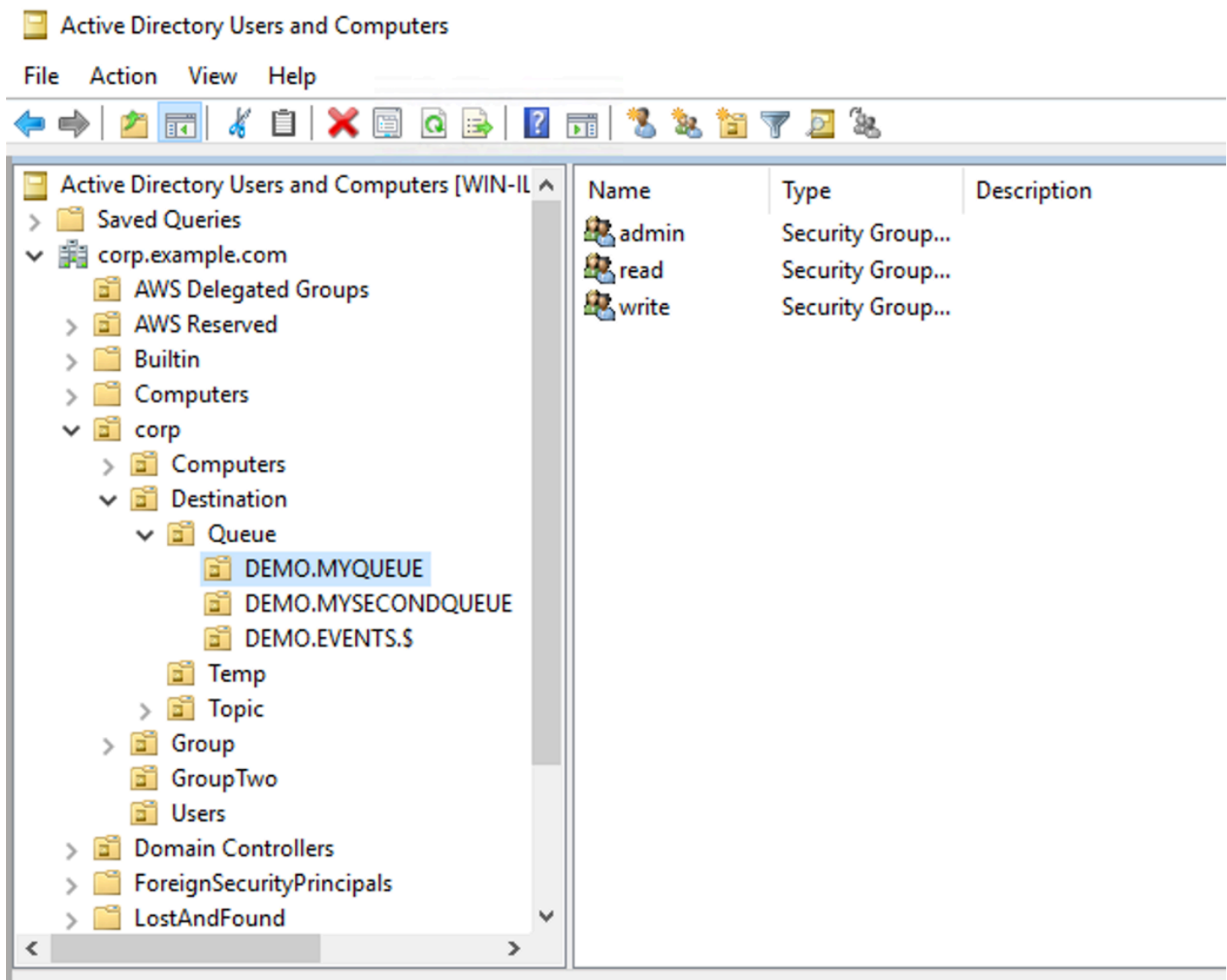
```
OU=DEMO.MYQUEUE,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



Sicherheitsgruppen

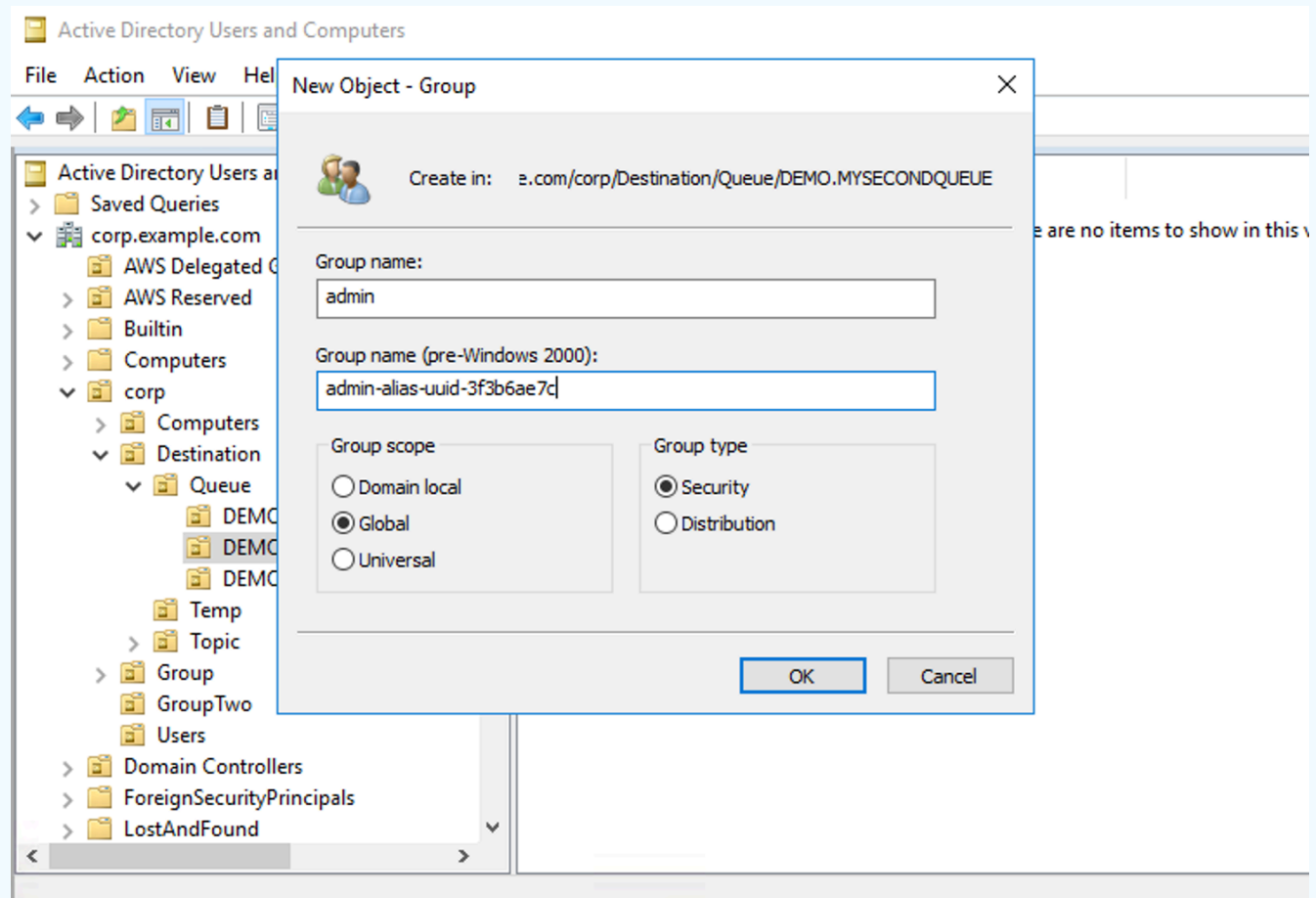
Innerhalb jeder Organisationseinheit, die ein Ziel oder einen Platzhalter darstellt, müssen Sie drei Sicherheitsgruppen erstellen. Wie bei allen Berechtigungen in ActiveMQ handelt es sich hierbei um Lese-/Schreib-/Administratorberechtigungen. Weitere Informationen zu den Funktionen der einzelnen Berechtigungen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

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



Note

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



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

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

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

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

Erstellen und Verwalten von ActiveMQ-Broker-Benutzern

Ein ActiveMQ-Benutzer ist 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 die [ActiveMQ-Webkonsole](#) zuzugreifen.

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

Note

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

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

Themen

- [Um einen neuen Benutzer zu erstellen](#)
- [So bearbeiten Sie einen vorhandenen Benutzer](#)
- [So löschen Sie einen vorhandenen Benutzer](#)

Um einen neuen Benutzer zu erstellen

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Wählen Sie in der Broker-Liste den Namen Ihres Brokers (z. B. MyBroker). Klicken Sie auf und danach auf `View details` (Details anzeigen).

Auf der Seite **MyBroker** werden im Abschnitt Benutzer alle Benutzer dieses Brokers aufgeführt.

	Username ▼	Console access	Groups	Pending modifications
<input type="radio"/>	paolo.santos	No	Devs	
<input type="radio"/>	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](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).

So bearbeiten Sie einen vorhandenen Benutzer

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Wählen Sie in der Broker-Liste den Namen Ihres Brokers (z. B. MyBroker) und klicken Sie auf und danach auf View details (Details anzeigen).

Auf der Seite **MyBroker** werden im Abschnitt Benutzer alle Benutzer dieses Brokers aufgeführt.

	Username ▼	Console access	Groups	Pending modifications
<input type="radio"/>	paolo.santos	No	Devs	
<input type="radio"/>	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](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).

So löschen Sie einen vorhandenen Benutzer

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Wählen Sie in der Broker-Liste den Namen Ihres Brokers (z. B. MyBroker) klicken Sie auf und danach auf View details (Details anzeigen).

Auf der Seite **MyBroker** werden im Abschnitt Benutzer alle Benutzer dieses Brokers aufgeführt.

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

3. Wählen Sie Ihre Anmeldeinformationen (z. B. **MyUser**) und dann Löschen aus.
4. Um die Löschung des Benutzers zu bestätigen, wählen Sie im Dialogfeld delete **MyUser?** die Option Delete (Löschen).

⚠ Important

Das Vornehmen von Änderungen an einem Benutzer wendet nicht sofort die Änderungen auf den Benutzer an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder [den Broker neu starten](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).

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.

Themen

- [Herstellen einer Verbindung mit Amazon MQ](#)
- [Sicherstellen einer effektiven Amazon MQ-Leistung](#)
- [Vermeiden von langsamen Neustarts durch Wiederherstellung vorbereiteter XA-Transaktionen](#)

Herstellen einer Verbindung mit Amazon MQ

Die folgenden Entwurfsmuster können die Effektivität der Verbindung Ihrer Anwendung mit Ihrem Amazon MQ-Broker verbessern.

Themen

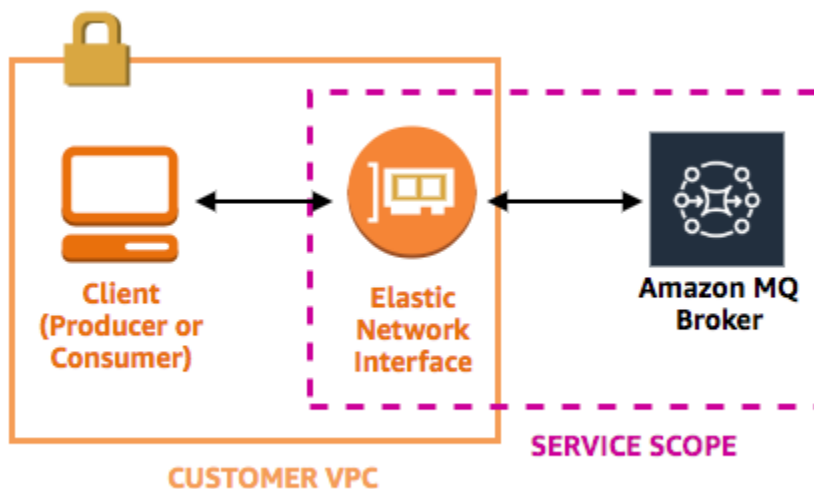
- [Verändern oder löschen Sie auf keinen Fall die Amazon MQ Elastic Network-Schnittstelle](#)
- [Verwenden Sie immer Verbindungspools](#)
- [Immer Failover-Transport verwenden, um Verbindungen zu mehreren Broker-Endpunkten einzurichten](#)
- [Vermeiden Sie die Nachrichtenauswahl](#)
- [Virtuelle Ziele gegenüber dauerhaften Abonnements bevorzugen](#)
- [Wenn Sie Amazon VPC Peering verwenden, vermeiden Sie Client-IPs im CIDR-Bereich 10.0.0.0/16](#)

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 [Getting Started with Amazon MQ](#)-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-3i45j6k17819-1.mq.us-east-2.amazonaws.com:61617,ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-east-2.amazonaws.com:61617)?randomize=true
```

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 Client-IPs im CIDR-Bereich **10.0.0.0/16**

Wenn Sie Amazon VPC Peering zwischen der On-Premise-Infrastruktur und Ihrem Amazon MQ -Broker einrichten, dürfen Sie keine Clientverbindungen mit IPs im CIDR-Bereich konfigurieren **10.0.0.0/16**.

Sicherstellen einer effektiven Amazon MQ-Leistung

Die folgenden Entwurfsmuster können die Effizienz und Leistung Ihres Amazon MQ-Broker verbessern.

Themen

- [Gleichzeitige Speicherung und Bereitstellung für Warteschlangen mit langsamen Konsumenten deaktivieren](#)
- [Auswählen des richtigen Broker-Instance-Typs für den besten Durchsatz](#)
- [Auswählen des richtigen Broker-Speichertyps für den besten Durchsatz](#)
- [Korrekte Konfiguration Ihres Netzwerk von Brokern](#)

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 Beispielformatierung 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.

Zur Überwachung nicht aufgelöster vorbereiteter XA-Transaktionen 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.

Regionsübergreifende Datenreplikation für Amazon MQ für ActiveMQ

Amazon MQ für ActiveMQ bietet eine Funktion zur regionsübergreifenden Datenreplikation (CRDR), die eine asynchrone Nachrichtenreplikation vom Primär-Broker in einer primären AWS-Region zum Replikat-Broker 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.

Dieser Abschnitt enthält Tutorials zur Einrichtung der regionsübergreifenden Datenreplikation mit Amazon MQ für ActiveMQ.

Themen

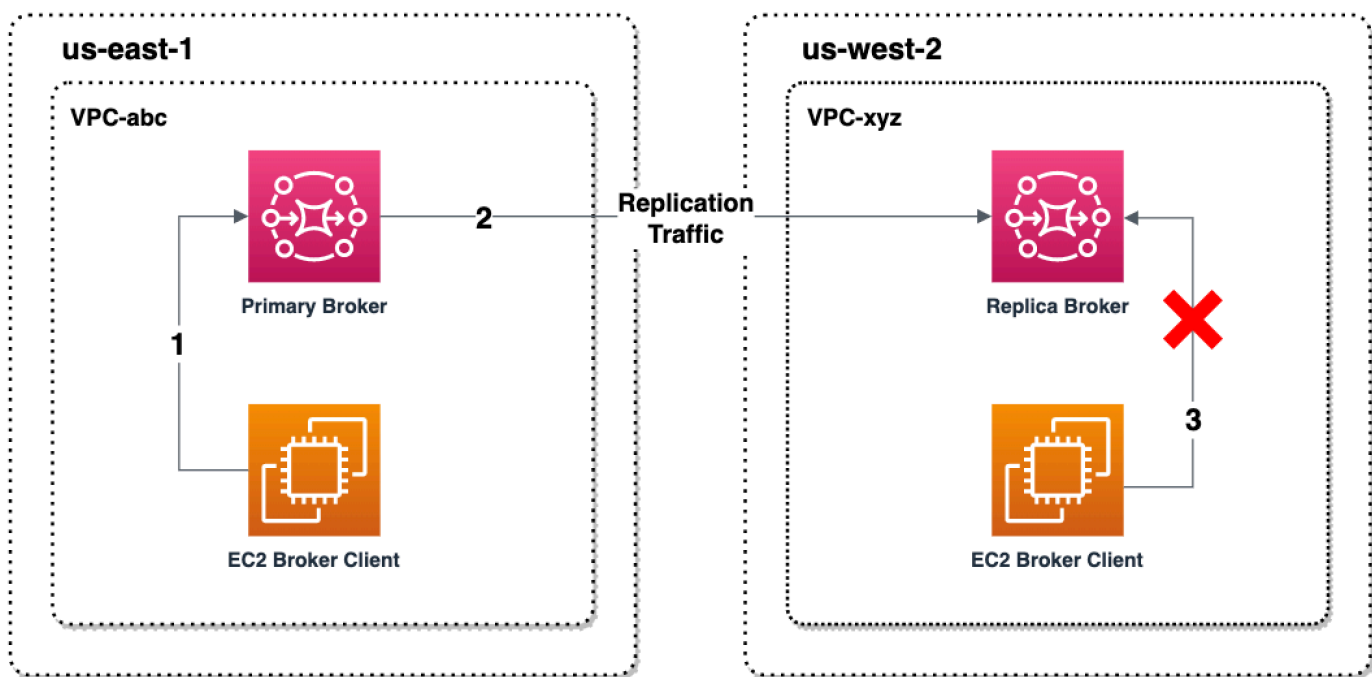
- [Primär- und Replikat-Broker in Amazon MQ](#)
- [Erstellen und Löschen eines regionsübergreifenden Datenreplikations-Brokers](#)
- [Initiieren eines Switchovers oder Failovers, um den Replikat-Broker in die Rolle des Primär-Brokers hochzustufen](#)

- [Regionsübergreifende Datenreplikationsmetriken in Amazon CloudWatch](#)

Primär- und Replikat-Broker in Amazon MQ

Sie können Primär- und Replikat-Broker für die asynchrone Datenreplikation vom Primär-Broker in einer primären AWS-Region zum Replikat-Broker 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 [Erstellen und Löschen eines regionsübergreifenden Datenreplikations-Brokers](#).

Note

Nur für aktive/Standby-Broker verfügbar.

Erstellen und Löschen eines regionsübergreifenden Datenreplikations-Brokers

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 Replikat-Broker in Amazon MQ](#).

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

Themen

- [Voraussetzungen](#)
- [Schritt 1 \(Optional\): Erstellen eines neuen Primär-Brokers](#)
- [Schritt 2: Erstellen eines Replikats eines vorhandenen Brokers](#)
- [Löschen eines CRDR-Brokers](#)

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 [Broker Architecture](#).
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. Brokernamen sind für andere AWS-Services 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-Services 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-Services 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-Services 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 [Erstellen und Herstellen einer Verbindung zu einem ActiveMQ-Broker](#).

Löschen eines CRDR-Brokers

Um einen Primär-CRDR-Broker oder einen Replikat-CRDR-Broker zu löschen, müssen Sie die Broker zunächst entkoppeln und anschließend neu starten. Die folgenden Anweisungen zeigen, wie Sie die Broker mithilfe der AWS-Managementkonsole entkoppeln und anschließend neu starten.

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

Initiieren eines Switchovers oder Failovers, um den Replikat-Broker in die Rolle des Primär-Brokers hochzustufen

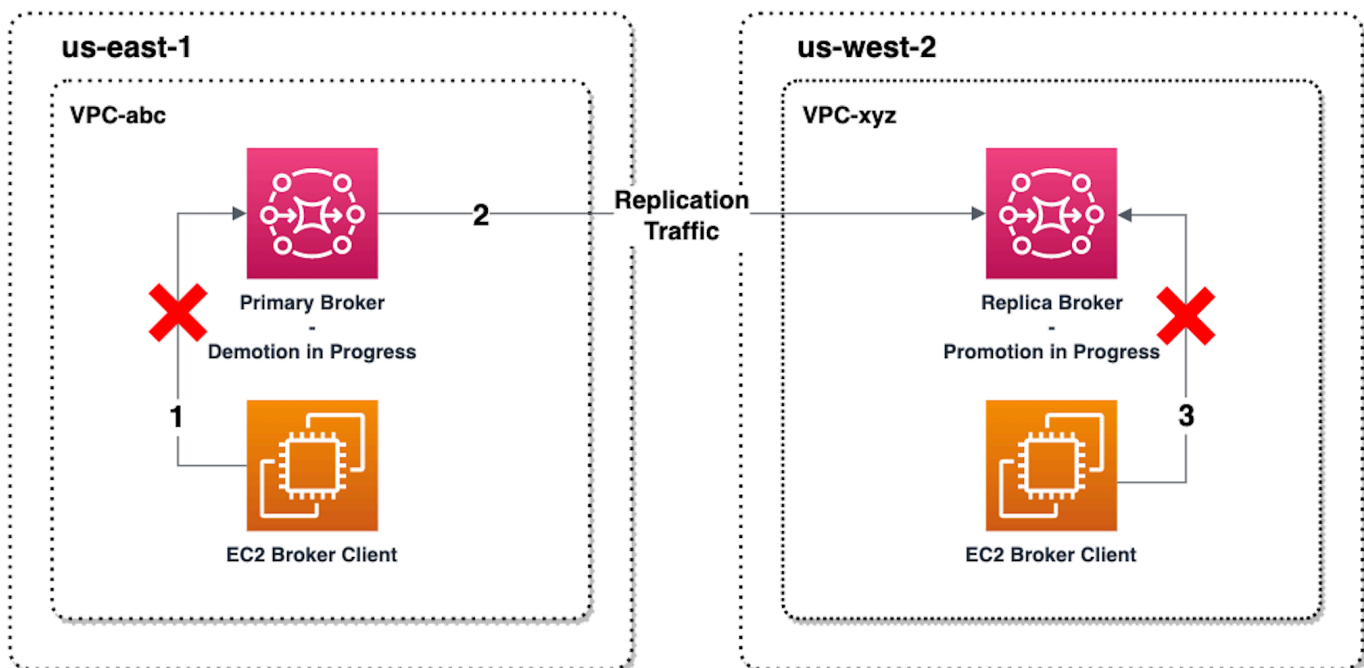
Sie können ein Switchover oder Failover initiieren, wenn Sie den Replikat-Broker in die Rolle des Primär-Brokers hochstufen möchten. Wenn Sie den Replikat-Broker hochstufen, wird der Primär-Broker in die Rolle des Replikat-Brokers herabgestuft.

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.

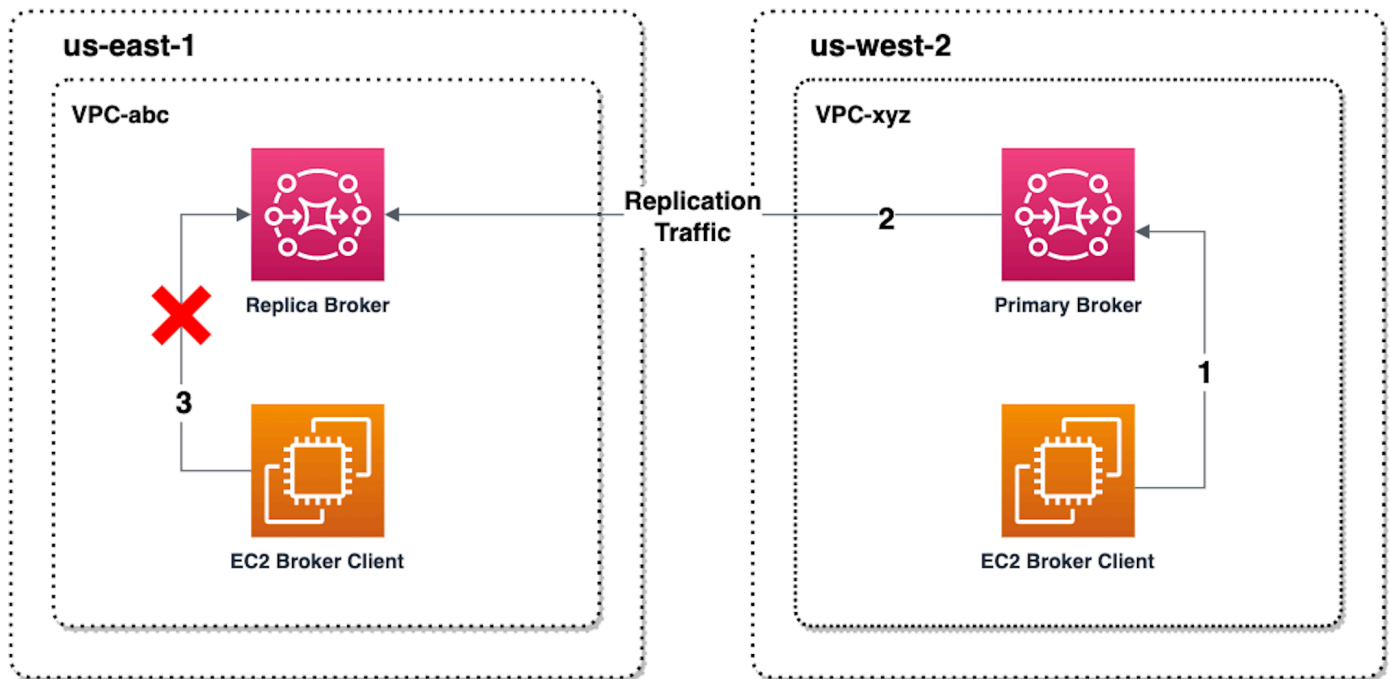
Bei einem Failover hat die Verfügbarkeit Vorrang vor der Konsistenz. Es wird nicht garantiert, dass die Broker nach Abschluss dieses Failover-Vorgangs den gleichen Status haben. Bei einem Failover ist der Replikat-Broker garantiert sofort verfügbar, um den Client-Datenverkehr zu bearbeiten, ohne

auf die Synchronisierung der Replikationsdaten oder auf das Signal zum Herunterfahren des Primär-Brokers zu warten. Der Erfolg des Failovers hängt weder vom Zustand der ursprünglichen primären Region noch vom Netzwerk zwischen den Regionen ab.

Das folgende Diagramm veranschaulicht einen Switchover, bei dem keiner der beiden Broker Clientverbindungen annimmt, während die Replikationswarteschlange geleert wird und die Broker-Status synchronisiert werden. Bei diesem Prozess kann der Client in der VPC des Primär-Brokers keine weiteren Zustandsänderungen vornehmen, während ein Failover ausgeführt wird, und der Primär-Broker wird zu einem Replikat heruntergestuft. 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.

Regionsübergreifende Datenreplikationsmetriken in Amazon CloudWatch

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

CRDR-Zeitstempel

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

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

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

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

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

Kontingente in Amazon MQ für ActiveMQ

In diesem Thema werden Kontingente innerhalb von Amazon MQ aufgeführt. Viele der folgenden Kontingente können für bestimmte AWS-Konten geändert werden. Weitere Informationen zur Beantragung einer Erhöhung eines Limits finden Sie unter [AWS Service-Kontingente](#) in der Allgemeinen Amazon Web Services-Referenz. Aktualisierte Limits sind auch nach Anwendung der Limit-Erhöhung nicht sichtbar. Weitere Informationen zum Anzeigen aktueller Verbindungslimits in Amazon CloudWatch finden Sie unter [Überwachen von Amazon MQ-Brokern mit Amazon CloudWatch](#).

Note

Kontingente in Amazon MQ für RabbitMQ finden Sie unter [Kontingente in Amazon MQ für RabbitMQ](#).

Themen

- [Broker](#)
- [Konfigurationen](#)
- [Benutzer](#)
- [Datenspeicherung](#)
- [API-Drosselung](#)

Broker

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ für ActiveMQ-Broker aufgeführt.

Limit	Beschreibung
Broker-Name	<ul style="list-style-type: none"> Muss in der Broker-Region und 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.
Anzahl der Broker, pro Region	50
Wire-Level-Verbindungen pro Protokoll für kleineren Broker	300 für mq.*.micro Instance-Typ-Broker.
Wire-Level-Verbindungen pro Protokoll für größeren Broker	2.000 für mq.*.large Instance-Typ-Broker.
Anzahl der Netzwerkkonnektoren	20
Sicherheitsgruppen pro Broker	5
ActiveMQ-Ziele (Warteschlangen und Themen), die in überwacht werden CloudWatch	CloudWatch überwacht nur die ersten 1000 Ziele.
In überwachte RabbitMQ-Ziele (Warteschlangen) CloudWatch	CloudWatch überwacht nur die ersten 500 Ziele, geordnet nach Anzahl der Verbraucher.
Tags pro Broker	50

Konfigurationen

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ für ActiveMQ-Konfigurationen aufgeführt.

Limit	Beschreibung
Konfigurationsname	<ul style="list-style-type: none">• Er muss 1–150 Zeichen umfassen.• 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 für ActiveMQ-Broker-Benutzer aufgeführt.

Limit	Beschreibung
Username	<ul style="list-style-type: none">• 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. (,).

Limit	Beschreibung
Passwort	<ul style="list-style-type: none"> • Es muss 12–250 Zeichen umfassen. • Es darf nur Zeichen aus den darstellbaren ASCII-Zeichen enthalten. • Es muss mindestens 4 eindeutige Zeichen enthalten. • Es darf keine Kommas enthalten. (,).
Benutzer pro Broker (einfache Auth)	250
Gruppen pro Benutzer (einfache Auth)	20

Datenspeicherung

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ für ActiveMQ-Datenspeicher aufgeführt.

Limit	Beschreibung
Speicherkapazität pro kleinerem Broker	20 GB fürmq.*.micro Instance-Typ-Broker. Weitere Informationen zu Amazon MQ Instance-Typen finden Sie unter Broker instance types .
Speicherkapazität pro Broker	200 GB fürmq.*.*large 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	50 GB. Weitere Informationen zur Verwendung der Job-Planer finden Sie unter JobSchedulerUsage imApache ActiveMQ -API-Dokumentation.

Limit	Beschreibung
Temporäre Speicherkapazität pro kleineren Broker.	5 GBmq.*.micro Instance-Typ-Broker.
Temporäre Speicherkapazität pro größeren Broker.	50 GB für mq.*.*large Instance-Typ-Broker.

API-Drosselung

Die folgenden Drosselungskontingente werden pro AWS Konto und für alle Amazon MQ-APIs aggregiert, um die Servicebandbreite aufrechtzuerhalten. Weitere Informationen zu Amazon MQ - APIs finden Sie im [Amazon MQ REST-API-Referenz](#).

Important

Diese Kontingente gelten nicht für Amazon MQ for ActiveMQ oder Amazon MQ für RabbitMQ Broker Messaging-APIs. Amazon MQ drosselt z. B. nicht das Senden und Empfangen von Nachrichten.

API-Aufrust-Limit	API-Ratenlimits
100	15

Arbeiten mit Amazon MQ für RabbitMQ

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 AWS Management Console, Amazon MQ-REST-API oder der AWS Command Line Interface erstellen, verwalten oder löschen.

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 API-Referenz](#).

Themen

- [RabbitMQ-Engine](#)
- [RabbitMQ-Tutorials](#)
- [Best Practices für Amazon MQ für RabbitMQ](#)
- [Kontingente in Amazon MQ für RabbitMQ](#)

RabbitMQ-Engine

Dieser Abschnitt beschreibt die grundlegenden Elemente eines RabbitMQ-Brokers und die unterstützten Plug-ins und bietet einen Überblick über die Optionen der RabbitMQ-Brokerarchitektur auf Amazon MQ.

Themen

- [Grundelemente](#)
- [Broker-Architektur](#)
- [Amazon MQ für RabbitMQ-Brokerkonfigurationen](#)
- [Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen](#)

Grundelemente

Dieser Abschnitt bietet eine Einführung in wichtige Konzepte zum Verständnis von RabbitMQ on Amazon MQ.

Themen

- [Broker](#)
- [Standardeinstellungen für Broker](#)
- [Broker-Instance-Typen](#)
- [Konfigurationen](#)
- [Benutzer](#)
- [Plug-ins](#)
- [Richtlinien](#)

Broker

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Die kombinierte Beschreibung der Broker-Instanceclass(m5,t3) undsize(large,micro) ist einBroker-Instance-Typ(zum Beispielmq.m5.large). Weitere Informationen finden Sie unter [Broker instance types](#).

- Ein Single-Instance-Broker besteht aus einem Broker in einer Availability Zone hinter einem Network Load Balancer (NLB). Der Broker kommuniziert mit Ihrer Anwendung und mit einem Amazon EBS-Speichervolume.
- Ein Cluster-Bereitstellung ist eine logische Gruppierung von drei RabbitMQ-Broker-Knoten hinter einem Network Load Balancer, wobei jeder Benutzer, Warteschlangen und ein verteilter Status über mehrere Availability Zones (AZ) verfügt.

Weitere Informationen finden Sie unter [Broker-Architektur](#).

Sie können automatische Upgrades auf Unterversionen aktivieren, damit Upgrades auf neue Unterversionen der Broker-Engine ausgeführt werden, sobald neue Versionen von RabbitMQ-Engine veröffentlicht werden. Automatische Upgrades werden während der-Wartungsfensterdefiniert durch den Wochentag, die Tageszeit (im 24-Stunden-Format) und die Zeitzone (standardmäßig UTC).

Unterstützte Protokolle

Sie können auf Ihre RabbitMQ-Broker mithilfe von [jede Programmiersprache, die RabbitMQ unterstützt](#) und durch Aktivieren von TLS für die folgenden Protokolle:

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

Listener-Ports

Von Amazon MQ verwaltete RabbitMQ Broker unterstützen die folgenden Listener-Ports für die Konnektivität auf Anwendungsebene über amqps, sowie Clientverbindungen über die RabbitMQ-Webkonsole und die Management-API.

- Listener-Port 5671 – Wird für Verbindungen verwendet, die über die sichere AMQP-URL hergestellt werden. Zum Beispiel, für einen Broker mit Broker-IDb-c8352341-ec91-4a78-ad9c-a43f23d325bb, der in derus-west-2Region bereitgestellt ist, ist dies die komplette BrokeramqpURL: `b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com:5671`.
- Listener-Ports 443 und 15671 – Beide Listener-Ports können austauschbar für den Zugriff auf einen Broker über die RabbitMQ-Webkonsole oder die Verwaltung-API verwendet werden.

Attribute

Ein RabbitMQ-Broker verfügt über mehrere Attribute:

- Ein Name. Beispiel: `MyBroker`
- Eine ID. Beispiel: `b-1234a5b6-78cd-901e-2fgh-3i45j6k17819`
- Einen Amazon-Ressourcennamen (ARN). Beispiel: `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819`
- Eine URL der RabbitMQ-Webkonsole. Beispiel: `https://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.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-3i45j6k17819-1.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](#)

Standardeinstellungen für Broker

Wenn Sie einen Amazon MQ für RabbitMQ Broker erstellen, wendet Amazon MQ einen Standardsatz von Broker-Richtlinien und vhost-Limits an, um die Leistung Ihres Brokers zu optimieren. Amazon MQ wendet Vhost-Beschränkungen nur auf den Standardwert (/) vhost an. Amazon MQ wendet keine Standardrichtlinien auf neu erstellte vhosts an. Wir empfehlen, diese Standardwerte für alle neuen und bestehenden Broker beizubehalten. Sie können diese Standardwerte jedoch jederzeit ändern, überschreiben oder löschen.

