



Guía para desarrolladores

Amazon MQ



Amazon MQ: Guía para desarrolladores

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Amazon MQ?	1
Características de Amazon MQ	1
¿Cómo puedo comenzar a utilizar Amazon MQ?	2
¿Cómo puedo proporcionar comentarios a Amazon MQ?	3
Configuración	4
Paso 1: requisitos previos	4
Inscríbese en un Cuenta de AWS	4
Creación de un usuario con acceso administrativo	5
Crea un usuario y obtén tus AWS credenciales	6
Paso 3: Prepararse para usar los códigos de ejemplo	8
Siguiendo pasos	9
Introducción: Creación y conexión con un agente de ActiveMQ	10
Creación de un agente de ActiveMQ	10
Introducción: Cómo crear un agente de RabbitMQ y conectarse a él	13
Crear un agente de RabbitMQ	13
Administración de un agente	16
Conexión con Amazon MQ	16
Puntos de conexión de servicio	16
Puntos de conexión de los agentes	17
Conexión a Amazon MQ mediante puntos de conexión de doble pila (IPv4 e IPv6)	17
Conexión a Amazon MQ mediante PrivateLink de AWS	17
Autenticación y autorización	18
Autenticación y autorización de Amazon MQ para RabbitMQ	19
Autenticación y autorización de Amazon MQ para ActiveMQ	20
Actualización de la versión del motor	21
Actualización manual de la versión del motor	22
Actualización del tipo de instancia	24
Almacenamiento	28
Diferencias entre tipos de almacenamiento	28
Configuración de un agente privado	30
Configurar un agente privado en el Consola de administración de AWS	31
Acceso a la consola web de un agente de Amazon MQ sin accesibilidad pública	31
Programación del mantenimiento del agente	32
Reiniciar un agente	36

Para reiniciar un agente de Amazon MQ	36
Eliminación de un agente	37
Eliminación de un agente de Amazon MQ	37
Estados del agente	37
Etiquetado	38
Adición de etiquetas en la consola de Amazon MQ	39
Amazon MQ para ActiveMQ	40
Amazon MQ para agentes de ActiveMQ	40
Agente	40
Usuario	43
Implementación de un agente	44
Agente de una sola instancia	44
agente activo/en espera	45
Red de agentes	46
¿Cómo funciona una red de agentes?	47
¿Cómo administra las credenciales una red de agentes?	47
Entre regiones	48
Conmutación por error dinámica con conectores de transporte	49
Tipos de instancias	50
Configuraciones de agentes	51
Atributos	52
Uso de archivos de configuración XML Spring	52
Creación de una configuración	53
Edición de una revisión de la configuración	56
Elementos permitidos	58
Atributos permitidos	61
Colecciones permitidas	74
Atributos de elemento secundarios	80
Replicación de datos entre regiones	87
Agentes principales y de réplicas	88
Creación de un agente de CRDR	89
Eliminación de un agente de CRDR	93
Promoción de un agente de CRDR	93
Métricas	96
Tutoriales de ActiveMQ	98
Creación y configuración de una red de agentes	98

Conexión de una aplicación Java con el agente	104
Integración de agentes de ActiveMQ con LDAP	110
Paso 3: (Opcional) Conectarse a una AWS Lambda función	126
Creación de un usuario del agente de ActiveMQ	128
Edición de un usuario de agente de ActiveMQ	130
Eliminación de un usuario de agente de ActiveMQ	131
Ejemplos prácticos de Java	131
Administración de versiones	143
Versiones admitidas del motor de Amazon MQ para ActiveMQ	144
Actualizaciones de la versión del motor	145
Lista de versiones del motor compatibles	145
Prácticas recomendadas de Amazon MQ para ActiveMQ	145
No modifique ni elimine nunca la interfaz de red elástica de Amazon MQ	145
Usar siempre el grupo de conexiones	146
Usar siempre el transporte de conmutación por error para conectarse a puntos de enlace de varios agentes	147
Evitar usar los selectores de mensajes	148
Preferir los destinos virtuales a las suscripciones duraderas	148
Si utiliza el emparejamiento de Amazon VPC, evite el cliente IPs en el rango CIDR 10.0.0.0/16	148
Desactivar el almacenamiento y el envío simultáneos en colas con consumidores lentos	148
Elegir el tipo de instancia de agente correcto para obtener el mejor desempeño	149
Elegir el tipo de almacenamiento de agente correcto para obtener el mejor rendimiento	150
Configurar la red de agentes correctamente	151
Para evitar los reinicios lentos, recupere las transacciones XA preparadas	151
Amazon MQ para RabbitMQ	154
Broker	154
Puertos de agente de escucha	154
Atributos	42
Administración de versiones	156
Lista de versiones del motor compatibles	157
RabbitMQ 4	157
Compatibilidad de versiones	160
Actualizaciones de versiones	161
Implementación de un agente RabbitMQ	161
Agente de una sola instancia	162

Implementación de clúster	162
Tipos de instancias	164
Tipos de instancias para la implementación de clústeres m7g	165
Tipos de instancias para la implementación de una sola instancia de m7g	166
Tipos de instancias para la implementación de una sola instancia mq .m5	167
Tipos de instancias para la implementación de clústeres mq .m5	168
Directrices de dimensionamiento	169
Límites de recursos predeterminados	170
Límite máximo de recursos	174
Valores predeterminados del agente	178
Configuraciones de agentes	184
Atributos	52
Creación de una configuración	185
Edición de una revisión de la configuración	188
Valores configurables	189
Autenticación y autorización	206
Autenticación y autorización simple	19
OAuth Autenticación y autorización 2.0	19
Autenticación y autorización de IAM	19
Autenticación y autorización de LDAP	19
Autenticación y autorización HTTP	20
Autenticación con certificado SSL	20
Autenticación y autorización simple	208
OAuth Autenticación y autorización 2.0	210
Autenticación y autorización de IAM	211
Autenticación y autorización HTTP	213
Autenticación con certificado SSL	215
Autenticación y autorización LDAP	219
Plugins	221
Complemento de administración RabbitMQ	222
el complemento Shovel (pala)	222
Complemento Federation (federación)	223
Complemento de intercambio Hash consistente	224
OAuth Complemento 2.0	225
Complemento LDAP	225
Complemento HTTP	225

Complemento de certificado SSL	225
complemento aws	226
Complemento JMS Topic Exchange	226
Protocolos	226
Soporte de JMS	227
Cliente RabbitMQ JMS	227
Compatible con JMS 1.1, 2.0 y 3.1 APIs	227
Autenticación y autorización	228
Interoperabilidad con las colas AMQP de RabbitMQ	228
Políticas	228
Colas de cuórum	233
Migración a colas de cuórum	234
Configuración de políticas	235
Prácticas recomendadas	236
Prácticas recomendadas de Amazon MQ para RabbitMQ	237
Configuración del agente	237
Fiabilidad de los mensajes	239
Optimización del rendimiento	243
Resiliencia de la red	248
Tutoriales de RabbitMQ	250
Edición de las preferencias del agente	250
Uso de Python Pika con Amazon MQ para RabbitMQ	251
Resolver la sincronización de colas en pausa	258
Reducir el número de conexiones y canales	265
Paso 2: Conectar al agente una aplicación basada en una JVM	266
Paso 3: (Opcional) Conectarse a una AWS Lambda función	270
Uso de la autenticación y autorización de OAuth 2.0	273
Uso de la autenticación y la autorización de IAM	281
Uso de la autenticación y la autorización de LDAP	286
Uso de la autenticación y la autorización HTTP	293
Uso de la autenticación con certificado SSL	297
Uso de mTLS para el AMQP y los puntos finales de administración	304
Conexión de la aplicación JMS	309
Seguridad	313
Protección de datos	314
Cifrado	315

Cifrado en reposo	315
Cifrado en tránsito	325
Identity and Access Management	326
Público	327
Autenticación con identidades	327
Administración del acceso con políticas	329
Cómo funciona Amazon MQ con IAM	331
Ejemplos de políticas basadas en identidades	336
Autenticación y autorización de las API	339
Autenticación y autorización del agente	345
AWS políticas gestionadas	347
Cómo utilizar roles vinculados a servicios	348
Resolución de problemas	355
Validación de conformidad	357
Resiliencia	357
Seguridad de la infraestructura	357
Prácticas recomendadas de seguridad	358
Preferir agentes sin accesibilidad pública	358
Configurar siempre una asignación de autorizaciones	358
Bloqueo de protocolos innecesarios	359
Registro y supervisión	360
Acceder a CloudWatch las métricas	360
Acceder a las CloudWatch métricas mediante el Consola de administración de AWS	361
Métricas para ActiveMQ	361
Métricas de Amazon MQ para ActiveMQ	361
Métricas de destino (cola y tema) de ActiveMQ	367
Métricas para RabbitMQ	371
Métricas de agentes de RabbitMQ	371
Dimensiones para las métricas de agentes de RabbitMQ	375
Métricas de nodos de RabbitMQ	375
Dimensiones para métricas de nodos de RabbitMQ	376
Métricas de colas de RabbitMQ	377
Dimensiones para métricas de colas de RabbitMQ	378
Métricas de redes de RabbitMQ	378
Dimensiones los agentes de RabbitMQ	379
Configuración de registros de Amazon MQ para RabbitMQ	380

Registrar llamadas a la API mediante CloudTrail	380
Información de Amazon MQ en CloudTrail	381
Ejemplo de entrada del archivo de registro de Amazon MQ	383
Configuración de registros de Amazon MQ para ActiveMQ	385
Comprensión de la estructura del inicio de sesión en los registros CloudWatch	385
Agregar el permiso CreateLogGroup a su usuario de Amazon MQ	386
Configurar una política basada en recursos para Amazon MQ.	387
Prevención de la sustitución confusa entre servicios	389
Resolución de problemas	391
Los grupos de registros no aparecen en CloudWatch	391
Los flujos de registro no aparecen en los grupos de CloudWatch registros	391
Cuotas	392
Agentes	392
Configuraciones	393
Usuarios	394
Almacenamiento de datos	395
Limitación de API	396
Resolución de problemas	398
Solución de problemas de ActiveMQ en Amazon MQ	398
Solución de problemas de RabbitMQ en Amazon MQ	398
Solución de problemas generales de Amazon MQ	401
No puedo conectarme con la consola web o los puntos de enlace de mi agente.	401
Excepciones SSL	407
Creé un agente pero me dio error.	408
Mi agente se reinició y no estoy seguro de porqué.	408
Solución de problemas de ActiveMQ en Amazon MQ	409
Recuperando registros CloudWatch	409
Conexión al agente tras un reinicio	410
Algunos clientes no se pueden conectar	410
Excepción JSP en la consola web	411
Solución de problemas de RabbitMQ en Amazon MQ	412
No puedo ver las métricas de mis colas o hosts virtuales en CloudWatch.	412
¿Cómo habilito los complementos de RabbitMQ en Amazon MQ?	412
No puedo cambiar la configuración de Amazon VPC para el agente.	413
Las implementaciones de clúster han pausado mis sincronizaciones de colas.	413

El agente de una sola instancia de Amazon MQ para RabbitMQ se encuentra en un bucle de reinicio.	413
He perdido el acceso a todas las cuentas de administrador de mi agente.	414
BROKER_ENI_DELETED	414
BROKER_OOM	414
RABBITMQ_MEMORY_ALARM	416
Paso 1: Diagnostique una alarma de memoria elevada	417
Paso 2: Abordar y prevenir una alarma de memoria elevada	419
RABBITMQ_INVALID_KMS_KEY	421
Diagnóstico y solución de INVALID_KMS_KEY	421
RABBITMQ_DISK_ALARM	422
Diagnóstico y direccionamiento de la alarma de límite de disco	423
RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE	424
Diagnosticar y abordar la alarma de cambio de tipo de instancia	424
RABBITMQ_INVALID_ASSUME_ROLE	425
Diagnóstico y tratamiento de RABBITMQ_INVALID_ASSUME_ROLE	425
RABBITMQ_INVALID_ARN_LDAP	426
Diagnóstico y tratamiento de RABBITMQ_INVALID_ARN_LDAP	426
RABBITMQ_INVALID_ARN_HTTP	427
Diagnóstico y tratamiento de RABBITMQ_INVALID_ARN_HTTP	428
RABBITMQ_INVALID_ARN_SSL	428
Diagnóstico y tratamiento de RABBITMQ_INVALID_ARN_SSL	429
RABBITMQ_INVALID_ARN	430
Diagnóstico y tratamiento de RABBITMQ_INVALID_ARN	430
Recursos relacionados	432
Recursos de Amazon MQ	432
Recursos de Amazon MQ para ActiveMQ	433
Recursos de Amazon MQ para RabbitMQ	433
Notas de la versión	435
.....	cdlxxvii

¿Qué es Amazon MQ?

Amazon MQ es un servicio de agente de mensajes administrado en [Apache ActiveMQ](#) Classic y [RabbitMQ](#) que facilita la configuración y el funcionamiento de los agentes de mensajes. Puede crear un nuevo agente de Amazon MQ con los protocolos de mensajería estándar del sector o migrar los agentes de mensajes existentes a Amazon MQ sin tener que volver a escribir el código de mensajería.

Un agente es un entorno de agente de mensajes que se ejecuta en Amazon MQ. Es el componente básico de Amazon MQ. Un agente de mensajes permite a las aplicaciones y los componentes de software comunicarse mediante diferentes lenguajes de programación, sistemas operativos y protocolos de mensajería formales. Puede utilizar los agentes Amazon MQ para la comunicación entre aplicaciones y componentes nativos en la nube a gran escala.

Temas

- [Características de Amazon MQ](#)
- [¿Cómo puedo comenzar a utilizar Amazon MQ?](#)
- [¿Cómo puedo proporcionar comentarios a Amazon MQ?](#)

Características de Amazon MQ

Mantenimiento administrado y actualizaciones de versiones

Amazon MQ realiza [actualizaciones de versiones](#) y [mantenimiento](#) para un agente de mensajes durante el [período de mantenimiento](#) programado.

Supervisión de agentes con CloudWatch

Amazon MQ y [Amazon CloudWatch](#) están integrados, por lo que puede ver y analizar métricas correspondientes a sus agentes y colas. Puede ver y analizar métricas desde la consola de Amazon MQ, la consola de CloudWatch, la línea de comandos y la API. Las métricas se recopilan automáticamente y se envían a CloudWatch cada minuto.

Seguridad

Amazon MQ proporciona el [cifrado](#) de los mensajes en reposo y en tránsito. Las conexiones con el intermediario utilizan SSL y el acceso se puede restringir a un punto de conexión privado dentro de su Amazon VPC. Además, puede utilizar [AWS Identity and Access Management](#) (IAM) para

controlar las acciones que los grupos y usuarios de IAM pueden realizar en los agentes específicos de Amazon MQ.

Colas de cuórum para RabbitMQ en Amazon MQ

Las [colas de cuórum](#) son un tipo de cola reflejada compuesta por un nodo líder (réplica principal) y nodos seguidores (otras réplicas). Cada nodo se encuentra en una zona de disponibilidad diferente, por lo que, si un nodo no está disponible temporalmente, la entrega de mensajes continúa con una réplica líder recién elegida en otra zona de disponibilidad. Las colas de cuórum son útiles para gestionar los mensajes dudosos, que se producen cuando un mensaje falla y se vuelve a poner en cola varias veces.

Replicación de datos entre regiones para ActiveMQ en Amazon MQ

La [replicación de datos entre regiones](#) (CRDR) permite la replicación asíncrona de mensajes desde el agente primario de una región de AWS principal al agente de réplicas de una región de réplicas. Al emitir una solicitud de conmutación por error a la API de Amazon MQ, el agente de réplicas actual se asciende al rol de agente principal y el agente principal actual se degrada al rol de réplicas.