Amazon MQ erstellt Richtlinien und Limits basierend auf dem Instance-Typ und dem Broker-Bereitstellungsmodus, den Sie beim Erstellen Ihres Brokers auswählen. Die Standardrichtlinien werden gemäß dem Bereitstellungsmodus wie folgt benannt:

- Einzelne Instance – AWS-DEFAULT-POLICY-SINGLE-INSTANCE
- Cluster-Bereitstellung–AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ

Für [Single-Instance-Broker](#) festgelegt 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 auf 1 für Broker-Standardwerte fest. Um eine eigene benutzerdefinierte Richtlinie für Cluster zu erstellen, weisen Sie einen Prioritätswert zu, der größer als 1 ist.

Note

In Clusterbereitstellungen ha-mode und ha-sync-mode Broker-Richtlinien sind für die klassische Spiegelung und Hochverfügbarkeit (HA) erforderlich.

Wenn Sie die Standardeinstellung AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ-Richtlinie verwenden, verwendet Amazon MQ die ha-a11-AWS-OWNED-DO-NOT-DELETE-Richtlinie mit dem Prioritätswert 0. Dadurch wird sichergestellt, dass die erforderlichen ha-mode und ha-sync-mode-Richtlinien weiterhin in Kraft sind. Wenn Sie Ihre eigene benutzerdefinierte Richtlinie erstellen, hängt Amazon MQ automatisch ha-mode und ha-sync-mode zu Ihren Richtliniendefinitionen an.

Themen

- [Richtlinien- und Grenzbeschreibungen](#)
- [Empfohlene Standardwerte](#)

Richtlinien- und Grenzbeschreibungen

In der folgenden Liste werden die Standardrichtlinien und -beschränkungen beschrieben, die Amazon MQ für einen neu erstellten Broker anwendet. Die Werte für `max-length`, `max-queues`, und `max-connections` variieren je nach Instance-Typ und Bereitstellungsmodus Ihres Brokers. Diese Werte werden im Feld Abschnitts [Empfohlene Standardwerte](#) erstellt.

- **queue-mode: lazy**(Richtlinie) — Aktiviert Lazy-Warteschlangen. Standardmäßig halten Warteschlangen einen In-Memory-Cache von Nachrichten, so dass der Broker Nachrichten so schnell wie möglich an Verbraucher senden kann. Dies kann dazu führen, dass der Broker der Speicher ausläuft und einen Alarm mit hohem Speicher auslöst. Lazy Queues versuchen, Nachrichten so früh wie möglich auf den Datenträger zu verschieben. Dies bedeutet, dass unter normalen Betriebsbedingungen weniger Meldungen im Speicher gespeichert werden. Amazon MQ für RabbitMQ kann mithilfe von Lazy Queues viel größere Messaging-Lasten und längere Warteschlangen unterstützen. Beachten Sie, dass in bestimmten Anwendungsfällen Broker mit faulen Warteschlangen möglicherweise geringfügig langsamer ausgeführt werden. Dies liegt daran, dass Nachrichten vom Datenträger zu Broker verschoben werden, anstatt Nachrichten aus einem In-Memory-Cache zu übermitteln.

Bereitstellungsmodi

Ein Single-Instance-Cluster

- **max-length: *number-of-messages***(Richtlinie) — Legt ein Limit für die Anzahl der Nachrichten in einer Warteschlange fest. In Clusterbereitstellungen verhindert das Limit die angehaltene Warteschlangensynchronisierung in Fällen wie Broker-Neustarts oder im Anschluss an ein Wartungsfenster.

Bereitstellungsmodi


Cluster

- **overflow: reject-publish**(policy) — Erzwingt Warteschlangen mit einem `max-length` Um neue Nachrichten abzulehnen, nachdem die Anzahl der Nachrichten in der Warteschlange den `max-length` Wert erreicht. Um sicherzustellen, dass Nachrichten nicht verloren gehen, wenn sich eine Warteschlange in einem Überlaufzustand befindet, müssen Clientanwendungen, die Nachrichten an den Broker [Herausgeber bestätigt](#) implementieren. Weitere Informationen zur


Implementierung von Publisher-Bestätigungen finden Sie unter [Herausgeber bestätigt](#) auf der RabbitMQ-Website.

 Bereitstellungsmodi
Cluster

- **max-queues:** *number-of-queues-per-vhost*(vhost-Limit) — Legt das Limit für die Anzahl der Warteschlangen in einem Broker fest. Ähnlich wie bei `max-length`-Richtliniendefinition verhindert die Begrenzung der Anzahl der Warteschlangen in Clusterbereitstellungen die angehaltene Warteschlangensynchronisierung nach Broker-Neustarts oder Wartungsfenstern. Durch 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

Note

Die `max-length` und `max-queue` Standardlimits werden basierend auf einer durchschnittlichen Nachrichtengröße von 5 kB getestet und ausgewertet. Wenn Ihre Nachrichten deutlich größer als 5 kB sind, müssen Sie die `max-length` und `max-queue`-Beschränkungen.

In der folgenden Tabelle finden Sie die Standardgrenzwerte für einen neu erstellten Broker. Amazon MQ wendet diese Werte entsprechend dem Instance-Typ und dem Bereitstellungsmodus des Brokers an.

Instance-Typ	Bereitstellungsmodus	max-length	max-queues	max-connections
t3.micro	Single-Instance	N/A	500	500
m5.large	Single-Instance	N/A	20 000	4.000
	Cluster	8 000 000	4.000	15 000
m5.xlarge	Single-Instance	N/A	30 000	8 000
	Cluster	9 000 000	5,000	20 000
m5.2xlarge	Single-Instance	N/A	60 000	15 000
	Cluster	10 000 000	6 000	40 000
m5.4xlarge	Single-Instance	N/A	150 000	30 000
	Cluster	12 000 000	10.000	100 000


Broker-Instance-Typen

Important

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

Important

In den Availability Zones euc1-az2 können Sie derzeit keine t2.micro-, m4.large- oder m5.*-Broker erstellen.

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
mq.t3.micro	2	1	Niedrig	<p>Verwenden Sie den mq.t3.micro -Instance-Typ für eine grundlegende Auswertung von Amazon MQ. Dieser Instance-Typ (nur bei Single-Instance-Brokern) ist für das AWSkostenlose Kontingent</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important</p> <p>Der mq.t3.micro -Instance-Typ unterstützt die Cluster-Bereitstellung nicht.</p> </div>
mq.m5.large	2	8	Hoch	<p>Verwenden Sie die mq.m5.large -Instance für reguläre Entwicklungs-, Test- und</p>

Instance-Typ	vCPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Hinweise
				Produktions-Workloads.
mq.m5.xlarge	4	16	Hoch	Verwenden Sie Instances vom Typ mq.m5.xlarge , mq.m5.2xlarge und mq.m5.4xlarge für reguläre Entwicklungs-, Test- und Produktions-Workloads, die einen höheren Durchsatz erfordern.
mq.m5.2xlarge	8	32	Hoch	
mq.m5.4xlarge	16	64	Hoch	

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.

Wichtig

Das Vornehmen von Änderungen an einer Konfiguration nichtwenden 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](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).

Derzeit ist es nicht möglich, eine Konfiguration zu löschen.

Weitere Informationen zum Erstellen, Bearbeiten und Verwalten von Konfigurationen finden Sie unter:

- [Creating and applying broker configurations](#)
- [RabbitMQ Broker Configurations](#)

Um die Änderungen an Ihren Konfigurationen nachzuverfolgen, können Sie Konfigurationsrevisionen erstellen. Weitere Informationen finden Sie unter [Creating and applying broker configurations](#).

Attribute

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

- Einen Namen (MyConfiguration)
- Eine ID (c-1234a5b6-78cd-901e-2fgh-3i45j6k17819)
- Einen Amazon-Ressourcennamen (ARN) (arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k17819)

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](#)


Benutzer

Jede AMQP 0-9-1 Clientverbindung hat einen zugeordneten Benutzer, der authentifiziert werden muss. Jede Clientverbindung zielt auch auf einen virtuellen Host (vhost) ab, für den der Benutzer über eine Reihe von Berechtigungen verfügen muss. Ein Benutzer kann die Berechtigung haben, Warteschlangen und Exchanges in einem Vhost zu konfigurieren, schreiben, und zu lesen. Benutzeranmeldeinformationen und der Ziel-vhost werden bei der Verbindungsherstellung angegeben.

Wenn Sie zum ersten Mal einen Broker für Amazon MQ für RabbitMQ erstellen, verwendet Amazon MQ die von Ihnen angegebenen Anmeldeinformationen, um einen RabbitMQ-Benutzer mit dem `administrator`-Tag zu erstellen. Sie können dann Benutzer über die RabbitMQ [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 für RabbitMQ unterstützt den Benutzernamen „Gast“ nicht und löscht das Standard-Gastkonto, wenn Sie einen neuen Broker erstellen. Amazon MQ löscht auch regelmäßig alle vom Kunden erstellten Konten namens „Gast“.

Um einen neuen Benutzer mit der RabbitMQ-Management-API zu erstellen, verwenden Sie den folgenden API-Endpunkt und den folgenden Anforderungstext. Ersetzen Sie den *Benutzernamen* und das *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-Services zugänglich, einschließlich - CloudWatch Protokollen. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.
- Wenn Sie das Administratorpasswort vergessen haben, das Sie bei der Erstellung des Brokers festgelegt haben, können Sie Ihre Anmeldeinformationen nicht zurücksetzen. Wenn Sie mehrere Administratoren erstellt haben, können Sie sich über einen anderen Administratorbenutzer anmelden und Ihre Anmeldeinformationen zurücksetzen oder neu

erstellen. Wenn Sie nur einen Administratorbenutzer haben, müssen Sie den Broker löschen und einen neuen mit neuen Anmeldeinformationen erstellen. Wir empfehlen, Nachrichten zu lesen oder zu sichern, bevor Sie den Broker löschen.

Der `tags`-Schlüssel ist obligatorisch und besteht aus einer durch Kommas getrennten Liste von Tags für den Benutzer. Amazon MQ unterstützt `administrator`-, `management`-, `monitoring`- und `polycymaker`-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` mit Ihren Informationen. Für den Standard-`vhost/`, verwenden Sie `%2F`.

```
POST /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](#).

Plug-ins

Amazon MQ für RabbitMQ unterstützt das [RabbitMQ Management Plugin](#), das die Management-API und die RabbitMQ-Webkonsole ausführt. Sie können die Webkonsole und die Management-API zum Erstellen und Verwalten von Broker-Benutzern und -Richtlinien verwenden.

Neben dem Management Plugin unterstützt Amazon MQ für RabbitMQ auch die folgenden Plug-ins.

Themen

- [Shovel Plugin](#)
- [Federation Plugin](#)
- [Consistent Hash Exchange Plugin](#)

Shovel Plugin

Von Amazon MQ verwaltete Makler unterstützen die [RabbitMQ Shovel](#), sodass Sie Nachrichten aus Warteschlangen und Exchanges auf einer Broker-Instance in eine andere verschieben können. Sie können Shovel verwenden, um lose gekoppelte Broker zu verbinden und Nachrichten von Knoten mit schwereren Nachrichtenladungen zu verteilen.

Von Amazon MQ verwaltete RabbitMQ Broker unterstützen dynamische Shovels. Dynamische Shovels werden mit Laufzeitparametern konfiguriert und können jederzeit programmgesteuert über eine Clientverbindung gestartet und gestoppt werden. Sie können z. B. mithilfe der RabbitMQ-Management-API eine PUT-Anforderung an den folgenden API-Endpunkt erstellen, um eine dynamische Shovel zu konfigurieren. Im Beispiel kann `{vhost}` durch den Namen des vhost des Brokers und `{name}` durch den Namen der neuen dynamischen Shovel ersetzt werden.

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

Im Anforderungstext müssen Sie entweder eine Warteschlange oder einen Exchange angeben, aber nicht beides. In diesem Beispiel unten wird eine dynamische Shovel zwischen einer in `src-queue` angegebenen lokalen Warteschlange und einer in `dest-queue` definierten Remote-Warteschlange konfiguriert. Auf ähnliche Weise können Sie die Parameter `src-exchange` und `dest-exchange` verwenden, um eine Shovel zwischen zwei Exchanges zu konfigurieren.


```
{
  "value": {
    "src-protocol": "amqp091",
    "src-uri": "amqp://localhost",
    "src-queue": "source-queue-name",
    "dest-protocol": "amqp091",
    "dest-uri": "amqps://b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-
west-2.amazonaws.com:5671",
    "dest-queue": "destination-queue-name"
  }
}
```

Important

Sie können die Shovel zwischen Warteschlangen oder Exchanges nicht konfigurieren, wenn das Shovel-Ziel ein privater Broker ist. Sie können die Shovel nur zwischen Warteschlangen

oder Exchanges in öffentlichen Brokern oder zwischen einer Quelle in einem privaten Broker und einem Ziel in einem öffentlichen Broker konfigurieren.

Weitere Informationen zur Verwendung dynamischer Shovels finden Sie unter dem [RabbitMQ Dynamic Shovel Plugin](#).


 Note

Amazon MQ unterstützt die Verwendung statischer Shoveln nicht.

Federation Plugin

Amazon MQ unterstützt Verbund-Exchange und -Warteschlangen. Mit Verbund können Sie den Nachrichtenfluss zwischen Warteschlangen, Exchanges und Verbrauchern auf separaten Brokern replizieren. Verbundwarteschlangen und Exchanges verwenden point-to-point Links, um eine Verbindung 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. Mit der Management-API können Sie eine POSTAnforderung/`api/parameters/federation-upstream/%2f/my-upstream` mit folgendem Anforderungstext 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. Mit der Verwaltungs-API können Sie eine POSTAnforderungen/`api/policies/%2f/federate-me` mit folgendem Anforderungstext erstellen.

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

Note

Der Anforderungstext nimmt an, dass Exchanges auf dem Server mit `amq` beginnen. Verwenden von regulären Ausdrücken `^amq\\.` stellt sicher, dass der Verbund für alle Börsen aktiviert ist, deren Namen mit „amq“ beginnen. Die Exchanges auf Ihrem RabbitMQ-Server können unterschiedlich benannt werden.

Weitere Informationen zum Konfigurieren des Federation Plugins finden Sie unter [RabbitMQ Federation Plugin](#).

Consistent Hash Exchange Plugin

Standardmäßig unterstützt Amazon MQ für RabbitMQ das Exchange Plug-in „Consistent Hash“. Consistent Hash tauscht Routing-Nachrichten an Warteschlangen aus, basierend auf einem Hash-Wert, der aus dem Routing-Schlüssel einer Nachricht berechnet wird. Angesichts eines ziemlich gleichmäßigen Routingschlüssels können Consistent Hash Exchanges Nachrichten zwischen Warteschlangen relativ gleichmäßig verteilen.

Bei Warteschlangen, die an einen Consistent Hash Exchange gebunden sind, ist der Bindungsschlüssel ein `number-as-a-string`, der das Bindungsgewicht jeder Warteschlange bestimmt. Warteschlangen mit einer höheren Bindungsstärke erhalten eine proportional höhere Verteilung von Nachrichten aus dem Consistent Hash Exchange, an den sie gebunden sind. In einer Consistent Hash Exchange-Topologie können Publisher einfach Nachrichten in der Exchange veröffentlichen, aber

Verbraucher müssen explizit konfiguriert werden, um Nachrichten aus bestimmten Warteschlangen zu verwenden.

Weitere Informationen zu Consistent Hash Exchanges finden Sie unter [RabbitMQ Consistent Hash Exchange Type](#) auf der - GitHub Website.

Richtlinien

Sie können benutzerdefinierte Richtlinien und Limits mit von Amazon MQ empfohlenen Standardwerten anwenden. Wenn Sie die empfohlenen Standardrichtlinien und -grenzwerte gelöscht haben und sie neu erstellen möchten, oder Sie zusätzliche Vhosts erstellt haben und die Standardrichtlinien und -grenzwerte auf Ihre neuen Vhosts anwenden möchten, können Sie die folgenden Schritte ausführen.

Important

Um die folgenden Schritte ausführen zu können, benötigen Sie einen Amazon MQ - Broker-Benutzer mit Administratorberechtigungen. Sie können den Administratorbenutzer verwenden, der beim ersten Erstellen des Brokers erstellt wurde, oder einen anderen Benutzer, den Sie später erstellt haben. Die folgende Tabelle enthält die erforderlichen Administratorbenutzer-Tag und Berechtigungen als reguläre Ausdrücke (regex) Muster.


Tags	Lesen Sie regex	Konfigurieren von regex	REGEXP-Schreiben
administrator	.*	.*	.*

Weitere Informationen zum Erstellen von RabbitMQ-Benutzern und zum Verwalten von Benutzer-Tags und -berechtigungen finden Sie unter [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 `-Verbindungen` Wählen Sie im Abschnitt `die Option RabbitMQ Webkonsole-URL`. Die RabbitMQ-Webkonsole wird in einer neuen Browserregisterkarte oder `-fenster` geöffnet.
5. Melden Sie sich mit Ihrem Broker-Administratortnamen und `-Passwort` an der RabbitMQ-Webkonsole an.
6. Wählen Sie in der RabbitMQ-Webkonsole oben auf der Seite die Option `Admin`.
7. Klicken Sie auf der `Admin` Wählen Sie im rechten Navigationsbereich die Option `Richtlinien`.
8. Klicken Sie auf der `Richtlinien` können Sie eine Liste der aktuellen Broker-Benutzerrichtlinien sehen. Unter `Benutzerrichtlinien` erweitern Sie `Richtlinie` hinzufügen/aktualisieren.
9. Um eine neue Broker-Richtlinie zu erstellen, tun Sie das Folgende unter `Richtlinie` hinzufügen/aktualisieren:
 - a. Für `Virtueller 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 die `Virtueller Host` wird 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. Für `Übernehmen 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 auf `Zeichenfolge` aus der Dropdown-Liste.
 - **overflow=reject-publish**. Klicken Sie auf `Zeichenfolge` aus der Dropdown-Liste.

 Note

Gilt nicht für Single-Instance-Broker.


- **max-length=number-of-messages**. Ersetzen Sie durch *number-of-messages* den von [Amazon MQ empfohlenen Wert](#) entsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers, z. B. **8000000** für einen `-mq.m5.largeCluster`. 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 der Benutzerrichtlinien.

 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 der **Einschränkungen** Sie können eine Liste der aktuellen Grenzwerte für virtuelle Hosts. Unter **Grenzwerten für virtuelle Hosts** weiterlegen Sie **Festlegen** oder **Aktualisieren** eines virtuellen Hosts.

13. Um ein neues vhost-Limit zu erstellen, gehen Sie unter **Festlegen** oder **Aktualisieren** eines virtuellen Hosts wie folgt vor:

- Für **Virtueller 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 `/`.
- Für **Limit**, wählen Sie `max-connections` aus den Dropdown-Optionen.
- Für **Value**, geben Sie den [Amazon MQ empfohlenen Wert](#) entsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers ein, z. B. **15000** für einen `mq.m5.largeCluster`.
- Klicken Sie auf **Grenzwert 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 eincurl-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%2Fdurch den Vhost-Namen.

Note


Ersetzen Sie den *Benutzernamen* und das *Passwort* durch Ihre Administrator-Anmeldeinformationen. Ersetzen Sie durch *number-of-messages* den von [Amazon MQ empfohlenen Wert](#) entsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers. Ersetzen Sie den *Richtliniennamen* durch einen Namen für Ihre Richtlinie. Ersetzen *Endpunkt Broker* mit der URL, die Sie zuvor notiert haben.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \  
-d '{"pattern":".*", "priority":1, "definition":{"queue-mode":lazy, \  
  "overflow":"reject-publish", "max-length":"number-of-messages"} }' \  
broker-endpoint/api/policies/%2F/policy-name
```

7. Um zu bestätigen, dass die neue Richtlinie den Benutzerrichtlinien Ihres Brokers hinzugefügt wird, geben Sie folgenden curl-Befehl ein, um alle Broker-Richtlinien aufzulisten.

```
curl -i -u username:password broker-endpoint/api/policies
```

- Um ein neues max-connectionsvirtuelles 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%2Fdurch den Vhost-Namen.

 Note

Ersetzen Sie den *Benutzernamen* und das *Passwort* durch Ihre Administrator-Anmeldeinformationen. Ersetzen Sie *max-connections* durch den [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
```

- Um ein neues 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
```

- 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
```

Broker-Architektur

RabbitMQ Broker können als Single-Instance-Broker oder in einem Cluster-Bereitstellung. Für beide Bereitstellungsmodi bietet Amazon MQ eine hohe Haltbarkeit, indem seine Daten redundant gespeichert werden.

Sie können auf Ihre RabbitMQ-Broker mithilfe von [jede Programmiersprache, die RabbitMQ unterstützt](#) und durch Aktivieren von TLS für die folgenden Protokolle:

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

Themen

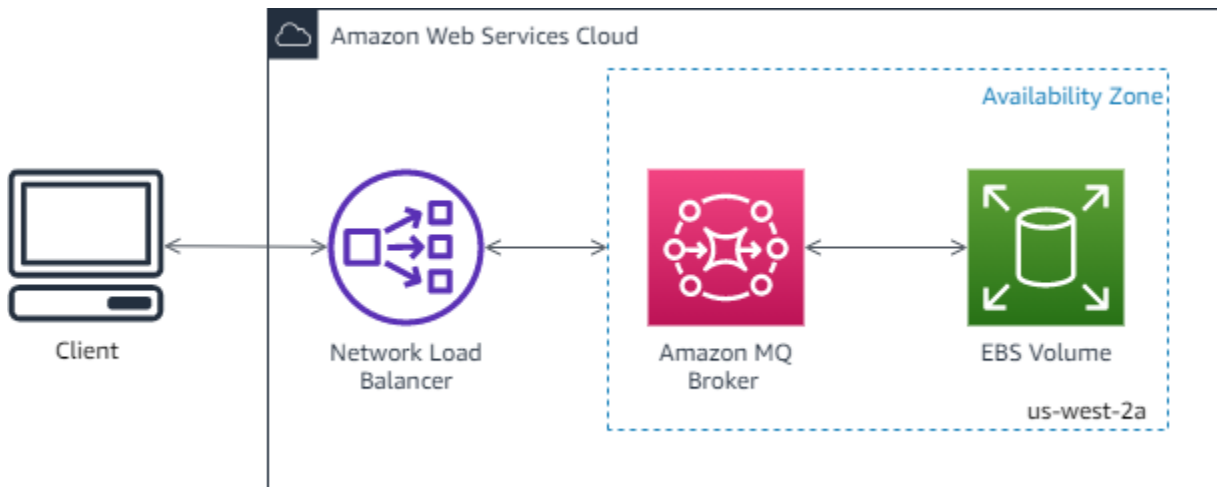
- [Single-Instance Broker](#)
- [Cluster-Bereitstellung für hohe Verfügbarkeit](#)

Single-Instance Broker

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.

Die Verwendung eines Network Load Balancer stellt sicher, dass Ihr Amazon MQ für RabbitMQ Broker-Endpunkt unverändert bleibt, wenn die Broker-Instance während eines Wartungsfensters oder aufgrund zugrunde liegender Amazon EC2 Hardwarefehler ersetzt wird. 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.



Cluster-Bereitstellung für hohe Verfügbarkeit

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 einem Hauptknoten und einen oder mehrere Spiegeln. 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` auf `all` und die `ha-sync-mode` auf `automatic` setzt. Dadurch wird sichergestellt, dass Daten auf alle Knoten im Cluster über verschiedene Availability Zones hinweg repliziert werden, um eine bessere Haltbarkeit zu gewährleisten.

Note

Während eines -Wartungsfensters wird die gesamte Wartung eines Clusters jeweils jeweils ein Knoten durchgeführt, wobei mindestens zwei laufende Knoten zu jeder Zeit beibehalten werden. Jedes Mal, wenn ein Knoten heruntergefahren wird, werden Clientverbindungen zu diesem Knoten getrennt und müssen wieder hergestellt werden. Sie müssen sicherstellen, dass der Clientcode so konzipiert ist, dass er automatisch wieder eine Verbindung mit dem Cluster herstellt. Weitere Informationen über den Wiederherstellungsprozess finden Sie unter [the section called “Automatische Wiederherstellung nach Netzwerkausfällen”](#).

Weil Amazon MQ `ha-sync-mode: automatic` während eines Wartungsfensters synchronisiert, werden die Warteschlangen synchronisiert, wenn jeder Knoten dem

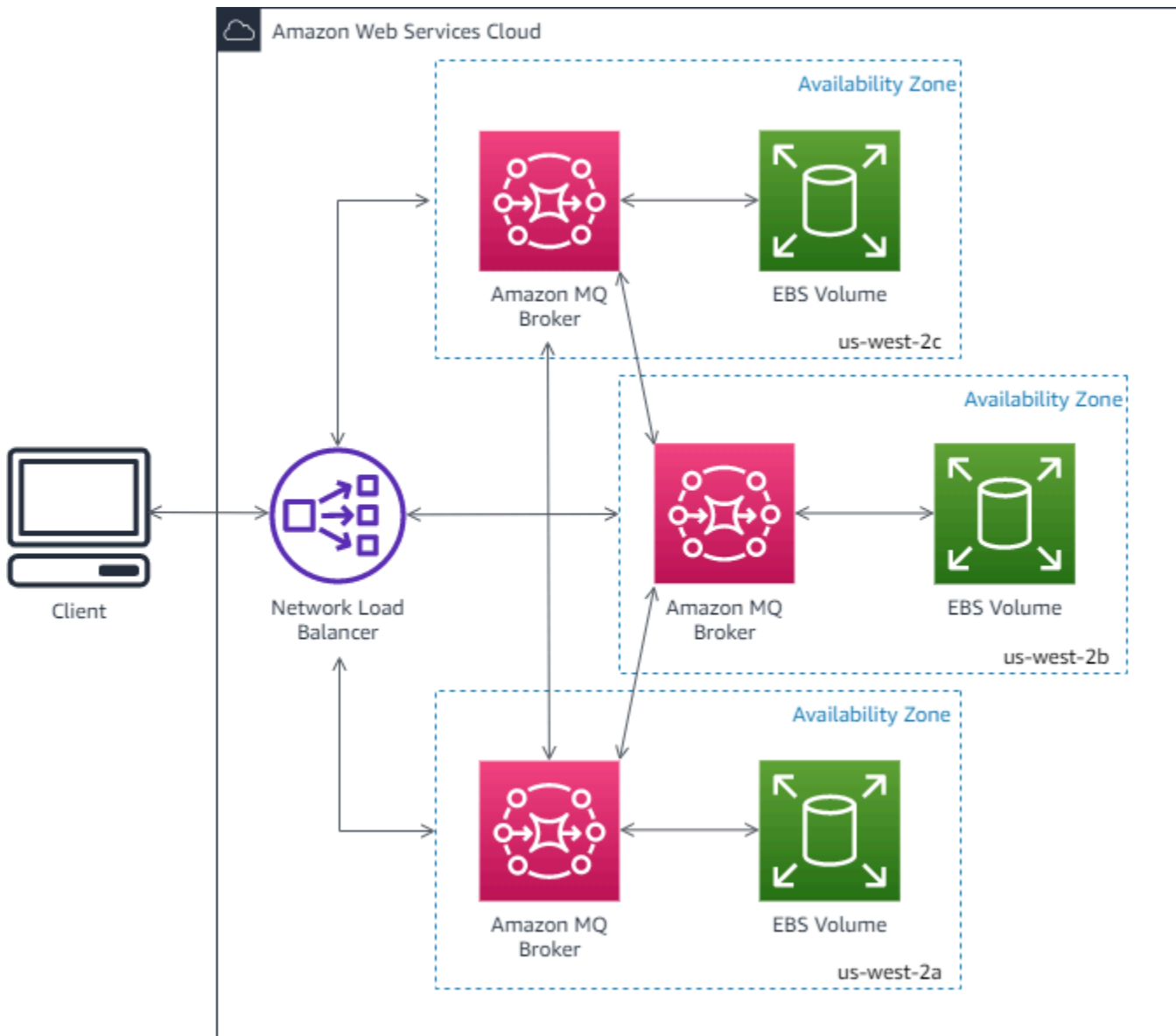
Cluster wieder beitrifft. Die Warteschlangen-Synchronisierung blockiert alle anderen Warteschlangen. Sie können die Auswirkungen der Warteschlangensynchronisierung während Wartungsfenstern verringern, indem Sie Warteschlangen kurz halten.

Die Standardrichtlinie sollte nicht gelöscht werden. Wenn Sie diese Richtlinie löschen, wird sie von Amazon MQ automatisch neu erstellt. 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 unter [Klassische gespiegelte Warteschlangen](#).

⚠ Important

Amazon MQ unterstützt nicht [Quorum-Warteschlangen](#). Das Aktivieren des Quorumwarteschlangen-Feature-Flags und das Erstellen von Quorumwarteschlangen führt zu Datenverlust.

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.



Amazon MQ für RabbitMQ-Brokerkonfigurationen

Eine Konfiguration enthält alle Einstellungen für Ihren RabbitMQ-Broker im Cuttlefish-Format. Sie können eine Konfiguration erstellen, bevor Sie Broker erstellen. Sie können die Konfiguration dann auf einen oder mehrere Broker anwenden.

Themen

- [Erstellen, Bearbeiten und Anwenden von RabbitMQ-Broker-Konfigurationen](#)
- [RabbitMQ-Konfigurationsrichtlinien](#)

Erstellen, Bearbeiten 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.

Weitere Informationen finden Sie hier:

- [Konfigurationen](#)
- [Amazon-MQ-Broker-Konfigurationslebenszyklus](#)

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

Themen

- [Eine neue Konfiguration erstellen](#)
- [Erstellen einer neuen Konfigurationsversion](#)
- [Eine Konfigurationsrevision auf Ihren Broker anwenden](#)
- [Eine Konfigurationsüberarbeitung bearbeiten](#)

Eine neue 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.

Amazon MQ ×

Brokers

Configurations

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

1. Wählen Sie in der Konfigurationsliste aus **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.

Die Speichern der Revision wird 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 Konfiguration nichtwenden 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](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).
Derzeit ist es nicht möglich, eine Konfiguration zu löschen.

Eine Konfigurationsrevision auf Ihren Broker anwenden

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

Amazon MQ ×

Brokers

Configurations

2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und wählen Sie dann Bearbeiten aus.
3. Wählen Sie auf der Seite Bearbeiten **MyBroker** 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

Ihr Broker ist offline, während er neu gestartet wird.


5. Wählen Sie Apply (Anwenden) aus.

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

Eine Konfigurationsüberarbeitung bearbeiten

1. Melden Sie sich bei der [Amazon-MQ-Konsole](#) an.


2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und wählen Sie dann Bearbeiten aus.
3. Wählen Sie auf der **MyBroker** Seite Bearbeiten aus.
4. Wählen Sie auf der Seite Bearbeiten **MyBroker** im Abschnitt Konfiguration eine Konfiguration und eine Revision und dann Bearbeiten aus.

 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 die Konfiguration verwendet (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.

Die Speichern der Revision wird 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 Konfiguration nichtwenden 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](#). Weitere Informationen finden Sie unter [Amazon-MQ-Broker-Konfigurationslebenszyklus](#).
Derzeit ist es nicht möglich, eine Konfiguration zu löschen.

RabbitMQ-Konfigurationsrichtlinien

Amazon MQ für RabbitMQ unterstützt jetzt das Erstellen und Anwenden von Konfigurationen auf Ihren RabbitMQ-Broker. Die standardmäßige Operatorrichtlinie auf jedem virtuellen Host enthält die folgenden empfohlenen HA-Eigenschaften:

```
name: default_operator_policy_AWS_managed
pattern: .*
apply-to: all
priority: 0
definition: {
  ha-mode: all
  ha-sync-mode: automatic
}
```

Änderungen der Betreiberrichtlinien über die AWS Management Console 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. Weitere Informationen zum Hinzufügen von Konfigurationen zu Ihrem Broker finden Sie unter [Creating and applying broker configurations](#).

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

RabbitMQ organisiert Versionsnummern gemäß der semantischen Versioning-Spezifikation als X.Y.Z. In Amazon MQ für RabbitMQ-Implementierungen bezeichnet X.Y die Hauptversion und Z stellt die Nebenversionsnummer dar. Amazon MQ betrachtet eine Versionsänderung als Hauptversionsänderung, wenn sich die Hauptversionsnummern ändern. Beispielsweise wird ein Upgrade von Version 3.8 auf 3.9 als Upgrade einer Hauptversion betrachtet. Eine Versionsänderung

gilt als unwesentlich, wenn sich nur die Unterversionsnummer ändert. Beispielsweise wird ein Upgrade von Version 3.8.23 auf 3.8.26 als Upgrade einer Unterversion betrachtet.

Wenn Sie einen neuen Amazon MQ für RabbitMQ Broker erstellen, können Sie jede unterstützte Version der RabbitMQ Engine angeben. Wenn Sie die verwenden, AWS Management Console um einen Broker zu erstellen, verwendet Amazon MQ automatisch die neueste Engine-Versionsnummer. Wenn Sie die AWS CLI oder die Amazon MQ API verwenden, um einen Broker zu erstellen, ist die Engine-Versionsnummer erforderlich. Wenn Sie keine Versionsnummer angeben, führt der Vorgang zu einer Ausnahme. Weitere Informationen hierzu finden Sie unter [create-broker](#) in der AWS CLI -Befehlsreferenz und [CreateBroker](#) in der Amazon-MQ-REST-API-Referenz.

Important

Amazon MQ unterstützt weder [Quorum-Warteschlangen](#) noch [-Streams](#). Das Aktivieren dieser Feature-Flag(s) und das Erstellen einer Quorum-Warteschlange oder eines Quorum-Streams führt zu Datenverlust.

Important

- Amazon MQ unterstützt die Verwendung von strukturierter Protokollierung in JSON, die in RabbitMQ 3.9 eingeführt wurde, nicht
- Amazon MQ für RabbitMQ empfiehlt neuen Brokern, die neueste unterstützte Nebenversion zu verwenden.
- RabbitMQ erlaubt nur inkrementelle Versionsaktualisierungen (z. B. von 3.9.x auf 3.10.x). Sie können Hauptversionen beim Aktualisieren nicht überspringen (z. B. von 3.8.x auf 3.10.x).

Amazon MQ für RabbitMQ unterstützt derzeit die folgenden Engine-Versionen:

Hauptversionen	Nebenversionen
RabbitMQ 3.11	<ul style="list-style-type: none"> • 3.11.28 (empfohlen)
RabbitMQ 3.10	<ul style="list-style-type: none"> • 3.10.25

Hauptversionen	Nebenversionen
RabbitMQ 3.9	<ul style="list-style-type: none"> • 3.9.27
RabbitMQ 3.8	<ul style="list-style-type: none"> • 3.8.34

Die folgenden Nebenversionen sind weiterhin für vorhandene Broker verfügbar, werden jedoch nicht für neue Broker empfohlen.

Vorhandene Nebenversionen

Hauptversionen	Nebenversionen
RabbitMQ 3.11	<ul style="list-style-type: none"> • 3.11.20 • 3.11.16
RabbitMQ 3.10	<ul style="list-style-type: none"> • 3.10.20 • 3.10.10
RabbitMQ 3.9	<ul style="list-style-type: none"> • 3.9.24 • 3.9.16 • 3.9.13
RabbitMQ 3.8	<ul style="list-style-type: none"> • 3.8.30 • 3.8.27 • 3.8.26 • 3.8.23 • 3.8.22 • 3.8.17 • 3.8.6

Aktualisierungen von Haupt- und Nebenversionen

Mit Amazon MQ können Sie steuern, wann Ihre Broker auf neue Versionen aktualisiert werden.

Wenn [Automatischer Unterversion-Upgrade](#) aktiviert ist, aktualisiert Amazon MQ Ihre Broker-Engine

automatisch auf neue Nebenversionen, sobald sie von Amazon MQ veröffentlicht und unterstützt werden.

Um ein Hauptversions-Upgrade durchzuführen, müssen Sie die Modulversionsnummer Ihres Brokers manuell aktualisieren. Neben- und Hauptversions-Upgrades finden gleichzeitig mit anderen Broker-Patching-Vorgängen während Ihres geplanten [Wartungsfensters](#) statt. Wenn Sie sich gegen ein automatisches Upgrade der Nebenversion entscheiden, können Sie Ihren Broker manuell auf eine neue unterstützte Nebenversion aktualisieren, indem Sie das gleiche Verfahren wie bei einem Hauptupgrade anwenden.

Weitere Informationen zum Aktualisieren Ihrer Broker-Voreinstellungen zum Aktivieren oder Deaktivieren von Nebenversions-Upgrades und zum manuellen Aktualisieren des Brokers finden Sie unter [the section called “Upgrade der Engine-Version”](#).

Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern müssen die gespiegelten Warteschlangen während des Neustarts synchronisiert werden. Bei längeren Warteschlangen kann der Synchronisierungsprozess der Warteschlange länger dauern. Während des Synchronisierungsprozess der Warteschlange ist die Warteschlange für Konsumenten und Produzenten nicht verfügbar. Wenn der Synchronisierungsprozess der Warteschlange abgeschlossen ist, wird der Broker wieder verfügbar. Um die Auswirkungen zu minimieren, empfehlen wir ein Upgrade während einer Zeit mit geringem Datenverkehr. Weitere Informationen zu bewährten Methoden für Versions-Upgrades finden Sie unter [Best Practices für Amazon MQ for RabbitMQ](#).

Unterstützte Engine-Versionen auflisten

Sie können alle unterstützten Engine-Neben- und Hauptversionen mit dem [describe-broker-instance-options](#) AWS CLI Befehl 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 für RabbitMQ zu filtern, und `mq.m5.largeInstance`-Typ, ersetzen `motor-Typ` mit `RABBITMQ` und `Instance-Typ` mit `mq.m5.large`.

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 unter [RabbitMQ-Tutorials](#) im Handbuch „RabbitMQ“.

Themen

- [Bearbeiten von Broker-Einstellungen](#)
- [Verwenden von Python Pika mit Amazon MQ for RabbitMQ](#)
- [Auflösen der Synchronisierung von RabbitMQ angehaltener Warteschlangensynchronisierung](#)

Bearbeiten von Broker-Einstellungen

Sie können Ihre Brokereinstellungen bearbeiten, z. B. CloudWatch Protokolle aktivieren oder deaktivieren, indem Sie die AWS Management Console.

RabbitMQ-Broker-Optionen bearbeiten

1. Melden Sie sich bei der [Amazon MQ-Konsole](#) an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und danach wählen Sie **Bearbeiten**.
3. Wählen Sie auf der Seite **MyBroker** bearbeiten im Abschnitt Spezifikationen eine Broker-Engine-Version oder einen Broker-Instance-Typ.
4. Im Abschnitt CloudWatch Logs, klicken Sie auf die Umschaltfläche, um allgemeine Protokolle zu aktivieren oder zu deaktivieren. Keine weiteren erforderlichen Schritte.

Note

- Für RabbitMQ-Broker verwendet Amazon MQ automatisch eine Service-Linked Role (SLR), um allgemeine Protokolle in CloudWatch zu veröffentlichen. Weitere Informationen finden Sie unter [the section called “Verwenden von servicegebundenen Rollen”](#)
- Amazon MQ unterstützt keine Überwachungsprotokollierung für RabbitMQ-Broker.

5. Konfigurieren Sie im Abschnitt **Wartung** den **Wartungszeitplan** für Ihren Broker:

Um Upgrades auf neue Versionen Ihres Brokers vorzunehmen, wenn sie von AWS veröffentlicht werden, wählen Sie **Automatische Upgrades von Unterversionen aktivieren**. Automatische Upgrades werden während der **Wartungsfenster** definiert durch den **Wochentag**, die **Tageszeit** (im 24-Stunden-Format) und die **Zeitzone** (standardmäßig UTC).

6. Wählen Sie **Änderungen einplanen**.

Note

Wenn Sie nur **Automatische kleinere Aktualisierungen aktivieren** wählen, wechselt die Schaltfläche zu **Speichern**, da kein Neustart des Brokers erforderlich ist.

Ihre Einstellungen werden zu der angegebenen Zeit auf Ihren Broker angewendet.

Verwenden von Python Pika mit Amazon MQ for RabbitMQ

Das folgende Tutorial zeigt, wie Sie einen [Python-Pika](#)-Client mit TLS einrichten können, der für die Verbindung zu einem Amazon-MQ-for-RabbitMQ-Broker konfiguriert ist. Pika ist eine Python-Implementierung des AMQP-0-9-1-Protokolls für RabbitMQ. Dieses Tutorial führt Sie durch die Installation von Pika, das Deklarieren einer Warteschlange, das Einrichten eines Herausgebers für das Senden von Nachrichten an den Standardaustausch des Brokers und das Einrichten eines Verbrauchers, der Nachrichten aus der Warteschlange erhält.

Themen

- [Voraussetzungen](#)
- [Berechtigungen](#)
- [Schritt eins: Erstellen Sie einen einfachen Python-Pika-Client](#)
- [Schritt zwei: Erstellen Sie einen Herausgeber und senden Sie eine Nachricht](#)
- [Schritt drei: Erstellen Sie einen Verbraucher und erhalten Sie eine Nachricht](#)
- [Schritt vier: \(Optional\) Richten Sie eine Ereignisschleife ein und konsumieren Sie Nachrichten](#)
- [Als nächstes](#)

Voraussetzungen

Um die Schritte dieses Tutorials auszuführen, benötigen Sie Folgendes:

- Einen Amazon-MQ-for-RabbitMQ-Broker. Weitere Informationen finden Sie unter [Erstellen eines Amazon-MQ-for-RabbitMQ-Brokers](#).
- [Python 3](#) für Ihr Betriebssystem installieren.
- [Pika](#) mithilfe von Python `pip` installiert. Öffnen Sie zum Installieren von Pika ein neues Terminalfenster und führen Sie Folgendes aus.

```
$ python3 -m pip install pika
```

Berechtigungen

Für dieses Tutorial benötigen Sie mindestens einen Amazon-MQ-for-RabbitMQ-Brokerbenutzer mit der Berechtigung, an einen Vhost zu schreiben und von ihm zu lesen. Die folgende Tabelle enthält die erforderlichen Mindestberechtigungen als reguläre Ausdrucksmuster (regex).

Tags (Markierungen)	Konfigurieren von regex	REGEXP-Schreiben	Lesen Sie regex
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 regex-Muster bereitstellen, die den Zugriff des Benutzers auf bestimmte Warteschlangen einschränken. Zum Beispiel, wenn Sie das Lese-regex-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 unter [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. Zum Beispiel `us-west-2`.
3. Führen Sie den folgenden Befehl im selben Verzeichnis aus, in dem Sie `publisher.py` erstellt haben.

```
$ python3 publisher.py
```

Wenn der Code erfolgreich ausgeführt wird, wird die folgende Meldung in Ihrem Terminalfenster angezeigt.

```
Trying to declare queue(hello world queue)...  
Sent message. Exchange: , Routing Key: hello world queue, Body: b'Hello World!'
```

Schritt drei: Erstellen Sie einen Verbraucher und erhalten Sie eine Nachricht

Gehen Sie wie folgt vor, um einen Verbraucher zu erstellen, der eine einzelne Nachricht aus der Warteschlange erhält.

1. Kopieren Sie den Inhalt des folgenden Codebeispiels und speichern Sie es lokal als `consumer.py` im selben Verzeichnis.

```
from basicClient import BasicPikaClient

class BasicMessageReceiver(BasicPikaClient):

    def get_message(self, queue):
        method_frame, header_frame, body = self.channel.basic_get(queue)
        if method_frame:
            print(method_frame, header_frame, body)
            self.channel.basic_ack(method_frame.delivery_tag)
            return method_frame, header_frame, body
        else:
            print('No message returned')

    def close(self):
        self.channel.close()
        self.connection.close()

if __name__ == "__main__":

    # Create Basic Message Receiver which creates a connection
    # and channel for consuming messages.
    basic_message_receiver = BasicMessageReceiver(
        "<broker-id>",
        "<username>",
        "<password>",
        "<region>"
    )

    # Consume the message that was sent.
    basic_message_receiver.get_message("hello world queue")

    # Close connections.
    basic_message_receiver.close()
```

Ähnlich wie bei dem Herausgeber, den Sie im vorherigen Schritt erstellt haben, erbt `BasicMessageReceiver` von `BasicPikaClient` und implementiert zusätzliche Methoden zum Empfangen einer einzelnen Nachricht und zum Schließen von Verbindungen.

2. In der `if __name__ == "__main__":`-Anweisung, ersetzen Sie die Parameter, die an den `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 einem Hauptreplikat auf einem Knoten und einem oder mehreren Mirror. 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 Spiegelungen (`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` während `mirror-1` nicht verfügbar ist, muss repliziert werden `mirror-1`.

Weitere Informationen zum Spiegelung finden Sie unter [Klassische gespiegelte Warteschlangen](#) auf der RabbitMQ-Website.

Wartung und Warteschlangensynchronisierung

Während [Wartungsfenstern](#) führt Amazon MQ alle Wartungsarbeiten jeweils einen Knoten aus, um sicherzustellen, dass der Broker betriebsbereit bleibt. Daher müssen Warteschlangen möglicherweise synchronisiert werden, wenn jeder Knoten den Vorgang fortsetzt. Während der Synchronisierung werden Nachrichten, die auf Spiegelungen repliziert werden müssen, vom entsprechenden Amazon Elastic Block Store (Amazon EBS) -Volume in den Speicher geladen, um in Batches verarbeitet zu werden. Durch die Verarbeitung von Nachrichten in Batches können Warteschlangen schneller synchronisiert werden.

Wenn Warteschlangen kurz gehalten werden und Nachrichten klein sind, werden die Warteschlangen erfolgreich synchronisiert und wie erwartet fortgesetzt. Wenn sich die Datenmenge in einem Batch jedoch dem Speicherlimit des Knotens nähert, löst der Knoten einen Alarm mit hohem Speicher aus, der die Warteschlangen-Synchronisierung pausiert. Sie können die Speicherauslastung bestätigen, indem Sie die `RabbitMemUsed` und `RabbitMqMemLimit` [Metriken von Brokerknoten in CloudWatch](#). Die Synchronisierung kann erst abgeschlossen werden, wenn Nachrichten verbraucht oder gelöscht oder die Anzahl der Nachrichten im Stapel reduziert wird.

Note

Die Reduzierung der Stapelgröße der Warteschlangensynchronisierung kann zu einer höheren Anzahl von Replikationstransaktionen führen.

Um eine angehaltene Warteschlangensynchronisierung aufzulösen, führen Sie die Schritte in diesem Lernprogramm aus, in dem veranschaulicht wird, wie eine `ha-sync-batch-size`-Richtlinie angewendet wird, und starten Sie die Warteschlangen-Synchronisierung neu.

Themen

- [Voraussetzungen](#)
- [Schritt 1: Wenden Sie eine `ha-sync-batch-size` Richtlinie an](#)
- [Schritt 2: Starten Sie die Warteschlangen-Synchronisierung](#)
- [Nächste Schritte](#)
- [Zugehörige Ressourcen](#)

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 (regex) Muster.

Tags (Markierungen)	Lesen Sie regex	Konfigurieren von regex	REGEXP-Schreiben
administrator	.*	.*	.*

Weitere Informationen zum Erstellen von RabbitMQ-Benutzern und zum Verwalten von Benutzer-Tags und -Berechtigungen finden Sie unter [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 `sha-a11-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 Nummer aus der Dropdown-Liste.

 Note


Möglicherweise müssen Sie den Wert von `ha-sync-batch-size` basierend auf der Anzahl und Größe der nicht synchronisierten Nachrichten in Ihren Warteschlangen anpassen.

- **ha-mode=all**. Klicken Sie auf Zeichenfolge aus 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 auf Zeichenfolge aus 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 der Benutzerrichtlinien 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 die RabbitMQ Webkonsole URL. Dies ist der Broker-Endpunkt, den Sie in einer HTTP-Anforderung verwenden.
5. Öffnen Sie ein neues Terminal- oder Befehlszeilenfenster Ihrer Wahl.

- Um eine neue Broker-Richtlinie zu erstellen, geben Sie Folgendes ein `curl`-Befehl. Dieser Befehl nimmt an, dass eine Warteschlange auf der `/vhost`, der als `%2F` encodiert ist.

 Note

Ersetzen Sie den *Benutzernamen* und das *Passwort* durch Ihre Broker-Administratoranmeldeinformationen. Möglicherweise müssen Sie den Wert von `ha-sync-batch-size(100)` basierend auf der Anzahl und Größe der nicht synchronisierten Nachrichten in Ihren Warteschlangen anpassen. Ersetzen Sie den Broker-Endpunkt durch die URL, die Sie zuvor notiert haben.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \  
-d '{"pattern":".*", "priority":1, "definition":{"ha-sync-batch-size":100, "ha-  
mode":"all", "ha-sync-mode":"automatic"}}' \  
https://b-589c045f-f81n-4ab0-a89c-co62e1c32ef8.mq.us-west-2.amazonaws.com/api/  
policies/%2Fbatch-size-policy
```


- Um zu bestätigen, dass die neue Richtlinie den Benutzerrichtlinien Ihres Brokers hinzugefügt wird, geben Sie folgenden `curl`-Befehl, um alle Broker-Richtlinien aufzulisten.

```
curl -i -u username:password https://b-589c045f-f81n-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 Option Queues (Warteschlangen).
2. Klicken Sie auf die Seite Queues (Warteschlangen), und suchen Sie Ihre angehaltene Warteschlange unter Alle Warteschlangen. In der Funktionen-Spalte, sollte die Warteschlange den Namen der neuen Richtlinie auflisten, die Sie erstellt haben (z. B. `batch-size-policy`).
3. Um den Synchronisierungsprozess mit einer reduzierten Stapelgröße neu zu starten, wählen Sie Synchronisation neu starten.

Note

Wenn die Synchronisation angehalten wird und nicht erfolgreich abgeschlossen wird, versuchen Sie, den `ha-sync-batch-size`-Wert zu reduzieren und starten Sie die Warteschlangen-Synchronisierung erneut.

Nächste Schritte

- Sobald Ihre Warteschlange erfolgreich synchronisiert wurde, können Sie die Speichermenge überwachen, die Ihre RabbitMQ-Knoten verwenden, indem Sie die Amazon CloudWatch Metrik `RabbitMQMemUsed`. Sie können auch die `RabbitMQMemLimit`-Metrik, um das Speicherlimit eines Knotens zu überwachen. Weitere Informationen finden Sie unter [Zugreifen auf CloudWatch Metriken für Amazon MQ](#) und [Protokollierung und Überwachung von Amazon MQ für RabbitMQ Broker](#).
- Um eine angehaltene Warteschlangensynchronisierung zu verhindern, empfehlen wir, Warteschlangen kurz zu halten und Nachrichten zu verarbeiten. Für Workloads mit größeren Nachrichtengrößen empfehlen wir außerdem, Ihren Broker-Instance-Typ auf eine größere Instance-Größe mit mehr Speicher zu aktualisieren. Weitere Informationen zu Broker-Instance-Typen und zum Bearbeiten von Broker-Voreinstellungen finden Sie unter [Instance-Typen von Amazon MQ für RabbitMQ](#) und [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 unter [the section called "Standardeinstellungen für Broker"](#).

Zugehörige Ressourcen

- [UpdateBrokerInput](#)— Verwenden Sie diese Broker-Eigenschaft, um einen Broker-Instance-Typ mithilfe der Amazon MQ-API zu aktualisieren.
- [Parameter und Richtlinien](#)(RabbitMQ Server Documentation) — Erfahren Sie mehr über RabbitMQ-Parameter und -Richtlinien auf der RabbitMQ-Website.
- [RabbitMQ-Management HTTP-API](#)— Erfahren Sie mehr über die RabbitMQ-Management-API.

Best Practices für Amazon MQ for RabbitMQ

Verwenden Sie dies als Referenz, um schnell Empfehlungen zur Maximierung der Leistung und Minimierung der Durchsatzkosten bei der Arbeit mit RabbitMQ-Brokern auf Amazon MQ zu finden.

Important

Amazon MQ unterstützt nicht [Quorum-Warteschlangen](#). Das Aktivieren des Quorumwarteschlangen-Feature-Flags und das Erstellen von Quorumwarteschlangen führt zu Datenverlust.

Important

Derzeit unterstützt Amazon MQ keine [Streams](#) oder die Verwendung der strukturierten Protokollierung in JSON, die in RabbitMQ 3.9.x eingeführt wurde.

Important

Amazon MQ für RabbitMQ unterstützt den Benutzernamen „Gast“ nicht und löscht das Standard-Gastkonto, wenn Sie einen neuen Broker erstellen. Amazon MQ löscht auch regelmäßig alle vom Kunden erstellten Konten namens „Gast“.

Themen

- [Lazy-Warteschlangen aktivieren](#)
- [Verwenden Sie persistente und dauerhafte Warteschlangen](#)

- [Warteschlangen kurz halten](#)
- [Bestätigung und Bestätigung konfigurieren](#)
- [Konfigurieren des Vorabrufs](#)
- [Konfigurieren von Celery](#)
- [Automatische Wiederherstellung nach Netzwerkausfällen](#)
- [Aktivieren von Classic Queue v2 für Ihren RabbitMQ-Broker](#)

Lazy-Warteschlangen aktivieren

Wenn Sie mit sehr langen Warteschlangen arbeiten, die große Mengen von Nachrichten verarbeiten, aktivieren Sie Lazy Warteschlangen, um die Gesamtleistung Ihres Brokers zu verbessern.

Das Standardverhalten von RabbitMQ besteht darin, Nachrichten im Speicher zwischenspeichern und sie nur dann auf die Festplatte zu verschieben, wenn der Broker mehr verfügbaren Speicher benötigt. Dieser Vorgang des Verschiebens von Nachrichten vom Speicher auf den Datenträger kann Zeit in Anspruch nehmen und verhindert, dass die Warteschlange Nachrichten verarbeitet werden. Das Aktivieren von Lazy-Queues kann sich erheblich auf die Beschleunigung des Verschiebens von Nachrichten auf die Festplatte auswirken, da Lazy Queues Nachrichten so schnell wie möglich auf der Festplatte speichern, wodurch weniger Nachrichten im Speicher zwischengespeichert werden.

Sie können Lazy-Queues aktivieren, indem Sie die `queue.declare`-Argumente zum Zeitpunkt der Deklaration oder durch Konfigurieren einer Richtlinie über die RabbitMQ-Verwaltungskonsole. Das folgende Beispiel veranschaulicht das Deklarieren einer Lazy-Queue mit der RabbitMQ Java-Client-Bibliothek.

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

Note

Durch Aktivieren von Lazy Queues können Festplatten-I/O-Operationen erhöht werden.

Verwenden Sie persistente und dauerhafte Warteschlangen

Persistente Nachrichten können dazu beitragen, Datenverlust in Situationen zu verhindern, in denen ein Broker abstürzt oder neu gestartet wird. Persistente Nachrichten werden auf die Festplatte geschrieben, sobald sie eintreffen. Im Gegensatz zu Lazy Queues werden jedoch persistente Nachrichten sowohl im Arbeitsspeicher als auch auf der Festplatte zwischengespeichert, es sei denn, der Broker benötigt mehr Speicher. In Fällen, in denen mehr Speicher benötigt wird, werden Nachrichten vom RabbitMQ-Broker-Mechanismus aus dem Speicher entfernt, der das Speichern von Nachrichten auf der Festplatte verwaltet, allgemein als Sitzungspersistenz 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.

```
boolean durable = true;
channel.queueDeclare("my_queue", durable, false, false, null);
```

Nachdem Sie die Warteschlange als dauerhaft konfiguriert haben, können Sie eine dauerhafte Nachricht an Ihre Warteschlange senden, indem Sie `MessageProperties` auf `PERSISTENT_TEXT_PLAIN` stellen, wie im folgenden Beispiel gezeigt.

```
import com.rabbitmq.client.MessageProperties;

channel.basicPublish("", "my_queue",
    MessageProperties.PERSISTENT_TEXT_PLAIN,
    message.getBytes());
```

Warteschlangen kurz halten

In Clusterbereitstellungen können Warteschlangen mit einer großen Anzahl von Nachrichten zu einer Überlastung der Ressourcen führen. Wenn ein Broker übermäßig ausgelastet ist, kann ein Neustart eines Amazon MQ für RabbitMQ Brokers zu weiteren Leistungseinbußen führen. Wenn ein Neustart durchgeführt wird, reagieren überlastete Broker möglicherweise nicht im `REBOOT_IN_PROGRESS` Zustand.

Während dem [Wartungsfenster](#) führt Amazon MQ alle Wartungsarbeiten jeweils einen Knoten aus, um sicherzustellen, dass der Broker betriebsbereit bleibt. Daher müssen Warteschlangen

möglicherweise synchronisiert werden, wenn jeder Knoten den Vorgang fortsetzt. Während der 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 bestätigen, indem Sie die Metriken `RabbitMemUsed` und `RabbitMqMemLimit` Broker-Knotens in [CloudWatch](#) vergleichen. 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 auf [Reduzierung der Batch-Größe der Warteschlangensynchronisation](#) anwenden.

Warning

Starten Sie einen Broker nicht neu, der hoch auf Ressourcen ausgeführt wird.

Wenn Sie einen Broker neu starten, wenn die Warteschlangensynchronisierung angehalten wird, initiiert der Broker den Synchronisierungsprozess neu. Dadurch kann die Broker-Ressourcen weiter verschlechtern, wenn Nachrichten vom Speicher in den Knotenspeicher übertragen werden, und dazu führen, dass der Broker im `REBOOT_IN_PROGRESS` Zustand unansprechbar wird.

Bestätigung und Bestätigung konfigurieren

Wenn eine Client-Anwendung die Bestätigung der Zustellung und des Verbrauchs von Nachrichten an den Broker sendet, wird sie als `Verbraucherbestätigung` bezeichnet. In ähnlicher Weise wird der Prozess der Bestätigung an einen Herausgeber als `Verlag bestätigen` bezeichnet. Sowohl die Bestätigung als auch die Bestätigung sind unerlässlich, um die Datensicherheit bei der Arbeit mit RabbitMQ-Brokern zu gewährleisten.

Die Bestätigung der Verbraucherzustellung wird in der Regel in der Clientanwendung konfiguriert. Bei der Arbeit mit AMQP 0-9-1 kann die Bestätigung aktiviert werden, indem die `basic.consume` konfiguriert oder eine Nachricht mit der `basic.code`-Methode abgerufen wird.

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 vorabruf, indem Sie [Vorabruf](#)-Einstellungen für eine Client-Anwendung konfigurieren.

Konfigurieren des Vorabrufs

Sie können den RabbitMQ-Prefetch-Wert verwenden, um zu optimieren, wie Ihre Verbraucher Nachrichten konsumieren. RabbitMQ implementiert den 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.

Konfigurieren von Celery

Python Celery sendet viele unnötige Nachrichten, die das Auffinden und Verarbeiten nützlicher Informationen erschweren können. Geben Sie den folgenden Befehl ein, um das Rauschen zu reduzieren und die Verarbeitung zu vereinfachen:

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

Automatische Wiederherstellung nach Netzwerkausfällen

Es wird empfohlen, die automatische Netzwerk Wiederherstellung immer zu aktivieren, um erhebliche Ausfallzeiten zu vermeiden, wenn Clientverbindungen zu RabbitMQ-Knoten fehlschlagen. Die RabbitMQ Java-Client-Bibliothek unterstützt standardmäßig automatische Netzwerk Wiederherstellung, beginnend mit Version `4.0.0`.

Die automatische Verbindungswiederherstellung wird ausgelöst, wenn eine nicht behandelte Ausnahme in der I/O-Schleife der Verbindung ausgelöst wird, wenn ein Timeout für den Socket-Lesevorgang erkannt wird oder wenn der Server eine [Herzschlag](#) 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 Netzwerk Wiederherstellung nicht aktiviert oder ausgelöst.

Aktivieren von Classic Queue v2 für Ihren RabbitMQ-Broker

Wir empfehlen, Classic Queue v2 (CQv2) für die Broker-Engine-Versionen 3.10 oder höher zu aktivieren, um die Leistung beispielsweise wie folgt zu verbessern:

- Reduzierung der Variation bei der Speichernutzung
- Verringerung des Speicherverbrauchs für die meisten Workloads
- Verbesserung der Verbraucherzustellung
- Erhöhung des Durchsatzes für Workloads, bei denen Verbraucher mit Produzenten Schritt halten

Um CQv2 verwenden zu können, müssen Sie zuerst das `classic_mirrored_queue_version` Feature-Flag aktivieren. Weitere Informationen zu Feature-Flags finden Sie unter [So aktivieren Sie Feature-Flags](#).

Um von CQv1 zu CQv2 zu migrieren, müssen Sie eine neue Warteschlangenrichtlinie erstellen oder eine vorhandene Warteschlangenrichtlinie bearbeiten, wobei die `queue-version` Richtlinienschlüsseldefinition auf festgelegt ist². Weitere Informationen zum Anwenden von Richtlinien finden Sie unter [Richtlinien](#). Weitere Informationen zur Aktivierung von CQv2 mit einer Warteschlangenrichtlinie finden Sie unter [Classic Queues](#) in der RabbitMQ-Dokumentation.

Die Migration von CQv1 zu CQv2 kann für Warteschlangen unter hoher Last speicherintensiv sein. Wir empfehlen, vor Beginn der Migration unsere anderen [bewährten Methoden zur Leistungsoptimierung](#) zu befolgen.

Wenn Sie eine Warteschlangenrichtlinie verwenden, führt das Löschen dieser Richtlinie dazu, dass die CQv2-Warteschlangen wieder auf CQv1 heruntergestuft werden. Das Herunterstufen von CQv2-Warteschlangen auf CQv1 wird nicht empfohlen, da RabbitMQ die Darstellung der Warteschlange auf der Festplatte konvertiert. Dies kann bei Warteschlangen mit großer Tiefe speicherintensiv und zeitaufwändig sein.

Kontingente in Amazon MQ für RabbitMQ

In diesem Thema werden Kontingente innerhalb von Amazon MQ aufgeführt. Viele der folgenden Kontingente 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 zum Anzeigen der aktuellen Verbindungslimits in Amazon CloudWatch finden Sie unter [Überwachung von Amazon-MQ-Brokern mit Amazon CloudWatch](#).

Themen

- [Broker](#)
- [Datenspeicherung](#)
- [API-Drosselung](#)

Broker

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ für RabbitMQ-Broker aufgeführt.

Limit	Beschreibung
Broker-Name	<ul style="list-style-type: none">• Muss in der Brokerregion und in Ihrem AWS-Konto eindeutig sein• Er muss 1–50 Zeichen umfassen.

Limit	Beschreibung
	<ul style="list-style-type: none"> • Es darf nur Zeichen aus den darstellbaren ASCII-Zeichen enthalten. • Er darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten.
Anzahl der Broker, pro Region	50
Sicherheitsgruppen pro Broker	5
In CloudWatch überwachte ActiveMQ-Ziele (Warteschlangen und Themen)	CloudWatch überwacht nur die ersten 1 000 Ziele.
In CloudWatch überwachte RabbitMQ-Ziele (Warteschlangen)	CloudWatch überwacht nur die ersten 500 Ziele, sortiert nach Anzahl der Verbraucher.
Tags pro Broker	50

Datenspeicherung

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ für RabbitMQ-Datenspeicher aufgeführt.

Limit	Beschreibung
Speicherkapazität pro kleinerem Broker	20 GB für mq.*.micro Instance-Typ-Broker. Weitere Informationen zu Amazon MQ Instance-Typen finden Sie unter Broker instance types .
Speicherkapazität pro Broker	200 GB für mq.*.*large Instance-Typ-Broker. Weitere Informationen zu Amazon MQ Instance-Typen finden Sie unter Broker instance types .

API-Drosselung

Die folgenden Drosselungskontingente werden pro AWS-Konto über alle Amazon MQ-APIs aggregiert, um die Service-Bandbreite beizubehalten. Weitere Informationen zu Amazon MQ -APIs finden Sie im [Amazon MQ REST-API-Referenz](#).

 **Important**

Diese Kontingente gelten nicht für Amazon MQ for ActiveMQ oder Amazon MQ für RabbitMQ Broker Messaging-APIs. Amazon MQ drosselt z. B. nicht das Senden und Empfangen von Nachrichten.

API-Aufrust-Limit	API-Ratenlimits
100	15

Sicherheit in Amazon MQ

Cloud-Sicherheit hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die eingerichtet wurden, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das [Modell der übergreifenden Verantwortlichkeit](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud selbst – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der [AWS-Compliance-Programme](#) regelmäßig. Informationen zu den Compliance-Programmen, die für Amazon MQ gelten, finden Sie unter [Im Rahmen des Compliance-Programms zugelassene AWS-Services](#).
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation zeigt Ihnen, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Amazon MQ einsetzen können. Die folgenden Themen veranschaulichen, wie Sie Amazon MQ zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie lernen auch, wie Sie andere AWS-Services verwenden, die Sie bei der Überwachung und Sicherung Ihrer Amazon MQ-Ressourcen unterstützen.

Themen

- [Datenschutz in Amazon MQ](#)
- [Identitäts- und Zugriffsverwaltung für Amazon MQ](#)
- [Compliance-Validierung für Amazon MQ](#)
- [Ausfallsicherheit bei Amazon MQ](#)
- [Infrastruktursicherheit in Amazon MQ](#)
- [Best Practices für die Sicherheit in Amazon MQ](#)

Datenschutz in Amazon MQ

Das AWS [Modell der übergreifenden Verantwortlichkeit](#) gilt für den Datenschutz in Amazon MQ. Wie in diesem Modell beschrieben, ist AWS verantwortlich für den Schutz der globalen Infrastruktur, in der die gesamte AWS Cloud ausgeführt wird. 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, AWS-Konto-Anmeldeinformationen zu schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden zu schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit AWS-Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder über eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

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, API, AWS CLI 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. Brokernamen und Benutzernamen sind für andere AWS-Services zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

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 wird in AWS Key Management Service (KMS) integriert, um transparente Server-seitige Verschlüsselung zu ermöglichen. Amazon MQ verschlüsselt Ihre Daten im Ruhezustand stets.

Wenn Sie einen Broker für Amazon MQ for ActiveMQ oder einen Broker für Amazon MQ for RabbitMQ erstellen können Sie den AWS KMS key angeben mit dem Amazon MQ Ihre ruhenden Daten verschlüsseln soll. Wenn Sie keinen KMS-Schlüssel angeben, erstellt Amazon MQ einen KMS-Schlüssel im Besitz von AWS 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.
- Von AWS verwalteter KMS-Schlüssel: Der von AWS verwaltete KMS-Schlüssel (aws/mq) ist ein KMS-Schlüssel in Ihrem Konto, der von Amazon MQ für Sie 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. Stattdessen empfehlen wir, den Broker zu löschen, wenn Sie Zugriffsrechte widerrufen müssen.
- Bei Brokern für Amazon MQ for ActiveMQ, die Amazon Elastic File System (EFS) zum Speichern von Nachrichtendaten verwenden, gilt: wenn Sie Amazon EFS die Berechtigung zum Verwenden der KMS-Schlüssel in Ihrem Konto entziehen, erfolgt dies nicht sofort.
- 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.
- Das Deaktivieren eines Schlüssels oder das Widerrufen einer Berechtigung erfolgt nicht sofort.

Wenn Sie einen [Single-Instance-Broker](#) mit einem KMS-Schlüssel für RabbitMQ erstellen, werden 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.

AWS CloudTrail-Protokolleintrag CreateGrant: Single-Instance-Broker

mq_grant

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
    "accountId": "111122223333",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "userName": "AmazonMqConsole"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2023-02-23T18:59:10Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "invokedBy": "mq.amazonaws.com"
},
"eventTime": "2018-06-28T22:23:46Z",
"eventSource": "amazonmq.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "203.0.113.0",
"userAgent": "PostmanRuntime/7.1.5",
"requestParameters": {
  "granteePrincipal": "mq.amazonaws.com",
  "keyId": "arn:aws:kms:us-east-1:316438333700:key/bdbe42ae-f825-4e78-a8a1-828d411c4be2",
  "retiringPrincipal": "mq.amazonaws.com",
  "operations": [
    "CreateGrant",
    "Decrypt",
    "GenerateDataKeyWithoutPlaintext",
```

```

        "ReEncryptFrom",
        "ReEncryptTo",
        "DescribeKey"
    ]
},
"responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}

```

EBS grant creation

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.