¿Cómo puedo comenzar a utilizar Amazon MQ?

Para empezar a utilizar ActiveMQ en Amazon MQ, consulte la siguiente documentación:

- [Introducción: Creación y conexión con un agente de ActiveMQ](#)
- [the section called “Implementación de un agente”](#)
- [Tutoriales de ActiveMQ](#)
- [the section called “Prácticas recomendadas de Amazon MQ para ActiveMQ”](#)

Para empezar a utilizar RabbitMQ en Amazon MQ, consulte la siguiente documentación:

- [Introducción: Cómo crear un agente de RabbitMQ y conectarse a él](#)
- [the section called “Implementación de un agente RabbitMQ”](#)
- [the section called “Tutoriales de RabbitMQ”](#)
- [the section called “Prácticas recomendadas de Amazon MQ para RabbitMQ”](#)

Para obtener más información acerca de las API REST de Amazon MQ, consulte la [Referencia de la API REST de Amazon MQ](#).

Para obtener más información acerca de los comandos AWS CLI de Amazon MQ, consulte [Amazon MQ en la Referencia de los comandos de la AWS CLI](#).

¿Cómo puedo proporcionar comentarios a Amazon MQ?

Agradecemos y alentamos sus comentarios sobre la documentación. Puede usar los íconos con el pulgar hacia arriba y el pulgar hacia abajo que hay a la derecha para enviar comentarios, o puede usar el formulario “Enviar comentarios” que aparece a continuación.

Para ponerse en contacto con el equipo de Amazon MQ, utilice el [foro de debate de Amazon MQ](#).

Configuración de Amazon MQ

Antes de poder utilizar Amazon MQ, debe llevar a cabo los siguientes pasos.

Temas

- [Paso 1: requisitos previos](#)
- [Paso 2: crea un usuario y obtén tus credenciales AWS](#)
- [Paso 3: Prepararse para usar los códigos de ejemplo](#)
- [Siguiendo pasos](#)

Paso 1: requisitos previos

Inscríbase en un Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abrir <https://portal.aws.amazon.com/billing/registro>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica o mensaje de texto e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea uno. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [Tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a <https://aws.amazon.com/> y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [Consola de administración de AWS](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Iniciar sesión como usuario raíz](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

Inicio de sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, use la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

Paso 2: crea un usuario y obtén tus credenciales AWS

Los usuarios necesitan acceso programático si quieren interactuar con personas AWS ajenas a. Consola de administración de AWS La forma de conceder el acceso programático depende del tipo de usuario que acceda. AWS

Para conceder acceso programático a los usuarios, elija una de las siguientes opciones.

¿Qué usuario necesita acceso programático?	Para	Mediante
IAM	(Recomendado) Utilice las credenciales de la consola como credenciales temporales para firmar las solicitudes programáticas dirigidas al AWS CLI AWS SDKs, o. AWS APIs	Siga las instrucciones de la interfaz que desea utilizar: • Para ello AWS CLI, consulte Iniciar sesión para el desarrollo AWS local en la Guía del AWS Command Line Interface usuario. • Para ello AWS SDKs, consulte Iniciar sesión para el desarrollo AWS local en la Guía de referencia de AWS SDKs and Tools.

¿Qué usuario necesita acceso programático?	Para	Mediante
Identidad del personal (Usuarios administrados en el IAM Identity Center)	Utilice credenciales temporales para firmar las solicitudes programáticas dirigidas al AWS CLI, AWS SDKs, o AWS APIs.	Siga las instrucciones de la interfaz que desea utilizar: • Para ello AWS CLI, consulte Configuración del AWS CLI uso AWS IAM Identity Center en la Guía del AWS Command Line Interface usuario. • Para AWS SDKs ver las herramientas y AWS APIs, consulte la autenticación del Centro de Identidad de IAM en la Guía de referencia de herramientas AWS SDKs y herramientas.
IAM	Utilice credenciales temporales para firmar las solicitudes programáticas dirigidas al AWS CLI AWS SDKs, o. AWS APIs	Siga las instrucciones de Uso de credenciales temporales con AWS recursos de la Guía del usuario de IAM.

¿Qué usuario necesita acceso programático?	Para	Mediante
IAM	(No recomendado) Utilice credenciales de larga duración para firmar las solicitudes programáticas dirigidas al AWS CLI AWS SDKs, o. AWS APIs	Siga las instrucciones de la interfaz que desea utilizar: • Para ello AWS CLI, consulte Autenticación con credenciales de usuario de IAM en la Guía del AWS Command Line Interface usuario. • Para obtener AWS SDKs información sobre las herramientas, consulte Autenticarse con credenciales de larga duración en la Guía de referencia de herramientas AWS SDKs y herramientas. • Para ello AWS APIs, consulte Administrar las claves de acceso para los usuarios de IAM en la Guía del usuario de IAM.

Paso 3: Prepararse para usar los códigos de ejemplo

Los siguientes tutoriales muestran cómo puede trabajar con los corredores de Amazon MQ mediante el Consola de administración de AWS uso de los corredores de Amazon MQ para ActiveMQ y Amazon MQ para RabbitMQ, y cómo conectarse a sus corredores de Amazon MQ para ActiveMQ de forma programática. Si quiere usar el código de ejemplo, debe instalar [Java Standard Edition Development Kit](#) y efectuar algunos cambios de configuración en el código de ejemplo.

También puede crear y administrar corredores mediante programación mediante la API REST de Amazon [MQ](#) y. AWS SDKs

Siguientes pasos

Ahora que está preparado para trabajar con Amazon MQ, empiece por [crear un agente](#). Según el tipo de motor de su agente, puede [conectar una aplicación Java con el agente de Amazon MQ para ActiveMQ](#) o utilizar la biblioteca de cliente Java de RabbitMQ para [conectar una aplicación basada en JVM con su agente de Amazon MQ para RabbitMQ](#).

Introducción: Creación y conexión con un agente de ActiveMQ

Un agente es un entorno de agente de mensajes que se ejecuta en Amazon MQ. Es el componente básico de Amazon MQ. La descripción combinada de la instancia del agente clase (m5) y tamaño (large, medium) se conoce como tipo de instancias del agente (por ejemplo, mq.m5.large). Para obtener más información, consulte [¿Qué es un agente de Amazon MQ para ActiveMQ?](#).

Creación de un agente de ActiveMQ

La primera tarea, y la más habitual, en Amazon MQ es la creación de un agente. El siguiente ejemplo muestra cómo puede utilizar la Consola de administración de AWS para crear un agente básico.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la página Select broker engine (Seleccionar motor de agente), elija Apache ActiveMQ.
3. En la página Select deployment and storage (Seleccionar implementación y almacenamiento), sección Deployment mode and storage type (Modo de implementación y tipo de almacenamiento), haga lo siguiente:
 - a. Elija el modo de implementación (por ejemplo, agente activo/en espera). Para obtener más información, consulte [Opciones de implementación de Amazon MQ para agentes de ActiveMQ](#).
 - Un agente de una sola instancia está compuesto por un agente en una zona de disponibilidad. El agente se comunica con su aplicación y con un volumen de almacenamiento de Amazon EBS o Amazon EFS. Para obtener más información, consulte [Opción 1: agentes de una sola instancia de Amazon MQ](#).
 - Un agente activo/en espera para conseguir una alta disponibilidad está compuesto por dos agentes en dos zonas de disponibilidad diferentes, configurados en un par redundante. Estos agentes se comunican de forma sincrónica con su aplicación y con Amazon EFS. Para obtener más información, consulte [Opción 2: active/standby agentes Amazon MQ para alta disponibilidad](#).
 - b. Elija el tipo de almacenamiento (por ejemplo, EBS). Para obtener más información, consulte [Storage](#).

 Note

Amazon EBS replica los datos dentro de una sola zona de disponibilidad y no admite el modo de implementación [ActiveMQ activa/en espera](#).

- c. Elija Siguiente.
4. En la página Configure settings (Configurar ajustes), sección Details (Detalles), haga lo siguiente:
 - a. Ingrese el nombre del agente.

 Important

No agregue información de identificación personal (PII) ni ninguna otra información confidencial en los nombres de agente. Los nombres de agente y los nombres de usuario están disponibles para otros servicios de AWS, incluido CloudWatch Logs. Los nombres de agente no están diseñados para usarse con información privada o confidencial.

 Note

En la sección Additional settings (Configuración adicional), también puede configurar lo siguiente:

- [Configuraciones](#)
- [Registros de CloudWatch](#)
- Acceso privado
- [Período de mantenimiento del agente](#)

- b. Elija el tipo de instancias del agente (por ejemplo, mq.m5.large). Para obtener más información, consulte [Broker instance types](#).
5. En la sección de acceso a la consola web de ActiveMQ, introduzca un nombre de usuario y una contraseña. Las siguientes restricciones se aplican a los nombres de usuario y las contraseñas de los agentes:

- El nombre de usuario solo puede contener caracteres alfanuméricos, guiones, puntos, guiones bajos y tildes (- . _ ~).
- Esta contraseña debe tener 12 caracteres como mínimo, debe contener al menos 4 caracteres únicos y no puede incluir comas, dos puntos ni signos de igual (,:=).

 Important

No agregue información de identificación personal (PII) ni ninguna otra información confidencial en los nombres de usuario de agente. Los nombres de usuario de agente están disponibles para otros servicios de AWS, incluido CloudWatch Logs. Los nombres de usuario de agente no están diseñados para usarse con información privada o confidencial.

6. Elija Implementar.

Mientras Amazon MQ crea su agente, muestra el estado Creation in progress (Creación en curso).

El proceso de creación del agente de tarda unos 15 minutos.

Cuando el agente se crea correctamente, Amazon MQ muestra el estado Running (En ejecución).

7. Seleccione **MyBroker**.

En la sección Connect (Conectar) de la página **MyBroker**, anote la URL de la consola web de [ActiveMQ](#), por ejemplo:

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

Además, anote los [puntos de enlace de protocolo de nivel de cable](#). A continuación, se muestra un ejemplo de un punto de enlace de OpenWire:

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

Introducción: Cómo crear un agente de RabbitMQ y conectarse a él

Un agente es un entorno de agente de mensajes que se ejecuta en Amazon MQ. Es el componente básico de Amazon MQ. La descripción combinada de la instancia del agente clase (m5) y tamaño (large, medium) se conoce como tipo de instancias del agente (por ejemplo, mq.m5.large). Para obtener más información, consulte [¿Qué es un agente de Amazon MQ para RabbitMQ?](#)

Crear un agente de RabbitMQ

La primera tarea, y la más habitual, en Amazon MQ es la creación de un agente. El siguiente ejemplo muestra cómo puede utilizar la Consola de administración de AWS para crear un agente básico.

Al crear un Amazon MQ para un agente de RabbitMQ, siga las [prácticas recomendadas de configuración del agente para RabbitMQ](#) a fin de maximizar el rendimiento del agente y optimizar la eficiencia del procesamiento de mensajes.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la página Select broker engine (Seleccionar motor del agente), elija RabbitMQ y, luego, Next (Siguiente).
3. En la página Select deployment mode (Seleccionar modo de implementación), elija el modo de implementación, por ejemplo, Cluster deployment (Implementación de clúster) y, luego, elija Next (Siguiente).
 - Un agente de una sola instancia está compuesto por un agente en una zona de disponibilidad situada detrás de un Network Load Balancer (NLB). El agente se comunica con su aplicación y con un volumen de almacenamiento de Amazon EBS. Para obtener más información, consulte [Opción 1: Agente de una sola instancia de Amazon MQ para RabbitMQ](#).
 - Una Implementación de clúster de RabbitMQ para alta disponibilidad es una agrupación lógica de tres nodos de agente de RabbitMQ situados detrás de un Network Load Balancer, cada uno de los cuales comparte usuarios, colas y un estado distribuido entre varias zonas de disponibilidad (AZ). Para obtener más información, consulte [Opción 2: Implementación de un clúster de Amazon MQ para RabbitMQ](#).
4. En la página Configure settings (Configurar ajustes), sección Details (Detalles), siga estos pasos:

- a. Ingrese el nombre del agente.

 Important

No agregue información de identificación personal (PII) ni ninguna otra información confidencial en los nombres de agente. Los nombres de agente y los nombres de usuario están disponibles para otros servicios de AWS, incluido CloudWatch Logs. Los nombres de agente no están diseñados para usarse con información privada o confidencial.

- b. Elija el tipo de instancias del agente (por ejemplo, mq.m7g.large). Para obtener más información, consulte [Broker instance types](#).

5. En la página Configure settings (Configurar ajustes), sección RabbitMQ access (Acceso a RabbitMQ), proporcione un nombre de usuario y una contraseña. Las siguientes restricciones se aplican a las credenciales de inicio de sesión de agente:
 - El nombre de usuario solo puede contener caracteres alfanuméricos, guiones, puntos y guiones bajos (- . _). Este valor no debe contener ningún carácter de tilde (~). Amazon MQ prohíbe utilizar guest como nombre de usuario.
 - Esta contraseña debe tener 12 caracteres como mínimo, debe contener al menos 4 caracteres únicos y no puede incluir comas, dos puntos ni signos de igual (,:=).

 Important

No agregue información de identificación personal (PII) ni ninguna otra información confidencial en los nombres de usuario de agente. Los nombres de usuario de agente están disponibles para otros servicios de AWS, incluido CloudWatch Logs. Los nombres de usuario de agente no están diseñados para usarse con información privada o confidencial.

 Note

En la sección Additional settings (Configuración adicional), también puede configurar lo siguiente:

- [Configuraciones](#)
- [Registros de CloudWatch](#)
- Acceso privado
- [Período de mantenimiento del agente](#)

6. Elija Siguiente.
7. En la página Review and create (Revisar y crear), puede revisar las opciones que seleccionó y modificarlas si es necesario.
8. Elija Create broker (Crear agente).

Mientras Amazon MQ crea su agente, muestra el estado Creation in progress (Creación en curso).

El proceso de creación del agente de tarda unos 15 minutos.

Cuando el agente se crea correctamente, Amazon MQ muestra el estado Running (En ejecución).

9. Seleccione **MyBroker**.

En la sección Connect (Conectar) de la página **MyBroker**, anote la URL de la [consola web de RabbitMQ](#) de su agente, por ejemplo:

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

Además, anote el [punto de enlace de Secure-AMQP](#). A continuación se muestra un ejemplo de un punto de enlace amqps que expone al puerto de agente de escucha 5671.

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

Administración de un agente de Amazon MQ

Tras crear un agente de Amazon MQ, podrá administrar y mantener sus distintos componentes.

Temas

- [Conexión con Amazon MQ](#)
- [Autenticación y autorización para agentes de Amazon MQ](#)
- [Actualización de una versión del motor del agente de Amazon MQ](#)
- [Actualización de un tipo de instancia del agente de Amazon MQ](#)
- [Tipos de almacenamiento de Amazon MQ para ActiveMQ](#)
- [Configuración de un agente de Amazon MQ privado](#)
- [Programación del período de mantenimiento de un agente de Amazon MQ](#)
- [Reiniciar un agente de Amazon MQ](#)
- [Eliminación de un agente de Amazon MQ](#)
- [Estados del agente de Amazon MQ](#)
- [Adición de etiquetas a los recursos de Amazon MQ](#)

Conexión con Amazon MQ

Puede conectarse a Amazon MQ desde otros servicios AWS mediante puntos de conexión de servicio y puntos de conexión de agentes.

Puntos de conexión de servicio

Se utilizan los siguientes métodos de conexión para la API del servicio Amazon MQ:

Dominios	Método de conexión
mq. <i>region</i> .amazonaws.com	IPv4
mq. <i>region</i> .api.aws	Doble pila (IPv4 e IPv6)
mq-fips. <i>region</i> .amazonaws.com	Solo FIPS con IPv4
mq-fips. <i>region</i> .api.aws	FIPS con pila doble

Puntos de conexión de los agentes

Los agentes de Amazon MQ utilizan los siguientes métodos de conexión:

Dominios	Método de conexión
<code>brokerId.mq.region.amazonaws.com</code>	IPv4
<code>brokerId.mq.region.on.aws</code>	Doble pila (IPv4 e IPv6)

 **Note**

Los agentes de Amazon MQ para ActiveMQ no admiten la doble pila.

Conexión a Amazon MQ mediante puntos de conexión de doble pila (IPv4 e IPv6)

Los puntos de conexión de doble pila admiten tráfico IPv4 e IPv6. Cuando realiza una solicitud a un punto de conexión de doble pila, la URL del punto de conexión resuelve a una dirección IPv4 o IPv6. Para obtener más información sobre los puntos de conexión de doble pila y FIPS, consulte la [Guía de referencia de SDK](#).

Amazon MQ solo es compatible con nombres de puntos de conexión de doble pila regionales, lo que implica que debe especificar la región AWS como parte del nombre. Los nombres de puntos de conexión de doble pila utilizan la siguiente convención de nomenclatura: `mq.region.api.aws`. Por ejemplo, el nombre del punto de conexión de doble pila para la región `eu-west-1` es `mq.eu-west-1.api.aws`.

Para obtener la lista completa de puntos de conexión de Amazon MQ, consulte la [Referencia general de AWS](#).

Conexión a Amazon MQ mediante PrivateLink de AWS

Los puntos de conexión de [PrivateLink de AWS](#) para la API de Amazon MQ compatibles con IPv4 e IPv6 proporcionan conectividad privada entre las nubes privadas virtuales (VPC) y la API de Amazon MQ sin exponer el tráfico a la Internet pública.

Note

La compatibilidad para PrivateLink solo está disponible para el punto de conexión de la API Amazon MQ, no para el punto de conexión del agente. Para obtener más información sobre cómo conectarse de forma privada a un punto de conexión de agente, consulte [Configuring a private Amazon MQ broker](#).

Para acceder a la API de Amazon MQ mediante PrivateLink, primero debe crear un [punto de conexión de VPC de interfaz](#) en la VPC específica desde la que desee conectarse. Al crear el punto de conexión de VPC, utilice el nombre del servicio `com.amazonaws.region.mq` o `com.amazonaws.region.mq-fips` para los puntos de conexión de FIPS.

Cuando llama a Amazon MQ mediante la CLI o el SDK de AWS, debe especificar la URL del punto de conexión para usar el nombre de dominio de doble pila: `mq.region.api.aws` o `mq-fips.region.api.aws`. PrivateLink para Amazon MQ no admite el nombre de dominio predeterminado que termina en `amazonaws.com`. Para obtener más información, consulte [Puntos de conexión de doble pila y FIPS](#) en la Guía de referencia de SDK.

El siguiente ejemplo de CLI muestra cómo llamar a `describe-broker-engine-type` en la región de Asia-Pacífico (Sídney) a través de un punto de conexión de VPC de Amazon MQ.

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

Para ver otras formas de configurar el punto de conexión en la CLI, consulte [Uso de puntos de conexión en la CLI de AWS](#)

También puede determinar el acceso de los usuarios a los puntos de conexión de VPC mediante políticas de punto de conexión de VPC. Para obtener más información, consulte [Controlar el acceso a puntos de conexión de VPC con políticas de punto de conexión](#).

Autenticación y autorización para agentes de Amazon MQ

Amazon MQ ofrece varios métodos de autenticación y autorización para proteger su infraestructura de mensajería de acuerdo con los requisitos de su organización.

Autenticación y autorización de Amazon MQ para RabbitMQ

Amazon MQ para RabbitMQ admite los siguientes métodos de autenticación y autorización:

Autenticación y autorización simple

En este método, los usuarios del agente se almacenan internamente en el agente de RabbitMQ y se administran a través de la consola web o la API de administración. Los permisos para vhosts, intercambios, colas y temas se configuran directamente en RabbitMQ. Este es el método predeterminado. Para obtener más información, consulte Autenticación y autorización [sencillas](#).

OAuth Autenticación y autorización 2.0

En este método, los usuarios intermediarios y sus permisos los administra un proveedor de identidades (IdP) OAuth 2.0 externo. La autenticación de los usuarios y los permisos de recursos para vhosts, bolsas, colas y temas se centralizan mediante el sistema Scope del proveedor OAuth 2.0. Esto simplifica la administración de usuarios y permite la integración con los sistemas de identidad existentes. Para obtener más información, consulte [Autenticación y autorización OAuth 2.0](#).

Autenticación y autorización de IAM

En este método, los usuarios del broker se autentican con credenciales de AWS IAM mediante la federación saliente de [IAM](#). Las credenciales de IAM se utilizan para obtener los tokens JWT del AWS Security Token Service (STS), y estos tokens JWT sirven como tokens 2.0 para la autenticación. OAuth Este método aprovecha la compatibilidad OAuth 2.0 existente en Amazon MQ para RabbitMQ, AWS donde actúa como proveedor de identidad 2.0. OAuth AWS IAM gestiona la autenticación de los usuarios, mientras que los permisos de recursos para vhosts, bolsas, colas y temas se gestionan mediante políticas de IAM y alias de ámbito configurados en RabbitMQ. [Para obtener más información, consulte Autenticación y autorización de IAM](#).

Autenticación y autorización de LDAP

En este método, los usuarios intermediarios y sus permisos se administran mediante un servicio de directorio LDAP externo. La autenticación de los usuarios y los permisos de recursos se centralizan a través del servidor LDAP, lo que permite a los usuarios acceder a RabbitMQ con sus credenciales de servicio de directorio existentes. [Para obtener más información, consulte Autenticación y autorización LDAP](#).

Autenticación y autorización HTTP

En este método, los usuarios del broker y sus permisos se administran mediante un servidor HTTP externo. La autenticación de los usuarios y los permisos de recursos se centralizan a través del servidor HTTP, lo que permite a los usuarios acceder a RabbitMQ con su propio proveedor de autenticación y autorización. Para obtener más información sobre este método, consulte [Autenticación y autorización HTTP](#).

Autenticación con certificado SSL

Amazon MQ admite el TLS mutuo (mTLS) para los corredores de RabbitMQ. El complemento de autenticación SSL utiliza los certificados de cliente de las conexiones mTLS para autenticar a los usuarios. En este método, los usuarios del broker se autentican mediante certificados de cliente X.509 en lugar de credenciales de nombre de usuario y contraseña. El certificado del cliente se valida con una autoridad de certificación (CA) de confianza y el nombre de usuario se extrae de un campo del certificado, como el nombre común (CN) o el nombre alternativo del sujeto (SAN). Este método proporciona una autenticación sólida sin transmitir las credenciales a través de la red. Para obtener más información, consulte [Autenticación con certificado SSL](#).

Note

RabbitMQ admite varios métodos de autenticación y autorización para su uso simultáneo. Por ejemplo, puede habilitar la autenticación OAuth 2.0 y la autenticación simple (interna). Para obtener más información, consulte la sección del tutorial OAuth 2.0 sobre cómo [habilitar la autenticación OAuth 2.0 y la autenticación simple \(interna\)](#) y la documentación sobre el control de [acceso de RabbitMQ](#).

Amazon MQ recomienda crear un usuario interno al probar las configuraciones de autenticación. Esto permite validar la configuración de acceso mediante la API de administración de RabbitMQ. [Para obtener más información, consulte Validación de acceso.](#)

Autenticación y autorización de Amazon MQ para ActiveMQ

Amazon MQ para ActiveMQ admite los siguientes métodos de autenticación y autorización:

Autenticación y autorización simple

En este método, los usuarios de los corredores se crean y administran a través de la consola o API de Amazon MQ. Los usuarios se pueden configurar con permisos específicos para acceder a las

colas, los temas y la consola web ActiveMQ. Para obtener más información sobre este método, consulte [Creación de un usuario de ActiveMQ broker](#).

Autenticación y autorización LDAP

En este método, los usuarios intermediarios se autentican mediante las credenciales almacenadas en el servidor LDAP. Puede agregar, eliminar y modificar usuarios y asignar permisos a temas y colas a través del servidor LDAP, lo que proporciona autenticación y autorización centralizadas. Para obtener más información sobre este método, consulte [Integración de agentes ActiveMQ con LDAP](#).

Actualización de una versión del motor del agente de Amazon MQ

Amazon MQ proporciona habitualmente nuevas versiones de motor de agente para todos los tipos de motores de agente compatibles. Las nuevas versiones de motor incluyen parches de seguridad, correcciones de errores y otras mejoras del motor del agente.

Amazon MQ organiza los números de versión según la especificación semántica del control de versiones como X.Y.Z. En las implementaciones de Amazon MQ, X indica la versión principal, Y representa el número de versión secundaria y Z hace referencia al número de versión del parche. Amazon MQ admite dos tipos de actualizaciones:

- Actualización de versión principal: se produce cuando cambian los números de la versión de motor principal. Por ejemplo, la actualización de la versión 3.13 de RabbitMQ a la versión 4.2 se considera una actualización de la versión principal.
- Actualización de versión secundaria: se produce cuando solo cambia el número de la versión de motor secundaria. Por ejemplo, actualizar desde la versión 3.11 a la versión 3.12 se considera una actualización de versión menor.

Puede actualizar manualmente su agente en cualquier momento a la siguiente versión principal o secundaria compatible. Amazon MQ gestiona la actualización a la última versión de parche compatible para todos los corredores durante el período de [mantenimiento](#) programado. Las actualizaciones de las versiones, tanto manuales como automáticas, se producen durante el período de mantenimiento programado o después de [reiniciar el broker](#). Amazon MQ actualiza su agente a la siguiente versión secundaria cuando la versión secundaria actual llega al fin del soporte.

Actualización manual de la versión del motor

Puede actualizar la versión de motor de un bróker mediante la Consola de administración de AWS, la AWS CLI, o la API de Amazon MQ.

Consola de administración de AWS

Para actualizar la versión de motor de un bróker mediante la Consola de administración de AWS:

1. En la página de detalles del agente, elija **Edit** (Editar).
2. Dentro de **Specifications** (Especificaciones), en **Broker engine version** (Versión de motor del agente), elija el número de la versión nueva de la lista desplegable.
3. Desplácese a la parte inferior de la página y elija **Schedule modifications** (Programar modificaciones).
4. En la página **Schedule broker modifications** (Programar modificaciones del agente), para **When to apply modifications** (Cuándo aplicar modificaciones), elija una de las siguientes opciones.
 - Seleccione **After the next reboot** (Después del próximo reinicio) si desea que Amazon MQ complete la actualización de versiones durante el próximo período de mantenimiento programado.
 - Seleccione **Immediately** (De inmediato) si desea reiniciar el agente y actualizar la versión del motor inmediatamente.

Important

Los agentes de instancia única están desconectados mientras se reinician. En el caso de los agentes de clústeres, solo un nodo está inactivo a la vez mientras el agente se reinicia.

5. Seleccione **Apply** (Aplicar) para terminar de aplicar los cambios.

AWS CLI

Para actualizar la versión de motor de un bróker mediante la AWS CLI:

1. Use el comando [update-broker](#) de la CLI y especifique los siguientes parámetros, como se muestra en el ejemplo.

- `--broker-id`: ID único que genera Amazon MQ para el agente. Puede consultar el ID a partir del ARN de su agente. Por ejemplo, si se le da el ARN `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, el ID del agente sería `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- `--engine-version`: número de versión de motor al que se va a actualizar el motor del agente.

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

2. (Opcional) Utilice el comando CLI [reboot-broker](#) para reiniciar el broker si desea actualizar la versión del motor inmediatamente.

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

Si no desea reiniciar el agente y aplicar los cambios de inmediato, Amazon MQ lo actualizará durante el próximo periodo de mantenimiento programado.

Important

Los agentes de instancia única están desconectados mientras se reinician. En el caso de los agentes de clústeres, solo un nodo está inactivo a la vez mientras el agente se reinicia.

API de Amazon MQ

Para actualizar la versión de motor de un agente a través de la API de Amazon MQ

1. Use la operación [UpdateBroker](#) de la API. Especifique `broker-id` como parámetro de ruta. En los siguientes ejemplos, se asume que el agente está en la región `us-west-2`. Para obtener más información acerca de los puntos de conexión de Amazon MQ disponibles, consulte [cuotas y puntos de conexión de Amazon MQ](#) en Referencia general de AWS

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

Authorization: *authorization-string*

Utilice `engineVersion` en la carga de la solicitud para especificar el número de versión al que se va a actualizar el agente.

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

2. (Opcional) Utilice la operación de la [RebootBroker](#) API para reiniciar el broker si desea actualizar la versión del motor de forma inmediata. `broker-id` se especifica como parámetro de ruta.

```
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
```

Si no desea reiniciar el agente y aplicar los cambios de inmediato, Amazon MQ lo actualizará durante el próximo periodo de mantenimiento programado.

Important

Los agentes de instancia única están desconectados mientras se reinician. En el caso de los agentes de clústeres, solo un nodo está inactivo a la vez mientras el agente se reinicia.

Actualización de un tipo de instancia del agente de Amazon MQ

Important

Las instancias `mq.m7g.x` solo están disponibles en Amazon MQ para agentes de RabbitMQ. Los agentes de Amazon MQ para ActiveMQ solo usan instancias `mq.m5.x`.

La descripción combinada de la instancia del agente clase (m7g) y tamaño (large) es un tipo de instancias del agente (por ejemplo, mq.m7g.large). Al elegir un tipo de instancia, es importante tener en cuenta los factores que afectarán al rendimiento del agente:

- el número de clientes y colas
- el volumen de mensajes enviados
- mensajes guardados en la memoria
- mensajes redundantes

Se recomienda usar los tipos de instancias de agente más pequeños (mq.m7g.medium) para probar el rendimiento de las aplicaciones. Asimismo, se aconsejan los tipos de instancias de agente más grandes (mq.m7g.large o superiores) para los niveles de producción de clientes y colas, rendimientos altos, los mensajes en memoria y los mensajes redundantes.

Le recomendamos que actualice a un tipo de instancia mayor (es decir, de micro a large) si tiene problemas de rendimiento o si va a pasar de un entorno de pruebas a un entorno de producción. Para actualizar el tipo de instancia, puede usar la Consola de administración de AWS, la AWS CLI o la API de Amazon MQ.

Consola de administración de AWS

Para actualizar a un tipo de instancia más grande a través de la Consola de administración de AWS, haga lo siguiente:

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En el panel de navegación izquierdo, elija Brokers (Agentes) y seleccione el agente que desea actualizar de la lista.
3. En la página de detalles del agente, elija Edit (Editar).
4. En Specifications (Especificaciones), para el tipo de instancia del agente, elija el nuevo tipo de instancia en la lista desplegable.
5. En la parte inferior de la página y elija Schedule modifications (Programar modificaciones).
6. En la página Schedule broker modifications (Programar modificaciones del agente), para When to apply modifications (Cuándo aplicar modificaciones), elija una de las siguientes opciones.
 - Seleccione After the next reboot (Después del próximo reinicio) si desea que Amazon MQ complete la actualización durante el próximo período de mantenimiento programado.

- Seleccione **Immediately** (De inmediato) si desea reiniciar el agente y actualizar el tipo de instancia inmediatamente.

 Important

Los agentes de instancia única están desconectados mientras se reinician. En el caso de los agentes de clústeres, solo un nodo está inactivo a la vez mientras el agente se reinicia.