AWS CloudTrail-Protokolleintrag CreateGrant: 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 keys](#) Sie im AWS Key Management ServiceEntwicklerhandbuch.

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 überträgt nur die VPC Ihres Brokers. 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 zur Daten während der Übertragung finden Sie unter [Verschlüsseln von Daten in Ruhe und während der Übertragung](#).

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 über 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_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA

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) ist ein AWS-Service, mit dem ein Administrator den Zugriff auf AWS-Ressourcen sicher steuern kann. IAM-Administratoren steuern, wer authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) ist, Amazon MQ-Ressourcen zu nutzen. IAM ist ein AWS-Service, den Sie ohne zusätzliche Kosten verwenden können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Funktionsweise von Amazon MQ mit IAM](#)
- [Beispiele für identitätsbasierte Amazon MQ-Richtlinien](#)
- [API-Authentifizierung und Amazon MQ-Autorisierung für](#)

- [AWSverwaltete Richtlinien für Amazon MQ](#)
- [Verwendung von serviceverknüpften Rollen für Amazon MQ](#)
- [Fehlerbehebung für Amazon MQ-Identität und -Zugriff](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, unterscheidet sich je nach Ihrer Arbeit in Amazon MQ.

Service-Benutzer – Wenn Sie den Amazon MQ-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie zur Ausführung von Aufgaben weitere Amazon MQ-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Featuresweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie auf eine Funktion in Amazon MQ nicht zugreifen können, siehe [Fehlerbehebung für Amazon MQ-Identität und -Zugriff](#).

Service-Administrator – Wenn Sie in Ihrem Unternehmen für die Amazon MQ-Ressourcen zuständig sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon MQ. Ihre Aufgabe besteht darin, zu bestimmen, auf welche Amazon-MQ-Funktionen und -Ressourcen Ihre Service-Nutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit Amazon MQ verwenden kann, finden Sie unter [Funktionsweise von Amazon MQ mit IAM](#).

IAM-Administrator – Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Amazon MQ verfassen können. Beispiele für identitätsbasierte Amazon MQ-Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Amazon MQ-Richtlinien](#).

Authentifizierung mit Identitäten

Die Authentifizierung ist die Art und Weise, wie Sie sich mit Ihren Anmeldeinformationen bei AWS anmelden. Die Authentifizierung (Anmeldung bei AWS) muss als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle erfolgen.

Sie können sich bei AWS als Verbundidentität mit Anmeldeinformationen anmelden, die über eine Identitätsquelle bereitgestellt werden. Benutzer von AWS IAM Identity Center. (IAM Identity Center),

die Single-Sign-on-Authentifizierung Ihres Unternehmens und Anmeldeinformationen für Google oder Facebook sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie auf AWS mithilfe des Verbunds zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich bei der AWS Management Console oder beim AWS-Zugriffportal anmelden. Weitere Informationen zum Anmelden bei AWS finden Sie unter [Anmelden bei Ihrem AWS-Konto](#) im Benutzerhandbuch von AWS-Anmeldung.

Bei programmgesteuerten Zugriff auf AWS bietet AWS ein Software Development Kit (SDK) und eine Command Line Interface (CLI, Befehlszeilenschnittstelle) zum kryptographischen Signieren Ihrer Anfragen mit Ihren Anmeldeinformationen. Wenn Sie keine AWS-Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode zum eigenen Signieren von Anforderungen finden Sie unter [Signieren von AWS-API-Anforderungen](#) im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise die Verwendung von Multi-Faktor Authentifizierung (MFA), um die Sicherheit Ihres Kontos zu verbessern. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center-Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

AWS-Konto-Stammbenutzer

Wenn Sie ein AWS-Konto neu erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen des Kontos hat. Diese Identität wird als AWS-Konto-Root-Benutzer bezeichnet. Für den Zugriff auf den Root-Benutzer müssen Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, die zur Erstellung des Kontos verwendet wurden. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Benutzer und Gruppen

Ein [IAM-Benutzer](#) ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder eine einzelne Anwendung. Wenn möglich, empfehlen wir,

temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

IAM roles (IAM-Rollen)

Eine [IAM-Rolle](#) ist eine Identität in Ihrem AWS-Konto mit spezifischen Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der AWS Management Console übernehmen, indem Sie [Rollen wechseln](#). Sie können eine Rolle annehmen, indem Sie eine AWS CLI oder AWS-API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wenn eine Verbundidentität authentifiziert wird, wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center-Benutzerhandbuch.

- Temporäre IAM-Benutzerberechtigungen – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- Kontoübergreifender Zugriff – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. In einigen AWS-Services können Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff – Einige AWS-Services verwenden Features in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Service kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- Forward access sessions (FAS) – Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle zum Ausführen von Aktionen in AWS verwenden, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen AWS-Service aufruft, in Kombination mit der Anforderung an den AWS-Service, Anforderungen an nachgelagerte Services zu stellen. FAS-Anforderungen werden nur dann gestellt, wenn ein Service eine Anforderung erhält, die Interaktionen mit anderen AWS-Services oder Ressourcen erfordert, um abgeschlossen werden zu können. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Forward Access Sessions \(FAS\)](#).
- Servicerolle – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- Serviceverbundene Rolle – Eine serviceverbundene Rolle ist ein Typ von Servicerolle, die mit einem AWS-Service verknüpft ist. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem AWS-Konto angezeigt und gehören zum Service. Ein IAM-Administrator kann die Berechtigungen für serviceverbundene Rollen anzeigen, aber nicht bearbeiten.

- Anwendungen in Amazon EC2 – Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und AWS CLI- oder AWS-API-Anforderungen durchführen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle einer EC2-Instance zuzuweisen und die Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Für die Zugriffssteuerung in AWS erstellen Sie Richtlinien und weisen diese den AWS-Identitäten oder -Ressourcen zu. Eine Richtlinie ist ein Objekt in AWS, das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS-JSON-Richtlinien festlegen, wer zum Zugriff auf was berechtigt ist. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole` -Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Benutzerinformationen über die AWS Management Console, die AWS CLI oder die AWS -API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem AWS-Konto anfügen können. Verwaltete Richtlinien umfassen von AWS verwaltete und von Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien 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. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder AWS-Services umfassen.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Service. Sie können verwaltete AWS-Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3, AWS WAF und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. Weitere Informationen zu ACLs finden Sie unter [Zugriffssteuerungsliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Ein ausdrückliches Ablehnen in einer dieser Richtlinien setzt das Zulassen außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Service-Kontrollrichtlinien (SCPs)** – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OE) in AWS Organizations angeben. AWS Organizations ist ein Service für die Gruppierung und zentrale Verwaltung mehrerer AWS-Konten Ihres Unternehmens. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. SCPs schränken Berechtigungen für Entitäten in Mitgliedskonten einschließlich des jeweiligen Root-Benutzer des AWS-Kontos ein. Weitere Informationen zu Organizations und SCPs finden Sie unter [Featuresweise von SCPs](#) im AWS Organizations-Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen dazu, wie AWS die Zulässigkeit einer Anforderung ermittelt, wenn mehrere Richtlinientypen beteiligt sind, finden Sie unter [Logik für die Richtlinienauswertung](#) 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 Überblick über das Zusammenwirken von Amazon MQ und anderen AWS-Services mit IAM finden Sie unter [AWS-Services, die mit IAM](#) funktionieren im IAM-Benutzerhandbuch.

Amazon MQ verwendet IAM zum Erstellen, Aktualisieren und Löschen von Operationen, jedoch eine native ActiveMQ-Authentifizierung für Broker. Weitere Informationen finden Sie unter [Integration von ActiveMQ-Brokern in LDAP](#).

Themen

- [Identitätsbasierte Amazon MQ-Richtlinien](#)
- [Ressourcenbasierte Amazon MQ -Richtlinien](#)
- [Autorisierung auf der Basis von Amazon MQ-Tags](#)
- [Amazon MQ 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 mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörige AWS-API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Richtlinienaktionen in Amazon MQ verwenden das folgende Präfix vor der Aktion: `mq:`. Um einem Benutzer beispielsweise die Berechtigung zum Ausführen einer Amazon MQ-Instance mit der `AmazonMQCreateBroker`-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 unter [Von Amazon MQ definierte Aktionen](#) im IAM-Benutzerhandbuch.

Ressourcen

Administratoren können mit AWS-JSON-Richtlinien festlegen, welche Personen zum Zugriff auf welche Ressourcen berechtigt sind. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource`- oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"

```

In Amazon MQ wird der primäre AWS-Ressourcen sind ein Amazon MQ Nachrichtenbroker und seine Konfiguration. Diesen Ressourcen und Bedingungen sind eindeutigen Amazon MQ-Ressourcennamen (ARN) zugeordnet, wie in der folgenden Tabelle gezeigt.

Ressourc entypen	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-Ressourcennamen \(ARNs\) und AWS-Service-Namespaces](#).

Um beispielsweise den Broker namens MyBroker mit brokerId b-1234a5b6-78cd-901e-2fgh-3i45j6k17819 in Ihrer Anweisung anzugeben, verwenden Sie den folgenden ARN:

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

```

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 die ARNs durch Kommata voneinander.

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

Eine Liste der Amazon MQ-Ressourcentypen und ihrer 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 festlegen, wer zum Zugriff auf was berechtigt ist. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, wertet AWS die Bedingung mittels einer logischen OR-Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und servicespezifische Bedingungsschlüssel. Eine Liste aller globalen AWS-Bedingungsschlüssel finden Sie unter [Globale AWS-Bedingungskontextschlüssel](#) 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 oder [Bedingungsschlüssel für Amazon MQ](#) im IAM-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
aws:ResourceTag/\${TagKey}	Filtert Aktionen basierend auf den Tags, die der Ressource zugeordnet sind.	Zeichenfolge
aws: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 [Beispiele für identitätsbasierte Amazon MQ-Richtlinien](#).

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 [Bedingungelement](#) einer Richtlinie Informationen an, indem Sie die Bedingungsschlüssel `mq:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder `aws:TagKeys` verwenden.

Amazon MQ unterstützt Richtlinien, die auf Tags basieren. Sie können z. B. den Zugriff auf alle Amazon MQ-Ressourcen einschränken, die ein Tag mit dem Schlüssel `environment` und dem Wert `production` enthalten:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "mq:DeleteBroker",
        "mq:RebootBroker",
        "mq>DeleteTags"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/environment": "production"
        }
      }
    }
  ]
}
```

Mit dieser Richtlinie Deny Sie die Möglichkeit, einen Amazon-MQ-Broker zu löschen oder neu zu starten, der das Tag `environment/production` enthält.

Weitere Informationen zum Markieren finden Sie unter:

- [Markieren von Ressourcen](#)
- [Zugriffssteuerung mit IAM-Tags](#)

Amazon MQ IAM-Rollen

Eine [IAM-Rolle](#) ist eine Entität in Ihrem AWS-Konto mit spezifischen Berechtigungen.

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. Temporäre Sicherheitsanmeldeinformationen erhalten Sie durch Aufrufen von AWS STS-API-Vorgängen wie [AssumeRole](#) oder [GetFederationToken](#).

Amazon MQ unterstützt die Verwendung temporärer Anmeldeinformationen.

Servicerollen

Diese Funktion ermöglicht einem Service das Annehmen einer [Servicerolle](#) in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Services, 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 Services beeinträchtigen.

Amazon MQ unterstützt Servicerollen.

Beispiele für identitätsbasierte Amazon MQ-Richtlinien

Benutzer und Rollen besitzen standardmäßig keine Berechtigungen zum Erstellen oder Ändern von Amazon-MQ-Ressourcen. Sie können auch keine Aufgaben ausführen, die die AWS Management Console-, AWS CLI- oder AWS-API benutzen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den IAM-Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von Richtlinien auf der JSON-Registerkarte](#) im IAM-Benutzerhandbuch.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der Amazon MQ-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien können festlegen, ob jemand Amazon-MQ-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder daraus löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine

Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie AWS-kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS-verwaltete Richtlinien](#) oder [AWS-verwaltete Richtlinien für Auftragsfunktionen](#) im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS-Service, wie beispielsweise AWS CloudFormation, verwendet werden. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Bedarf einer Multi-Faktor-Authentifizierung (MFA) – Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der Amazon MQ-Konsole

Um auf die Amazon MQ-Konsole zugreifen zu können, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen Ihnen das Auflisten und Anzeigen von Details zu Amazon MQ-Ressourcen in Ihrem AWS-Konto gestatten. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (IAM-Benutzer oder -Rollen) mit dieser Richtlinie.

Um sicherzustellen, dass diese Entitäten dennoch die Amazon MQ-Konsole verwenden können, fügen Sie den Entitäten auch die folgende von AWS verwaltete Richtlinie an. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

```
AmazonMQReadOnlyAccess
```

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die den API-Operation entsprechen, die Sie ausführen möchten.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    }
  ]
}
```

```
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

API-Authentifizierung und Amazon MQ-Autorisierung für

Amazon MQ verwendet die standardmäßige AWS-Anforderungssignatur für die API-Authentifizierung. Weitere Informationen dazu finden Sie unter [Signieren von AWS API-Anforderungen](#) im Allgemeinen AWS-Referenz.

Note

Derzeit unterstützt Amazon MQ keine IAM-Authentifizierung unter Verwendung ressourcenbasierter Berechtigungen oder ressourcenbasierter Richtlinien.

Um AWS-Benutzer für die Arbeit mit Brokern, Konfigurationen und Benutzern zu autorisieren, müssen Sie die IAM-Richtlinienberechtigungen bearbeiten.

Themen

- [Erforderliche IAM-Berechtigungen zum Erstellen eines Amazon MQ-Brokers](#)
- [Amazon MQ REST API-Berechtigungen-Referenz](#)
- [Unterstützte Berechtigungen auf Ressourcenebene für Amazon MQ-API-Aktionen](#)

Erforderliche IAM-Berechtigungen zum Erstellen eines Amazon MQ-Brokers

Um einen Broker zu erstellen, müssen Sie entweder die AmazonMQFullAccess-IAM-Richtlinie verwenden oder die folgenden EC2-Berechtigungen in Ihre IAM-Richtlinie aufnehmen.

Die folgende benutzerdefinierte Richtlinie besteht aus zwei Anweisungen (eine bedingte), die Berechtigungen zum Ändern der Ressourcen erteilen, die Amazon MQ benötigt, um einen ActiveMQ-Broker zu erstellen.

Wichtig

- Die `ec2:CreateNetworkInterface`-Aktion ist erforderlich, damit Amazon MQ eine Elastic Network-Schnittstelle (Elastic Network Interface, ENI) in Ihrem Konto für Sie erstellen kann.
- Die `ec2:CreateNetworkInterfacePermission`-Aktion erlaubt es Amazon MQ, die ENI an einen ActiveMQ-Broker anzufügen.
- Der `ec2:AuthorizedService`-Bedingungsschlüssel stellt sicher, dass ENI-Berechtigungen nur Amazon MQ-Service-Konten gewährt werden.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "mq:*",
      "ec2:CreateNetworkInterface",
      "ec2>DeleteNetworkInterface",
      "ec2:DetachNetworkInterface",
      "ec2:DescribeInternetGateways",
      "ec2:DescribeNetworkInterfaces",
      "ec2:DescribeRouteTables",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeSubnets",
      "ec2:DescribeVpcs"
    ],
    "Effect": "Allow",
    "Resource": "*"
  }], {
    "Action": [
      "ec2:CreateNetworkInterfacePermission",
```

```

        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfacePermissions"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "ec2:AuthorizedService": "mq.amazonaws.com"
        }
    }
}
]]
}

```

Weitere Informationen finden Sie unter [Schritt 2: Erstellen eines Benutzers und Abrufen Ihrer AWS-Anmeldeinformationen](#) und [Verändern oder löschen Sie auf keinen Fall die Amazon MQ Elastic Network-Schnittstelle](#).

Amazon MQ REST API-Berechtigungen–Referenz

Die folgende Tabelle listet Amazon MQ REST-APIs und die entsprechenden IAM-Berechtigungen auf.

Amazon MQ REST-APIs und erforderliche Berechtigungen

Amazon MQ REST APIs	Erforderliche Berechtigungen
CreateBroker	mq:CreateBroker
CreateConfiguration	mq:CreateConfiguration
CreateTags	mq:CreateTags
CreateUser	mq:CreateUser
DeleteBroker	mq>DeleteBroker
DeleteUser	mq>DeleteUser
DescribeBroker	mq:DescribeBroker
DescribeConfiguration	mq:DescribeConfiguration

Amazon MQ REST APIs	Erforderliche Berechtigungen
DescribeConfigurationRevision	mq:DescribeConfigurationRevision
DescribeUser	mq:DescribeUser
ListBrokers	mq:ListBrokers
ListConfigurationRevisions	mq:ListConfigurationRevisions
ListConfigurations	mq:ListConfigurations
ListTags	mq:ListTags
ListUsers	mq:ListUsers
RebootBroker	mq:RebootBroker
UpdateBroker	mq:UpdateBroker
UpdateConfiguration	mq:UpdateConfiguration
UpdateUser	mq:UpdateUser

Unterstützte Berechtigungen auf Ressourcenebene für Amazon MQ-API-Aktionen

Berechtigungen auf Ressourcenebene bedeutet, dass Sie angeben können, für welche Ressourcen die Benutzer Aktionen ausführen dürfen. Amazon MQ unterstützt teilweise Berechtigungen auf Ressourcenebene. Bei bestimmten Amazon MQ-Aktionen können Sie kontrollieren, wann die Benutzer diese Aktionen verwenden dürfen. Dies basiert auf Bedingungen, die erfüllt sein müssen, oder auf bestimmten Ressourcen, die von den Benutzern verwendet werden dürfen.

In der folgenden Tabelle werden die Amazon MQ-API-Aktionen aufgeführt, die Berechtigungen auf Ressourcenebene derzeit unterstützen, sowie die unterstützten Ressourcen, Ressourcen-ARNs und Bedingungsschlüssel für jede Aktion.

Important

Falls eine Amazon MQ-API-Aktion nicht in dieser Tabelle genannt wird, unterstützt sie keine Berechtigungen auf Ressourcenebene. Wenn eine Amazon MQ-API-Aktion Berechtigungen

auf Ressourcenebene nicht unterstützt, können Sie den Benutzern die Berechtigung zur Verwendung dieser Aktion erteilen, müssen aber für das Ressourcenelement in der Richtlinienanweisung ein Sternchen * als Platzhalterzeichen einfügen.

API-Aktion	Ressourcentypen (*erforderlich)
CreateConfiguration	Konfigurationen*
CreateTags	Broker , Konfigurationen
CreateUser	Broker*
DeleteBroker	Broker*
DeleteUser	Broker*
DescribeBroker	Broker*
DescribeConfiguration	Konfigurationen*
DescribeConfigurationRevision	Konfigurationen*
DescribeUser	Broker*
ListConfigurationRevisions	Konfigurationen*
ListConfigurationRevisions	Konfigurationen*
ListTags	Broker , Konfigurationen
ListUsers	Broker*
RebootBroker	Broker*
UpdateBroker	Broker*

API-Aktion	Ressourcentypen (*erforderlich)
UpdateConfiguration	Konfigurationen*
UpdateUser	Broker*

AWSverwaltete Richtlinien für Amazon MQ

Eine von AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von AWS erstellt und verwaltet wird. Von AWS verwaltete Richtlinien stellen Berechtigungen für viele häufige Anwendungsfälle bereit, damit Sie beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS-verwaltete Richtlinien möglicherweise nicht die geringsten Berechtigungen für Ihre spezifischen Anwendungsfälle gewähren, da sie für alle AWS-Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [kundenverwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Die Berechtigungen, die in den von AWS verwalteten Richtlinien definiert sind, können nicht geändert werden. Wenn AWS Berechtigungen aktualisiert, die in einer von AWS verwalteten Richtlinie definiert werden, wirkt sich das Update auf alle Prinzipalidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert am wahrscheinlichsten eine von AWS verwaltete Richtlinie, wenn ein neuer AWS-Service gestartet wird oder neue API-Operationen für bestehende Services verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

AWSVerwaltete Richtlinie: AmazonMQServiceRolePolicy

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 unter [the section called "Serviceverknüpfte Rollenberechtigungen für Amazon MQ"](#).

Änderungen von Amazon MQ an von AWS verwalteten Richtlinien

Anzeigen von Details zu Aktualisierungen für AWS-verwaltete Richtlinien für Amazon MQ, seit dieser Dienst mit der Verfolgung dieser Änderungen im März 2021 begonnen hat. Um automatische Warnungen über Änderungen an dieser Seite erhalten, abonnieren Sie den RSS-Feed auf der Amazon MQ-[Dokumentverlauf](#)-Seite.

Änderung	Beschreibung	Datum
Amazon MQ hat mit der Verfolgung von Änderungen begonnen	Amazon MQ hat mit der Verfolgung von Änderungen für seine AWS-verwaltete Richtlinien begonnen.	5. Mai 2021

Verwendung von serviceverknüpften Rollen für Amazon MQ

Amazon MQ verwendet AWS Identity and Access Management (IAM) [serviceverknüpfte Rollen](#). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Amazon MQ verknüpft ist. Serviceverknüpfte Rollen werden von Amazon MQ vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle macht die Einrichtung von Amazon MQ einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon MQ definiert die Berechtigungen seiner serviceverknüpften Rollen, und sofern nicht anders definiert, kann nur Amazon MQ seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Amazon MQ-Ressourcen, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS-Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Yes (Ja) in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Amazon MQ

Amazon MQ verwendet die serviceverknüpfte Rolle namens `AWSServiceRoleForAmazonMQ` – Amazon MQ verwendet diese serviceverknüpfte Rolle, um AWS-Dienstleistungen in Ihrem Namen.

Die servicegebundene Rolle `AWSServiceRoleForRDS` vertraut den folgenden Services, die diese Rolle übernehmen:

- `mq.amazonaws.com`

Amazon MQ verwendet die Berechtigungsrichtlinie [AmazonMQServiceRolePolicy](#), die serviceverknüpfte Rolle `AWSServiceRoleForAmazonMQ` enthält, um die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `ec2:CreateVpcEndpoint` auf der `vpc`-Ressource.
- Aktion: `ec2:CreateVpcEndpoint` auf der `subnet`-Ressource.
- Aktion: `ec2:CreateVpcEndpoint` auf der `security-group`-Ressource.
- Aktion: `ec2:CreateVpcEndpoint` auf der `vpc-endpoint`-Ressource.
- Aktion: `ec2:DescribeVpcEndpoints` auf der `vpc`-Ressource.
- Aktion: `ec2:DescribeVpcEndpoints` auf der `subnet`-Ressource.
- Aktion: `ec2:CreateTags` auf der `vpc-endpoint`-Ressource.
- Aktion: `logs:PutLogEvents` auf der `log-group`-Ressource.
- Aktion: `logs:DescribeLogStreams` auf der `log-group`-Ressource.
- Aktion: `logs:DescribeLogGroups` auf der `log-group`-Ressource.
- Aktion: `CreateLogStream` auf der `log-group`-Ressource.
- Aktion: `CreateLogGroup` auf der `log-group`-Ressource.

Wenn Sie einen Amazon-MQ-für-RabbitMQ-Broker erstellen, erlaubt die `AmazonMQServiceRolePolicy`-Berechtigungsrichtlinie Amazon MQ die Durchführung der folgenden Aufgaben in Ihrem Namen.

- Erstellen Sie einen Amazon-VPC-Endpunkt für den Broker mithilfe der von Ihnen bereitgestellten Amazon VPC, des Subnetzes und der Sicherheitsgruppe. Sie können den für Ihren Broker erstellten Endpunkt verwenden, um sich über die RabbitMQ-Verwaltungskonsole, die Verwaltungs-API oder programmatisch mit dem Broker zu verbinden.
- Erstellen Sie Protokollgruppen und veröffentlichen Sie Broker-Protokolle in Amazon CloudWatch Logs.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcEndpoints"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVpcEndpoint"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:vpc/*",
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVpcEndpoint"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:vpc-endpoint/*"
      ],
      "Condition": {
```

```

        "StringEquals": {
            "aws:RequestTag/AMQManaged": "true"
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateTags"
        ],
        "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
        "Condition": {
            "StringEquals": {
                "ec2:CreateAction": "CreateVpcEndpoint"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DeleteVpcEndpoints"
        ],
        "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
        "Condition": {
            "StringEquals": {
                "ec2:ResourceTag/AMQManaged": "true"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "logs:PutLogEvents",
            "logs:DescribeLogStreams",
            "logs:DescribeLogGroups",
            "logs:CreateLogStream",
            "logs:CreateLogGroup"
        ],
        "Resource": [
            "arn:aws:logs:*:*:log-group:/aws/amazonmq/*"
        ]
    }
]

```

```
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter [servicegebundene Rollenberechtigungen](#) im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Amazon MQ

Sie müssen eine servicegebundene Rolle nicht manuell erstellen. Wenn Sie das erste Mal einen Broker erstellen, erstellt Amazon MQ eine serviceverknüpfte Rolle, die AWS-Services in Ihrem Namen aufruft. Alle nachfolgenden Broker, die Sie erstellen, verwenden dieselbe Rolle, und es wird keine neue Rolle erstellt.

Important

Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Funktionen verwendet. Weitere Informationen finden Sie unter [Eine neue Rolle ist in meinem IAM-Konto erschienen](#).

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen.

Sie können auch die IAM-Konsole verwenden, um eine serviceverknüpfte Rolle mit dem Anwendungsfall Amazon MQ zu erstellen. Erstellen Sie in der AWS CLI oder der AWS-API eine servicegebundene Rolle mit dem Servicenamen `mq.amazonaws.com`. Weitere Informationen finden Sie unter [Erstellen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch. Wenn Sie diese servicegebundene Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Bearbeiten einer serviceverknüpften Rolle für Amazon MQ

Amazon MQ erlaubt es Ihnen nicht, die serviceverknüpfte Rolle `AWSServiceRoleForAmazonMQ` zu bearbeiten. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon MQ

Wenn Sie eine Funktion oder einen Service, die bzw. der eine servicegebundene Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte

juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre servicegebundene Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Note

Wenn der Amazon MQ-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt der Löschvorgang möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie Amazon MQ-Ressourcen, die von der Rolle `AWSServiceRoleForAWSLicenseManagerRole` verwendet werden:

- Löschen Sie Ihre Amazon MQ -Broker mit dem AWS Management Console, Amazon MQ CLI oder Amazon MQ API. Weitere Informationen zum Löschen von Brokern finden Sie unter [???](#).

So löschen Sie die servicegebundene Rolle mit IAM

Verwenden Sie die IAM-Konsole, die AWS CLI oder die AWS-API, um die serviceverknüpfte Rolle „`AWSServiceRoleForIVSRecordToS3`“ zu löschen. Weitere Informationen finden Sie unter [Löschen einer servicegebundenen Rolle](#) im IAM-Leitfaden

Unterstützte Regionen für Amazon MQ serviceverknüpfte Rollen

Amazon MQ unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [AWS-Regionen und Endpunkte](#).

Name der Region	Regions-ID	Amazon MQ Support
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Osaka)	ap-northeast-3	Ja

Name der Region	Regions-ID	Amazon MQ Support
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Kanada (Zentral)	ca-central-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Irland)	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](#)
- [Ich bin nicht zur Ausführung von iam:PassRole autorisiert](#)
- [Ich möchte Personen außerhalb meines AWS-Kontos Zugriff auf meine Amazon MQ-Ressourcen erteilen](#)

Ich bin nicht autorisiert, eine Aktion in Amazon MQ auszuführen

Wenn die AWS Management Console Ihnen mitteilt, dass Sie nicht zur Ausführung einer Aktion autorisiert sind, 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 Benutzer `mateojackson` versucht, die Konsole zum Anzeigen von Details zu einem *Widget* zu verwenden, jedoch nicht über `mq:GetWidget`-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 zur Ausführung von `iam:PassRole` autorisiert

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 erlauben die Übergabe einer vorhandenen Rolle an diesen Service, sodass keine neue Servicerolle oder serviceverknüpfte Rolle erstellt werden muss. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Service.