7. Seleccione **Apply** (Aplicar) para terminar de aplicar los cambios.

AWS CLI

Para actualizar el tipo de instancia de un agente a través de la AWS CLI

1. Use el comando [modify-broker](#) de la CLI y especifique los siguientes parámetros, como se muestra en el ejemplo.
 - `--broker-id`: ID único que genera Amazon MQ para el agente.
 - `--host-instance-type`: número de versión de motor al que se va a actualizar el motor del agente.

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

2. (Opcional) Utilice el comando [reboot-broker](#) de la CLI para reiniciar su agente si desea actualizar el tipo de instancia de inmediato.

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

Si no desea reiniciar el agente y aplicar los cambios de inmediato, Amazon MQ lo actualizará durante el próximo período de mantenimiento programado.

⚠ Important

Los agentes de instancia única están desconectados mientras se reinician. En el caso de los agentes de clústeres, solo un nodo está inactivo a la vez mientras el agente se reinicia.

API de Amazon MQ

Para actualizar el tipo de instancia de un agente a través de la API de Amazon MQ

1. Utilice la operación [UpdateBroker](#) de la API. Especifique `broker-id` como parámetro de ruta. En los siguientes ejemplos, se asume que el agente está en la región `us-west-2`. Para obtener más información acerca de los puntos de conexión de Amazon MQ disponibles, consulte [cuotas y puntos de conexión de Amazon MQ](#) en Referencia general de AWS

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

Utilice `host-instance-type` en la carga útil de la solicitud para especificar el tipo de instancia al que se va a actualizar el agente.

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

2. (Opcional) Utilice la operación [RebootBroker](#) de la API para reiniciar su agente si desea actualizar la versión de motor de inmediato. Se especifica `broker-id` como parámetro de ruta.

```
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
```

Si no desea reiniciar el agente y aplicar los cambios de inmediato, Amazon MQ lo actualizará durante el próximo período de mantenimiento programado.

 Important

Los agentes de instancia única están desconectados mientras se reinician. En el caso de los agentes de clústeres, solo un nodo está inactivo a la vez mientras el agente se reinicia.

Tipos de almacenamiento de Amazon MQ para ActiveMQ

Amazon MQ para ActiveMQ es compatible con Amazon Elastic File System (EFS) y Amazon Elastic Block Store (EBS). De forma predeterminada, los agentes de ActiveMQ utilizan Amazon EFS para el almacenamiento del agente. Para aprovechar el alto nivel de durabilidad y la replicación en varias zonas de disponibilidad, utilice Amazon EFS. Para aprovechar la baja latencia y el alto rendimiento, utilice Amazon EBS.

 Important

- Puede utilizar Amazon EBS únicamente con la familia de tipos de instancias de agente mq.m5.
- Si bien puede cambiar el tipo de instancia de agente, no puede cambiar el tipo de almacenamiento de agente después de crear el agente.
- Amazon EBS replica los datos dentro de una sola zona de disponibilidad y no admite el modo de implementación [ActiveMQ activa/en espera](#).

Diferencias entre tipos de almacenamiento

La tabla que se incluye a continuación proporciona información general resumida de las diferencias entre los tipos de almacenamiento en memoria, de Amazon EFS y Amazon EBS para agentes de ActiveMQ.

Storage Type	Persistencia	Ejemplo de caso de uso	Número máximo aproximado de mensajes puestos en cola por productor , por segundo (mensaje de 1 KB)	Replicación
En memoria	No persistente	• Cotizaciones de acciones • Actualizaciones de datos de ubicación • Datos modificados con frecuencia	5 000	Ninguno
Amazon EBS	Persistente	• Volúmenes elevados de texto • Procesamiento de pedidos	500	Varias copias dentro de una única zona de disponibilidad (AZ)
Amazon EFS	Persistente	Transacciones financieras	80	Múltiples copias en múltiples AZs

El almacenamiento de mensajes en memoria proporciona la latencia más baja y el rendimiento más alto. Sin embargo, los mensajes se pierden durante el reemplazo de instancias o el reinicio del agente.

Amazon EFS está diseñado para ofrecer una alta durabilidad y se replica en varios sitios AZs para evitar la pérdida de datos que se produzca por el fallo de algún componente individual o por un problema que afecte a la disponibilidad de una zona de disponibilidad. Amazon EBS tiene un rendimiento optimizado y se replica en varios servidores dentro de una única zona de disponibilidad.

Configuración de un agente de Amazon MQ privado

Un agente privado no tiene accesibilidad pública y no se puede acceder a él desde fuera de su VPC. Antes de configurar un agente privado, consulte la siguiente información sobre VPCs las subredes y los grupos de seguridad:

- VPCs
 - Las subredes y los grupos de seguridad de un agente deben estar en la misma VPC.
 - Cuando utilice un agente privado, es posible que vea direcciones IP que no configuró con su VPC. Estas direcciones IP pertenecen a la infraestructura de Amazon MQ y no requieren ninguna acción.
- Subredes
 - Si las subredes están dentro de una VPC compartida, la VPC debe pertenecer a la misma cuenta que creó el agente.
 - Si no se proporciona ninguna subred, se usarán las subredes predeterminadas de la VPC predeterminada.
 - No se pueden cambiar las subredes utilizadas una vez creado el agente.
 - En el caso de los clústeres y active/standby los agentes, las subredes deben estar en distintas zonas de disponibilidad.
 - En el caso de los agentes de instancia única, puede especificar qué subred utilizar y el agente se creará dentro de la misma zona de disponibilidad.
- Grupos de seguridad
 - Si no se proporciona ningún grupo de seguridad, se utilizarán los grupos de seguridad predeterminados de la VPC predeterminada.
 - La instancia única, el clúster y active/standby los corredores requieren al menos un grupo de seguridad (por ejemplo, el grupo de seguridad predeterminado).

Note

Los agentes públicos de RabbitMQ no utilizan subredes ni grupos de seguridad.

- Una vez creado el agente, no se puede cambiar el grupo de seguridad utilizado. Los propios grupos de seguridad aún se pueden modificar.

Configurar un agente privado en el Consola de administración de AWS

Para configurar un agente privado, comience [creando un nuevo agente](#) en la Consola de administración de AWS. A continuación, en la sección Network settings (Configuración de red), para configurar la conectividad de su agente, haga lo siguiente:

1. Elija el acceso privado para su agente. Para conectarse a un corredor privado, puede usar IPv4 IPv6, o doble pila (IPv4 y IPv6). Para obtener más información, consulte [Connecting to Amazon MQ](#).
2. A continuación, elija Usar la VPC, las subredes y los grupos de seguridad predeterminados, o elija Seleccionar las VPC, subredes y grupos de seguridad existentes. Si no desea utilizar la VPC, las subredes o los grupos de seguridad predeterminados o existentes, debe crear uno nuevo para conectarse al agente privado.

Note

Para el acceso de un agente privado, el método de conexión será el mismo que el tipo de IP seleccionado de la subred. Una vez creado el agente, el punto de conexión de VPC no se puede cambiar y siempre tendrá el tipo de IP de las subredes seleccionadas. Si desea usar un nuevo tipo de IP, debe crear un agente nuevo.

Note

Amazon MQ para ActiveMQ no utiliza puntos de conexión de VPC. Al crear un agente de ActiveMQ por primera vez, Amazon MQ aprovisiona una interfaz de red elástica (ENI) en la VPC. Los grupos de seguridad se ubican en el ENI y se pueden usar tanto para agentes públicos como privados.

Acceso a la consola web de un agente de Amazon MQ sin accesibilidad pública

Cuando desactivas la accesibilidad pública para tu corredor, el ID de AWS cuenta que creó el corredor puede acceder al corredor privado. Si desactiva la accesibilidad pública para su agente, debe realizar los siguientes pasos para obtener acceso a la consola web del agente.

1. Cree una instancia EC2 de Linux en `public-vpc` (con una dirección IP pública, si es necesario).
2. Para verificar que la VPC está configurada correctamente, establezca una conexión `ssh` a la instancia EC2 y utilice el comando `curl` con el URI del agente.
3. Desde su equipo, cree un túnel `ssh` a la instancia EC2 utilizando la ruta a su archivo de clave privada y la dirección IP de la instancia EC2 pública. Por ejemplo:

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

Se inicia un servidor proxy de avance en su equipo.

4. Instale un cliente proxy, por ejemplo, [FoxyProxy](#) en su máquina.
5. Configure el cliente de proxy mediante los siguientes ajustes:
 - Para el tipo de proxy, especifique `SOCKS5`.
 - Para la dirección IP, nombre de DNS y nombre de servidor, especifique `localhost`.
 - Para el puerto, especifique `8080`.
 - Elimine todos los patrones de URL existentes.
 - Para el patrón de URL, especifique `*.mq.*.amazonaws.com*`
 - Para el tipo de conexión, especifique `HTTP(S)`.

Cuando habilita su cliente de proxy, puede obtener acceso a la consola web en su equipo.

Important

Si utiliza un agente privado, es posible que vea direcciones IP que no configuró con su VPC. Estas direcciones IP pertenecen a la infraestructura de RabbitMQ en Amazon MQ y no requieren ninguna acción.

Programación del período de mantenimiento de un agente de Amazon MQ

Amazon MQ realiza de forma periódica el mantenimiento del hardware, el sistema operativo o el software del motor de un agente de mensajes durante el período de mantenimiento. Por

ejemplo, si cambió el tipo de instancias del agente, Amazon MQ aplicará los cambios durante el próximo período de mantenimiento programado. La duración del mantenimiento puede durar hasta dos horas, dependiendo de las operaciones programadas para el agente de mensajes. Puede minimizar el tiempo de inactividad durante un período de mantenimiento seleccionando un modo de implementación del agente con alta disponibilidad en varias zonas de disponibilidad (AZ).

Amazon MQ para ActiveMQ proporciona implementaciones en modo [activo/en espera](#) para alta disponibilidad. En el modo activo/en espera, Amazon MQ realiza operaciones de mantenimiento a una instancia cada vez, de manera que al menos una instancia permanezca disponible. Además, puede configurar una [red de agentes](#) con períodos de mantenimiento repartidos a lo largo de la semana. Amazon MQ para RabbitMQ proporciona implementaciones de [clúster](#) para alta disponibilidad. En las implementaciones de clúster, Amazon MQ realiza operaciones de mantenimiento a un nodo cada vez, manteniendo al menos dos nodos en funcionamiento en todo momento.

Cuando se crea el agente por primera vez, puede programar que el período de mantenimiento se realice una vez por semana, a una hora determinada. Solo puede ajustar el período de mantenimiento de un agente hasta cuatro veces antes del próximo período de mantenimiento programado. Una vez completado el período de mantenimiento del agente, Amazon MQ restablece el límite, lo que le permite ajustar de nuevo la programación antes de que se produzca el siguiente período de mantenimiento. La disponibilidad del agente no se ve afectada al ajustar su período de mantenimiento.

Para ajustar el período de mantenimiento del agente, puede utilizar la Consola de administración de AWS, la AWS CLI o la API de Amazon MQ.

Ajuste del período de mantenimiento del agente a través de la Consola de administración de AWS

Para ajustar el período de mantenimiento del agente mediante el comando Consola de administración de AWS

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En el panel de navegación izquierdo, elija Brokers (Agentes) y seleccione el agente que desea actualizar de la lista.
3. En la página de detalles del agente, elija Edit (Editar).
4. En Maintenance (Mantenimiento), haga lo siguiente.

- a. En Start day (Día de inicio), elija un día de la semana, por ejemplo, Sunday (Domingo) en la lista desplegable.
- b. En Start time (Hora de inicio), elija la hora y minuto del día que desea programar para el próximo período de mantenimiento del agente, por ejemplo, 12:00.

 Note

Las opciones de hora de inicio están configuradas en la zona horaria UTC+0.

5. A continuación, seleccione Programar modificaciones. A continuación, seleccione Después del siguiente reinicio o Inmediatamente. Si selecciona After the next reboot (Tras el siguiente reinicio), se actualizará inmediatamente el período de mantenimiento sin reiniciar el agente. Si selecciona Inmediatamente, el agente se reiniciará de inmediato.
6. En la página de detalles del agente, en Maintenance window (período de mantenimiento), compruebe que aparezca la nueva programación preferida.

Ajuste del período de mantenimiento del agente a través de la AWS CLI

Para ajustar el período de mantenimiento del agente a través de la AWS CLI

1. Use el comando [update-broker](#) de la CLI y especifique los siguientes parámetros, como se muestra en el ejemplo.
 - `--broker-id`: ID único que genera Amazon MQ para el agente. Puede consultar el ID a partir del ARN de su agente. Por ejemplo, si se le da el ARN `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, el ID del agente sería `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
 - `--maintenance-window-start-time`: parámetros que determinan la hora de inicio del período de mantenimiento semanal proporcionada en la siguiente estructura.
 - `DayOfWeek` – el día de la semana, en la siguiente sintaxis: MONDAY | TUESDAY | WEDNESDAY | THURSDAY | FRIDAY | SATURDAY | SUNDAY
 - `TimeOfDay`: la hora, en formato de 24 horas.
 - `TimeZone`: (Opcional) la zona horaria, en el formato país/ciudad o de diferencia respecto a UTC. De forma predeterminada, se configura en UTC.

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

2. (Opcional) Utilice el comando [describe-broker](#) de la CLI para comprobar que el período de mantenimiento se hay actualizado correctamente.

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

Programación del período de mantenimiento del agente a través de la API de Amazon MQ

Para ajustar el período de mantenimiento del agente a través de la API de Amazon MQ

1. Utilice la operación [UpdateBroker](#) de la API. Especifique `broker-id` como parámetro de ruta. En los siguientes ejemplos, se asume que el agente está en la región `us-west-2`. Para obtener más información acerca de los puntos de conexión de Amazon MQ disponibles, consulte [cuotas y puntos de conexión de Amazon MQ](#) en Referencia general de AWS

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

Utilice el parámetro `maintenanceWindowStartTime` y tipo de recurso [WeeklyStartTime](#) en la carga de la solicitud.

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

2. (Opcional) Utilice la operación [DescribeBroker](#) de la API para comprobar que el período de mantenimiento se haya actualizado correctamente. Se especifica `broker-id` como parámetro de ruta.

```
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
```

Reiniciar un agente de Amazon MQ

Para aplicar una configuración nueva a un agente, puede reiniciarlo.

Note

Si su agente de ActiveMQ no responde, puede reiniciarlo para que pueda recuperarse de un estado de error.

En el siguiente ejemplo, se muestra cómo reiniciar un agente de Amazon MQ mediante la Consola de administración de AWS.

Para reiniciar un agente de Amazon MQ.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, elija el nombre de su corredor (por ejemplo, MyBroker).
3. En la **MyBroker** página, elija Acciones y reinicie el agente.

Important

Los agentes de instancia única estarán desconectados mientras se reinician. Los agentes del clúster estarán disponibles, pero cada nodo se reiniciará de uno en uno.

4. En el cuadro de diálogo Reboot broker, seleccione Reboot.

El proceso para reiniciar un agente tarda unos 5 minutos. Si el reinicio incluye cambios en el tamaño de la instancia o se realiza en un agente con una profundidad de cola elevada, el proceso de reinicio puede durar más tiempo.

Eliminación de un agente de Amazon MQ

Si no utilizas un agente de Amazon MQ (y no tienes previsto utilizarlo en un futuro próximo), se recomienda eliminarlo de Amazon MQ para reducir tus costes. AWS

El siguiente ejemplo muestra cómo eliminar un agente a través de la Consola de administración de AWS.

Eliminación de un agente de Amazon MQ

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, seleccione su agente (por ejemplo MyBroker) y, a continuación, elija Eliminar.
3. ¿En la opción Eliminar **MyBroker**? cuadro de diálogo, escriba `delete` y, a continuación, elija Eliminar.

El proceso de eliminación de un agente tarda unos 5 minutos.

Estados del agente de Amazon MQ

La condición actual de un agente se indica mediante un estado. En la siguiente tabla, se muestran los estados de un agente de Amazon MQ.

Consola	API	Description (Descripción)
Error de creación	CREATION_FAILED	No se pudo crear el agente.
Creación en curso	CREATION_IN_PROGRESS	El agente se está creando.
Eliminación en curso	DELETION_IN_PROGRESS	El agente se está eliminando.
Reinicio en curso	REBOOT_IN_PROGRESS	El agente se está reiniciando.

Consola	API	Description (Descripción)
Ejecutar	RUNNING	El agente es operativo.
Se requiere una acción crítica	CRITICAL_ACTION_REQUIRED	El agente se está ejecutando, pero se encuentra en un estado reducido y requiere una acción inmediata. Puede encontrar instrucciones para resolver el problema si elige el código de acción requerido de la lista en Resolución de problemas .

Adición de etiquetas a los recursos de Amazon MQ

Para organizar e identificar los recursos de Amazon MQ para asignación de costos, puede agregar etiquetas de metadatos que identifiquen el propósito de un agente o configuración. Esto es útil especialmente cuando dispone de muchos agentes. Puede utilizar las etiquetas de asignación de costos para organizar la factura de AWS de modo que refleje su propia estructura de costos. Para ello, regístrese para obtener una factura de su cuenta de AWS que incluya los valores y claves de etiquetas. Para obtener más información, consulte [Configuración de un informe de asignación de costos mensual](#) en la Guía del usuario de AWS Billing.

Por ejemplo, podría agregar etiquetas que representen el centro de costos y el objetivo de sus recursos de Amazon MQ:

Recurso	Clave	Valor
Broker1	Cost Center	34567
	Stack	Production
Broker2	Cost Center	34567
	Stack	Production

Recurso	Clave	Valor
Broker3	Cost Center	12345
	Stack	Development

Este plan de etiquetado le permite agrupar dos agentes que realizan tareas relacionadas en el mismo centro de costos, mientras etiqueta un agente no relacionado con una etiqueta de asignación de costos distinta.

Adición de etiquetas en la consola de Amazon MQ

Puede agregar etiquetas rápidamente a los recursos que está creando en la consola de Amazon MQ siguiendo estos pasos:

1. En la página Create a broker (Crear un agente), seleccione Additional settings (Configuración adicional).
2. En Tags (Etiquetas), seleccione Add tag (Añadir etiqueta).
3. Introduzca un par Key (Clave) y Value (Valor).
4. (Opcional) Seleccione Add tag (Añadir etiqueta) para añadir varias etiquetas a su agente.
5. Seleccione Create broker (Crear agente).

Para añadir etiquetas en el momento de crear una configuración:

1. En la página Create configuration (Crear configuración), seleccione Advanced (Avanzado).
2. En Tags (Etiquetas) en la página Create configuration (Crear configuración), seleccione Add tag (Añadir etiqueta).
3. Introduzca un par Key (Clave) y Value (Valor).
4. (Opcional) Seleccione Add tag (Añadir etiqueta) para añadir varias etiquetas a su configuración.
5. Seleccione Create configuration (Crear configuración).

Tras añadir etiquetas, puede verlas, editarlas y eliminarlas de sus recursos en la consola de Amazon MQ. También puede ver las etiquetas de sus recursos mediante la API de REST. Para obtener más información, consulte [Referencia de la API REST de Amazon MQ](#).

Uso de Amazon MQ para ActiveMQ

Amazon MQ facilita la creación de un agente de mensajes con los recursos informáticos y de almacenamiento que se adapten a sus necesidades. Puede crear, administrar y eliminar agentes mediante la Consola de administración de AWS, la API REST de Amazon MQ o la AWS Command Line Interface.

Los agentes de Amazon MQ para ActiveMQ pueden implementarse como agentes de una sola instancia o agentes activos/en espera. Para ambos modos de implementación, Amazon MQ proporciona un alto nivel de durabilidad, ya que almacenan sus datos de forma redundante.

Note

Amazon MQ utiliza [Apache KahaDB](#) como su almacén de datos. No se admiten otros almacenes de datos, como JDBC y LevelDB.

Para obtener acceso a sus agentes, puede usar [cualquier lenguaje de programación que admita ActiveMQ](#) y habilitar TLS explícitamente para los siguientes protocolos:

- [AMQP](#)
- [MQTT](#)
- MQTT a través de [WebSocket](#)
- [OpenWire](#)
- [STOMP](#)
- STOMP a través de WebSocket

Para obtener más información acerca de las API REST de Amazon MQ, consulte la [Referencia de la API REST de Amazon MQ](#).

Amazon MQ para agentes de ActiveMQ

¿Qué es un agente de Amazon MQ para ActiveMQ?

Un agente es un entorno de agente de mensajes que se ejecuta en Amazon MQ. Es el componente básico de Amazon MQ. La descripción combinada de la instancia del agente clase (m5) y tamaño

(`large`, `medium`) se conoce como tipo de instancias del agente (por ejemplo, `mq.m5.large`). Para obtener más información, consulte [Broker instance types](#).

- Un agente de una sola instancia está compuesto por un agente en una zona de disponibilidad. El agente se comunica con su aplicación y con un volumen de almacenamiento de Amazon EBS o Amazon EFS.
- Un agente activo/en espera está compuesto por dos agentes en dos zonas de disponibilidad diferentes, configurados en un par redundante. Estos agentes se comunican de forma sincrónica con su aplicación y con Amazon EFS.

Para obtener más información, consulte [Opciones de implementación de Amazon MQ para agentes de ActiveMQ](#).

Puede habilitar las actualizaciones automáticas de versiones secundarias para obtener las nuevas versiones secundarias del motor del agente a medida que Apache las publique. Las actualizaciones automáticas se producen durante el período de mantenimiento definido por el día de la semana, la hora del día (en formato de 24 horas) y la zona horaria (UTC de forma predeterminada).

Para obtener más información acerca de cómo se crean y se administran los agentes, consulte los siguientes temas:

- [Introducción: Creación y conexión con un agente de ActiveMQ](#)
- [Agentes](#)
- [Broker statuses](#)

Protocolos de nivel de cable admitidos

Para obtener acceso a sus agentes, puede usar [cualquier lenguaje de programación que admita ActiveMQ](#) y habilitar TLS explícitamente para los siguientes protocolos:

- [AMQP](#)
- [MQTT](#)
- MQTT a través de [WebSocket](#)
- [OpenWire](#)
- [STOMP](#)
- STOMP a través de WebSocket

Atributos

Un agente de ActiveMQ tiene varios atributos, por ejemplo:

- Un nombre (MyBroker)
- Un ID (b-1234a5b6-78cd-901e-2fgh-3i45j6k17819)
- Un nombre de recurso de Amazon (ARN) (arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819)
- Una URL de ActiveMQ Web Console (<https://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:8162>)

Para obtener más información, consulte el tema sobre la [consola web](#) en la documentación de Apache ActiveMQ.

Important

Si especifica un mapa de autorizaciones que no incluya al grupo `activemq-webconsole`, no puede utilizar la consola web de ActiveMQ porque el grupo no está autorizado a enviar mensajes al agente de Amazon MQ ni para recibir mensajes de este.

- Puntos de enlace de protocolo de nivel de conexión:
 - `amqp+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:5671`
 - `mqtt+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:8883`
 - `ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:61617`

Note

Es un punto de enlace de OpenWire.

- `stomp+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:61614`
- `wss://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:61619`

Para obtener más información, consulte el tema sobre la [configuración de transportes](#) en la documentación de Apache ActiveMQ.

Note

En el caso de un agente activo/en espera, Amazon MQ proporciona dos URL de la consola web de ActiveMQ, pero solo hay una URL activa a la vez. Del mismo modo, Amazon MQ proporciona dos puntos de enlace para cada protocolo de nivel de cable, pero solo hay un punto de enlace activo en cada par a la vez. Los sufijos -1 y -2 denotan un par redundante.

Para ver una lista completa de los atributos del agente, consulte lo siguiente en la Referencia de la API REST de Amazon MQ:

- [ID de operación de REST: Broker](#)
- [ID de operación de REST: Brokers](#)
- [ID de operación de REST: Reboot](#)

Usuarios de agentes

Un usuario de ActiveMQ es una persona o una aplicación que pueden acceder a las colas y los temas de un agente de ActiveMQ. Puede configurar a los usuarios para que tengan permisos específicos. Por ejemplo, puede permitir que algunos usuarios accedan a la [consola web de ActiveMQ](#).

Un grupo es una etiqueta semántica. Puede asignar un grupo a un usuario y configurar permisos para que los grupos envíen, reciban y administren colas y temas específicos.

Important

Si realiza cambios en un usuario, los cambios no se aplica al usuario de inmediato. Para aplicar los cambios, debe espere al próximo período de mantenimiento o [reiniciar el agente](#).

Para obtener información sobre los usuarios y los grupos, consulte lo siguiente en la documentación de Apache ActiveMQ:

- [Autorización](#)
- [Ejemplo de autorización](#)

Para obtener más información acerca de cómo crear, editar y eliminar usuarios de ActiveMQ, consulte los siguientes temas:

- [Creación de un usuario del agente de ActiveMQ](#)
- [Usuarios](#)

Atributos de usuario

Para ver una lista completa de los atributos de usuario, consulte lo siguiente en la Referencia de la API REST de Amazon MQ:

- [ID de operación de REST: User](#)
- [ID de operación de REST: Users](#)

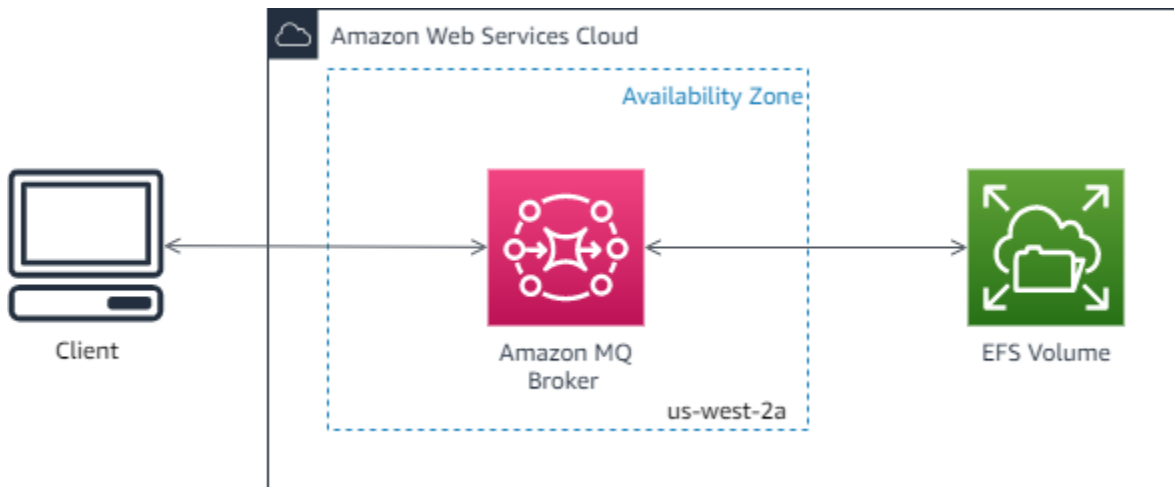
Opciones de implementación de Amazon MQ para agentes de ActiveMQ

Amazon MQ ofrece opciones de implementación de clústeres e instancias únicas para los agentes.

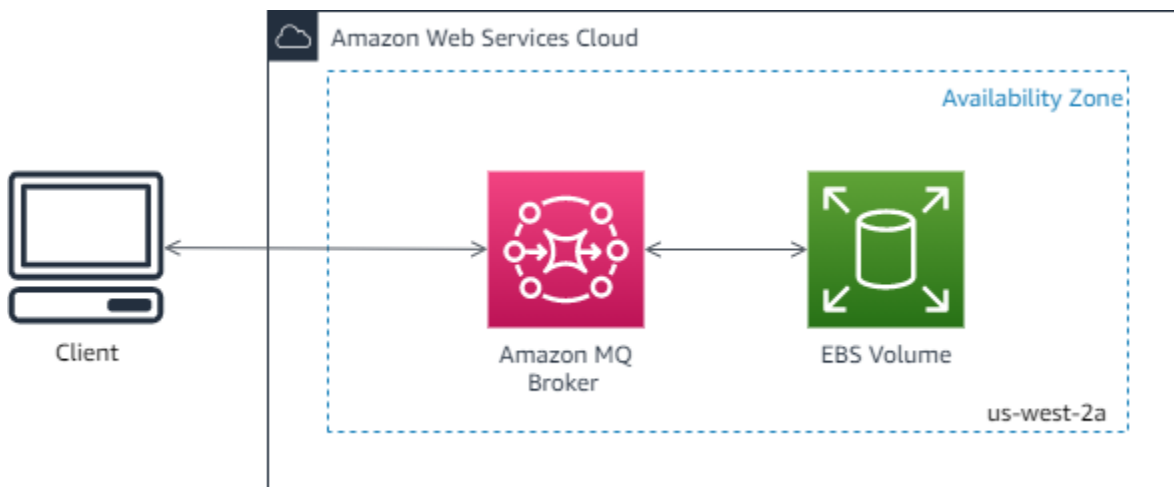
Opción 1: agentes de una sola instancia de Amazon MQ

Un agente de una sola instancia está compuesto por un agente en una zona de disponibilidad. El agente se comunica con su aplicación y con un volumen de almacenamiento de Amazon EBS o Amazon EFS. Los volúmenes de almacenamiento de Amazon EFS están diseñados para ofrecer el máximo nivel de durabilidad y disponibilidad mediante el almacenamiento de datos de forma redundante en varias zonas de disponibilidad (AZs). Amazon EBS proporciona almacenamiento a nivel de bloque optimizado para baja latencia y alto rendimiento. Para obtener más información acerca de las opciones de almacenamiento, consulte [Storage](#).

El siguiente diagrama ilustra un agente de instancia única con almacenamiento de Amazon EFS replicado en varias AZs



El siguiente diagrama ilustra un agente de una sola instancia con almacenamiento de Amazon EBS replicado en varios servidores dentro de una única zona de disponibilidad.