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 Service.

```
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.

Wenden Sie sich an Ihren AWS-Administrator, falls Sie weitere Unterstützung benötigen. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen odzur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS-Kontos Zugriff auf meine Amazon MQ-Ressourcen erteilen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, 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 zum Gewähren des Zugriffs auf Ihre Ressourcen für alle Ihre AWS-Konten finden Sie unter [Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen Ihrer AWS-Konto](#) im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie AWS-Konten-Drittanbieter Zugriff auf Ihre Ressourcen bereitstellen, finden Sie unter [Gewähren des Zugriffs auf AWS-Konten von externen Benutzern](#) im IAM-Benutzerhandbuch.
- 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 kontenübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

Compliance-Validierung für Amazon MQ


Externe Prüfer bewerten im Rahmen verschiedener AWS-Compliance-Programme die Sicherheit und Compliance von Amazon MQ. Zu diesen Programmen gehören SOC, PCI, HIPAA und andere.

Informationen darüber, ob ein AWS-Service in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter [AWS-Services in Geltungsbereich nach Compliance-Programm](#). Wählen Sie das Compliance-Programm, das Sie interessiert. Allgemeine Informationen finden Sie unter [AWS-Compliance-Programme](#).

Sie können Auditberichte von Drittanbietern unter AWS Artifact herunterladen. Weitere Informationen finden Sie unter [Berichte herunterladen in AWS Artifact](#).

Ihre Compliance-Verantwortung bei der Verwendung von AWS-Services ist von der Sensibilität Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften abhängig. AWS stellt die folgenden Ressourcen zur Unterstützung der Compliance bereit:

- [Kurzanleitungen für Sicherheit und Compliance](#) – In diesen Bereitstellungsleitfäden werden Überlegungen zur Architektur erörtert und Schritte zum Bereitstellen von Basisumgebungen auf AWS zur Verfügung gestellt, die auf Sicherheit und Compliance ausgerichtet sind.
- [Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance in Amazon Web Services](#) – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe von AWS HIPAA-berechtigte Anwendungen erstellen können.

 Note

Nicht alle AWS-Services sind HIPAA-berechtigt. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [AWS-Compliance-Ressourcen](#) – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort relevant sein.
- [AWS-Compliance-Leitfäden für Kunden](#) – Verstehen Sie das Modell der geteilten Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Methoden zum Schutz von AWS-Services zusammengefasst und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- [Auswertung von Ressourcen mit Regeln](#) im AWS Config-Entwicklerhandbuch – Der AWS Config-Service bewertet, wie gut Ihre Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften übereinstimmen.
- [AWS Security Hub](#) – Dieser AWS-Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus innerhalb von AWS. Security Hub verwendet Sicherheitskontrollen, um Ihre AWS-Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).

- [AWS Audit Manager](#) – Dieser AWS-Service hilft Ihnen, Ihre AWS-Nutzung kontinuierlich zu überprüfen, um den Umgang mit Risiken und die Compliance von Branchenstandards zu vereinfachen.

Ausfallsicherheit bei Amazon MQ

Im Zentrum der globalen AWS-Infrastruktur stehen die AWS-Regionen und Availability Zones (Verfügbarkeitszonen, AZs). AWS Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und -Availability Zones finden Sie unter [Globale AWS-Infrastruktur](#).

Infrastruktursicherheit in Amazon MQ

Als verwalteter Service ist durch die globalen Verfahren zur Gewährleistung der Netzwerksicherheit von AWS geschützt. Informationen zu AWS-Sicherheitsdiensten und wie AWS die Infrastruktur schützt, finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS-Umgebung anhand der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastrukturschutz](#) im Security Pillar AWS Well-Architected Framework.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf 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 Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Best Practices für die Sicherheit in Amazon MQ

Die folgenden Entwurfsmuster können die Sicherheit Ihres Amazon MQ-Broker verbessern.

Themen

- [Broker ohne öffentlichen Zugriff bevorzugen](#)
- [Immer eine Autorisierungszuordnung konfigurieren](#)
- [Unnötige Protokolle mit VPC-Sicherheitsgruppen bockieren](#)

Weitere Informationen dazu, wie Amazon MQ Ihre Daten verschlüsselt, sowie eine Liste der unterstützten Protokolle finden Sie unter [Datenschutz](#).

Broker ohne öffentlichen Zugriff bevorzugen

Für Broker ohne öffentliche Zugänglichkeit ist kein Zugriff von außerhalb Ihrer [VPC](#) möglich. Dies reduziert die Anfälligkeit Ihres Brokers für DDoS-Angriffe (Distributed Denial of Service) aus dem öffentlichen Internet ganz wesentlich. Weitere Informationen finden Sie unter [Zugreifen auf die Webkonsole eines Brokers ohne öffentliche Zugänglichkeit](#) in diesem Handbuch und unter [Vorbereitung auf DDoS-Angriffe durch die Verringerung der Angriffsfläche](#) im AWS-Blog zur Sicherheit.

Immer eine Autorisierungszuordnung konfigurieren

Da für ActiveMQ standardmäßig keine Autorisierungszuordnung konfiguriert ist, kann jeder authentifizierte Benutzer eine Aktion auf dem Broker ausführen. Daher ist es eine bewährte Methode, Berechtigungen nach Gruppe einzuschränken. Weitere Informationen finden Sie unter [authorizationEntry](#).

Important

Wenn Sie eine Autorisierungszuordnung angeben, die `dieactivemq-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 blockieren

Um die Sicherheit zu erhöhen, sollten Sie die Verbindungen von unnötigen Protokolle und Ports, indem Sie Ihre Amazon VPC-Sicherheitsgruppe ordnungsgemäß konfigurieren. Beispielsweise können Sie zur Einschränkung des Zugriffs auf die meisten Protokolle bei gleichzeitiger Gewährung des Zugriffs zu OpenWire und zur Webkonsole den Zugriff lediglich auf 61617 und 8162 erlauben. Dies begrenzt Gefahren durch die Blockierung von Protokollen, die Sie nicht verwenden, während OpenWire und die Webkonsole normal funktionieren können.

Erlauben Sie nur die Protokoll-Ports, die Sie verwenden.

- AMQP: 5671
- MQTT: 8883
- OpenWire: 61617
- STOMP: 61614
- WebSocket: 61619

Weitere Informationen finden Sie unter:

- [Configure Additional Broker Settings](#)
- [Sicherheitsgruppen für Ihre VPC](#)
- [Standardsicherheitsgruppe für Ihre VPC](#)
- [Arbeiten mit Sicherheitsgruppen](#)

Überwachen und Protokollieren von Amazon MQ-Brokern

Die Überwachung ist ein wichtiger Teil der Aufrechterhaltung von Zuverlässigkeit, Verfügbarkeit und Performance Ihrer AWS-Lösungen. Sie sollten Überwachungsdaten von allen Teilen Ihrer AWS-Lösung sammeln, damit Sie einen Multi-Point-Fehler leichter debuggen können, falls ein solcher auftritt. AWS bietet verschiedene Tools zur Überwachung von Amazon MQ und anderer Ressourcen und zur Reaktion auf potenzielle Vorfälle.

Themen

- [Zugreifen auf CloudWatch Metriken für Amazon MQ](#)
- [Überwachung von Amazon MQ-Brokern mit Amazon CloudWatch](#)
- [Protokollieren von Amazon MQ-API-Aufrufen mit AWS CloudTrail](#)
- [Konfigurieren von Amazon MQ zur Veröffentlichung von Protokollen in Amazon CloudWatch Logs](#)

Zugreifen auf CloudWatch Metriken für Amazon MQ

Amazon MQ und Amazon CloudWatch sind integriert, sodass Sie mit Metriken für Ihre ActiveMQ-Broker und die Zieladressen des Brokers anzeigen und analysieren können (Wartschlangen und Themen). Sie können Ihre Amazon MQ Metriken über die CloudWatch Konsole, die AWS CLI oder die CloudWatch CLI an. Die CloudWatch-Metriken für Amazon MQ werden automatisch vom Broker abgefragt und jede Minute an CloudWatch übergeben.

Eine vollständige Liste der Amazon MQ-Metriken finden Sie unter [Monitoring Amazon MQ using CloudWatch](#).

Weitere Informationen zu CloudWatch-Alarmen und Statusänderungen finden Sie unter [Erstellen von Amazon CloudWatch-Alarmen](#) im Amazon CloudWatch-Benutzerhandbuch.

Note

Für Amazon MQ-Metriken, die über CloudWatch gemeldet werden, fallen keine Gebühren an. Diese Metriken werden im Rahmen des Amazon MQ-Service bereitgestellt.

Für ActiveMQ-Broker überwacht CloudWatch nur die ersten 1 000 Ziele.

Für RabbitMQ-Broker überwacht CloudWatch nur die ersten 500 Ziele, sortiert nach Anzahl der Verbraucher.

Themen

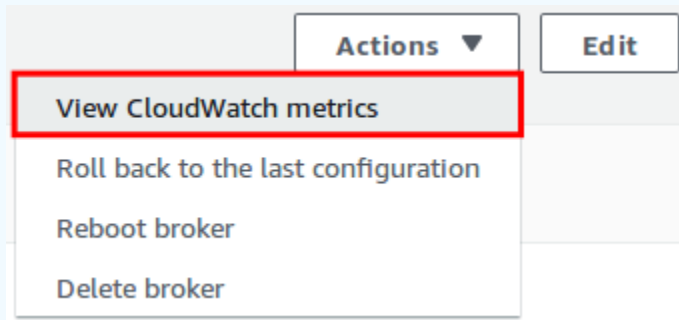
- [AWS Management Console](#)
- [AWS Command Line Interface](#)
- [Amazon CloudWatch API](#)

AWS Management Console

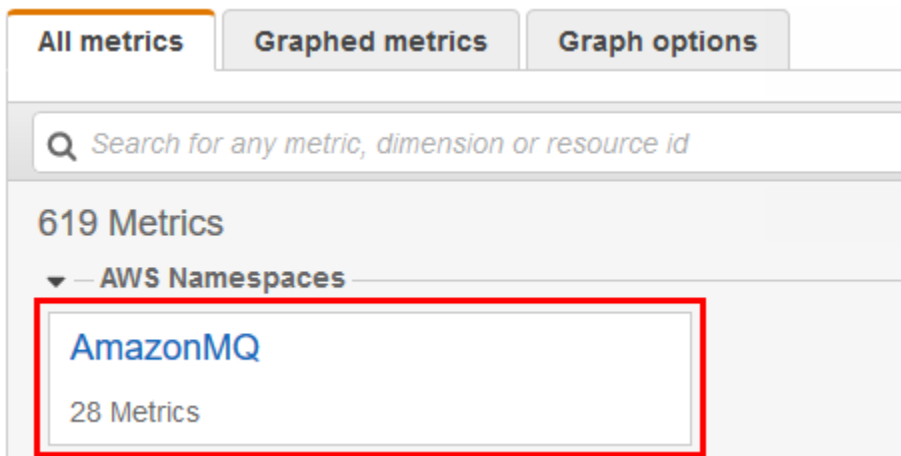
Das folgende Beispiel zeigt, wie Sie auf CloudWatch Metriken für Amazon MQ mithilfe der zugreifenAWS Management Console.

Note

Wenn Sie sich bereits bei der Amazon MQ-Konsole angemeldet haben, wählen Sie auf der Seite Details des Brokers Aktionen, CloudWatch-Metriken anzeigen.



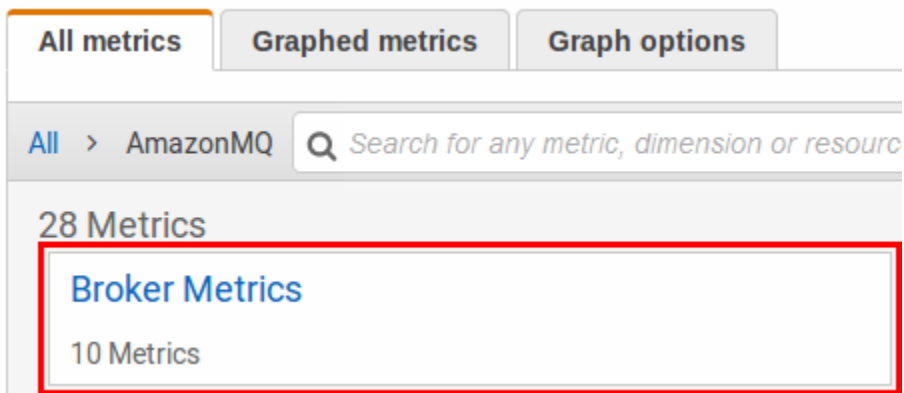
1. Melden Sie sich beim an [CloudWatch Konsole](#).
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace der AmazonMQ-Metrik aus.



4. Wählen Sie eine der folgenden Metrik-Dimensionen aus:

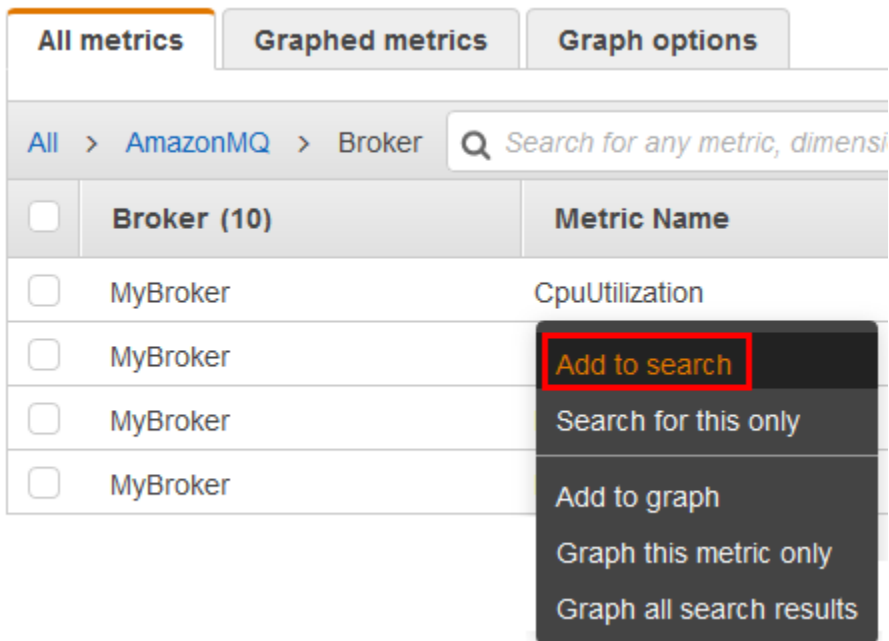
- Broker-Metriken
- Warteschlangenmetriken nach Broker
- Themenbezogene Metriken nach Broker

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.



AWS Command Line Interface

Für den Zugriff auf Amazon MQ-Metriken mithilfe der AWS CLI, verwenden Sie den Befehl [get-metric-statistics](#).

Weitere Informationen finden Sie unter [Abrufen von Statistiken für eine bestimmte Ressource](#) im Amazon CloudWatch-Benutzerhandbuch.

Amazon CloudWatch API

Für den Zugriff auf Amazon MQ Metriken mithilfe der CloudWatch API verwenden Sie die [GetMetricStatistics](#)Aktion

Weitere Informationen finden Sie unter [Abrufen von Statistiken für eine bestimmte Ressource](#) im Amazon CloudWatch-Benutzerhandbuch.

Überwachung von Amazon MQ-Brokern mit Amazon CloudWatch

Amazon MQ und Amazon CloudWatch sind integriert, sodass Sie mit CloudWatch Metriken für Ihren ActiveMQ-Broker und die Ziele des Brokers (Warteschlangen und Themen) anzeigen und analysieren können. Sie können Ihre Amazon MQ Metriken über die CloudWatch Konsole, die AWS CLI oder

die CloudWatch CLI anzeigen und analysieren. Die CloudWatch-Metriken für Amazon MQ werden automatisch vom Broker abgefragt und jede Minute an CloudWatch übergeben.

Weitere Informationen finden Sie unter [Zugreifen auf CloudWatch Metriken für Amazon MQ](#).

Note

Die folgenden Statistiken gelten für alle Metriken:

- Average
- Minimum
- Maximum
- Sum

Der AWS/AmazonMQ-Namespace enthält die folgenden Metriken.

Themen


- [Protokollieren und Überwachen von Amazon MQ für ActiveMQ Broker](#)
- [Protokollierung und Überwachung von Amazon MQ für RabbitMQ Broker](#)

Protokollieren und Überwachen von Amazon MQ für ActiveMQ Broker

Amazon MQ für ActiveMQ Metriken

Metrik	Einheit	Beschreibung
AmqpMaximumConnections	Anzahl	Die maximale Anzahl von Clients, die Sie über AMQP mit Ihrem Broker verbinden können. Weitere Informationen zu Verbindungskontingenten finden Sie unter Quotas in Amazon MQ .
BurstBalance	Prozent	Der Anteil der verbleibenden Burst-Gutschriften auf

Metrik	Einheit	Beschreibung
		<p>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-Guthaben in Amazon EBS finden Sie unter I/O-Guthaben und Burst-Performance.</p>

Metrik	Einheit	Beschreibung
CpuCreditBalance	Guthaben (vCPU-Minuten)	<p> Important</p> <p>Diese Metrik ist nur für den Broker-Instance-Typ <code>mq.t2.micro</code> verfügbar.</p> <p>Die Metriken für CPU-Guthaben sind nur mit fünfminütigen Intervallen verfügbar.</p> <p>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.</p> <p>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.</p>


Metrik	Einheit	Beschreibung
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 Festplattenspeichers, der vom Speicher des Aufgabenschedulers belegt wird.
JournalFilesForFastRecovery	Anzahl	Die Anzahl der Journaldateien, die nach einem sauberen Shutdown erneut abgespielt werden.

Metrik	Einheit	Beschreibung
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 mit 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	Beschreibung
OpenwireMaximumConnections	Anzahl	Die maximale Anzahl von Clients, die Sie über OpenWire mit Ihrem Broker verbinden können. 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	Beschreibung
TotalProducerCount	Anzahl	Die Gesamtzahl der Nachrichtenproduzenten, 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 Schreibvorgänge.
WsMaximumConnections	Anzahl	Die maximale Anzahl von Clients, die Sie über WebSocket mit Ihrem Broker verbinden können. Weitere Informationen zu Verbindungskontingenten finden Sie unter Quotas in Amazon MQ .

Dimensionen für ActiveMQ-Broker-Metriken

Dimension	Beschreibung
Broker	Der Name des Brokers <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Ein Broker mit einer einzigen Instance hat das Suffix -1. Ein aktiver/Standby-Broker für hohe Verfügbarkeit hat die</p> </div>

Dimension	Beschreibung
	Suffixe -1 und -2 für sein redundantes Paar.

ActiveMQ-Ziel-Metriken (Warteschlange und Thema)


Important


Die folgenden Metriken enthalten minutenweise Zählungen für den CloudWatch-Abfragezeitraum.

- EnqueueCount
- ExpiredCount
- DequeueCount
- DispatchCount
- InFlightCount

Zum Beispiel hat in einem 5-minütigen [CloudWatch Zeitraum](#), EnqueueCount fünf Zählwerte, jeweils für 1 Minute des Zeitraums. Die Statistiken Minimum und Maximum bieten den niedrigsten und höchsten Wert pro Minute während des angegebenen Zeitraums.

Metrik	Einheit	Beschreibung
ConsumerCount	Anzahl	Die Anzahl der Verbraucher, die das Ziel abonniert haben.
EnqueueCount	Anzahl	Die Anzahl der Nachrichten, die zum Ziel gesendet werden, pro Minute.
EnqueueTime	Zeit (Millisekunden)	Die End-to-End-Latenz zwischen der Ankunft einer Nachricht beim Broker bis

Metrik	Einheit	Beschreibung
		<p>zu ihrer Zustellung an einen Verbraucher.</p> <div data-bbox="1068 331 1510 1461" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> Note</p> <p>EnqueueTime misst weder die End-to-End-Latenz ab dem Zeitpunkt, an dem eine Nachricht von einem Produzenten gesendet wird, noch die Latenz ab dem Empfang einer Nachricht durch einen Broker, bis sie vom Broker bestätigt wird. 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.</p> </div>
ExpiredCount	Anzahl	Die Anzahl der Nachrichten pro Minute, die nicht übermittelt werden konnten, da sie abgelaufen sind.
DispatchCount	Anzahl	Die Anzahl der Nachrichten pro Minute, die an Verbraucher gesendet wurden.

Metrik	Einheit	Beschreibung
DequeueCount	Anzahl	Die Anzahl der Nachrichten, die von Verbrauchern bestätigt wurden.
InFlightCount	Anzahl	Die Anzahl der an Verbraucher gesendeten Nachrichten, die nicht bestätigt wurden.
ReceiveCount	Anzahl	Die Anzahl der Nachrichten, die vom Remote-Broker für einen Duplex-Netzwerk-Connector 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. <div data-bbox="1068 1234 1507 1451" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important Die Metrik gilt nur für Warteschlangen.</p> </div>
TotalEnqueueCount	Anzahl	Die Gesamtzahl der Nachrichten, die an den Broker gesendet wurden.
TotalDequeueCount	Anzahl	Die Gesamtanzahl der Nachrichten, die von Clients verwendet wurden.

Note

Die Metriken `TotalEnqueueCount` und `TotalDequeueCount` enthalten Nachrichten zu Beratungsthemen. Weitere Informationen zu Nachrichten zu Beratungsthemen finden Sie in der [ActiveMQ-Dokumentation](#).


Dimensionen für ActiveMQ Ziel-Metriken (Warteschlange und Thema)

Dimension	Beschreibung
Broker	Der Name des Brokers. <div data-bbox="857 785 1474 1066" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>Ein Broker mit einer einzigen Instance hat das Suffix -1. Ein aktiver/Standby-Broker für hohe Verfügbarkeit hat die Suffixe -1 und -2 für sein redundantes Paar.</p> </div>
Topic oder Queue	Der Name des Themas oder der Warteschlange.
NetworkConnector	Der Name des Netzwerk-Connectors.

Protokollierung und Überwachung von Amazon MQ für RabbitMQ Broker

RabbitMQ-Broker-Metriken

Metrik	Einheit	Beschreibung
ExchangeCount	Anzahl	Die Gesamtzahl der auf dem Broker konfigurierten Börsen.

Metrik	Einheit	Beschreibung
QueueCount	Anzahl	Die Gesamtanzahl der auf dem Broker konfigurierten Warteschlangen.
ConnectionCount	Anzahl	Die Gesamtzahl der auf dem Broker hergestellt wurden.
ChannelCount	Anzahl	Die Gesamtzahl der auf dem Broker festgelegten Kanäle.
ConsumerCount	Anzahl	Die Gesamtzahl der Verbraucher, die mit dem Broker verbunden sind.
MessageCount	Anzahl	Die Gesamtzahl der Nachrichten in der Warteschlange. <div data-bbox="1068 961 1510 1369" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"><p> Note</p><p>Die produzierte Anzahl ist die Gesamtsumme der bereitgestellten und unerkannten Nachrichten auf dem Broker.</p></div>
MessageReadyCount	Anzahl	Die Gesamtzahl der bereitgestellten Nachrichten in den Warteschlangen.
MessageUnacknowledgedCount	Anzahl	Die Gesamtzahl der nicht bestätigten Nachrichten in den Warteschlangen.

Metrik	Einheit	Beschreibung
PublishRate	Anzahl	<p>Die Rate, mit der Nachrichten an den Broker veröffentlicht werden.</p> <p>Die erzeugte Zahl stellt die Anzahl der Nachrichten pro Sekunde zum Zeitpunkt der Probenahme dar.</p>
ConfirmRate	Anzahl	<p>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.</p> <p>Die erzeugte Zahl stellt die Anzahl der Nachrichten pro Sekunde zum Zeitpunkt der Probenahme dar.</p>
AckRate	Anzahl	<p>Die Rate, mit der Nachrichten von den Verbrauchern anerkannt werden.</p> <p>Die erzeugte Zahl stellt die Anzahl der Nachrichten pro Sekunde zum Zeitpunkt der Probenahme dar.</p>

Metrik	Einheit	Beschreibung
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	Beschreibung
RabbitMQDiskFreeLimit	Bytes	Das Festplattenlimit für einen RabbitMQ-Broker. Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar. Diese Metrik unterscheidet sich je nach Instance-Größe. Weitere Informationen zu Amazon MQ Instance-Typen, finden Sie unter the section called “Instance-Typen von Amazon MQ für RabbitMQ”
RabbitMQDiskFree	Bytes	Das Gesamt-Volumen des freien Speicherplatzes, der in einem RabbitMQ-Broker verfügbar ist. Wenn die Datenträgenutzung seinen Grenzwert überschreitet, blockiert der Cluster alle Herstellerverbindungen. Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar.

Metrik	Einheit	Beschreibung
RabbitMQFdUsed	Anzahl	Anzahl der verwendeten Datei-Deskriptoren Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikergebnisse aller drei RabbitMQ-Knoten dar.
RabbitMQIOReadAverageTime	Anzahl	Die durchschnittliche Zeit (in Millisekunden), die RabbitMQ für einen Lesevorgang benötigt. Der Wert ist proportional zur Nachrichtengröße.
RabbitMQIOWriteAverageTime	Anzahl	Die durchschnittliche Zeit (in Millisekunden), die RabbitMQ für einen Schreibvorgang benötigt. Der Wert ist proportional zur Nachrichtengröße.

Abmessungen für RabbitMQ-Broker-Metriken

Dimension	Beschreibung
Broker	Der Name des Brokers.

RabbitMQ-Knoten-Metriken

Metrik	Einheit	Beschreibung
SystemCpuUtilization	Prozent	Der Prozentsatz der zugewiesenen Amazon EC2-

Metrik	Einheit	Beschreibung
		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 Limit hinausgeht, blockiert der Cluster alle Herstellerverbindungen.
RabbitMQDiskFreeLimit	Bytes	Das Festplattenlimit für einen RabbitMQ-Knoten. Diese Metrik unterscheidet sich je nach Instance-Größe. Weitere Informationen zu Amazon MQ Instance-Typen, finden Sie unter the section called "Instance-Typen von Amazon MQ für RabbitMQ"
RabbitMQDiskFree	Bytes	Das Gesamt-Volumen des freien Speicherplatzes, der in einem RabbitMQ-Knoten verfügbar ist. Wenn die Datenträgernutzung seinen Grenzwert überschreitet, blockiert der Cluster alle Herstellerverbindungen.
RabbitMQFdUsed	Anzahl	Anzahl der verwendeten Datei-Deskriptoren

Abmessungen für RabbitMQ-Knotenmetriken

Dimension	Beschreibung
Node	<p>Der Name des Knotens.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>Ein Knotenname besteht aus zwei Teilen: einem Präfix (üblicher weiserabbit) und einen Hostnamen . Zum Beispiel ist <code>rabbit@ip-10-0-0-230.us-west-2.compute.internal</code> ein Knotenname mit dem Präfix <code>rabbit</code> und dem Hostnamen <code>ip-10-0-0-230.us-west-2.compute.internal</code> .</p> </div>
Broker	Der Name des Brokers.

RabbitMQ-Warteschlangen-Metriken

Metrik	Einheit	Beschreibung
ConsumerCount	Anzahl	Die Anzahl der Verbraucher, die die Warteschlange abonniert haben.
MessageReadyCount	Anzahl	Die Anzahl der Nachrichten, die derzeit zugestellt werden können.
MessageUnacknowledgedCount	Anzahl	Die Anzahl der Nachrichten, für die der Server auf die Bestätigung wartet.

Metrik	Einheit	Beschreibung
MessageCount	Anzahl	Die Gesamtzahl für MessageReadyCount und MessageUnacknowledgedCount (auch als Warteschlangentiefe bezeichnet).

Dimensionen für RabbitMQ-Queue-Metriken

Note

Amazon MQ 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	Beschreibung
Queue	Der Name der Warteschlange.
VirtualHost	Der Name des virtuellen Hosts.
Broker	Der Name des Brokers.

Protokollieren von Amazon MQ-API-Aufrufen mit AWS CloudTrail

Amazon MQ ist in AWS CloudTrail integriert. Dies ist ein Service, der die Amazon MQ-Aufrufe eines Benutzers, einer Rolle oder eines AWS-Service protokolliert. CloudTrail erfasst alle API-Aufrufe für Amazon MQ-Broker und Konfigurationen als Ereignisse, einschließlich Aufrufen von der Konsole und von Code-Aufrufen an die Amazon MQ APIs. Weitere Informationen über CloudTrail finden Sie im [AWS CloudTrail-Leitfaden](#).

Note

CloudTrail protokolliert keine API-Aufrufe im Zusammenhang mit ActiveMQ-Operationen (z. B. das Senden und Empfangen von Nachrichten) oder an die ActiveMQ-Webkonsole. Um Informationen zu ActiveMQ-Operationen zu protokollieren, können Sie [Amazon MQ so konfigurieren, dass allgemeine und Prüfprotokolle in Amazon CloudWatch Logs veröffentlicht werden](#).

Anhand der von CloudTrail erfassten Informationen können Sie eine bestimmte Anforderung an eine Amazon MQ-API, die IP-Adresse des Anforderers, die Identität des Anforderers, das Datum und die Uhrzeit der Anforderung usw. identifizieren. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail-Ereignissen an einen Amazon S3-Bucket, einschließlich Ereignisse für aktivieren. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole trotzdem in Ereignisverlauf anzeigen. Weitere Informationen finden Sie unter [Übersicht zum Erstellen eines Trails](#) im [AWS CloudTrail-Benutzerhandbuch](#).

Amazon MQ-Informationen in CloudTrail

CloudTrail wird beim Erstellen Ihres AWS-Kontos für Sie aktiviert. Wenn die unterstützte Ereignisaktivität in Amazon MQ eintritt, wird diese Aktivität in einem CloudTrail-Ereignis zusammen mit anderen AWS-Serviceereignissen im Ereignisverlauf aufgezeichnet. Sie können die neuesten Ereignisse für Ihr AWS-Konto anzeigen, durchsuchen und herunterladen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail-Ereignisverlauf](#) im AWS CloudTrail-Benutzerhandbuch.


Ein Trail ermöglicht es CloudTrail, Protokolldateien in einem Amazon-S3-Bucket bereitzustellen. Sie können einen Trail erstellen, um eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS-Konto zu pflegen. Wenn Sie einen Trail in der AWS Management Console anlegen, gilt dieser standardmäßig für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen AWS-Regionen und stellt die Protokolldateien in dem angegebenen Amazon S3-Bucket bereit. Sie können auch andere AWS-Services konfigurieren, um die in den CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie in folgenden Themen im AWS CloudTrail-Benutzerhandbuch:

- [Von CloudTrail unterstützte Dienste und Integrationen](#)
- [Konfigurieren von Amazon-SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen](#)


- [Empfangen von CloudTrail-Protokolldateien aus mehreren Konten](#)

Amazon MQ unterstützt die Protokollierung der Anforderungsparameter und der Antworten für die folgenden APIs als Ereignisse in CloudTrail-Protokolldateien:

- [CreateConfiguration](#)
- [DeleteBroker](#)
- [DeleteUser](#)
- [RebootBroker](#)
- [UpdateBroker](#)

 Note

Die RebootBroker-Protokolldateien werden protokolliert, wenn Sie den Broker neu starten. Während des Wartungsfensters wird der Service automatisch neu gestartet, und die RebootBroker-Protokolldateien werden nicht protokolliert.

 Important

Für die GET-Methoden der folgenden APIs werden Anforderungsparameter protokolliert, die Antworten aber sind geschwärzt:

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

Für die folgenden APIs sind die Anforderungsparameter `data` und `password` mit Sternchen ausgeblendet (***):

- [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 Anforderung von einem anderen AWS-Service ausgeführt?

Weitere Informationen finden Sie unter [CloudTrail-userIdentity-Element](#) im AWS CloudTrailBenutzerhandbuch.

Beispiel für einen Amazon MQ-Protokolldateieintrag

Ein Trail ist eine Konfiguration, durch die Ereignisse als Protokolldateien an den angegebenen Amazon S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten.

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 CloudTrail-Protokolleinträge kein geordnetes Stacktrace öffentlicher APIs sind, erscheinen die Informationen nicht in einer bestimmten Reihenfolge.

```
{  
  "eventVersion": "1.06",
```

```
"userIdentity": {
  "type": "IAMUser",
  "principalId": "AKIAIOSFODNN7EXAMPLE",
  "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
  "accountId": "111122223333",
  "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
  "userName": "AmazonMqConsole"
},
"eventTime": "2018-06-28T22:23:46Z",
"eventSource": "amazonmq.amazonaws.com",
"eventName": "CreateBroker",
"awsRegion": "us-west-2",
"sourceIPAddress": "203.0.113.0",
"userAgent": "PostmanRuntime/7.1.5",
"requestParameters": {
  "engineVersion": "5.15.9",
  "deploymentMode": "ACTIVE_STANDBY_MULTI_AZ",
  "maintenanceWindowStartTime": {
    "dayOfWeek": "THURSDAY",
    "timeOfDay": "22:45",
    "timeZone": "America/Los_Angeles"
  },
  "engineType": "ActiveMQ",
  "hostInstanceType": "mq.m5.large",
  "users": [
    {
      "username": "MyUsername123",
      "password": "****",
      "consoleAccess": true,
      "groups": [
        "admins",
        "support"
      ]
    },
    {
      "username": "MyUsername456",
      "password": "****",
      "groups": [
        "admins"
      ]
    }
  ],
  "creatorRequestId": "1",
  "publiclyAccessible": true,
```



```
    "securityGroups": [
      "sg-a1b234cd"
    ],
    "brokerName": "MyBroker",
    "autoMinorVersionUpgrade": false,
    "subnetIds": [
      "subnet-12a3b45c",
      "subnet-67d8e90f"
    ]
  },
  "responseElements": {
    "brokerId": "b-1234a5b6-78cd-901e-2fgh-3i45j6k17819",
    "brokerArn": "arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819"
  },
  "requestID": "a1b2c345-6d78-90e1-f2g3-4hi56jk7l890",
  "eventID": "a12bcd3e-fg45-67h8-ij90-12k34d5l16mn",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Konfigurieren von Amazon MQ zur Veröffentlichung von Protokollen in Amazon CloudWatch Logs

Amazon MQ ist in Amazon CloudWatch Logs integriert, ein Service, der Ihre Protokolldateien aus einer Vielzahl von Quellen überwacht, speichert und öffnet. Sie können zum Beispiel [CloudWatch-Alarme](#) konfigurieren, um Benachrichtigungen über [Broker-Neustarts](#) zu erhalten oder Fehler bei der [ActiveMQ-Broker-Konfiguration](#) zu beheben. Weitere Informationen zu CloudWatch Logs finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).

Themen

- [Konfigurieren von Amazon MQ für ActiveMQ-Protokolle](#)
- [Konfigurieren von Amazon MQ für RabbitMQ-Protokolle](#)

Konfigurieren von Amazon MQ für ActiveMQ-Protokolle

Um Amazon MQ das Veröffentlichen von Protokollen in CloudWatch Logs zu erlauben, müssen Sie [Ihrem Amazon MQ-Benutzer](#) eine Berechtigung hinzufügen und eine ressourcenbasierte Richtlinie für [Amazon MQ konfigurieren](#), bevor Sie den Broker erstellen oder neu starten.

Im Folgenden werden die Schritte zum Konfigurieren von CloudWatch-Protokollen für Ihre ActiveMQ-Broker beschrieben.

Themen

- [Grundlegendes zum Aufbau der Protokollierung in CloudWatch Logs](#)
- [Hinzufügen der CreateLogGroup-Berechtigung zu Ihrem Amazon-MQ-Benutzer](#)
- [Konfigurieren einer ressourcenbasierten Richtlinie für Amazon MQ.](#)
- [Dienstübergreifende Confused-Deputy-Prävention](#)
- [Fehlerbehebung bei der CloudWatch Logs Konfiguration](#)

Grundlegendes zum Aufbau der Protokollierung in CloudWatch Logs

Sie können die allgemeine Protokollierung und die Audit-Protokollierung aktivieren, wenn Sie die [erweiterten Broker-Einstellungen konfigurieren](#), einen Broker erstellen oder einen Broker bearbeiten.

Durch die allgemeine Protokollierung wird die Standardprotokollierungsebene INFO aktiviert (DEBUG-Protokollierung wird nicht unterstützt). Sie 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-3i45j6k17819/general
```

Durch die [Audit-Protokollierung](#) wird die Protokollierung von Verwaltungsaktionen mit JMX oder mit der ActiveMQ-Webkonsole aktiviert und es wird `audit.log` in einer Protokollgruppe in Ihrem CloudWatch-Konto veröffentlicht. Die Protokollgruppe hat ein Format, das in etwa aussieht wie folgt:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k17819/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-3i45j6k17819-1.log
```

```
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-2.log
```

Die Suffixe -1 und -2 kennzeichnen einzelne Broker-Instances. Weitere Informationen finden Sie unter [Arbeiten mit Protokollgruppen und Protokoll-Streams](#) 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 Berechtigung `logs:CreateLogGroup` verfügt.

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 Beispiel [IAM-basierte Richtlinie](#) Erteilen der Berechtigung für `logs:CreateLogGroup` für Benutzer, denen diese Richtlinie angehängt ist.

```
{
  "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 unter [CreateLogGroup](#) 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 CloudWatch Logs veröffentlichen.

Um es Amazon MQ zu ermöglichen, Protokolle in Ihrer CloudWatch Logs-Protokollgruppe zu veröffentlichen, konfigurieren Sie eine ressourcenbasierte Richtlinie, um Amazon MQ Zugriff auf die folgenden CloudWatch Logs-API-Aktionen zu gewähren:

- [CreateLogStream](#) – Erstellt einen CloudWatch-Logs-Protokollstream für die angegebene Protokollgruppe.
- [PutLogEvents](#) - Stellt Ereignisse für den angegebenen CloudWatch Logs-Protokollstream bereit.

Im folgenden Beispiel gewährt die ressourcenbasierte Richtlinie die Berechtigung für `logs:CreateLogStream` und `logs:PutLogEvents` an AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "mq.amazonaws.com" },
      "Action": [ "logs:CreateLogStream", "logs:PutLogEvents" ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/amazonmq/*"
    }
  ]
}
```

Diese ressourcenbasierte Richtlinie muss mithilfe von AWS CLI konfiguriert werden, wie im folgenden Befehl gezeigt. Im Beispiel, ersetzen Sie `us-east-1` mit Ihren eigenen Informationen.

```
aws --region us-east-1 logs put-resource-policy --policy-name AmazonMQ-logs \
--policy-document "{\"Version\": \"2012-10-17\", \"Statement\": [{ \"Effect\": \"Allow\", \"Principal\": { \"Service\": \"mq.amazonaws.com\" },
```

```
\ "Action": [\"logs:CreateLogStream\", \"logs:PutLogEvents\"], \"Resource\":  
  \":arn:aws:logs:*:*:log-group:\ /aws\amazonmq\/*\" ]}]}"
```

Note

Da in diesem Beispiel das Präfix `/aws/amazonmq/` verwendet wird, müssen Sie die ressourcenbasierte Richtlinie nur einmal pro AWS-Konto und Region konfigurieren.

Dienstübergreifende 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 der dienstübergreifende Identitätswechsel zu Confused-Deputy-Problem führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen die Verwendung globalen Bedingungskontextschlüssel [aws:SourceArn](#) und [aws:SourceAccount](#) in Ihrer ressourcenbasierten Amazon-MQ-Richtlinie, um den Zugriff auf CloudWatch Logs 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 veranschaulicht eine ressourcenbasierte Richtlinie, die den Zugriff auf CloudWatch Logs auf einen einzelnen Amazon-MQ-Broker beschränkt.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": "logs:CreateLogStream",  
      "Resource": "arn:aws:logs:*:*:log-group:\ /aws\amazonmq\/*",  
      "Effect": "Deny"  
    }  
  ]  
}
```

```

{
  "Effect": "Allow",
  "Principal": {
    "Service": "mq.amazonaws.com"
  },
  "Action": [
    "logs:CreateLogStream",
    "logs:PutLogEvents"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/amazonmq/*",
  "Condition": {
    "StringEquals": {
      "aws:SourceAccount": "123456789012",
      "aws:SourceArn": "arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819"
    }
  }
}

```

Sie können Ihre ressourcenbasierte Richtlinie auch so konfigurieren, dass der Zugriff auf CloudWatch Logs auf alle Broker in einem Konto beschränkt wird, wie im Folgenden dargestellt.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "mq.amazonaws.com"
        ]
      },
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/amazonmq/*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:mq:*:123456789012:broker:*"
        }
      },
    }
  ]
}

```

```
    "StringEquals": {
      "aws:SourceAccount": "123456789012"
    }
  }
}
]
```

Weitere Informationen über das Sicherheitsproblem des verwirrten Stellvertreters finden Sie unter [Das Problem des verwirrten Stellvertreters](#) im Benutzerhandbuch.

Fehlerbehebung bei der CloudWatch Logs Konfiguration

Es kann vorkommen, dass sich CloudWatch Logs unvorhergesehen verhält. In diesem Abschnitt erhalten Sie einen Überblick über häufige Probleme und deren Lösungen.

Protokollgruppen werden in CloudWatch nicht angezeigt

[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 in CloudWatch-Protokollgruppen nicht angezeigt

[Konfigurieren einer ressourcenbasierten Richtlinie für Amazon MQ](#). Dies ermöglicht es Ihrem Broker, seine Protokolle zu veröffentlichen.

Konfigurieren von Amazon MQ für RabbitMQ-Protokolle

Wenn Sie die CloudWatch-Protokollierung für Ihre RabbitMQ-Broker aktivieren, verwendet Amazon MQ eine service-verknüpfte Rolle, um allgemeine Protokolle in CloudWatch zu veröffentlichen.

Wenn beim Erstellen eines Brokers keine Rolle mit Amazon MQ vorhanden ist, erstellt Amazon MQ automatisch eine Rolle. Alle nachfolgenden RabbitMQ-Broker verwenden dieselbe service-verknüpfte Rolle, um Protokolle in CloudWatch zu veröffentlichen.

Weitere Informationen zu serviceverknüpften Rollen finden Sie unter [Verwenden serviceverknüpfter Rollen](#) im AWS Identity and Access Management Benutzerhandbuch. Weitere Informationen darüber, wie Amazon MQ serviceverknüpfte Rollen verwendet, finden Sie unter [the section called "Verwenden von servicegebundenen Rollen"](#).

Kontingente in Amazon MQ

In diesem Thema werden Kontingente innerhalb von Amazon MQ aufgeführt. Viele der folgenden Kontingente 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 zum Anzeigen der aktuellen Verbindungslimits in Amazon CloudWatch finden Sie unter [Überwachung von Amazon-MQ-Brokern mit Amazon CloudWatch](#).



Themen

- [Broker](#)
- [Konfigurationen](#)
- [Benutzer](#)
- [Datenspeicherung](#)
- [API-Drosselung](#)

Broker


In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ-Brokern aufgeführt.

Limit	Beschreibung
Broker-Name	<ul style="list-style-type: none">• Er muss in Ihrem AWS-Konto eindeutig sein.• Er muss 1–50 Zeichen umfassen.• Er 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	<div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p> Important Gilt nicht für RabbitMQ-Broker.</p> </div> <p>300 für <code>mq.*.micro</code> Instance-Typ-Broker.</p>
Wire-Level-Verbindungen pro Protokoll für größeren Broker	<div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p> Important Gilt nicht für RabbitMQ-Broker.</p> </div> <p>2.000 für <code>mq.*.large</code> Instance-Typ-Broker.</p>
Sicherheitsgruppen pro Broker	5
ActiveMQ-Ziele (Warteschlangen und Themen), die in CloudWatch überwacht werden	CloudWatch überwacht nur die ersten 1 000 Ziele.
In CloudWatch überwachte RabbitMQ-Ziele (Warteschlangen)	CloudWatch überwacht nur die ersten 500 Ziele, sortiert nach Anzahl der Verbraucher.
Tags pro Broker	50

Konfigurationen

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ-Konfigurationen aufgeführt.

 **Important**
Gilt nicht für RabbitMQ-Broker.

Limit	Beschreibung
Konfigurationsname	<ul style="list-style-type: none"> • Er muss 1–150 Zeichen umfassen. • Er 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.

Important

Gilt nicht für RabbitMQ-Broker.




Limit	Beschreibung
Benutzername	<ul style="list-style-type: none"> • Er muss 1–100 Zeichen umfassen. • Er 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. (,).

Limit	Beschreibung
Passwort	<ul style="list-style-type: none"> • Es muss 12–250 Zeichen umfassen. • Es darf nur Zeichen aus den darstellbaren ASCII-Zeichen enthalten. • Es muss mindestens 4 eindeutige Zeichen enthalten. • Er darf keine Kommas enthalten. (,).
Benutzer pro Broker (einfache Auth)	250
Gruppen pro Benutzer (einfache Auth)	20

Datenspeicherung

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit der Amazon MQ-Datenspeicherung aufgeführt.

Limit	Beschreibung
Speicherkapazität pro kleinerem Broker	20 GB für mq.*.micro Instance-Typ-Broker. Weitere Informationen zu Amazon MQ Instance-Typen finden Sie unter Broker instance types .
Speicherkapazität pro Broker	200 GB für mq.*.*large Instance-Typ-Broker. Weitere Informationen zu Amazon MQ Instance-Typen finden Sie unter Broker instance types .
Job-Scheduler-Nutzungslimit pro Broker gestützt von Amazon EBS	

Limit	Beschreibung
	<div data-bbox="829 212 1507 380" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #ffe6e6;"> <p> Important Gilt nicht für RabbitMQ-Broker.</p> </div> <p data-bbox="829 449 1474 625">50 GB. Weitere Informationen zur Verwendung der Job-Planner finden Sie unter JobSchedulerUsage im Apache ActiveMQ -API-Dokumentation.</p>
<p data-bbox="110 674 727 751">Temporäre Speicherkapazität pro kleineren Broker.</p>	<div data-bbox="829 705 1507 873" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #ffe6e6;"> <p> Important Gilt nicht für RabbitMQ-Broker.</p> </div> <p data-bbox="829 942 1419 978">5 GBmq.*.micro Instance-Typ-Broker.</p>
<p data-bbox="110 1024 727 1102">Temporäre Speicherkapazität pro größeren Broker.</p>	<div data-bbox="829 1056 1507 1224" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #ffe6e6;"> <p> Important Gilt nicht für RabbitMQ-Broker.</p> </div> <p data-bbox="829 1293 1495 1371">50 GB für mq.*.*large Instance-Typ-Broker.</p>

API-Drosselung

Die folgenden Drosselungskontingente werden pro AWS-Konto über alle Amazon MQ-APIs aggregiert, um die Service-Bandbreite beizubehalten. Weitere Informationen zu Amazon MQ -APIs finden Sie im [Amazon MQ REST-API-Referenz](#).

⚠ Important

Diese Kontingente gelten nicht für Amazon MQ for ActiveMQ oder Amazon MQ für 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.

Inhalt

- [Allgemeine Problembehebung](#)
 - [Ich kann keine Verbindung zu meiner Broker-Webkonsole oder -Endpunkten herstellen.](#)
 - [Mein Broker läuft und ich kann die Konnektivität mit telnet bestätigen, aber meine Clients können keine Verbindung herstellen und geben SSL-Ausnahmen zurück.](#)
 - [Ich habe einen Broker erstellt, aber die Brokererstellung ist fehlgeschlagen.](#)
 - [Mein Broker wurde neu gestartet und ich bin mir nicht sicher, warum.](#)
- [Fehlerbehebung: Amazon MQ für ActiveMQ](#)
 - [Ich kann keine allgemeinen oder Überwachungsprotokolle für meinen Broker in CloudWatch Logs sehen, obwohl ich die Protokollierung aktiviert habe.](#)
 - [Nach dem Neustart oder dem Wartungsfenster des Brokers kann ich keine Verbindung zu meinem Broker herstellen, obwohl der Status RUNNING lautet. Warum?](#)
 - [Ich sehe, dass einige meiner Clients eine Verbindung zum Broker herstellen, während andere keine Verbindung herstellen können.](#)
 - [Ich sehe beim Ausführen von Operationen die Ausnahme org.apache.jasper.JasperException: An exception occurred processing JSP page auf der ActiveMQ-Konsole.](#)
- [Fehlerbehebung: Amazon MQ für RabbitMQ](#)
 - [In CloudWatch können keine Metriken für meine Warteschlangen oder virtuellen Hosts angezeigt werden.](#)
 - [Wie aktiviere ich Plugins in Amazon MQ für RabbitMQ?](#)
 - [Ich kann die Amazon-VPC-Konfiguration für den Broker nicht ändern.](#)
- [Fehlerbehebung: Erforderliche Codes für die Amazon-MQ-Aktion](#)
 - [Amazon MQ für RabbitMQ: Alarm über hohe Speicherauslastung](#)
 - [Diagnostizieren eines Alarms über hohe Speicherauslastung mit der RabbitMQ-Webkonsole](#)
 - [Diagnose eines Alarms über hohe Speicherauslastung mithilfe von Amazon-MQ-Metriken](#)
 - [Umgang mit dem Alarm über hohe Speicherauslastung](#)
 - [Reduzierung der Anzahl der Verbindungen und Kanäle](#)

- [Umgang mit pausierten Warteschlangensynchronisierungen in Clusterbereitstellungen](#)
- [Umgang mit Neustartschleifen in Einzel-Instance-Brokern](#)
- [Verhindern von Alarmen über hohe Speicherauslastung](#)
- [Amazon MQ für RabbitMQ: Ungültiger AWS Key Management Service-Schlüssel](#)
 - [Diagnose und Behandlung von INVALID_KMS_KEY](#)
- [Amazon MQ for ActiveMQ: Elastic-Network-Schnittstellenalarm wurde gelöscht](#)
- [Amazon MQ for ActiveMQ: Alarm -wegen zu geringem Arbeitsspeicher für Broker](#)
- [Amazon MQ für RabbitMQ: Festplattenlimit-Alarm](#)
 - [Diagnose und Behebung eines Festplattenlimit-Alarms](#)

Allgemeine Problembehebung

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.](#)
- [Mein Broker läuft und ich kann die Konnektivität mit telnet bestätigen, aber meine Clients können keine Verbindung herstellen und geben SSL-Ausnahmen zurück.](#)
- [Ich habe einen Broker erstellt, aber die Brokererstellung ist fehlgeschlagen.](#)
- [Mein Broker wurde neu gestartet und ich bin mir nicht sicher, warum.](#)

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 Subnetzes [Zugriffskontrolllisten \(ACL\)](#)-Regeln. Wenn Sie benutzerdefinierte Netzwerk-ACLs erstellt haben, müssen Sie möglicherweise die Netzwerk-ACL-Regeln ändern, um Zugriff auf Ihren Broker zu gewähren. Weitere Informationen zu Amazon VPC Netzwerken finden Sie unter [Einrichten 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 Modultypen gruppiert, da Amazon MQ für ActiveMQ und Amazon MQ für RabbitMQ unterschiedliche Ports für Verbindungen verwenden.

Amazon MQ für ActiveMQ

- Webkonsole — Port 8162
- OpenWire — Port 61617
- AMQP — Port 5671
- STOMP – Portierung von 61614
- MQTT-Port 8883
- WSS — Port 61619

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.

Amazon MQ for ActiveMQ

So testen Sie die Netzwerkkonnektivität Ihres Amazon MQ-Brokers

1. Öffnen Sie ein Terminal-Fenster oder eine Eingabeaufforderung.
2. Führen Sie Folgendes aus: `nslookup`, um Ihren Broker-DNS-Eintrag abzufragen. Für [aktive/Standby-Funktion](#)-Bereitstellungen verwenden, testen Sie sowohl die aktiven als auch die Standby-Endpunkte. Die Aktiv-/Standby-Endpunkte werden mit einem Suffix, `-1` oder `-2` der eindeutigen Broker-ID hinzugefügt. Ersetzen Sie den Endpunkt durch Ihre Informationen.

```
$ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Wird der Befehl erfolgreich ausgeführt, wird Ihnen eine Ausgabe ähnlich der folgenden angezeigt:


```
Non-authoritative answer:
Server: dns-resolver-corp-sfo-1.sfo.corp.amazon.com
Address: 172.10.123.456

Name: ec2-12-345-123-45.us-west-2.compute.amazonaws.com
Address: 12.345.123.45
Aliases: b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Die aufgelöste IP-Adresse sollte mit den in der Amazon MQ Konsole angegebenen IP-Adressen übereinstimmen. Dies 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. Ersetzen Sie `port` durch die

Port-Nummer 8162 für die Webkonsole oder andere Wire-Level-Ports verwenden, um zusätzliche Protokolle nach Bedarf zu testen.

 Note

Für Active-/Standby-Bereitstellungen erhalten Sie eine `Connect failed` Fehlermeldung, wenn Sie `telnet` mit dem Standby-Endpunkt. 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 `BrokerGeneral`-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`-Pfadttest 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 Ihrer [Elastic Network-Schnittstelle](#), und bestätigen Sie, dass der Status `in-use` ist. [Erstellen eines Amazon VPC Flow-Protokolls](#) für die Netzwerkschnittstelle jeder Instance und überprüfen Sie die generierten Flow-Protokolle. Suchen Sie nach den IP-Adressen des Brokers, wenn Sie die `telnet` und vergewissern Sie sich, dass die Verbindungspakete `ACCEPTED`,

einschließlich eines Rücksendepakets. Weitere Informationen und ein Beispiel für ein Flow Log finden Sie unter [Beispiele für Flow-Protokolldatensätze](#) im 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>
    ...
```

Amazon MQ for RabbitMQ

So testen Sie die Netzwerkkonnektivität Ihres Amazon MQ-Brokers

1. Öffnen Sie ein Terminal-Fenster oder eine Eingabeaufforderung.
2. Führen Sie 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-3i45j6k17819-1.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 *port* durch Port 443 für die Webkonsole und 5671 ersetzen, um die Wire-Level-AMQP-Verbindung zu testen.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.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-3i45j6k17819-1.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 `BrokerConnection.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 `dielnet`-Pfadtest erneut. Möglicherweise müssen Sie diesen Vorgang wiederholen, wenn die Verbindung geschlossen wird, bevor Ihr Broker neue Verbindungsmetriken in CloudWatch veröffentlicht hat. Metriken werden alle fünf Minuten veröffentlicht.
 - Für Broker ohne öffentliche Zugänglichkeit, wenn `dielnet`-Befehl fehlschlägt, überprüfen Sie den Status Ihrer [Elastic Network-Schnittstellen](#), und bestätigen Sie,

dass der Status `in-use`. [Erstellen eines Amazon VPC Flow-Protokolls](#) für jede Netzwerkschnittstelle und überprüfen Sie die generierten Flussprotokolle. Suchen Sie nach den privaten IP-Adressen des Brokers, wenn `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 unter [Beispiele für Flow-Protokolldatensätze](#) im Entwicklerhandbuch für Amazon VPC.

 Note

Dieser Schritt gilt nicht für Amazon MQ-Broker für RabbitMQ-Broker mit öffentlicher Zugänglichkeit.

5. Führen Sie Folgendes aus: `curl`, um die Konnektivität zur RabbitMQ Admin-Webkonsole zu überprüfen.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com:443/index.html
```

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. Zum Erstellen muss der Broker entweder die von AWS verwaltete IAM-Richtlinie `AmazonMQFullAccess` verwenden oder über den richtigen Satz von Amazon-EC2-Berechtigungen in Ihrer benutzerdefinierten IAM-Richtlinie verfügen. Weitere Informationen zu den erforderlichen Amazon EC2 Berechtigungen finden Sie unter [IAM-Berechtigungen erforderlich, um einen Amazon MQ-Broker zu erstellen](#) aus.
- 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 über Broker-Wartungsfenster finden Sie unter [the section called “Wartung eines Brokers”](#).
- Ihr Broker-Instance-Typ ist möglicherweise nicht für Ihre Anwendungs-Workload geeignet. Beispiel: Ausführen eines Produktions-Workloads auf einem `mq.t2.micro` kann dazu führen, dass dem Broker keine Ressourcen mehr zur Verfügung stehen. Eine hohe CPU-Auslastung oder eine hohe Broker-Speicherauslastung kann dazu führen, dass ein Broker unerwartet neu gestartet wird.

Verwenden Sie die folgenden CloudWatch-Metriken für Ihren Engine-Typ, um zu sehen, wie viel CPU und Arbeitsspeicher von Ihrem Broker genutzt wird.

- Amazon MQ für ActiveMQ— Überprüfen Sie `CpuUtilization` Der Prozentsatz der zugewiesenen Amazon EC2 Recheneinheiten, die zurzeit vom Broker genutzt werden. Überprüfen `HeapUsage` Sie den Prozentsatz des ActiveMQ JVM-Speicherlimits, den der Broker derzeit verwendet.
- Amazon MQ für RabbitMQ— Überprüfen Sie `SystemCpuUtilization` Der Prozentsatz der zugewiesenen Amazon EC2 Recheneinheiten, die zurzeit vom Broker genutzt werden. `CheckRabbitMQMemUsed` 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 Ihre Workload finden Sie unter [Broker instance types](#).

Fehlerbehebung: Amazon MQ für ActiveMQ

Verwenden Sie die Informationen in diesem Abschnitt, um häufige Probleme zu diagnostizieren und zu lösen, die beim Arbeiten mit Amazon MQ for ActiveMQ Broker auftreten können.

Inhalt

- [Ich kann keine allgemeinen oder Überwachungsprotokolle für meinen Broker in CloudWatch Logs sehen, obwohl ich die Protokollierung aktiviert habe.](#)
- [Nach dem Neustart oder dem Wartungsfenster des Brokers kann ich keine Verbindung zu meinem Broker herstellen, obwohl der Status RUNNING lautet. Warum?](#)
- [Ich sehe, dass einige meiner Clients eine Verbindung zum Broker herstellen, während andere keine Verbindung herstellen können.](#)
- [Ich sehe beim Ausführen von Operationen die Ausnahme `org.apache.jasper.JasperException: An exception occurred processing JSP page` auf der ActiveMQ-Konsole.](#)

Ich kann keine allgemeinen oder Überwachungsprotokolle für meinen Broker in CloudWatch Logs sehen, obwohl ich die Protokollierung aktiviert habe.

Wenn Sie keine Protokolle für Ihren Broker in CloudWatch Logs anzeigen können, gehen Sie folgendermaßen 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. Überprüfen Sie, ob Sie eine ressourcenbasierte Richtlinie konfiguriert haben, damit Amazon MQ Protokolle in CloudWatch Logs veröffentlichen kann. Um es Amazon MQ zu ermöglichen, Protokolle in Ihrer CloudWatch-Logs-Protokollgruppe zu veröffentlichen, konfigurieren Sie eine ressourcenbasierte Richtlinie, um Amazon MQ Zugriff auf die folgenden CloudWatch-Logs-API-Aktionen zu gewähren:
 - [CreateLogStream](#) – Erstellt einen CloudWatch-Logs-Protokollstream für die angegebene Protokollgruppe.
 - [PutLogEvents](#) - Stellt Ereignisse für den angegebenen CloudWatch Logs-Protokollstream bereit.

Weitere Informationen zum Konfigurieren von Amazon MQ für ActiveMQ für die Veröffentlichung von Protokollen in CloudWatch Logs finden Sie unter [Konfigurieren der Protokollierung](#).

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 Metriken auf CloudWatch für Ihren Amazon MQ für ActiveMQ Broker überwachen:

- **StoragePercentUsage** – Große Prozentsätze bei oder nahe 100 % können dazu führen, dass der Broker Verbindungen ablehnt.
- **JournalFilesForFullRecovery** – Gibt die Anzahl der Journaldateien an, die nach einem unreinen Shutdown und Neustart erneut abgespielt werden. Ist der Wert zunehmend bzw. konstant höher als Eins, weist dies auf ungelöste Transaktionen hin, die nach dem Neustart Verbindungsprobleme verursachen können.
- **OpenTransactionCount** – Eine Zahl größer als Null nach einem Neustart zeigt an, dass der Broker versucht, zuvor verbrauchte Nachrichten zu speichern, was zu Verbindungsproblemen führt.

Um dieses Problem zu beheben, empfehlen wir Ihnen, Ihre XA-Transaktionen mit einem `rollback()` oder `commit()` zu lösen. Weitere Informationen sowie ein Codebeispiel zum Lösen von XA-Transaktionen mit `rollback()`, finden Sie unter [Wiederherstellen von XA-Transaktionen](#).

Ich sehe, dass einige meiner Clients eine Verbindung zum Broker herstellen, während andere keine Verbindung herstellen können.

Wenn Ihr Broker im RUNNING-Status ist und einige Clients sich erfolgreich mit dem Broker verbinden können, während andere dies nicht tun können, haben Sie möglicherweise das Limit an [Wire-Level-Verbindungen](#) für den Broker erreicht. Gehen Sie wie folgt vor, um zu überprüfen, ob Sie das Wire-Level-Verbindungslimit erreicht haben:

- Überprüfen Sie die allgemeinen Broker-Protokolle für Ihren Amazon-MQ-für-ActiveMQ-Broker in CloudWatch Logs. Wenn das Limit erreicht wurde, sehen Sie `Reached Maximum Connections` in den Broker-Protokollen. Weitere Informationen zum Konfigurieren von CloudWatch Logs für Amazon-MQ-für-ActiveMQ-Broker finden Sie unter [the section called “Grundlegendes zum Aufbau der Protokollierung in CloudWatch Logs”](#).

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: Amazon MQ für RabbitMQ

Verwenden Sie die Informationen in diesem Abschnitt, um häufige Probleme zu diagnostizieren und zu lösen, die beim Arbeiten mit Amazon MQ for RabbitMQ Broker auftreten können.

Inhalt

- [In CloudWatch können keine Metriken für meine Warteschlangen oder virtuellen Hosts angezeigt werden.](#)
- [Wie aktiviere ich Plugins in Amazon MQ für RabbitMQ?](#)
- [Ich kann die Amazon-VPC-Konfiguration für den Broker nicht ändern.](#)

In CloudWatch können keine Metriken für meine Warteschlangen oder virtuellen Hosts angezeigt werden.

Wenn Sie keine Metriken für Ihre Warteschlangen oder virtuellen Hosts in CloudWatch anzeigen können, überprüfen Sie, ob Ihre Warteschlangen- oder virtuellen Hostnamen Leerzeichen, Tabulatorzeichen 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 Amazon MQ für RabbitMQ?

Amazon MQ für RabbitMQ unterstützt derzeit nur die RabbitMQ-Plugins für Verwaltung, Schaufel, Verbund und konsistenten Hash-Austausch, die standardmäßig aktiviert sind. Weitere Informationen zur Verwendung unterstützter Plugins finden Sie unter [the section called “Plug-ins”](#).

Ich kann die 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.

Fehlerbehebung: Erforderliche Codes für die Amazon-MQ-Aktion

Amazon MQ gibt eine Ausnahme für bestimmte API-Vorgänge zurück, z. B. [RebootBroker](#), wenn sich Ihr Broker in einem instabilen Zustand befindet und eine Reihe von Aktionen erforderlich sind,

um ihn wieder in einen stabilen Zustand zu überführen. Zu den Ausnahmen gehören spezifische erforderliche Aktionscodes, die Ihnen helfen, die Hauptursache zu identifizieren und das Problem zu beheben, um Ihren Broker wiederherzustellen.

Verwenden Sie die folgende Themenliste, um den erforderlichen Aktionscode zu identifizieren, den Sie erhalten haben, und erfahren Sie mehr über die Schritte, die wir zur Lösung Ihres Problems empfehlen.

Erforderliche Aktionscodes

- [Amazon MQ für RabbitMQ: Alarm über hohe Speicherauslastung](#)
- [Amazon MQ für RabbitMQ: Ungültiger AWS Key Management Service-Schlüssel](#)
- [Amazon MQ for ActiveMQ: Elastic-Network-Schnittstellenalarm wurde gelöscht](#)
- [Amazon MQ for ActiveMQ: Alarm -wegen zu geringem Arbeitsspeicher für Broker](#)
- [Amazon MQ für RabbitMQ: Festplattenlimit-Alarm](#)

Amazon MQ für RabbitMQ: Alarm über hohe Speicherauslastung

RabbitMQ löst einen Alarm über hohe Speicherauslastung aus, wenn die Speicherauslastung des Brokers, identifiziert durch CloudWatch-Metrik `RabbitMQMemUsed`, das Speicherlimit überschreitet, identifiziert durch `RabbitMQMemLimit`. `RabbitMQMemLimit` wurde von Amazon MQ festgelegt und wurde speziell unter Berücksichtigung des für jeden Host-Instance-Typ verfügbaren Arbeitsspeichers abgestimmt.

Ein Broker von Amazon MQ for RabbitMQ, der einen Alarm über hohe Speicherauslastung ausgelöst hat, blockiert alle Clients, die Nachrichten veröffentlichen. Aufgrund der hohen Speicherauslastung kann es bei Ihrem Broker auch andere Probleme geben, die die Diagnose und Auflösung des Alarms erschweren.

Broker für Einzel-Instances, die aufgrund der hohen Speicherauslastung den Start nicht abschließen können, gelangen möglicherweise in eine Neustartschleife, bei der die Interaktionen mit dem Broker begrenzt sind. In Clusterbereitstellungen kann es bei Warteschlangen zu einer pausierten Synchronisierung von Nachrichten zwischen Replikaten auf verschiedenen Knoten kommen. Pausierte Warteschlangensynchronisierungen verhindern den Verbrauch von Nachrichten aus Warteschlangen und müssen separat angesprochen werden, während der Speicheralarm aufgelöst wird.

Amazon MQ startet einen Broker nicht neu, der einen Alarm über hohe Speicherauslastung hat, und gibt eine Ausnahme für [RebootBroker](#)-API-Operationen zurück, solange der Broker weiterhin den Alarm auslöst.

Die Informationen in diesem Abschnitt helfen Ihnen bei der Diagnose und Behebung von RabbitMQ-Alarmen über hohe Speicherauslastung, die von Ihrem Broker ausgelöst werden.

Note

Nachdem Sie die erforderlichen Maßnahmen ergriffen haben, kann mehrere Stunden dauern, bis der Status `RABBITMQ_MEMORY_ALARM` gelöscht wird.

Note

Sie können einen Broker nicht von einem `mq.m5.`-Instance-Typ auf einen `mq.t3.micro`-Instance-Typ herunterstufen. Wenn Sie ein Downgrade durchführen möchten, müssen Sie Ihren Broker löschen und einen neuen erstellen.

Themen

- [Diagnostizieren eines Alarms über hohe Speicherauslastung mit der RabbitMQ-Webkonsole](#)
- [Diagnose eines Alarms über hohe Speicherauslastung mithilfe von Amazon-MQ-Metriken](#)
- [Umgang mit dem Alarm über hohe Speicherauslastung](#)
- [Reduzierung der Anzahl der Verbindungen und Kanäle](#)
- [Umgang mit pausierten Warteschlangensynchronisierungen in Clusterbereitstellungen](#)
- [Umgang mit Neustartschleifen in Einzel-Instance-Brokern](#)
- [Verhindern von Alarmen über hohe Speicherauslastung](#)

Diagnostizieren eines Alarms über hohe Speicherauslastung mit der RabbitMQ-Webkonsole

Die RabbitMQ-Webkonsole kann detaillierte Informationen zur Speicherauslastung für jeden Knoten generieren und anzeigen. Sie finden diese Informationen durch das folgende Verfahren:

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die RabbitMQ-Webkonsole Ihres Brokers.
2. Auf der RabbitMQ-Konsole wählen Sie auf der Seite Übersicht den Namen eines Knotens aus der Knoten-Liste aus.
3. Wählen Sie auf der Detailseite des Knotens die Option Details zum Speicher, um den Abschnitt zu erweitern und die Informationen zur Speicherauslastung des Knotens anzuzeigen.

Die Informationen zur Speicherauslastung, die RabbitMQ in der Webkonsole bereitstellt, können Ihnen helfen, festzustellen, welche Ressourcen möglicherweise zu viel Speicher verbrauchen und zum Alarm über hohe Speicherauslastung beitragen. Weitere Informationen zu den über die RabbitMQ-Web-Konsole verfügbaren Speicherauslastung finden Sie unter [Gründe für die Speichernutzung](#) auf der Website der RabbitMQ-Server-Dokumentation.

Diagnose eines Alarms über hohe Speicherauslastung mithilfe von Amazon-MQ-Metriken

Amazon MQ aktiviert standardmäßig Metriken für Ihren Broker. Sie können Ihre [Broker-Metriken anzeigen](#), 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-CloudWatch-Metrik	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

Amazon-MQ-CloudWatch-Metrik	Grund für eine hohe Speicherauslastung	
	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 werden, auch im Knotenspeicher gespeichert, und eine hohe Anzahl von Kanälen kann zu einem Alarm über hohe Speicherauslastung führen.	

Amazon-MQ-CloudWatch-Metrik	Grund für eine hohe Speicherauslastung
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.

Umgang mit dem Alarm über hohe Speicherauslastung

Für jeden Mitwirkenden, den Sie identifizieren, empfehlen wir die folgenden Aktionen, um den Alarm über hohe Speicherauslastung des Brokers zu mildern und aufzulösen.

Grund für eine hohe Speicherauslastung	Amazon MQ-Empfehlung	
<p>Die Anzahl der Nachrichten in der Warteschlange ist zu hoch.</p>	<p>Führen Sie eine der folgenden Aktionen aus:</p> <ul style="list-style-type: none">• Verbrauchen Sie Nachrichten, die in den Warteschlangen veröffentlicht wurden.• Löschen Sie Nachrichten aus den Warteschlangen.• Löschen Sie die Warteschlangen aus Ihrem Broker.	
<p>Die Anzahl der auf dem Broker konfigurierten Warteschlangen ist zu hoch.</p>	<p>Reduzieren Sie die Anzahl der Warteschlangen.</p>	
<p>Die Anzahl der auf dem Broker hergestellten Verbindungen ist zu hoch.</p>	<p>Reduzieren Sie die Anzahl der Verbindungen. Weitere Informationen finden Sie unter the section called “Reduzierung der Anzahl der Verbindungen und Kanäle”.</p>	
<p>Die Anzahl der auf dem Broker festgelegten Kanäle ist zu hoch.</p>	<p>Reduzieren Sie die Anzahl der Kanäle. Weitere Informationen finden Sie unter the section called “Reduzierung der Anzahl der Verbindungen und Kanäle”.</p>	
<p>Die Anzahl der Verbraucher, die mit dem Broker verbunden sind, ist zu hoch.</p>	<p>Reduzieren Sie die Gesamtzahl der Verbraucher, die mit dem Broker verbunden sind.</p>	

Grund für eine hohe Speicherauslastung	Amazon MQ-Empfehlung	
Die Veröffentlichungsrate von Nachrichten ist zu hoch.	Reduzieren Sie die Rate, mit der Nachrichten an den Broker veröffentlicht werden.	
Die Rate der Clientverbindungsversuche ist zu hoch.	Reduzieren Sie die Häufigkeit, mit der Clients versuchen, sich mit dem Broker zu verbinden, um Nachrichten zu veröffentlichen oder zu konsumieren, oder konfigurieren Sie den Broker.	

Reduzierung der Anzahl der Verbindungen und Kanäle

Verbindungen zu Ihrem Broker von Amazon MQ for RabbitMQ können entweder von Ihren Clientanwendungen oder durch manuelles Schließen über die RabbitMQ-Webkonsole geschlossen werden. Um eine Verbindung mit der RabbitMQ-Webkonsole zu schließen, gehen Sie wie folgt vor.

1. Melden Sie sich bei AWS Management Console an und öffnen Sie die RabbitMQ-Webkonsole Ihres Brokers.
2. Wählen Sie auf der RabbitMQ-Konsole die Registerkarte Verbindungen.
3. Wählen Sie auf der Seite Verbindungen unter Alle Verbindungen den Namen der Verbindung aus, die Sie aus der Liste schließen möchten.
4. Wählen Sie auf der Seite der Verbindungsdetails die Option Diese Verbindung schließen aus, um den Abschnitt zu erweitern, wählen Sie dann Schließen erzwingen aus. Optional können Sie den Standardtext für den Grund durch eine eigene Beschreibung ersetzen. Amazon MQ for RabbitMQ gibt den von Ihnen angegebenen Grund an den Client zurück, wenn Sie die Verbindung schließen.
5. Klicken Sie im Dialogfeld auf OK, um die Verbindung zu bestätigen und zu schließen.

Wenn Sie eine Verbindung schließen, werden alle Kanäle, die mit einer geschlossenen Verbindung verbunden sind, ebenfalls geschlossen.

Note

Ihre Clientanwendungen sind möglicherweise so konfiguriert, dass sie Verbindungen zum Broker automatisch wiederherstellen, nachdem sie geschlossen wurden. In diesem Fall reicht das Schließen von Verbindungen von der Broker-Webkonsole nicht aus, um die Verbindungs- oder Kanalanzahl zu reduzieren.

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.

Umgang mit pausierten Warteschlangensynchronisierungen in Clusterbereitstellungen

Während Sie sich um die Alarme über hohe Speicherauslastung von RabbitMQ kümmern, stellen Sie möglicherweise fest, dass Nachrichten in einer oder mehreren Warteschlangen nicht verbraucht werden können. Diese Warteschlangen synchronisieren möglicherweise Nachrichten zwischen Knoten, in denen die jeweiligen Warteschlangen für die Veröffentlichung und den Verbrauch nicht verfügbar sind. Warteschlangensynchronisierungen können aufgrund des Alarms über hohe Speicherauslastung pausiert werden und sogar zum Arbeitsspeicheralarm beitragen.

Informationen zum Stoppen und erneuten Versuchen der Synchronisierung von pausierten Warteschlangen finden Sie unter [the section called “Beheben der angehaltenen Warteschlangensynchronisierung”](#).

Umgang mit Neustartschleifen in Einzel-Instance-Brokern

Ein Einzel-Instance-Broker von Amazon MQ for RabbitMQ, der einen Alarm über hohe Speicherauslastung auslöst, läuft Gefahr, dass er nicht verfügbar ist, wenn er neu gestartet wird und nicht genügend Arbeitsspeicher zum Starten hat. Dies kann dazu führen, dass RabbitMQ in eine Neustartschleife gelangt und weitere Interaktionen mit dem Broker solange verhindert, bis das Problem behoben ist. Wenn sich Ihr Broker in einer Neustartschleife befindet, können Sie die von Amazon MQ empfohlenen Aktionen, die zuvor in diesem Abschnitt beschrieben wurden, nicht anwenden, um den Alarm über hohe Speicherauslastung zu beheben.

Um Ihren Broker wiederherzustellen, empfehlen wir, auf einen größeren Instance-Typ mit mehr Arbeitsspeicher zu aktualisieren. Im Gegensatz zu Clusterbereitstellungen können Sie einen Einzel-Instance-Broker aktualisieren, während ein Alarm über hohe Speicherauslastung auftritt, da während eines Neustarts keine Warteschlangensynchronisierungen zwischen Knoten durchgeführt werden müssen.

Verhindern von Alarmen über hohe Speicherauslastung

Für jeden von Ihnen identifizierten Faktor empfehlen wir die folgenden Maßnahmen zur Verhinderung und Verringerung des Auftretens von RabbitMQ-Alarmen über hohe Speicherauslastung.