Opción 2: active/standby agentes Amazon MQ para alta disponibilidad

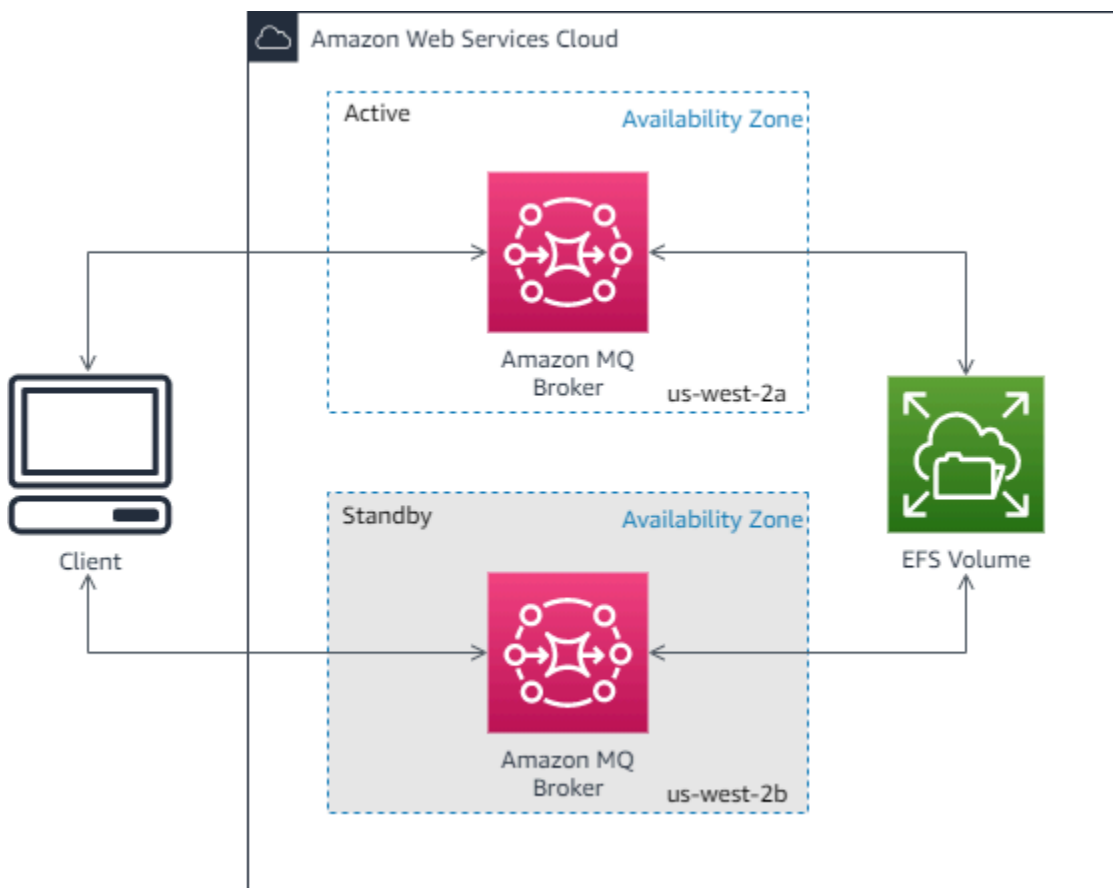
Un agente activo/en espera está compuesto por dos agentes en dos zonas de disponibilidad diferentes, configurados en un par redundante. Estos agentes se comunican de forma sincrónica con su aplicación y con Amazon EFS. Los volúmenes de almacenamiento de Amazon EFS están diseñados para ofrecer el máximo nivel de durabilidad y disponibilidad mediante el almacenamiento de datos de forma redundante en varias zonas de disponibilidad (AZs). Para obtener más información, consulte [Storage](#).

Normalmente, solo una de las instancias del agente está activa en todo momento, mientras que la otra se mantiene en espera. Si una de las instancias del agente no funciona correctamente o se encuentra en mantenimiento, Amazon MQ tarda un breve instante en poner la instancia inactiva fuera de servicio. Esto permite que la instancia en espera que está en buen estado se active y comience

a aceptar comunicaciones entrantes. Los periodos de mantenimiento y los reinicios del agente que inicie provocarán una conmutación por error. Cuando se reinicia un agente, la conmutación por error solamente tarda unos segundos.

Para un active/standby bróker, Amazon MQ proporciona dos ActiveMQ Web Console URLs, pero solo hay una URL activa a la vez. Del mismo modo, Amazon MQ proporciona dos puntos de enlace para cada protocolo de nivel de cable, pero solo hay un punto de enlace activo en cada par a la vez. Los sufijos -1 y -2 denotan un par redundante. Para los puntos de conexión de protocolo de nivel de cable, debería permitir que la aplicación se conecte a cualquiera de los puntos de conexión mediante el [transporte de conmutación por error](#).

En el siguiente diagrama, se muestra un active/standby intermediario con almacenamiento de Amazon EFS replicado en varios AZs.



Red de agentes de Amazon MQ

Amazon MQ es compatible con la característica de red de agentes de ActiveMQ.

Una red de corredores se compone de varios corredores o active/standby corredores de una sola instancia activos simultáneamente. La creación de una red de agentes puede aumentar la disponibilidad, la tolerancia a errores y el equilibrio de carga con varias instancias de agentes.

¿Cómo funciona una red de agentes?

Una red de los agentes se establece conectando un agente a otro mediante conectores de red. Un conector de red proporciona mensajes bajo demanda de un agente a otro. Los conectores de red se configuran en la configuración del agente como conexiones no dúplex o dúplex. Para conexiones no dúplex, los mensajes se reenvían solo desde un agente al otro. En el caso de las conexiones dúplex, los mensajes se reenvían en ambos sentidos entre ambos agentes.

Si el conector de red está configurado como dúplex, también se reenvían mensajes desde Agente2 a Agente1.

Puede utilizar conexiones dúplex y no dúplex en una red de agentes. Es posible que desee introducir una conexión dúplex con otro agente para mejorar el tráfico o evitar un aumento del límite. Las conexiones dúplex también son útiles para realizar migraciones parciales desde entornos locales a agentes gestionados de Amazon MQ.

¿Cómo administra las credenciales una red de agentes?

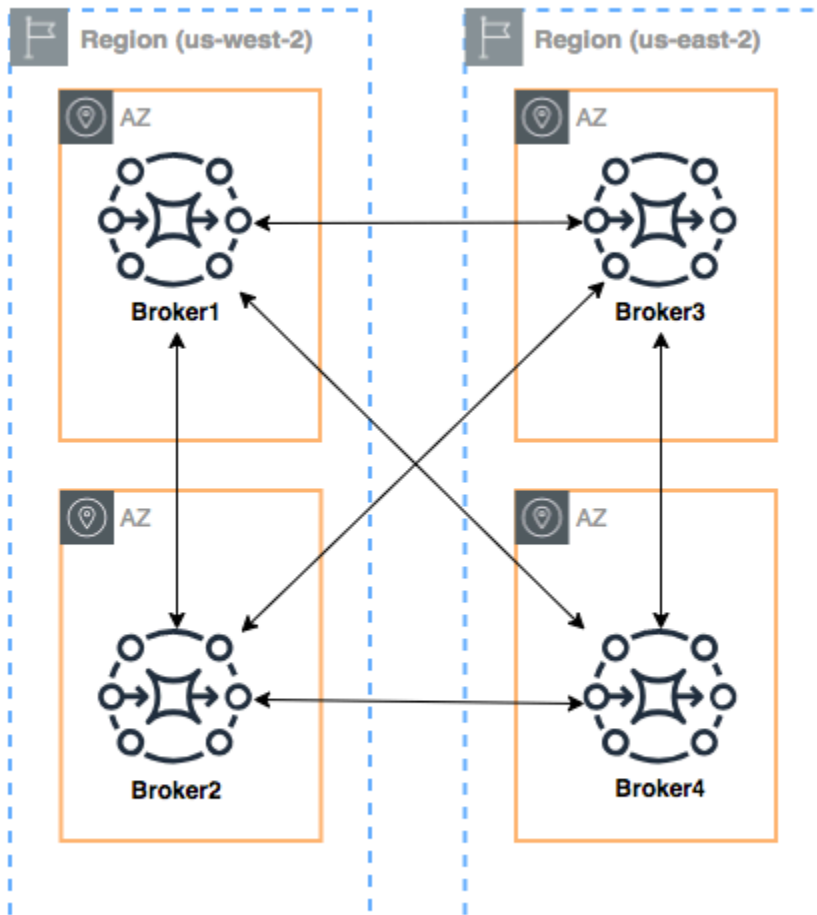
Para que un agente A se conecta a un agente B en una red, el agente A debe utilizar credenciales válidas, como cualquier otro productor o consumidor. En lugar de proporcionar una contraseña en la configuración de `<networkConnector>` del agente A, debe crear primero un usuario en el agente A con los mismos valores que otro usuario del agente B (se trata de usuarios independientes y únicos que comparten los mismos valores de nombre de usuario y contraseña). Cuando se especifica el atributo `userName` en la configuración `<networkConnector>`, Amazon MQ agregará la contraseña de forma automática en tiempo de ejecución.

Important

No especifique el atributo `password` para `<networkConnector>`. No recomendamos almacenar contraseñas de texto no cifrado en los archivos de configuración del agente, dado que esto hace que las contraseñas estén visibles en la consola de Amazon MQ. Para obtener más información, consulte [Configure Network Connectors for Your Broker](#).

Entre regiones

Para configurar una red de corredores que abarque AWS regiones, despliegue corredores en esas regiones y configure los conectores de red para los puntos finales de esos corredores.



Por configurar una red de agentes como en este ejemplo, podría añadir entradas `networkConnectors` a las configuraciones de `Agente1` y `Agente4` que hacen referencia a los puntos de enlace de nivel de cable de esos agentes.

Conectores de red para el `Agente1`:

```
<networkConnectors>
  <networkConnector name="1_to_2" userName="myCommonUser" duplex="true"
    uri="static:(ssl://b-987615k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)"/>
  <networkConnector name="1_to_3" userName="myCommonUser" duplex="true"
```

```

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

```

Conector de red para el Agente2:

```

<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>

```

Conectores de red para el Agente4:

```

<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-west-2.amazonaws.com:61617)"/>
</networkConnectors>

```

Conmutación por error dinámica con conectores de transporte

Además de configurar los elementos de `networkConnector`, puede configurar las opciones de `transportConnector` del agente para permitir la conmutación por error dinámica y reequilibrar las conexiones cuando se añaden o quitan agentes en la red.

```

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

```

En este ejemplo, tanto `updateClusterClients` como `rebalanceClusterClients` se han establecido en `true`. En este caso, se proporcionará a los clientes una lista de los agentes de la red y se les pedirá que se reequilibren si se une un nuevo agente.

Opciones disponibles:

- `updateClusterClients`: pasa información a los clientes sobre los cambios efectuados en la red de topología del agente.
- `rebalanceClusterClients`: hace que los clientes se reequilibren entre los agentes si se añade un nuevo agente a una red de agentes.
- `updateClusterClientsOnRemove`: actualiza los clientes con información sobre la topología cuando un agente abandona una red de agentes.

Si `updateClusterClients` está establecido en `true`, se pueden configurar los clientes para que se conecten a un único agente de la red de agentes.

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

Cuando un nuevo corredor se conecte, recibirá una lista de todos los corredores URIs de la red. Si se produce un error al conectar el agente, se puede cambiar dinámicamente a uno de los agentes que se proporcionaron al conectarse.

Para obtener más información acerca de la conmutación por error, consulte [Broker-side Options for Failover](#) en la documentación de Active MQ.

Amazon MQ para tipos de instancia de agentes de ActiveMQ

La descripción combinada de la instancia del agente clase (m5) y tamaño (large, medium) se conoce como tipo de instancias del agente (por ejemplo, `mq.m5.large`). En la siguiente tabla se indican los tipos de instancia del agente de Amazon MQ disponibles para agentes de ActiveMQ.

Amazon MQ avisa con al menos 90 días de antelación antes de que finalice el soporte de un tipo de instancia. Le recomendamos actualizar su agente a un nuevo tipo de instancia antes de la fecha de fin del soporte para evitar posibles interrupciones.

Important

No puede crear agentes a partir en `t2.micro` o `mq.m4.large` a partir del 17 de marzo de 2025.

Tipo de instancia	vCPU	Memoria (GiB)	Uso recomendado	Almacenamiento	Fin del soporte en Amazon MQ
mq.t3.micro	2	1	Evaluación	EFS	
mq.m5.large	2	8	Producción	EFS o EBS	
mq.m5.xlarge	4	16	Producción	EFS o EBS	
mq.m5.2xlarge	8	32	Producción	EFS o EBS	
mq.m5.4xlarge	16	64	Producción	EFS o EBS	

Para obtener más información acerca de consideraciones de desempeño, consulte [Elegir el tipo de instancia de agente correcto para obtener el mejor desempeño](#).

Configuraciones del agente de Amazon MQ para ActiveMQ

Una configuración contiene todos los ajustes del agente de ActiveMQ en formato XML (similar al archivo `activemq.xml` de ActiveMQ). Puede crear una configuración antes de crear los agente. Luego, puede aplicar la configuración a uno o más agentes.

Important

Si realiza cambios en una configuración, los cambios no se aplican al agente de inmediato. Para aplicar los cambios, debe esperar al próximo periodo de mantenimiento o [reiniciar el agente](#).

Solo puede eliminar una configuración mediante la API `DeleteConfiguration`. Para obtener más información, consulte [Configuraciones](#) en la Referencia de la API de Amazon MQ.

Atributos

La configuración de un agente tiene varios atributos, por ejemplo:

- Un nombre (MyConfiguration)
- Un ID (c-1234a5b6-78cd-901e-2fgh-3i45j6k17819)
- Un Nombre de recurso de Amazon (ARN) (arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k17819)

Para ver una lista completa de los atributos de configuración, consulte lo siguiente en la Referencia de la API REST de Amazon MQ:

- [ID de operación de REST: Configuration](#)
- [ID de operación de REST: Configurations](#)

Para ver una lista completa de los atributos de revisión de la configuración, consulte lo siguiente:

- [ID de operación de REST: Configuration Revision](#)
- [ID de operación de REST: Configuration Revisions](#)

Uso de archivos de configuración XML Spring

Para configurar los agentes de ActiveMQ se usan archivos [XML Spring](#). Pueden configurarse diversos aspectos del agente de ActiveMQ, como los destinos predefinidos, las políticas de destino, las políticas de autorización y los complementos. Amazon MQ controla alguno de estos elementos de configuración, como el almacenamiento y el transporte de red. Otras opciones de configuración, como la creación de redes de agentes, no se admiten actualmente.

El conjunto completo de las opciones de configuración admitidas se especifica en los esquemas XML de Amazon MQ. Para descargar los archivos zip de los esquemas admitidos, utilice los siguientes enlaces.

- [amazon-mq-active-mq-5.19.1.xsd.zip](#)
- [amazon-mq-active-mq-5.18.4.xsd.zip](#)
- [amazon-mq-active-mq-5.17.6.xsd.zip](#)
- [amazon-mq-active-mq-5.16.7.xsd.zip](#)

- [amazon-mq-active-mq-5.15.16.xsd.zip](#)

Puede usar estos esquemas para validar y sanear los archivos de configuración. Con Amazon MQ también puede cargar archivos XML para proporcionar configuraciones. Cuando se carga un archivo XML, Amazon MQ lo limpia de forma automática y elimina los parámetros de configuración no válidos y prohibidos según el esquema.

Note

Para los atributos, solo se pueden usar valores estáticos. Amazon MQ elimina de la configuración los elementos y los atributos que contienen referencias a elementos, variables y expresiones de Spring.

Creación de una configuración del agente de Amazon MQ para ActiveMQ

Una configuración contiene todos los ajustes del agente de ActiveMQ en formato XML (similar al archivo `activemq.xml` de ActiveMQ). Puede crear una configuración antes de crear los agente. Luego, puede aplicar la configuración a uno o más agentes. Puede aplicar una configuración inmediatamente o durante un período de mantenimiento.

En el ejemplo siguiente, se muestra cómo crear y aplicar la configuración de un agente de Amazon MQ mediante la Consola de administración de AWS.

Important

Solo puede eliminar una configuración mediante la API `DeleteConfiguration`. Para obtener más información, consulte [Configuraciones](#) en la Referencia de la API de Amazon MQ.

Crear una nueva configuración

Para crear una nueva configuración de un agente, cree primero la nueva configuración.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En el lado izquierdo, expanda el panel de navegación y elija **Configurations** (Configuraciones).

Amazon MQ ×

Brokers

Configurations

3. En la página Configurations (Configuraciones), seleccione Create configuration (Crear configuración).
4. En la página Create configuration (Crear configuración), en la sección Details (Detalles), escriba el nombre de la configuración en Configuration name (Nombre de configuración) (por ejemplo, MyConfiguration) y seleccione una versión del motor del agente en Broker engine (Motor del agente).