Grund für eine hohe Speicherauslastung	Amazon-MQ-Empfehlung	
Die Anzahl der Nachrichten in der Warteschlange ist zu hoch.	<p>Gehen Sie wie folgt vor:</p> <ul style="list-style-type: none"> Aktivieren Sie Verzögerungswarteschlangen. Legen Sie den Grenzwert für die Warteschlangentiefe fest oder verringern Sie ihn. 	
Die Anzahl der auf dem Broker konfigurierten Warteschlangen ist zu hoch.	Legen Sie den Grenzwert für die Warteschlangenanzahl fest oder verringern Sie ihn.	
Die Anzahl der auf dem Broker hergestellten Verbindungen ist zu hoch.	Legen Sie den Grenzwert für die Verbindungsanzahl fest oder verringern Sie ihn.	
Die Anzahl der auf dem Broker festgelegten Kanäle ist zu hoch.	Legen Sie eine maximale Anzahl von Kanälen pro Verbindung für Clientanwendungen fest.	
Die Anzahl der Verbraucher, die mit dem Broker verbunden sind, ist zu hoch.	Legen Sie einen geringen Vorabrufgrenzwert für Verbraucher fest.	

Grund für eine hohe Speicherauslastung	Amazon-MQ-Empfehlung	
Die Rate der Clientverbindungsversuche ist zu hoch.	Verwenden Sie langlebigere Verbindungen, um die Anzahl und Häufigkeit von Verbindungsversuchen zu reduzieren.	

Nachdem der Arbeitsspeicheralarm Ihres Brokers behoben wurde, können Sie Ihren Host-Instance-Typ auf eine Instance mit zusätzlichen Ressourcen aktualisieren. Informationen zum Aktualisieren des Instance-Typs Ihres Brokers finden Sie unter [UpdateBrokerInput](#) in der Referenz zur Amazon MQ REST API.

Eine vollständige Liste der Broker-Instance-Typen finden Sie unter [the section called "Instance-Typen von Amazon MQ für RabbitMQ"](#).

Amazon MQ für RabbitMQ: Ungültiger AWS Key Management Service-Schlüssel

Amazon MQ für RabbitMQ gibt den Code `INVALID_KMS_KEY` für erforderliche kritische Aktionen aus, wenn ein Broker, der mit einem vom Kunden verwalteten AWS KMS key (CMK) erstellt wurde, feststellt, dass der AWS Key Management Service (KMS)-Schlüssel deaktiviert ist. 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

Wenn Sie den Fehlercode INVALID_KMS_KEY für erforderliche Aktionen diagnostizieren und beheben möchten, müssen Sie die AWS-Befehlszeilenschnittstelle (CLI) und die AWS Key Management Service-Konsole verwenden.

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 AWS Key Management Service-Konsole an.
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.

Wenn Sie überprüfen möchten, ob die erforderlichen Erteilungen immer noch mit dem KMS-Schlüssel des Brokers verknüpft sind, rufen Sie die `ListGrant`-Methode auf, um sich zu vergewissern, dass `mq_rabbit_grant` und `mq_grant` vorhanden sind. Wenn die KMS-Ertelung 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-Ertelung 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.

Amazon MQ for ActiveMQ: Elastic-Network-Schnittstellenalarm wurde gelöscht

Amazon MQ for ActiveMQ löst einen `BROKER_ENI_DELETED`-Alarm aus, wenn Sie die Elastic-Network-Schnittstelle (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

Ihren Broker führen. Wenn Sie die Netzwerkschnittstelle löschen möchten, müssen Sie zuerst den Broker löschen.

Amazon MQ for ActiveMQ: Alarm -wegen zu geringem Arbeitsspeicher für Broker

Amazon MQ for ActiveMQ löst einen `BROKER_OOM`-Alarm aus, wenn der Broker aufgrund unzureichender Speicherkapazität einer Neustartschleife unterzogen wird. 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 anzeigen, indem Sie auf die Amazon-CloudWatch-Konsole zugreifen oder die CloudWatch-API verwenden. Die folgenden Metriken sind beim Diagnostizieren des `ActiveMQ-BROKER_OOM`-Alarms nützlich:

Amazon-MQ-CloudWatch-Metrik	Grund für eine hohe Speicherauslastung	
TotalMessageCount	Nachrichten werden im Speicher gespeichert, bis sie verbraucht oder verworfen werden. Eine hohe Nachrichtenanzahl kann auf eine Überauslastung der Ressourcen hinweisen und zu einem Alarm über hohe Speicherauslastung führen.	
HeapUsage	Der prozentuale Anteil am ActiveMQ JVM-Speicherlimit, der vom Broker derzeit genutzt wird. Ein höherer Prozentsatz weist darauf hin, dass der Broker erhebliche Ressource	

Amazon-MQ-CloudWatch-Metrik	Grund für eine hohe Speicherauslastung
	n 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-Rechenheiten, die zurzeit vom Broker genutzt werden.
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 bei [Bewährte Methoden für Amazon MQ for ActiveMQ](#).

Amazon MQ für RabbitMQ: Festplattenlimit-Alarm

Der Datenträgerlimit-Alarm ist ein Hinweis darauf, dass das von einem RabbitMQ-Knoten verwendete Festplattenvolumen aufgrund einer hohen Anzahl von Nachrichten, die beim Hinzufügen neuer Nachrichten nicht verbraucht wurden, gesunken ist. RabbitMQ löst einen Festplattenlimit-Alarm aus, wenn der freie Datenträgerspeicher des Brokers, der anhand der Amazon-CloudWatch-Metrik `RabbitMQDiskFree` identifiziert wurde, das Festplattenlimit erreicht, das durch `RabbitMQDiskFreeLimit` identifiziert wurde. `RabbitMQDiskFreeLimit` wird von Amazon MQ festgelegt und wurde unter Berücksichtigung des verfügbaren Festplattenspeichers für jeden Broker-Instance-Typ definiert.

Ein Broker von Amazon MQ für RabbitMQ, der einen Festplattenlimit-Alarm ausgelöst hat, steht für die Veröffentlichung neuer Nachrichten nicht mehr zur Verfügung. Wenn RabbitMQ in einem Cluster ausgeführt wird, gilt der Festplattenalarm clusterweit. Wenn ein Knoten das Limit unterschreitet, werden eingehende Nachrichten von allen anderen Knoten blockiert. Aufgrund der mangelnden Festplattenspeichers können bei Ihrem Broker auch andere Probleme auftreten, die die Diagnose und Auflösung des Alarms erschweren.

Amazon MQ startet einen Broker 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-Alarmes

Amazon MQ aktiviert standardmäßig Metriken für Ihren Broker. Sie können [Ihre Broker-Metriken anzeigen](#), indem Sie auf die Amazon-CloudWatch-Konsole zugreifen oder indem Sie die CloudWatch-API verwenden. `MessageCount` ist eine Metrik, die für die Diagnose des Festplattenlimit-Alarmes von RabbitMQ nützlich ist. Nachrichten werden im Speicher gespeichert, bis sie verwendet oder verworfen werden. Eine hohe Nachrichtenanzahl weist auf eine Überauslastung des Festplattenspeichers hin und kann zu einem Festplattenalarm führen.

Verwenden Sie die Amazon MQ Managementkonsole, damit Sie den Festplattenlimit-Alarm diagnostizieren können, um:

- Verwenden Sie Nachrichten, die in den Warteschlangen veröffentlicht wurden.
- Löschen Sie Nachrichten aus den Warteschlangen.
- Löschen Sie die Warteschlangen aus Ihrem Broker.

 Note

Es kann mehrere Stunden dauern, bis der 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.

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	<p>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.</p> <p>Mit der AWS::AmazonMQ::Configuration -Ressource können Sie Amazon MQ-Konfigurationen erstellen, Konfigurationsänderungen hinzufügen oder Benutzer ändern und Informationen über die angegebene Konfiguration zurückgeben.</p>
Regionen und Endpunkte	Informationen zu Amazon MQ-Regionen und -Endpunkten
Produktseite	Hauptwebsite für Informationen zu Amazon MQ.

Ressource	Beschreibung
Diskussionsforum	Ein auf der Community basierendes Forum, das für Entwickler eingerichtet wurde, um technische Fragen zu Amazon MQ zu klären
AWS Informationen zu Premium Support	Die Hauptwebsite mit Informationen zu AWS Premium Support ist ein persönlicher und reaktionsschneller Supportkanal. Er bietet Ihnen Hilfe beim Konfigurieren und Verwenden von Anwendungen auf AWS Infrastructure Services.

Amazon MQ für ActiveMQ-Ressourcen

In der folgenden Tabelle werden nützliche Ressourcen für die Arbeit mit Apache ActiveMQ aufgeführt.

Ressource	Beschreibung
Apache ActiveMQ – Handbuch Erste Schritte	Die offizielle Dokumentation für Apache ActiveMQ.
ActiveMQ in Aktion	Ein Handbuch für Apache ActiveMQ, das den Aufbau von JMS-Nachrichten, Verbindungsselementen, Mitteilungspersistenz, Authentifizierung und Autorisierung abdeckt.
Cross-Language-Clients	Eine Liste der Programmiersprachen und der entsprechenden Apache ActiveMQ-Bibliotheken. Siehe auch ActiveMQ-Client und QpidJMS-Client .

Amazon MQ für RabbitMQ-Ressourcen

In der folgenden Tabelle werden nützliche Ressourcen für die Arbeit mit RabbitMQ aufgeführt.

Ressource	Beschreibung
Das RabbitMQ Handbuch „Erste Schritte“	Die offizielle Dokumentation für RabbitMQ.
RabbitMQ Client-Bibliotheken und Entwickler-Tools	Ein Leitfaden zu den offiziell unterstützten Client-Bibliotheken und Devloper-Tools für die Arbeit mit RabbitMQ unter Verwendung einer Vielzahl von Programmiersprachen und Plattformen.
Bewährte Methoden für RabbitMQ	Bewährte Methoden und Empfehlungen für die Arbeit mit RabbitMQ.

Versionshinweise zu Amazon MQ

Die folgende Tabelle listet neu eingeführte und verbesserte Amazon MQ-Funktionen auf. Änderungen am Amazon MQ Entwicklerhandbuch finden Sie unter [Amazon MQ Dokumentverlauf](#).

Datum	Aktualisierung der Dokumentation
4. März 2023	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ 3.11.28.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none"> • Versionshinweise zu RabbitMQ 3.11.28 im RabbitMQ-Server- GitHub Repository • RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>
19. Januar 2024	<p>Amazon MQ für RabbitMQ unterstützt den Benutzernamen „Gast“ nicht und löscht das Standard-Gastkonto, wenn Sie einen neuen Broker erstellen . Amazon MQ löscht auch regelmäßig alle vom Kunden erstellten Konten namens „Gast“ .</p>
15. Dezember 2023	<p>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.</p>
11. Dezember 2023	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ 3.10.25.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none"> • Versionshinweise zu RabbitMQ 3.10.25 im RabbitMQ-Server- GitHub Repository • RabbitMQ-Änderungsprotokoll

Datum	Aktualisierung der Dokumentation
	Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen .
26. Oktober 2023	<p>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.</p> <p>Weitere Informationen zur Aktualisierung Ihres ActiveMQ-Brokers finden Sie unter Verwalten von Amazon MQ für ActiveMQ Engine-Versionen.</p>
27. September 2023	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ 3.11.20.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.11.20 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>

Datum	Aktualisierung der Dokumentation
27. Juli 2023	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ 3.11.16.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.11.16 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>
27. Juli 2023	<p>Amazon MQ für RabbitMQ unterstützt jetzt das Erstellen und Anwenden von Konfigurationen auf Ihren RabbitMQ-Broker.</p> <p>Weitere Informationen zum Hinzufügen von Konfigurationen zu Ihrem Broker finden Sie unter RabbitMQ Broker Configurations.</p> <p>Weitere Informationen über dieses Feature finden Sie unter:</p> <ul style="list-style-type: none">• Richtlinien für Betreiber• Änderungen der Richtlinien für Betreiber

Datum	Aktualisierung der Dokumentation
23. Juni 2023	<p>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.</p> <p>Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• 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• Arbeiten mit Spring XML-Konfigurationsdateien
21. Juni 2023	<p>Amazon MQ für ActiveMQ bietet jetzt eine regionsübergreifende Datenreplikationsfunktion (CRDR), die eine asynchrone Nachrichtenreplikation vom Primär-Broker in einer primären AWS Region zum Replikat-Broker 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.</p> <p>Informationen zu den ersten Schritten mit CRDR finden Sie unter Regionsübergreifende Datenreplikation für Amazon MQ für ActiveMQ im Entwicklerhandbuch.</p>
18. Mai 2023	<p>Amazon MQ ist jetzt in den folgenden Regionen verfügbar:</p> <ul style="list-style-type: none">• Asien-Pazifik (Melbourne)• Asien-Pazifik (Hyderabad)• Europa (Spain)• Europa (Zürich) <p>Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.</p>

Datum	Aktualisierung der Dokumentation
14. April 2023	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ der Version 3.9.27.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.9.27 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>
14. April 2023	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.10.20.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.10.20 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>


Datum	Aktualisierung der Dokumentation
31. März 2023	<p>Amazon MQ für RabbitMQ hat Version 3.10.17 der RabbitMQ-Engine deaktiviert</p> <p>Das Team von Amazon MQ für RabbitMQ und die Open-Source-Maintainer von RabbitMQ haben ein Problem mit der RabbitMQ-Managementkonsole in Version 3.10.17 festgestellt. Amazon MQ hat diese Version zurückgezogen. Um die Auswirkungen dieses Problems zu mildern, erstellen Sie neue Broker mit Version 3.10.20, während wir daran arbeiten, eine neue Patch-Version von RabbitMQ zu unterstützen. Wir empfehlen, die Option Automatisches Unterversion-Upgrade zu aktivieren, um die neuesten Fehlerbehebungen, Sicherheitsupdates und Leistungsverbesserungen automatisch zu erhalten.</p> <p>Weitere Informationen zu verfügbaren Versionen von Amazon MQ für RabbitMQ finden Sie unter Engine-Versionen von Amazon MQ für RabbitMQ.</p>
1. März 2023	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.10.17.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.10.17 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>

Datum	Aktualisierung der Dokumentation
21. Februar 2023	<p>Amazon MQ für RabbitMQ lässt sich jetzt in AWS Key Management Service (KMS) integrieren, um serverseitige Verschlüsselung zu bieten. Sie können jetzt Ihren eigenen kundenverwalteten CMK auswählen oder einen - AWS verwalteten KMS-Schlüssel in Ihrem AWS KMS Konto verwenden. Weitere Informationen finden Sie unter Verschlüsselung im Ruhezustand.</p> <p>Amazon MQ unterstützt die Verwendung von AWS KMS Schlüsseln auf folgende Weise.</p> <ul style="list-style-type: none">• 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.• Von AWS verwalteter KMS-Schlüssel – Der von AWS verwaltete KMS-Schlüssel (aws/mq) ist ein KMS-Schlüssel in Ihrem Konto, der von Amazon MQ in Ihrem Namen 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	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.34.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.8.34 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>

Datum	Aktualisierung der Dokumentation
15. Dezember 2022	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ der Version 3.9.24.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.9.24 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>
13. Dezember 2022	<p>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.</p>
14. November 2022	<p>Amazon MQ für RabbitMQ unterstützt jetzt 3.10, eine Hauptversion der Engine. Sie können jetzt Classic Queues Version 2 (CQv2) für Ihre RabbitMQ-Warteschlangen aktivieren. Direkte Updates von 3.8 auf 3.10 werden nicht unterstützt. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.10.10• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>

Datum	Aktualisierung der Dokumentation
9. November 2022	<p>Amazon MQ unterstützt jetzt ActiveMQ 5.17.2, eine neue Engine-Unterversion. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.17.2 – Versionsseite• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Aktualisieren einer Amazon MQ-Broker-Engine-Version• Arbeiten mit Spring XML-Konfigurationsdateien
17. August 2022	<p>Amazon MQ unterstützt jetzt ActiveMQ 5.17.1, eine neue Hauptversion der Engine. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.17.1 – Versionsseite• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Aktualisieren einer Amazon MQ-Broker-Engine-Version• Arbeiten mit Spring XML-Konfigurationsdateien
14. Juli 2022	<p>Amazon MQ unterstützt jetzt ActiveMQ 5.16.5, eine neue Engine-Unterversion. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.16.5 – Versionsseite• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Arbeiten mit Spring XML-Konfigurationsdateien• Aktualisieren einer Amazon MQ-Broker-Engine-Version
4. Mai 2022	<p>Amazon MQ fügt inklusive Sprache für das <code>networkConnector</code> -Element in der Broker-Konfiguration hinzu.</p> <ul style="list-style-type: none">• Erstellen und Konfigurieren eines Amazon MQ-Netzwerks von Brokern

Datum	Aktualisierung der Dokumentation
25. April 2022	<p>Amazon MQ: Diese Version fügt den <code>CRITICAL_ACTION_REQUIRED</code> - Broker-Status und die <code>ActionRequired</code> -API-Eigenschaft hinzu. <code>CRITICAL_ACTION_REQUIRED</code> informiert Sie, wann Ihr Broker heruntergestuft wird. <code>ActionRequired</code> stellt Ihnen einen Code zur Verfügung, mit dem Sie im Entwicklerhandbuch Anweisungen zur Behebung des Problems finden können.</p> <ul style="list-style-type: none">• the section called “Fehlerbehebung: Erforderliche Codes für die Amazon-MQ-Aktion”• ActionRequired -Dokumentation in der Amazon-MQ-API-Referenz.
20. April 2022	<p>Amazon MQ unterstützt jetzt ActiveMQ 5.16.4, eine neue Engine-Unterversion. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.16.4 – Versionsseite• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Arbeiten mit Spring XML-Konfigurationsdateien• Aktualisieren einer Amazon MQ-Broker-Engine-Version
1. März 2022	<p>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.</p>
25. Februar 2022	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.27.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.8.27 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>

Datum	Aktualisierung der Dokumentation
16. Februar 2022	Amazon MQ ist jetzt in der Region Afrika (Kapstadt) verfügbar. Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
14. Februar 2022	<p>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.</p> <p>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 - GitHub Website.</p> <div data-bbox="402 800 1507 1066" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Derzeit unterstützt Amazon MQ keine Streams oder die Verwendung der strukturierten Protokollierung in JSON, die in RabbitMQ 3.9 eingeführt wurde.</p></div> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.9.13 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>


Datum	Aktualisierung der Dokumentation
07. Februar 2022	<p>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.</p> <p>Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• the section called “Protokollierung und Überwachung von Amazon MQ für RabbitMQ Broker”
18. Januar 2022	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.26.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.8.26 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>
13. Januar 2022	<p>Amazon MQ führt den <code>RABBITMQ_MEMORY_ALARM</code> -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.</p> <ul style="list-style-type: none">• the section called “ RABBITMQ_MEMORY_ALARM ”

Datum	Aktualisierung der Dokumentation
6. Januar 2022	<p>Wenn Sie CloudWatch Protokolle für Amazon MQ für ActiveMQ-Broker konfigurieren, unterstützt Amazon MQ die Verwendung der aws:SourceAccount globalen Bedingungskontextschlüssel aws:SourceArn und in ressourcenbasierten IAM-Richtlinien, um das Confused-Deputy-Problem zu vermeiden. Weitere Informationen finden Sie unter den folgenden Topics.</p> <ul style="list-style-type: none">• the section called “Dienstübergreifende Confused-Deputy-Prävention”
20. Dezember 2021	<p>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 überwachen können. Weitere Informationen finden Sie unter den folgenden Topics.</p> <ul style="list-style-type: none">• the section called “Protokollieren und Überwachen von Amazon MQ für ActiveMQ Broker”
16. November 2021	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.23.</p> <p>Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.8.23 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.</p>

Datum	Aktualisierung der Dokumentation
12. Oktober 2021	<p>Amazon MQ unterstützt jetzt ActiveMQ 5.16.3, eine neue Engine-Unterversion. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.16.3 – Versionsseite• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Aktualisieren einer Amazon MQ-Broker-Engine-Version• Arbeiten mit Spring XML-Konfigurationsdateien
8. September 2021	<p>Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.22.</p> <p>Diese Version enthält eine Korrektur für ein Problem mit Warteschlangen mit Time to Message TTL (Time to Live, Gültigkeitsdauer), identifiziert in der zuvor unterstützten Version, RabbitMQ 3.8.17. Wir empfehlen, Ihre vorhandenen Broker auf Version 3.8.22 zu aktualisieren.</p> <p>Weitere Informationen zu den Updates und Features in diesem Release finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.8.22 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll <p>Weitere Informationen zu den unterstützten Amazon MQ für RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen</p>
25. August 2021	<p>Amazon MQ für RabbitMQ hat die RabbitMQ-Engine-Version 3.8.17 vorübergehend deaktiviert, da ein Problem mit Warteschlangen mit pro Nachricht time-to-live (TTL) identifiziert wurde. Wir empfehlen die Verwendung der Version 3.8.11.</p>

Datum	Aktualisierung der Dokumentation
29. Juli 2021	<p>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:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.8.17 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll• Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen
16. Juli 2021	<p>Sie können jetzt das Wartungsfenster eines Amazon MQ-Brokers mithilfe der oder AWS Management Console AWS CLI der Amazon MQ-API anpassen. Weitere Informationen zu Broker-Wartungsfenstern finden Sie hier.</p> <ul style="list-style-type: none">• Verwalten eines Amazon MQ-Brokers
6. Juli 2021	<p>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:</p> <ul style="list-style-type: none">• Consistent Hash Exchange Plugin• RabbitMQ Consistent Hash Exchange-Typ im RabbitMQ GitHub -Repository
7. Juni 2021	<p>Amazon MQ unterstützt jetzt ActiveMQ 5.16.2, eine neue Hauptversion der Engine. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.16.2 – Versionsseite• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Aktualisieren einer Amazon MQ-Broker-Engine-Version• Arbeiten mit Spring XML-Konfigurationsdateien

Datum	Aktualisierung der Dokumentation
26. Mai 2021	Amazon MQ für RabbitMQ ist jetzt in den Regionen China (Peking) und China (Ningxia) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte .
18. Mai 2021	<p>Amazon MQ für RabbitMQ implementiert Broker-Standardwerte.</p> <p>Wenn Sie zum ersten Mal einen Broker erstellen, erstellt Amazon MQ eine Reihe von Broker-Richtlinien und Vhost-Limits basierend auf dem von Ihnen gewählten Instance-Typ und Bereitstellungsmodus, um die Leistung des Brokers zu optimieren. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Standardwerte für Amazon MQ für RabbitMQ Broker
5. Mai 2021	<p>Amazon MQ unterstützt jetzt ActiveMQ 5.15.15. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.15.15 – Versionsseite• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Arbeiten mit Spring XML-Konfigurationsdateien
5. Mai 2021	<p>Amazon MQ hat mit der Verfolgung von Änderungen an - AWS verwalteten Richtlinien begonnen. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• the section called “Von 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	<p>Amazon MQ unterstützt jetzt RabbitMQ 3.8.11. Weitere Informationen zu den in diesem Update enthaltenen Updates und Features finden Sie hier:</p> <ul style="list-style-type: none">• Versionshinweise zu RabbitMQ 3.8.11 im RabbitMQ-Server- GitHub Repository• RabbitMQ-Änderungsprotokoll• Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen


Datum	Aktualisierung der Dokumentation
01. April 2021	Amazon MQ ist jetzt in der Region Asien-Pazifik (Osaka) verfügbar. Informationen zu den verfügbaren Regionen finden Sie unter Amazon MQ Regionen und Endpunkte .
21. Dezember 2020	<p>Amazon MQ unterstützt jetzt ActiveMQ 5.15.14. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.15.14 – Versionshinweise• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Arbeiten mit Spring XML-Konfigurationsdateien• <div data-bbox="431 697 1510 1062" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> Important</p><p>Aufgrund eines bekannten Apache ActiveMQ Problems in dieser Version wird die neue Pause-Warteschlange in der ActiveMQ Webkonsole nicht mit Amazon MQ für ActiveMQ-Broker verwendet werden. Informationen zum Beheben dieses Problems finden Sie unter AMQ-8104.</p></div>

Datum	Aktualisierung der Dokumentation
4. November 2020	<p>Amazon MQ unterstützt jetzt RabbitMQ, ein beliebter Open-Source-Nachrichtenbroker. Auf diese Weise können Sie Ihre vorhandenen RabbitMQ-Message Broker zu migrieren, AWS ohne Code neu schreiben zu müssen.</p> <p>Amazon MQ für RabbitMQ verwaltet sowohl einzelne als auch geclusterte Nachrichtenbroker und übernimmt Aufgaben wie das Bereitstellen der Infrastruktur, das Einrichten des Brokers und das Aktualisieren der Software.</p> <ul style="list-style-type: none">• Amazon MQ unterstützt RabbitMQ 3.8.6. Weitere Informationen zu unterstützten Engine-Versionen finden Sie unter the section called “Versionsverwaltung.”.• Das AWS kostenlose Kontingent umfasst bis zu 750 Stunden einer Einzel-Instance-<code>mq.t3.micro</code> -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-Bibliothek 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 verfügbaren Regionen finden Sie in der AWS -Regionentabelle. <p>Um mit Amazon MQ zu beginnen, einen Broker zu erstellen und eine JVM-basierte Anwendung mit Ihrem RabbitMQ-Broker zu verbinden, siehe the section called “Erstellen und Verbinden mit einem RabbitMQ-Broker”.</p>
22. Oktober 2020	<p>Amazon MQ unterstützt ActiveMQ 5.15.13. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.15.13 – Versionshinweise• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Arbeiten mit Spring XML-Konfigurationsdateien

Datum	Aktualisierung der Dokumentation
30. September 2020	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 authentifizieren, 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 <code>diemq.t3.micro</code> -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: <ul style="list-style-type: none">• ActiveMQ 5.15.12 – Versionshinweise• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Arbeiten mit Spring XML-Konfigurationsdateien

Datum	Aktualisierung der Dokumentation
30. April 2020	<p>Amazon MQ unterstützt ein neues untergeordnetes Sammlungselement, <code>systemUsage</code>, für das Element <code>broker</code>. Weitere Informationen finden Sie unter systemUsage.</p> <p>Amazon MQ unterstützt auch drei neue Attribute für das untergeordnete <code>kahaDB</code>-Element.</p> <ul style="list-style-type: none">• <code>journalDiskSyncInterval</code> – Intervall (ms), wann eine Datenträger synchronisierung durchgeführt werden soll, wenn <code>journalDiskSyncStrategy=periodic</code>.• <code>journalDiskSyncStrategy</code> – konfiguriert die Richtlinie für die Datenträgersynchronisierung.• <code>preallocationStrategy</code> – konfiguriert, wie der Broker versucht, die Journaldateien vorab zuzuweisen, wenn eine neue Journaldatei benötigt wird. <p>Weitere Informationen finden Sie unter Attribute.</p>
3. März 2020	<p>Amazon MQ unterstützt zwei neue CloudWatch Metriken</p> <ul style="list-style-type: none">• <code>TempPercentUsage</code> – der Anteil des verfügbaren temporären Speichers, der von nicht persistenten Nachrichten verwendet wird.• <code>JobSchedulerStorePercentUsage</code> – der Anteil des Festplattenspeichers, der vom Speicher des Aufgaben-Schedulers belegt wird. <p>Weitere Informationen finden Sie unter Monitoring Amazon MQ using CloudWatch.</p>
4. Februar 2020	<p>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.</p>

Datum	Aktualisierung der Dokumentation
22. Januar 2020	<p>Amazon MQ unterstützt ActiveMQ 5.15.10. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.15.10 – Versionshinweise• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Arbeiten mit Spring XML-Konfigurationsdateien
19. Dezember 2019	<p>Amazon MQ in den Regionen EU (Stockholm) und Südamerika (São Paulo) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte.</p>


Datum	Aktualisierung der Dokumentation
16. Dezember 2019	<p>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.</p> <div data-bbox="402 541 1507 1041" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> Important</p><ul style="list-style-type: none">• Sie können Amazon EBS nur mit dem <code>mq.m5Broker</code>-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.</div> <p>Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Storage• Auswählen des richtigen Broker-Speichertyps für den besten Durchsatz• Die <code>storageType</code> -Eigenschaft der broker-instance-options -Ressource in der Amazon-MQ-REST-API-Referenz• Die <code>BurstBalance</code> -, <code>VolumeReadOps</code> - und <code>VolumeWriteOps</code> - Metriken im Amazon MQ für ActiveMQ Metriken-Abschnitt.
18. Oktober 2019	<p>Zwei Amazon- CloudWatch Metriken sind verfügbar: <code>TotalEnqueueCount</code> und <code>TotalDequeueCount</code> . Weitere Informationen finden Sie unter ActiveMQ-Ziel-Metriken (Warteschlange und Thema).</p>

Datum	Aktualisierung der Dokumentation
11. Oktober 2019	<p>Amazon MQ unterstützt jetzt Federal Information Processing Standard 140-2 (FIPS)-konforme Endpunkte in US-Handelsregionen.</p> <p>Weitere Informationen finden Sie unter den folgenden Topics:</p> <ul style="list-style-type: none">• Federal Information Processing Standard (FIPS) 140-2• Amazon-MQ-Regionen und -Endpunkte
30. September 2019	<p>Amazon MQ bietet jetzt die Möglichkeit, Ihre Broker durch Ändern des Host-Instance-Typs zu skalieren. Weitere Informationen finden Sie in der <code>hostInstanceType</code> -Eigenschaft von UpdateBrokerInput und in der <code>pendingHostInstanceType</code> -Eigenschaft von DescribeBrokerOutput.</p>
30. August 2019	<p>Sie können jetzt die einem Broker zugeordneten Sicherheitsgruppen sowohl in der Konsole als auch mit UpdateBrokerInput aktualisieren.</p>
22. Juli 2019	<p>Amazon MQ lässt sich in AWS Key Management Service (KMS) integrieren, um serverseitige Verschlüsselung zu bieten. Sie können jetzt Ihren eigenen kundenverwalteten CMK auswählen oder einen - AWS verwalteten KMS-Schlüssel in Ihrem AWS KMS Konto verwenden. Weitere Informationen finden Sie unter Verschlüsselung im Ruhezustand.</p> <p>Amazon MQ unterstützt die Verwendung von AWS KMS Schlüsseln auf folgende Weise.</p> <ul style="list-style-type: none">• AWS -eigener KMS-Schlüssel – Der Schlüssel gehört Amazon MQ und befindet sich nicht in Ihrem Konto.• Von AWS verwalteter KMS-Schlüssel – Der von AWS verwaltete KMS-Schlüssel (<code>aws/mq</code>) ist ein KMS-Schlüssel in Ihrem Konto, der von Amazon MQ in Ihrem Namen erstellt, verwaltet und verwendet wird.• Wählen Sie vorhandene kundenverwaltete CMK— Customer Managed CMKs werden von Ihnen in AWS Key Management Service (KMS) erstellt und verwaltet.

Datum	Aktualisierung der Dokumentation
19. Juni 2019	Amazon MQ ist in den Regionen Europa (Paris) und Asien-Pazifik (Mumbai) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte .
12. Juni 2019	Amazon MQ ist in der Region Kanada (Zentral) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte .
3. Juni 2019	Zwei neue Amazon- CloudWatch Metriken sind verfügbar: <code>EstablishedConnectionsCount</code> und <code>InactiveDurableSubscribers</code> . Weitere Informationen finden Sie hier: <ul style="list-style-type: none">• Monitoring Amazon MQ using CloudWatch• Amazon MQ für ActiveMQ Metriken
10. Mai 2019	Der Datenspeicher für neue <code>mq.t2.micro</code> -Instance-Typen wurde auf 20 GB beschränkt. Weitere Informationen finden Sie hier: <ul style="list-style-type: none">• 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: <ul style="list-style-type: none">• 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: <ul style="list-style-type: none">• Broker-Instance-Optionen• Broker-Engine-Typen



Datum	Aktualisierung der Dokumentation
8. April 2019	<p>Amazon MQ unterstützt ActiveMQ 5.15.9. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• ActiveMQ 5.15.9 – Versionshinweise• Verwalten von Amazon MQ für ActiveMQ Engine-Versionen• Arbeiten mit Spring XML-Konfigurationsdateien
4. März 2019	<p>Verbesserte Dokumentation für die Konfiguration des dynamischen Failover und die Anpassung von Clients für ein Netzwerk von Brokern. Aktivieren Sie das dynamische Failover durch die Konfiguration von <code>transportConnectors</code> zusammen mit den <code>networkConnectors</code>-Konfigurationsoptionen. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Dynamisches Failover mit Transport Connectors• Amazon MQ Netzwerk von Brokern• Amazon MQ Broker Configuration Parameters
27. Februar 2019	<p>Amazon MQ ist in der Region Europa (London), sowie in folgenden Regionen verfügbar:</p> <ul style="list-style-type: none">• 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	<p>Die Standardkonfiguration enthält jetzt eine Richtlinie zum Löschen inaktiver Ziele.</p>



Datum	Aktualisierung der Dokumentation
17. Januar 2019	Amazon MQ <code>mq.t2.micro</code> -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: <ul style="list-style-type: none">• Amazon MQ Netzwerk von Brokern• Creating and Configuring a Network of Brokers• Korrekte Konfiguration Ihres Netzwerk von Brokern• networkConnector• networkConnectionStartAsync
11. Dezember 2018	Amazon MQ unterstützt ActiveMQ 5.15.8, 5.15.6 und 5.15.0. <ul style="list-style-type: none">• Fehlerbehebungen und Verbesserungen in ActiveMQ:<ul style="list-style-type: none">• ActiveMQ 5.15.8 – Versionshinweise• ActiveMQ 5.15.7 – Versionshinweise
5. Dezember 2018	AWS unterstützt das Markieren von Ressourcen, um Ihre Kostenzuordnung nachzuverfolgen. 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 erweitert, um Amazon MQ als SOC-konformen Service einzuschließen.
15. Oktober 2018	<ul style="list-style-type: none">• 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 erweitert, um Amazon MQ als HIPAA-fähigen Service einzuschließen.

Datum	Aktualisierung der Dokumentation
27. September 2018	<p>Amazon MQ unterstützt ActiveMQ 5.15.6, zusätzlich zu 5.15.0. Weitere Informationen finden Sie hier:</p> <ul style="list-style-type: none">• Tutorial: Bearbeiten der Broker-Engine-Version, des Instance-Typs, von CloudWatch Logs und von Wartungseinstellungen• Fehlerbehebungen und Verbesserungen in der ActiveMQ-Dokumentation:<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Die folgenden Metriken sind verfügbar:<ul style="list-style-type: none">• <code>CurrentConnectionsCount</code>• <code>TotalConsumerCount</code>• <code>TotalProducerCount</code> <p>Weitere Informationen finden Sie im Abschnitt Amazon MQ für ActiveMQ Metriken.</p> <ul style="list-style-type: none">• Die IP-Adresse des Brokers wird auf der Seite Details angezeigt. <div data-bbox="431 1373 1507 1591" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"><p> Note</p><p>Für Broker mit deaktivierter öffentlicher Zugänglichkeit wird die interne IP-Adresse angezeigt.</p></div>

Datum	Aktualisierung der Dokumentation
30. August 2018	<p>Amazon MQ ist in der Region Asien-Pazifik (Singapur), sowie in folgenden Regionen verfügbar:</p> <ul style="list-style-type: none">• 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	<p>Sie können Amazon MQ so konfigurieren, dass allgemeine Protokolle und Prüfungsprotokolle in Amazon CloudWatch Logs veröffentlicht werden. Weitere Informationen finden Sie unter Configuring Amazon MQ to publish logs to Amazon CloudWatch Logs.</p>
25. Juli 2018	<p>Amazon MQ ist in den Regionen Asien-Pazifik (Tokio) und Asien-Pazifik (Seoul), sowie in folgenden Regionen verfügbar:</p> <ul style="list-style-type: none">• US East (Ohio)• USA Ost (Nord-Virginia)• USA West (Nordkalifornien)• USA West (Oregon)• Asien-Pazifik (Sydney)• Europe (Frankfurt)• Europa (Irland)
19. Juli 2018	<p>Sie können verwenden AWS CloudTrail , um Amazon MQ-API-Aufrufe zu protokollieren. Weitere Informationen finden Sie unter Logging Amazon MQ API calls using CloudTrail.</p>

Datum	Aktualisierung der Dokumentation
29. Juni 2018	<p>Zusätzlich zu <code>mq.t2.micro</code> und <code>mq.m4.large</code> stehen die folgenden Broker-Instance-Typen für reguläre Entwicklungs-, Test- und Produktions-Workloads zur Verfügung, die einen hohen Durchsatz erfordern:</p> <ul style="list-style-type: none">• <code>mq.m5.large</code>• <code>mq.m5.xlarge</code>• <code>mq.m5.2xlarge</code>• <code>mq.m5.4xlarge</code> <p>Weitere Informationen finden Sie unter Broker instance types.</p>
27. Juni 2018	<p>Amazon MQ ist in der Region USA West (Nordkalifornien), sowie in folgenden Regionen verfügbar:</p> <ul style="list-style-type: none">• US East (Ohio)• USA Ost (Nord-Virginia)• USA West (Oregon)• Asien-Pazifik (Sydney)• Europe (Frankfurt)• Europa (Irland)

Datum	Aktualisierung der Dokumentation
14. Juni 2018	<ul style="list-style-type: none">• Sie können die -AWS::Amazon MQ::Broker AWS CloudFormation Ressource verwenden, um die folgenden Aktionen auszuführen:<ul style="list-style-type: none">• 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. <div data-bbox="435 632 1507 894" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"><p> Note</p><p>Wenn Sie eine Eigenschaft des Eigenschaftstyps Amazon MQ Broker ConfigurationId oder Amazon MQ Broker User ändern, wird der Broker sofort neu gestartet.</p></div> <ul style="list-style-type: none">• Sie können die -AWS::Amazon MQ::Configuration AWS CloudFormation Ressource verwenden, um die folgenden Aktionen auszuführen:<ul style="list-style-type: none">• Erstellen einer Konfiguration.• Aktualisieren der angegebenen Konfiguration.• Rückgabe von Informationen über die angegebene Konfiguration. <div data-bbox="435 1209 1507 1430" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"><p> Note</p><p>Sie können verwenden, AWS CloudFormation um eine Amazon MQ-Konfiguration zu ändern, aber nicht zu löschen.</p></div>
7. Juni 2018	Die Amazon MQ-Konsole unterstützt Deutsch, Brasilianisches Portugiesisch, Spanisch, Italienisch und Traditionelles Chinesisch.
17. Mai 2018	Die Anzahl der Benutzer pro Broker ist auf 250 begrenzt. Weitere Informationen finden Sie unter Benutzer .
13. März 2018	Das Erstellen eines Brokers dauert ca. 15 Minuten. Weitere Informationen finden Sie unter Abschließen der Broker-Erstellung .

Datum	Aktualisierung der Dokumentation
1. März 2018	<ul style="list-style-type: none">• Sie können die gleichzeitige Speicherung und Bereitstellung für Apache KahaDB mit dem Attribut concurrentStoreAndDispatchQueues konfigurieren.• Die -CpuCreditBalance CloudWatch Metrik ist für den mq.t2.micro Broker-Instance-Typ verfügbar.
10. Januar 2018	<p>Die folgenden Änderungen wirken sich auf die Amazon MQ-Konsole aus:</p> <ul style="list-style-type: none">• In der Liste der Broker ist die Spalte Erstellung standardmäßig ausgeblendet. Wählen Sie zum Anpassen der Seitengröße und der Spalten  aus.• Wählen Sie auf der MyBroker Seite im Abschnitt Verbindungen den Namen Ihrer Sicherheitsgruppe aus oder  öffnen Sie die EC2-Konsole (anstelle der VPC-Konsole). Die EC2-Konsole ermöglicht eine intuitivere Konfiguration von Regeln für ein- und ausgehenden Datenverkehr. Weitere Informationen finden Sie im aktualisierten Abschnitt Eingehende Verbindungen aktivieren.
9. Januar 2018	<ul style="list-style-type: none">• 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.

Datum	Aktualisierung der Dokumentation
28. November 2017	<p>Dies ist die erste veröffentlichte Version von Amazon MQ und des Amazon MQ Entwicklerhandbuchs.</p> <ul style="list-style-type: none">• Amazon MQ ist in den folgenden Regionen verfügbar:<ul style="list-style-type: none">• US East (Ohio)• USA Ost (Nord-Virginia)• USA West (Oregon)• Asien-Pazifik (Sydney)• Europe (Frankfurt)• Europa (Irland) <p>Die Verwendung des <code>mq.t2.micro</code> -Instance-Typs hängt von CPU-Guthaben und Basisleistung ab – mit der Fähigkeit, die Baseline-Ebene zu überschreiten (weitere Informationen finden Sie in der CpuCredit Balance -Metrik). Wenn Ihre Anwendung Fixed Performance, erwägen Sie, einen <code>mq.m5.large</code> -Instance-Typ zu verwenden.</p> <ul style="list-style-type: none">• Sie können <code>mq.m4.large</code> - und <code>mq.t2.micro</code> -Broker erstellen. <p>Die Verwendung des <code>mq.t2.micro</code> -Instance-Typs hängt von CPU-Guthaben und Basisleistung ab – mit der Fähigkeit, die Baseline-Ebene zu überschreiten (weitere Informationen finden Sie in der CpuCredit Balance -Metrik). Wenn Ihre Anwendung Fixed Performance, erwägen Sie, einen <code>mq.m5.large</code> -Instance-Typ zu verwenden.</p> <ul style="list-style-type: none">• Sie können die Broker-Engine ActiveMQ 5.15.0 verwenden.• Sie können Broker auch programmgesteuert mit der Amazon MQ REST API und AWS SDKs erstellen und verwalten.• Sie können auf Ihre Broker mithilfe von jeder Programmiersprache, die ActiveMQ unterstützt zugreifen, und indem Sie TLS explizit für die folgenden Protokolle aktivieren:<ul style="list-style-type: none">• AMQP• MQTT• MQTT über WebSocket• OpenWire

Datum	Aktualisierung der Dokumentation
	<ul style="list-style-type: none"> • STOMP • STOMP über 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.

Amazon MQ Dokumentverlauf

In der folgenden Tabelle sind die wichtigen Änderungen am Amazon MQ-Entwicklerhandbuch aufgeführt. Weitere Informationen über neu in Amazon MQ eingeführte und verbesserte Funktionen finden Sie unter .

Datum	Aktualisierung der Dokumentation
22. August 2022	<p>Es wurden separate Kapitel für Amazon MQ für ActiveMQ und Amazon MQ für RabbitMQ-Engines erstellt. Diese übergeordneten Kapitel enthalten jetzt Engine-Details, Tutorials und bewährte Methoden:</p> <ul style="list-style-type: none"> • Working with Amazon MQ for ActiveMQ • Working with Amazon MQ for RabbitMQ
13. Januar 2022	<p>Es wurde ein neuer Abschnitt zur Fehlerbehebung hinzugefügt, der die Statuscodes auflistet, die Amazon MQ zurückgibt, wenn sich ein Broker in einem ungesunden Zustand befindet, sowie detaillierte Informationen zur Diagnose und Wiederherstellung des Brokers.</p> <ul style="list-style-type: none"> • the section called “Fehlerbehebung: Erforderliche Codes für die Amazon-MQ-Aktion”
8. November 2021	<p>Es wurde ein neues Tutorial hinzugefügt, das die Einrichtung eines Python-Pika-Clients mit Amazon-MQ-for-RabbitMQ-Brokern beschreibt.</p>


Datum	Aktualisierung der Dokumentation
8. Oktober 2021	<p>Die folgenden Themen zur Fehlerbehebung für Amazon MQ für ActiveMQ und Amazon MQ für RabbitMQ-Broker-Engines wurden hinzugefügt:</p> <ul style="list-style-type: none">• the section called “Verwenden von Python Pika mit Amazon MQ for RabbitMQ”
22. September 2021	<p>Die folgenden Themen zur Behebung häufiger Verbindungs- und Autorisierungsprobleme mit Amazon MQ für ActiveMQ-Broker wurden hinzugefügt:</p> <ul style="list-style-type: none">• Einige Clients können keine Verbindung herstellen• the section called “Wie aktiviere ich Plugins in Amazon MQ für RabbitMQ?”• the section called “Ich kann die Amazon-VPC-Konfiguration für den Broker nicht ändern.”
12. August 2021	<p>Der folgende Abschnitt wurde hinzugefügt, um die Problembehandlung häufig auftretender Probleme bei der Arbeit mit Amazon MQ-Brokern zu beschreiben.</p> <ul style="list-style-type: none">• Fehlerbehebung
29. Juli 2021	<p>Die folgenden Abschnitte wurden hinzugefügt, um die Versionsverwaltung von Amazon MQ für RabbitMQ und das Upgrade von Amazon MQ Brokern auf neue kleinere und größere Engine-Versionen zu beschreiben, sobald sie unterstützt werden.</p> <ul style="list-style-type: none">• the section called “Versionsverwaltung.”
21. Juli 2021	<p>Die folgenden Abschnitte wurden hinzugefügt, um die Verbindung eines Amazon MQ-Brokers AWS Lambda mit als Ereignisquelle zu beschreiben.</p> <ul style="list-style-type: none">• Connect your Amazon MQ for ActiveMQ broker to Lambda• Connect your Amazon MQ for RabbitMQ broker to Lambda


Datum	Aktualisierung der Dokumentation
16. Juli 2021	<p>Die folgenden Abschnitte wurden hinzugefügt, um Amazon MQ Broker-Wartungsfenster zu beschreiben und ein Wartungsfenster mithilfe der oder AWS Management Console AWS CLI der Amazon MQ-API anzupassen.</p> <ul style="list-style-type: none">• the section called “Wartung eines Brokers”
7. Juni 2021	<p>Die folgenden Abschnitte wurden hinzugefügt, um Amazon MQ für die ActiveMQ-Versionsverwaltung und das Upgrade von Amazon MQ-Brokern auf neue kleinere und größere Engine-Versionen zu beschreiben, so wie sie unterstützt werden.</p> <ul style="list-style-type: none">• the section called “Versionsverwaltung.”• the section called “Upgrade der Engine-Version”
18. Mai 2021	<p>Der folgende Abschnitt wurde hinzugefügt, um die Standardwerte für Amazon MQ für RabbitMQ Broker zu beschreiben</p> <ul style="list-style-type: none">• the section called “Standardeinstellungen für Broker”
5. Mai 2021	<p>Der folgende Abschnitt zur Beschreibung AWS verwalteter Richtlinien für Amazon MQ und Aktualisierungen dieser Richtlinien wurde hinzugefügt:</p> <ul style="list-style-type: none">• the section called “Von AWS verwaltete Richtlinien”
16. Februar 2021	<p>Der folgende Tutorial-Abschnitt für Amazon MQ für RabbitMQ wurde hinzugefügt:</p> <ul style="list-style-type: none">• the section called “Beheben der angehaltenen Warteschlangen-Synchronisierung”

Datum	Aktualisierung der Dokumentation
4. November 2020	<ul style="list-style-type: none">• Die folgenden Abschnitte wurden hinzugefügt, um Amazon MQ für RabbitMQ zu dokumentieren:<ul style="list-style-type: none">• the section called “Erstellen und Verbinden mit einem RabbitMQ-Broker”• the section called “RabbitMQ-Tutorials”• the section called “Best Practices für Amazon MQ for RabbitMQ”• the section called “RabbitMQ-Engine”• the section called “Konfigurieren von Amazon MQ für RabbitMQ-Protokolle”• the section called “Verwenden von servicegebundenen Rollen”• Zusätzliche Überarbeitungen an bestehenden Kapiteln und Abschnitten des Leitfadens wurden vorgenommen, um Amazon MQ für RabbitMQ-Unterstützung genau zu dokumentieren.
16. Dezember 2019	<ul style="list-style-type: none">• Die folgenden Abschnitte wurden hinzugefügt:<ul style="list-style-type: none">• Storage• Auswählen des richtigen Broker-Speichertyps für den besten Durchsatz• Die Informationen in den folgenden Abschnitten wurden überarbeitet:<ul style="list-style-type: none">• Broker• Broker instance types• Amazon MQ Single-Instance-Broker• Aktive/Standby Amazon MQ-Broker für hohe Verfügbarkeit• Create an ActiveMQ broker• Creating and configuring a broker
19. Juli 2019	<p>Inhalte zur Verschlüsselungsverwaltung wurden in den folgenden Abschnitten geändert und hinzugefügt:</p> <ul style="list-style-type: none">• Datenschutz in Amazon MQ• Verschlüsselung im Ruhezustand• Verschlüsselung während der Übertragung• EncryptionOptions

Datum	Aktualisierung der Dokumentation
22. April 2019	<p>Die folgenden Abschnitte für Tag-basierte Richtlinien und Berechtigungen auf Ressourcenebene wurden hinzugefügt:</p> <ul style="list-style-type: none">• Funktionsweise von Amazon MQ mit IAM• Unterstützte Berechtigungen auf Ressourcenebene für Amazon MQ-API-Aktionen
4. März 2019	<p>Verbesserte Dokumentation für die Konfiguration des dynamischen Failover und die Anpassung von Clients für ein Netzwerk von Brokern. Aktivieren Sie das dynamische Failover durch die Konfiguration von <code>transportConnectors</code> zusammen mit den <code>networkConnectors</code> -Konfigurationsoptionen.</p> <ul style="list-style-type: none">• Dynamisches Failover mit Transport Connectors• Amazon MQ Netzwerk von Brokern• Amazon MQ Broker Configuration Parameters
5. Januar 2019	<p>Verbesserte Dokumentation zu einigen pro-Minute-Metriken. Weitere Informationen finden Sie im Folgenden: ActiveMQ-Ziel-Metriken (Warteschlange und Thema)</p>
19. Dezember 2018	<ul style="list-style-type: none">• Die folgenden Abschnitte wurden hinzugefügt:<ul style="list-style-type: none">• Amazon MQ Netzwerk von Brokern• Creating and Configuring a Network of Brokers• Korrekte Konfiguration Ihres Netzwerk von Brokern• networkConnector• networkConnectionStartAsync• Das untergeordnete Sammlungselement <code>networkConnectors</code> wurde zum Abschnitt In Amazon MQ-Konfigurationen zulässige Elemente, untergeordnete Sammlungselemente und deren untergeordnete Attribute hinzugefügt.
11. Dezember 2018	<p>Die Dokumentation wurde aktualisiert, um die Verfügbarkeit von ActiveMQ Version 5.15.8 zu berücksichtigen.</p>


Datum	Aktualisierung der Dokumentation
5. Dezember 2018	Der Abschnitt Markieren von Ressourcen wurde hinzugefügt.
26. Oktober 2018	Der Abschnitt Vermeiden von langsamen Neustarts durch Wiederherstellung vorbereiteter XA-Transaktionen wurde hinzugefügt.
15. Oktober 2018	Der Abschnitt Quotas in Amazon MQ wurde aktualisiert.
1. Oktober 2018	Die Informationen im Abschnitt Nächste Schritte wurden korrigiert.
27. September 2018	<ul style="list-style-type: none"> • Der Abschnitt Tutorial: Bearbeiten der Broker-Engine-Version, des Instance-Typs, von CloudWatch Logs und von Wartungseinstellungen wurde hinzugefügt. • Die folgenden Abschnitte wurden aktualisiert: <ul style="list-style-type: none"> • Create an ActiveMQ broker • Configure Basic Broker Settings
18. September 2018	Der folgende Hinweis wurde dem Erstellen und Verwalten von ActiveMQ-Broker-Benutzern hinzugefügt: 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.
31. August 2018	<ul style="list-style-type: none"> • Die Terminologie für aktive/Standby-Broker wurde geklärt. Weitere Informationen finden Sie unter Aktive/Standby Amazon MQ-Broker für hohe Verfügbarkeit. • Die Terminologie für das Wartungsfenster wurde vereinfacht. Weitere Informationen finden Sie unter Amazon-MQ-Broker-Konfigurationslebenszyklus. • Der Abschnitt Configure Additional Broker Settings wurde neu geschrieben. • Die Abschnitte Amazon MQ für ActiveMQ Metriken und Listing brokers and viewing broker details wurden aktualisiert.
15. August 2018	Die Informationen im Abschnitt Create an ActiveMQ broker wurden korrigiert.


Datum	Aktualisierung der Dokumentation
13. August 2018	Der Abschnitt Zugreifen auf die Webkonsole eines Brokers ohne öffentliche Zugänglichkeit wurde hinzugefügt.
2. August 2018	<ul style="list-style-type: none"> • Der Abschnitt Fehlerbehebung bei der CloudWatch Logs Konfiguration wurde hinzugefügt. • In diesem Handbuch wurde durchgängig die folgende Warnung hinzugefügt: <div data-bbox="431 583 1507 947" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Important</p> <p>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.</p> </div>
1. August 2018	<p>Die Informationen in den folgenden Abschnitten wurden korrigiert:</p> <ul style="list-style-type: none"> • Grundlegendes zum Aufbau der Protokollierung in CloudWatch Logs • Connect a Java application to your broker
31. Juli 2018	<ul style="list-style-type: none"> • Das dreiminütige Demovideo wurde in den Abschnitt Getting Started with Amazon MQ verschoben. • Das dreiminütige Video mit den ersten Schritten wurde zum Abschnitt What is Amazon MQ? hinzugefügt.
30. Juli 2018	<ul style="list-style-type: none"> • Der Abschnitt Configuring Amazon MQ to publish logs to Amazon CloudWatch Logs wurde hinzugefügt. • Der Abschnitt Configure Additional Broker Settings wurde aktualisiert.
19. Juli 2018	<ul style="list-style-type: none"> • Der Abschnitt Logging Amazon MQ API calls using CloudTrail wurde hinzugefügt.

Datum	Aktualisierung der Dokumentation
5. Juli 2018	<ul style="list-style-type: none">• Zum Abschnitt <code>authorizationEntry</code> wurde ein untergeordnetes Immer eine Autorisierungszuordnung konfigurieren-Element hinzugefügt.• Die Informationen im Abschnitt Integration von ActiveMQ-Brokern in LDAP wurden präzisiert.• Die Informationen im Abschnitt API-Drosselung wurden präzisiert.
29. Juni 2018	<ul style="list-style-type: none">• Die Informationen im Abschnitt Broker instance types wurden aktualisiert.• Der Abschnitt Auswählen des richtigen Broker-Instance-Typs für den besten Durchsatz wurde hinzugefügt.
4. Juni 2018	<p>Zusätzlich zu GitHub, HTML, PDF und Kindle sind die Versionshinweise zum Amazon MQ-Entwicklerhandbuch als RSS-Feed verfügbar.</p> 
29. Mai 2018	<p>Folgende Änderungen wurden am Abschnitt Working Java Example vorgenommen:</p> <ul style="list-style-type: none">• Ein STOMP+WSS-Java-Beispiel wurde hinzugefügt. Der Java-Beispielcode für STOMP+WSS stellt eine Verbindung zu einem Broker her, erstellt eine Warteschlange und veröffentlicht und empfängt eine Nachricht.• Das MQTT-Java-Beispiel wurde verbessert.• Das OpenWire Java-Beispiel wurde verbessert.
24. Mai 2018	<p>Der Anschluss für den Endpunkt des Wire-Level-Protokolls wurde im MQTT-Java-Beispiel im Abschnitt Working Java Example korrigiert.</p>
22. Mai 2018	<p>Korrektur der Informationen in allen Abschnitten über Java-Abhängigkeiten.</p>
17. Mai 2018	<p>Die Informationen im Abschnitt Benutzer wurden korrigiert.</p>
15. Mai 2018	<p>Die Informationen im Abschnitt Sicherstellen einer effektiven Amazon MQ-Leistung wurden korrigiert.</p>

Datum	Aktualisierung der Dokumentation
8. Mai 2018	<ul style="list-style-type: none">• Amazon MQ REST API-Berechtigungen–Referenz wurde in einen eigenen Abschnitt verschoben.• Der Erforderliche IAM-Berechtigungen zum Erstellen eines Amazon MQ-Brokers Abschnitt wurde mit ein Beispiel für benutzerdefinierte IAM-Richtlinien erstellt.
7. Mai 2018	<ul style="list-style-type: none">• Im gesamten Handbuch wurde präzisiert, dass das Broker-Wartungsfenster 2 Stunden beträgt. Weitere Informationen finden Sie unter Amazon-MQ-Broker-Konfigurationslebenszyklus.• Es wurden Erklärungen hinzugefügt, warum für das Erstellen eines Brokers die Berechtigungen <code>ec2:CreateNetworkInterface</code> und <code>ec2:CreateNetworkInterfacePermission</code> erforderlich sind. Weitere Informationen finden Sie unter API-Authentifizierung und Amazon MQ-Autorisierung für .
1. Mai 2018	<p>Klärung der Informationen über das Wartungsfenster für aktive/Standby-Broker in den folgenden Abschnitten:</p> <ul style="list-style-type: none">• Aktive/Standby Amazon MQ-Broker für hohe Verfügbarkeit• Creating and configuring a broker• Creating and applying broker configurations
27. April 2018	<p>Die folgenden Abschnitte wurden neu geschrieben und der Java-Beispielcode wurde optimiert, um der Empfehlung zu entsprechen, das Verbindungspooling nur für Produzenten zu verwenden, nicht für Verbraucher:</p> <ul style="list-style-type: none">• Verwenden Sie immer Verbindungspools• Erstellen eines Nachrichtenproduzenten und Senden einer Nachricht• Erstellen eines Nachrichtenkonsumenten und Empfangen der Nachricht• AmazonMQExample.java
26. April 2018	<p>Im Abschnitt Working Java Example wurde ein MQTT Java-Beispiel hinzugefügt. Der Java-Beispielcode für MQTT stellt eine Verbindung zu einem Broker her, erstellt ein Topic und veröffentlicht und empfängt eine Nachricht.</p>

Datum	Aktualisierung der Dokumentation
4. April 2018	Der Abschnitt zur Kommunikation mit Amazon MQ wurde in Herstellen einer Verbindung mit Amazon MQ umbenannt.
3. April 2018	Die Informationen im Abschnitt Gleichzeitige Speicherung und Bereitstellung für Warteschlangen mit langsamen Konsumenten deaktivieren wurden korrigiert und präzisiert.
2. April 2018	Der Abschnitt „Gleichzeitige Speicherung und Bereitstellung von Warteschlangen in Amazon MQ“ wurde in den Abschnitt Gleichzeitige Speicherung und Bereitstellung für Warteschlangen mit langsamen Konsumenten deaktivieren verschoben.
27. März 2018	<ul style="list-style-type: none">• Das re:Invent-Einführungsvideo wurde durch ein dreiminütiges Demovideo im Abschnitt What is Amazon MQ? ersetzt.• Die folgenden Abschnitte wurden neu strukturiert:<ul style="list-style-type: none">• Broker Architecture• Funktionsweise von Amazon MQ• Amazon-MQ-Broker-Konfigurationslebenszyklus wurde in den Abschnitt Broker Architecture verschoben.
22. März 2018	In diesem Handbuch wurde die folgende Erklärung erläutert: Amazon MQ verschlüsselt Nachrichten im Ruhezustand und bei der Übertragung mithilfe von Verschlüsselungsschlüsseln, die sicher verwaltet und gespeichert werden. Weitere Informationen finden Sie im AWS Encryption SDK -Entwicklerhandbuch .
19. März 2018	Die folgende Anweisung in diesem Handbuch wurde verdeutlicht: Aktiv/Standby-Broker besteht aus zwei Brokern in zwei verschiedenen Availability Zones, die in einer Redundanz konfiguriert wurden. Diese Broker kommunizieren synchron mit Ihrer Anwendung und mit Amazon EFS.
15. März 2018	<ul style="list-style-type: none">• Der Abschnitt Amazon MQ Basic elements wurde umstrukturiert.

Datum	Aktualisierung der Dokumentation
12. März 2018	<ul style="list-style-type: none">• Die Informationen in den Abschnitten Best Practices für die Sicherheit in Amazon MQ und Herstellen einer Verbindung mit Amazon MQ wurden korrigiert und präzisiert.• Der Abschnitt Gleichzeitige Speicherung und Bereitstellung für Warteschlangen mit langsamen Konsumenten deaktivieren wurde hinzugefügt.• Warnungen wurden in ein Vorwort für den Abschnitt Konfigurieren erweiterter Broker-Einstellungen gruppiert.
9. März 2018	<ul style="list-style-type: none">• Die Informationen im Abschnitt Immer eine Autorisierungszuordnung konfigurieren wurden korrigiert und präzisiert.• Der Abschnitt authorizationEntry wurde hinzugefügt und der Abschnitt kahaDB wurde aktualisiert.
8. März 2018	<ul style="list-style-type: none">• Der Abschnitt Immer eine Autorisierungszuordnung konfigurieren wurde hinzugefügt.• Hinweise zu Broker-Suffixe wurden zum Abschnitt Monitoring Amazon MQ using CloudWatch hinzugefügt.
6. März 2018	<p>In diesem Handbuch wurde durchgängig der folgende Hinweis hinzugefügt:</p> <div data-bbox="402 1157 1507 1566" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Note</p><p>Die Verwendung des <code>mq.t2.micro</code> -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 erfordert, erwägen Sie, einen <code>mq.m5.large</code> -Instance-Typ zu verwenden.</p></div>

Datum	Aktualisierung der Dokumentation
1. März 2018	<ul style="list-style-type: none">• Die Metrik <code>CpuCreditBalance</code> wurde dem Abschnitt Amazon MQ für ActiveMQ Metriken hinzugefügt.• Der Abschnitt Amazon MQ-Attribute wurde hinzugefügt.• Links von Elementen im Abschnitt the section called “Zugelassene Elemente” zu deren Attributen und zu untergeordneten Sammlungselementen hinzugefügt.• Korrekturen am AWS Glossar in vorgenommenen GitHub.
28. Februar 2018	Die Bildanzeige in wurde korrigiert GitHub.
27. Februar 2018	<p>Zusätzlich zu HTML, PDF und Kindle ist das Amazon MQ-Entwicklerhandbuch auf verfügbar GitHub. Um Feedback zu hinterlassen, wählen Sie das GitHub Symbol in der oberen rechten Ecke.</p> 
26. Februar 2018	<ul style="list-style-type: none">• Regionen wurden in allen Beispielen und Diagrammen konsistent gestaltet.• Optimierte Links zu den AWS Konsolen- und Produktwebseiten.
22. Februar 2018	<p>Die in den folgenden Abschnitten enthaltenen Informationen wurden präzisiert und korrigiert:</p> <ul style="list-style-type: none">• Broker ohne öffentlichen Zugriff bevorzugen• Immer Failover-Transport verwenden, um Verbindungen zu mehreren Broker-Endpunkten einzurichten• API-Authentifizierung und Amazon MQ-Autorisierung für• Integration von ActiveMQ-Brokern in LDAP
21. Februar 2018	<p>Der Java-Code in den folgenden Abschnitten wurde korrigiert:</p> <ul style="list-style-type: none">• Working Java Example• Connect a Java application to your broker• Verwenden Sie immer Verbindungspools

Datum	Aktualisierung der Dokumentation
20. Februar 2018	Die Informationen in den Abschnitten „ Sicherheit in Amazon MQ “ und „Bewährte Methoden“ wurden korrigiert und präzisiert.
19. Februar 2018	<ul style="list-style-type: none">• Der Java-Code im Abschnitt Verwenden Sie immer Verbindungspools wurde korrigiert.• Restrukturierung und Erweiterung der Abschnitte „Bewährte Methoden“ und der Abschnitte „Sicherheit in Amazon MQ“.
16. Februar 2018	<ul style="list-style-type: none">• Der Abschnitt Best Practices für die Sicherheit in Amazon MQ wurde hinzugefügt.• Der Abschnitt Herstellen einer Verbindung mit Amazon MQ wurde aktualisiert.• Der Java-Code in den folgenden Abschnitten wurde korrigiert:<ul style="list-style-type: none">• Getting Started with Amazon MQ• AmazonMQExample.java
15. Februar 2018	<ul style="list-style-type: none">• Der Abschnitt „Bewährte-Methoden-Abschnitte“ wurde neu strukturiert und erweitert.• Die folgenden Abschnitte wurden aktualisiert:<ul style="list-style-type: none">• Wie sehen meine ersten Schritte mit Amazon MQ aus?• Nächste Schritte (Erste Schritte)• Related resources
14. Februar 2018	Die folgenden Abschnitte wurden aktualisiert: <ul style="list-style-type: none">• Quotas in Amazon MQ• API-Drosselung• Sicherheit in Amazon MQ
13. Februar 2018	<ul style="list-style-type: none">• Der Abschnitt Related resources wurde aktualisiert.• Der Abschnitt Quotas in Amazon MQ wurde aktualisiert.• Der Abschnitt Bitte geben Sie uns Feedback wurde hinzugefügt.

Datum	Aktualisierung der Dokumentation
25. Januar 2018	<ul style="list-style-type: none"> • Es wurde ein Fehler im Unterabschnitt Java-Abhängigkeiten hinzufügen des Abschnitts Working Java Example behoben. • Die Berechtigung für die REST-Operations-ID RebootBroker wird auf der IAM-Konsole korrekt als <code>mq:RebootBroker</code> aufgelistet.
24. Januar 2018	<ul style="list-style-type: none"> • Der Abschnitt Verändern oder löschen Sie auf keinen Fall die Amazon MQ Elastic Network-Schnittstelle wurde hinzugefügt. • Alle Diagramme in diesem Handbuch wurden aktualisiert. • Es wurden Links auf die Amazon MQ REST API Reference im gesamten Handbuch hinzugefügt sowie Links auf spezifische REST-APIs im Abschnitt API-Authentifizierung und Amazon MQ-Autorisierung für .
19. Januar 2018	Die Informationen im Abschnitt Amazon MQ für ActiveMQ-Ressourcen wurden aktualisiert.
18. Januar 2018	Die Informationen im Abschnitt Quotas in Amazon MQ wurden korrigiert und präzisiert.
17. Januar 2018	Die Empfehlung, virtuelle Ziele dauerhaften Abonnements vorzuziehen , wurde mit einer verbesserten Erklärung wieder eingesetzt.
11. Januar 2018	<ul style="list-style-type: none"> • Das Amazon MQ-Entwicklerhandbuch steht im Kindle-Format zur Verfügung, ebenso wie in HTML und PDF. • Die Informationen in den Abschnitten API-Authentifizierung und Amazon MQ-Autorisierung für und Schritt 2: Erstellen eines Benutzers und Abrufen Ihrer AWS-Anmeldeinformationen wurden korrigiert und präzisiert.
3. Januar 2018	<code>DescribeConfigurationRevision</code> wurde dem Abschnitt API-Authentifizierung und Amazon MQ-Autorisierung für hinzugefügt.
15. Dezember 2017	Die Empfehlung gegen dauerhafte Abonnements wurde aus dem Abschnitt „Bewährte-Methoden-Abschnitte“ entfernt.

Datum	Aktualisierung der Dokumentation
8. Dezember 2017	<ul style="list-style-type: none">• Die Voraussetzung Eingehende Verbindungen aktivieren wurde den Abschnitten Connecting a Java application to your broker und Working Java Example hinzugefügt.• In diesem Handbuch wurde durchgängig der folgende Hinweis hinzugefügt: Derzeit können Sie eine Konfiguration nicht löschen.
7. Dezember 2017	<ul style="list-style-type: none">• Der Code in dem AmazonMQExample.java wurde verbessert.• Der Abschnitt API-Authentifizierung und Amazon MQ-Autorisierung für wurde hinzugefügt.
5. Dezember 2017	<ul style="list-style-type: none">• Informationen im Abschnitt Monitoring Amazon MQ using CloudWatch wurden geklärt und korrigiert:<ul style="list-style-type: none">• Verbesserte Beschreibungen der Metrik.• Unterabschnitte Amazon MQ für ActiveMQ Metriken und Dimensionen für ActiveMQ-Broker-Metriken hinzugefügt.• Video „Einführung von Amazon MQ“ im Abschnitt What is Amazon MQ? hinzugefügt.
4. Dezember 2017	<ul style="list-style-type: none">• Die folgenden Informationen im Abschnitt Datenspeicherung hinzugefügt: Speicherkapazität pro Broker beträgt 200 GB.• Voraussetzungen dem Abschnitt Working Java Example hinzugefügt. (Die <code>activemq-client.jar</code> und <code>activemq-pool.jar</code> -Pakete sind für das Beispiel erforderlich. Weitere Informationen finden Sie unter Connecting a Java application to your broker).
1. Dezember 2017	<ul style="list-style-type: none">• Die Screenshots in allen Tutorials wurden aktualisiert und verbessert.• Die folgende Erklärung in diesem Handbuch wurde verdeutlicht: Änderungen an einer Konfigurationsrevision oder einem ActiveMQ Benutzer werden nicht sofort angewendet. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten. Weitere Informationen finden Sie unter Amazon-MQ-Broker-Konfigurationslebenszyklus.

AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.

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.