Note

Para obtener más información acerca de las versiones de motor de ActiveMQ compatibles con Amazon MQ para ActiveMQ, consulte [the section called “Administración de versiones”](#).

5. Seleccione Crear configuración.

Crear una revisión de configuración nueva

Tras crear una configuración de un agente, tendrá que editarla mediante una revisión de la configuración.

1. En la lista de configuraciones, elija **MyConfiguration**.

Note

La primera revisión de la configuración siempre se genera automáticamente cuando Amazon MQ crea la configuración.

En la **MyConfiguration** página, se muestran el tipo y la versión del motor de broker que utiliza la nueva revisión de configuración (por ejemplo, Apache ActiveMQ 5.15.16).

2. En la pestaña Configuration details, se muestra el número de revisión de la configuración, la descripción y la configuración del agente en formato XML.

Note

Cuando se edita la configuración actual, se crea una nueva revisión de configuración.

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. Elija Edit configuration (Editar configuración) y realice cambios en la configuración XML.
4. Seleccione Save.

Se muestra el cuadro de diálogo Save revision (Guardar revisión).

5. (Opcional) Escriba A description of the changes in this revision.
6. Seleccione Save.

De este modo, guardará la nueva revisión de la configuración.

Important

La consola de Amazon MQ elimina automáticamente los parámetros de configuración no válidos y prohibidos de acuerdo con un esquema. Para obtener más información y una lista completa de los parámetros XML permitidos, consulte [Amazon MQ Broker Configuration Parameters](#).

Aplicar una revisión de configuración al agente

Tras revisar la configuración, puede aplicarla a su agente.

1. En el lado izquierdo, expanda el panel de navegación y elija Brokers (Agentes).

Amazon MQ

Brokers

Configurations

2. En la lista de corredores, seleccione su corredor (por ejemplo MyBroker) y, a continuación, elija Editar.
3. En la **MyBroker** página de edición, en la sección Configuración, seleccione una configuración y una revisión y, a continuación, elija Programar modificaciones.
4. En la sección Schedule broker modifications (Programar modificaciones del agente), elija During the next scheduled maintenance window (Durante el siguiente periodo de mantenimiento programado) si desea aplicar las modificaciones durante el siguiente periodo de mantenimiento programado o Immediately (Inmediatamente) si desea aplicarlas inmediatamente.

Important

Los agentes de instancia única están desconectados mientras se reinician. En el caso de los agentes de clústeres, solo un nodo está inactivo a la vez mientras el agente se reinicia.

5. Seleccione Aplicar.

La revisión de la configuración se aplicará al broker en el momento especificado.

Edición de una revisión de configuración de Amazon MQ para ActiveMQ

Puede que desee editar una revisión de configuración después de aplicarla a su agente. Utilice las siguientes instrucciones para editar una revisión de configuración.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, seleccione su corredor (por ejemplo MyBroker) y, a continuación, elija Editar.
3. En la **MyBroker** página, elija Editar.
4. En la **MyBroker** página de edición, en la sección Configuración, seleccione una configuración y una revisión y, a continuación, elija Editar.

Note

A menos que seleccione una configuración cuando crea un agente, la primera revisión de la configuración siempre se crea automáticamente cuando Amazon MQ crea el agente.

En la **MyBroker** página, se muestran el tipo y la versión del motor de broker que utiliza la configuración (por ejemplo, Apache ActiveMQ 5.15.8).

5. En la pestaña Configuration details, se muestra el número de revisión de la configuración, la descripción y la configuración del agente en formato XML.

Note

Cuando se edita la configuración actual, se crea una nueva revisión de configuración.

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

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

```

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

```

6. Elija Edit configuration (Editar configuración) y realice cambios en la configuración XML.
7. Seleccione Save.

Se muestra el cuadro de diálogo Save revision (Guardar revisión).

8. (Opcional) Escriba A description of the changes in this revision.
9. Seleccione Save.

De este modo, guardará la nueva revisión de la configuración.

⚠ Important

La consola de Amazon MQ elimina automáticamente los parámetros de configuración no válidos y prohibidos de acuerdo con un esquema. Para obtener más información y una lista completa de los parámetros XML permitidos, consulte [Amazon MQ Broker Configuration Parameters](#).

Elementos permitidos en las configuraciones de Amazon MQ

A continuación, se muestra una lista detallada de los elementos permitidos en las configuraciones de Amazon MQ. Para obtener más información, consulte el tema sobre la [configuración de XML](#) en la documentación de Apache ActiveMQ.

Element
abortSlowAckConsumerStrategy (atributos)
abortSlowConsumerStrategy (atributos)
authorizationEntry (atributos)
authorizationMap (elementos de colección secundarios)
authorizationPlugin (elementos de colección secundarios)
broker (atributos) elementos de colección secundarios
cachedMessageGroupMapFactory (atributos)
compositeQueue (atributos) elementos de colección secundarios
compositeTopic (atributos) elementos de colección secundarios
constantPendingMessageLimitStrategy (atributos)
discarding (atributos)
discardingDLQBrokerPlugin (atributos)

Element

fileCursor

fileDurableSubscriberCursor

fileQueueCursor

filteredDestination [\(atributos\)](#)fixedCountSubscriptionRecoveryPolicy [\(atributos\)](#)fixedSizedSubscriptionRecoveryPolicy [\(atributos\)](#)forcePersistencyModeBrokerPlugin [\(atributos\)](#)individualDeadLetterStrategy [\(atributos\)](#)

lastImageSubscriptionRecoveryPolicy

messageGroupHashBucketFactory [\(atributos\)](#)mirroredQueue [\(atributos\)](#)

noSubscriptionRecoveryPolicy

oldestMessageEvictionStrategy [\(atributos\)](#)oldestMessageWithLowestPriorityEvictionStrategy [\(atributos\)](#)policyEntry [\(atributos\)](#) | [elementos de colección secundarios](#)policyMap [\(elementos de colección secundarios\)](#)prefetchRatePendingMessageLimitStrategy [\(atributos\)](#)

priorityDispatchPolicy

priorityNetworkDispatchPolicy

queryBasedSubscriptionRecoveryPolicy [\(atributos\)](#)

Element

queue ([atributos](#))

redeliveryPlugin ([atributos](#) | [elementos de colección secundarios](#))

redeliveryPolicy ([atributos](#))

redeliveryPolicyMap ([elementos de colección secundarios](#))

retainedMessageSubscriptionRecoveryPolicy ([elementos de colección secundarios](#))

roundRobinDispatchPolicy

sharedDeadLetterStrategy ([atributos](#) | [elementos de colección secundarios](#))

simpleDispatchPolicy

simpleMessageGroupMapFactory

statisticsBrokerPlugin

storeCursor

storeDurableSubscriberCursor ([atributos](#))

strictOrderDispatchPolicy

tempDestinationAuthorizationEntry ([atributos](#))

tempQueue ([atributos](#))

tempTopic ([atributos](#))

timedSubscriptionRecoveryPolicy ([atributos](#))

timeStampingBrokerPlugin ([atributos](#))

topic ([atributos](#))

transportConnector ([atributos](#))

Element
uniquePropertyMessageEvictionStrategy (atributos)
virtualDestinationInterceptor (elementos de colección secundarios)
virtualTopic (atributos)
vmCursor
vmDurableCursor
vmQueueCursor

Elementos y sus atributos permitidos en las configuraciones de Amazon MQ

A continuación, se muestra una lista detallada de los elementos y sus atributos permitidos en las configuraciones de Amazon MQ. Para obtener más información, consulte el tema sobre la [configuración de XML](#) en la documentación de Apache ActiveMQ.

Element	Atributo
abortSlowAckConsumerStrategy	abortConnection
	checkPeriod
	ignoreIdleConsumers
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	maxTimeSinceLastAck
	name

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

Element	Atributo
	<code>enableMessageExpirationOnActiveDurableSubs</code>
	<code>maxPurgedDestinationsPerSweep</code>
	<code>maxSchedulerRepeatAllowed</code>
	<code>monitorConnectionSplits</code>
	<u><code>networkConnectorStartAsync</code></u>
	<code>offlineDurableSubscriberTaskSchedule</code>
	<code>offlineDurableSubscriberTimeout</code>
	<code>persistenceThreadPriority</code>
	<code>persistent</code>
	<code>populateJMSXUserID</code>
	<code>producerSystemUsagePortion</code>
	<code>rejectDurableConsumers</code>
	<code>rollbackOnlyOnAsyncException</code>
	<code>schedulePeriodForDestinationPurge</code>
	<code>schedulerSupport</code>
	<code>splitSystemUsageForProducersConsumers</code>
	<code>taskRunnerPriority</code>
	<code>timeBeforePurgeTempDestinations</code>

Element	Atributo
	useAuthenticatedPrincipalForJMSXUserID
	useMirroredQueues
	useTempMirroredQueues
	useVirtualDestSubs
	useVirtualDestSubsOnCreation
	useVirtualTopics
cachedMessageGroupMapFactory	cacheSize
compositeQueue	concurrentSend
	copyMessage
	forwardOnly
	name
	sendWhenNotMatched
compositeTopic	concurrentSend
	copyMessage
	forwardOnly
	name
	sendWhenNotMatched
conditionalNetworkBridgeFilterFactory	rateDuration
	rateLimit
	replayDelay

Element	Atributo
	replayWhenNoConsumers
	selectorAware
	 Compatible con Apache ActiveMQ 5.16.x
constantPendingMessageLimit Strategy	limit
discarding	deadLetterQueue
	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
discardingDLQBrokerPlugin	processNonPersistent
	dropAll
	dropOnly
	dropTemporaryQueues
	dropTemporaryTopics
filteredDestination	reportInterval
	queue
	selector

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

Element	Atributo
mirroredQueue	copyMessage
	postfix
	prefix
oldestMessageEvictionStrategy	evictExpiredMessagesHighWatermark
oldestMessageWithLowestPriorityEvictionStrategy	evictExpiredMessagesHighWatermark
policyEntry	advisoryForConsumed
	advisoryForDelivery
	advisoryForDiscardingMessages
	advisoryForFastProducers
	advisoryForSlowConsumers
	advisoryWhenFull
	allConsumersExclusiveByDefault
	alwaysRetroactive
	blockedProducerWarningInterval
	consumersBeforeDispatchStarts
	cursorMemoryHighWaterMark
	doOptimizeMessageStorage
	durableTopicPrefetch
	enableAudit

Element	Atributo
	<code>expireMessagesPeriod</code>
	<code>gcInactiveDestinations</code>
	<code>gcWithNetworkConsumers</code>
	<code>inactiveTimeoutBeforeGC</code>
	<code>inactiveTimeoutBeforeGC</code>
	<code>includeBodyForAdvisory</code>
	<code>lazyDispatch</code>
	<code>maxAuditDepth</code>
	<code>maxBrowsePageSize</code>
	<code>maxDestinations</code>
	<code>maxExpirePageSize</code>
	<code>maxPageSize</code>
	<code>maxProducersToAudit</code>
	<code>maxQueueAuditDepth</code>
	<code>memoryLimit</code>
	<code>messageGroupMapFactoryType</code>
	<code>minimumMessageSize</code>
	<code>optimizedDispatch</code>
	<code>optimizeMessageStoreInFlightLimit</code>
	<code>persistJMSRedelivered</code>

Element	Atributo
	prioritizedMessages
	producerFlowControl
	queue
	queueBrowserPrefetch
	queuePrefetch
	reduceMemoryFootprint
	sendAdvisoryIfNoConsumers
	sendFailIfNoSpace
	sendFailIfNoSpaceAfterTimeout
	 Compatible con Apache ActiveMQ 5.16.4 y versiones posteriores
	sendDuplicateFromStoreToDLQ
	storeUsageHighWaterMark
	strictOrderDispatch
	tempQueue
	tempTopic
	timeBeforeDispatchStarts
	topic
	topicPrefetch

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

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

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

Element	Atributo
	prefix
	selectorAware
	setOriginalDestination
	transactedSend

Atributos de elemento principales de Amazon MQ

A continuación se ofrece una explicación detallada de los atributos de elemento principales. Para obtener más información, consulte el tema sobre la [configuración de XML](#) en la documentación de Apache ActiveMQ.

Temas

- [agente](#)

agente

`broker` es un elemento de colección principal.

Atributos

`networkConnectionStartAsincrónico`

Para mitigar la latencia de la red y permitir que otras redes se inicien de manera oportuna, utilice la etiqueta `<networkConnectionStartAsync>`. La etiqueta indica al agente que utilice un ejecutor para iniciar las conexiones de red en paralelo, de forma asincrónica a un inicio de agente.

Valor predeterminado: `false`

Configuración de ejemplo

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

Elementos, elementos de colección secundarios y sus elementos secundarios permitidos en las configuraciones de Amazon MQ

A continuación, se muestra una lista detallada de los elementos, los elementos de colección secundarios y sus elementos secundarios permitidos en las configuraciones de Amazon MQ. Para obtener más información, consulte el tema sobre la [configuración de XML](#) en la documentación de Apache ActiveMQ.

Element	Elemento de colección secundario	Elemento secundario
authorizationMap	authorizationEntries	authorizationEntry
		tempDestinationAuthorizationEntry
	defaultEntry	authorizationEntry
		tempDestinationAuthorizationEntry
authorizationPlugin	tempDestinationAuthorizationEntry	tempDestinationAuthorizationEntry
authorizationPlugin	map	authorizationMap
broker	destinationInterceptors	mirroredQueue
		virtualDestinationInterceptor
	destinationPolicy	policyMap
	destinations	queue
		tempQueue
		tempTopic
		topic

Element	Elemento de colección secundario	Elemento secundario
	networkConnectors	networkConnector
	persistenceAdapter	kahaDB
	plugins	authorizationPlugin
		discardingDLQBrokerPlugin
		forcePersistencyModeBrokerPlugin
		redeliveryPlugin
		statisticsBrokerPlugin
		timeStampingBrokerPlugin
	systemUsage	systemUsage
	transportConnector	name
		updateClusterClients
		rebalanceClusterClients
compositeQueue	forwardTo	updateClusterClientsOnRemove
		queue
		tempQueue
		tempTopic

Element	Elemento de colección secundario	Elemento secundario
compositeTopic	forwardTo	topic
		filteredDestination
		queue
		tempQueue
		tempTopic
policyEntry	deadLetterStrategy	topic
		filteredDestination
		discarding
		individualDeadLetterStrategy
		sharedDeadLetterStrategy
	destination	queue
		tempQueue
		tempTopic
		topic
	dispatchPolicy	priorityDispatchPolicy
		priorityNetworkDispatchPolicy
		roundRobinDispatchPolicy

Element	Elemento de colección secundario	Elemento secundario
		<code>simpleDispatchPolicy</code>
		<code>strictOrderDispatchPolicy</code>
		<code>clientIdFilterDispatchPolicy</code>
	<code>messageEvictionStrategy</code>	<code>oldestMessageEvictionStrategy</code>
		<code>oldestMessageWithLowestPriorityEvictionStrategy</code>
		<code>uniquePropertyMessageEvictionStrategy</code>
	<code>messageGroupMapFactory</code>	<code>cachedMessageGroupMapFactory</code>
		<code>messageGroupHashBucketFactory</code>
		<code>simpleMessageGroupMapFactory</code>
	<code>pendingDurableSubscriberPolicy</code>	<code>fileDurableSubscriberCursor</code>
		<code>storeDurableSubscriberCursor</code>
		<code>vmDurableCursor</code>
	<code>pendingMessageLimitStrategy</code>	<code>constantPendingMessageLimitStrategy</code>

Element	Elemento de colección secundario	Elemento secundario
	pendingQueuePolicy	prefetchRatePendingMessageLimitStrategy
		fileQueueCursor
		storeCursor
		vmQueueCursor
	pendingSubscriberPolicy	fileCursor
		vmCursor
	slowConsumerStrategy	abortSlowAckConsumerStrategy
		abortSlowConsumerStrategy
	subscriptionRecoveryPolicy	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy

Element	Elemento de colección secundario	Elemento secundario
		retainedMessageSubscriptionRecoveryPolicy
timedSubscriptionRecoveryPolicy		
policyMap	defaultEntry	policyEntry
	policyEntries	policyEntry
redeliveryPlugin	redeliveryPolicyMap	redeliveryPolicyMap
redeliveryPolicyMap	defaultEntry	redeliveryPolicy
	redeliveryPolicyEntries	redeliveryPolicy
retainedMessageSubscriptionRecoveryPolicy	wrapped	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy

Element	Elemento de colección secundario	Elemento secundario
		timedSubscriptionRecoveryPolicy
sharedDeadLetterStrategy	deadLetterQueue	queue
		tempQueue
		tempTopic
		topic
virtualDestinationInterceptor	virtualDestinations	compositeQueue
		compositeTopic
		virtualTopic

Atributos de elemento secundarios de Amazon MQ

A continuación se ofrece una explicación detallada de los atributos de elemento secundarios. Para obtener más información, consulte el tema sobre la [configuración de XML](#) en la documentación de Apache ActiveMQ.

Temas

- [authorizationEntry](#)
- [networkConnector](#)
- [kahaDB](#)
- [systemUsage](#)

authorizationEntry

`authorizationEntry` es un elemento secundario del elemento de colección secundario `authorizationEntries`.

Atributos

admin|read|write

Los permisos concedidos a un grupo de usuarios. Para obtener más información, consulte [Configurar siempre una asignación de autorizaciones](#).

Si especifica un mapa de autorizaciones que no incluya al grupo `activemq-webconsole`, no puede utilizar la consola web de ActiveMQ porque el grupo no está autorizado a enviar mensajes al agente de Amazon MQ ni para recibir mensajes de este.

Valor predeterminado: `null`

Configuración de ejemplo

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

Note

El grupo `activemq-webconsole` de ActiveMQ en Amazon MQ tiene permisos de administrador en todas las colas y temas. Todos los usuarios de este grupo tendrán acceso de administrador.

networkConnector

`networkConnector` es un elemento secundario del elemento de colección secundario `networkConnectors`.

Temas

- [Atributos](#)
- [Ejemplo de configuraciones](#)

Atributos

conduitSubscriptions

Especifica si una conexión de red en una red de agentes trata varios consumidores suscritos al mismo destino como un consumidor. Por ejemplo, si `conduitSubscriptions` se ha establecido en `true` y se conectan dos consumidores al agente B y consumen desde un destino, el agente B combina las suscripciones en una sola suscripción lógica a través de la conexión de red al agente A, de modo que solo se reenvía una única copia de un mensaje del agente A al agente B.

Note

Establecer `conduitSubscriptions` en `true` puede reducir el tráfico de red redundante. Sin embargo, el uso de este atributo puede tener implicaciones para el balanceo de carga de mensajes en los consumidores y podría causar un comportamiento incorrecto en ciertas situaciones (por ejemplo, con selectores de mensajes de JMS o con temas duraderos).

Valor predeterminado: `true`

duplex

Especifica si la conexión en la red de agentes se utiliza para producir y consumir mensajes. Por ejemplo, si un agente A crea una conexión al agente B en modo no dúplex, los mensajes pueden reenviarse solo desde el agente A al agente B. Sin embargo, si un agente A crea una conexión dúplex al agente B, entonces el agente B puede reenviar mensajes al agente A sin tener que configurar un `<networkConnector>`.

Valor predeterminado: `false`

name

El nombre del puente en la red de agentes.

Valor predeterminado: `bridge`

uri

El punto de enlace de protocolo de nivel de conexión para uno de los dos agentes (o para varios agentes) en una red de agentes.

Valor predeterminado: null

nombre de usuario

El nombre de usuario común a los agentes en una red de agentes.

Valor predeterminado: null

Ejemplo de configuraciones

 Note

Cuando se utiliza un `networkConnector` para definir una red de agentes, no incluya la contraseña para el usuario común a los agentes.

Una red de agentes con dos agentes

En esta configuración, dos agentes están conectados en una red de agentes. El nombre del conector de red es `connector_1_to_2`, el nombre de usuario común a los corredores es `myCommonUser`, la conexión es `duplex` y el URI del OpenWire punto final lleva el prefijo `static:`, lo que indica una one-to-one conexión entre los corredores.

```
<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>
```

Para obtener más información, consulte [Configure Network Connectors for Your Broker](#).

Una red de agentes con varios agentes

En esta configuración, varios agentes están conectados en una red de agentes. El nombre del conector de red es `connector_1_to_2`, el nombre de usuario común a los

intermediariosmyCommonUser, la conexión es y la lista de OpenWire terminales separados por comas lleva el prefijo duplexmasterslave:, lo que indica que URIs hay una conexión de conmutación por error entre los corredores. La conmutación por error de agente a agente no es aleatoria y los intentos de reconexión continúan indefinidamente.

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

Note

Le recomendamos que utilice el prefijo masterslave: para las redes de agentes. El prefijo es idéntico a la sintaxis static:failover:()?randomize=false&maxReconnectAttempts=0 más explícita.

Note

Esta configuración de XML no permite espacios.

kahaDB

kahaDB es un elemento secundario del elemento de colección secundario persistenceAdapter.

Atributos

concurrentStoreAndDispatchQueues

Especifica si se debe usar almacenamiento y envío simultáneo para colas. Para obtener más información, consulte [Desactivar el almacenamiento y el envío simultáneos en colas con consumidores lentos](#).

Valor predeterminado: true

cleanupOnStop

 Compatible con
Apache ActiveMQ 15.16.x y versiones posteriores

Si se desactiva, la recolección de basura y la limpieza no se llevan a cabo cuando se suspende al agente, lo que acelera el proceso de apagado. El aumento de la velocidad es útil en casos de bases de datos grandes o de programadores.

Valor predeterminado: `true`

journalDiskSyncInterval

Intervalo (ms) para cuándo realizar una sincronización de disco si `journalDiskSyncStrategy=periodic`. Para obtener más información, consulte la [documentación de Apache ActiveMQ kahaDB](#).

Valor predeterminado: `1000`

journalDiskSyncEstrategia

 Compatible con
Apache ActiveMQ 15.14.x y versiones posteriores

Configura la política de sincronización de disco. Para obtener más información, consulte la [documentación de Apache ActiveMQ kahaDB](#).

Valor predeterminado: `always`

Note

La [documentación de ActiveMQ](#) indica que la pérdida de datos está limitada a la duración de `journalDiskSyncInterval`, que tiene un valor predeterminado de 1 segundo. La pérdida de datos puede ser más larga que el intervalo, pero es difícil ser preciso. Sea precavido.

preallocationStrategy

Configura la forma en que el agente intentará preasignar los archivos de diario cuando se necesite un nuevo archivo de diario. Para obtener más información, consulte la [documentación de Apache ActiveMQ kahaDB](#).

Valor predeterminado: `sparse_file`

Configuración de ejemplo

Example

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

systemUsage

`systemUsage` es un elemento secundario del elemento de colección secundario `systemUsage`. Controla la cantidad máxima de espacio que usará el agente antes de ralentizar a los productores. Para obtener más información, consulte [Producer Flow Control \(Control del flujo del productor\)](#) en la documentación de Apache ActiveMQ.

Elemento secundario

memoryUsage

`memoryUsage` es un elemento secundario del elemento secundario `systemUsage`. Gestiona el uso de la memoria. Use `memoryUsage` para realizar un seguimiento de cuánto se está utilizando un elemento para que pueda controlar el uso del conjunto de trabajo de manera productiva. Para obtener más información, consulte [el esquema](#) de la documentación de Apache ActiveMQ.

Elemento secundario

`memoryUsage` es un elemento secundario del elemento secundario `memoryUsage`.

Atributo

percentOfJvmMontón

Número entero entre 0 (inclusive) y 70 (inclusive).

Valor predeterminado: 70

Atributos

sendFailIfNoSpace

Establece si un método `send()` debe fallar si no hay espacio libre. El valor predeterminado es `false`, lo que bloquea el método `send()` hasta que el espacio esté disponible. Para obtener más información, consulte el [esquema](#) de la documentación de Apache Active MQ.

Valor predeterminado: `false`

sendFailIfNoSpaceAfterTimeout

Valor predeterminado: `null`

Configuración de ejemplo

Example

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

Replicación de datos entre regiones para Amazon MQ para ActiveMQ

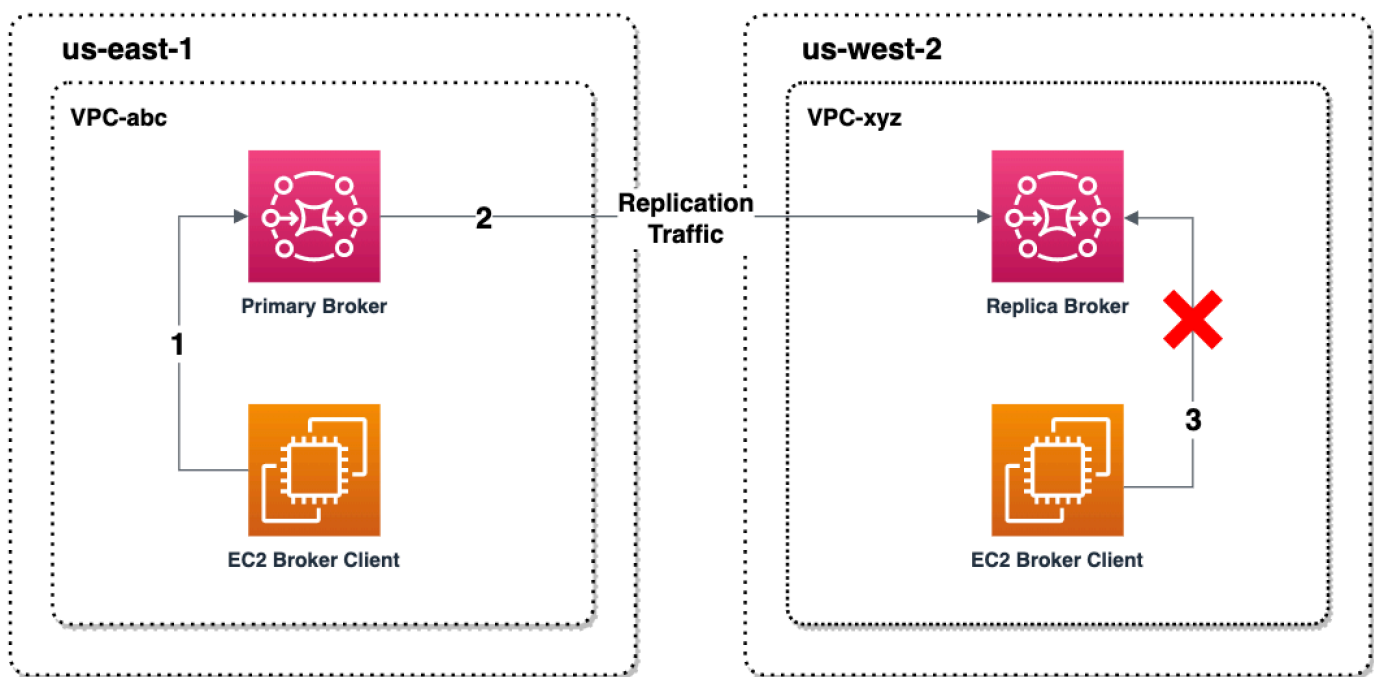
Amazon MQ para ActiveMQ ofrece una característica de replicación de datos entre regiones (CRDR) que permite la replicación asíncrona de mensajes desde el agente primario de una región

de AWS principal al agente de réplicas de una región de réplicas. Al emitir una solicitud de conmutación por error a la API de Amazon MQ, el agente de réplicas actual se asciende al rol de agente principal y el agente principal actual se degrada al rol de réplicas.

Agentes principales y de réplicas para la replicación de datos entre regiones

Puede crear agentes principales y de réplicas para la replicación asíncrona de datos desde el agente principal de una región de AWS primaria al agente de réplicas de una región de réplicas. La región principal consta de un par redundante de agentes activos o en espera denominado agente principal. La región secundaria consta de un par redundante de agentes activos o en espera denominado agente de réplicas.

En el siguiente diagrama se ilustra un agente de réplicas en una región secundaria que recibe datos replicados asíncronos del agente principal en la región principal.



Los agentes principales y de réplicas actúan como una solución de recuperación de datos entre regiones. Si el agente principal de la región principal falla, puede ascender a primario al agente de réplicas de la región secundaria iniciando una conmutación o conmutación por error. El antiguo agente principal se convierte entonces en el agente de réplicas y el antiguo agente de réplicas se asciende a agente principal. Para obtener instrucciones sobre la creación de un agente principal y de réplicas, consulte [Creación de un agente de replicación de datos entre regiones de Amazon MQ](#).

 Note

Solo disponible para agentes activos o en espera.
No disponible para colas duplicadas.

Creación de un agente de replicación de datos entre regiones de Amazon MQ

Con la replicación de datos entre regiones (CRDR), puede alternar entre agentes de mensajes de Amazon MQ para ActiveMQ en dos regiones de AWS según sea necesario. Puede designar un agente existente como agente principal y crear una réplica para este agente o crear un nuevo agente principal y de réplicas juntos. A continuación, puede ascender el agente de réplicas al rol de agente principal mediante la operación de la API Promote de Amazon MQ. Para obtener más información sobre los agentes principales y de réplicas, consulte [Agentes principales y de réplicas para la replicación de datos entre regiones](#).

Las siguientes instrucciones describen cómo puede crear y configurar un agente de réplicas mediante la consola de administración de Amazon MQ.

Temas

- [Requisitos previos](#)
- [Paso 1 \(opcional\): Crear un nuevo agente principal](#)
- [Paso 2: Crear una réplica de un agente existente](#)

Requisitos previos

Para utilizar la característica de replicación de datos entre regiones, debe revisar y cumplir los siguientes requisitos previos:

- Versión: la característica de replicación de datos entre regiones solo está disponible para agentes de Amazon MQ en la versión 5.17.6 y versiones más recientes.
- Región: la replicación de datos entre regiones se admite en las siguientes regiones: Este de EE. UU. (Ohio), Este de EE. UU. (Norte de Virginia), Oeste de EE. UU. (Oregón) y Oeste de EE. UU. (Norte de California).

- Tipo de instancia: la replicación de datos entre regiones solo está disponible para instancias de agente de tamaño `mq.m5.large` y superiores.
- Tipo de implementación: la replicación de datos entre regiones solo está disponible para agentes activos o en espera con implementación en zonas de disponibilidad múltiple.
- Estado de agente: solo puede crear un agente de réplicas para un agente principal con el estado de agente `Running`.

Paso 1 (opcional): Crear un nuevo agente principal

Crear un nuevo agente principal

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la página de agentes de la consola de Amazon MQ, elija Crear agentes.
3. En la página Select broker engine (Seleccionar motor de agente), elija Apache ActiveMQ.
4. En la página Select deployment and storage (Seleccionar implementación y almacenamiento), sección Deployment mode and storage type (Modo de implementación y tipo de almacenamiento), haga lo siguiente:
 - En Modo de implementación, elija Agente activo/en espera. Un Agente activo/en espera se compone de dos agentes en dos zonas de disponibilidad diferentes configurados en un par redundante. Estos agentes se comunican de forma sincrónica con su aplicación y con Amazon EFS. Para obtener más información, consulte [Opciones de implementación de Amazon MQ para agentes de ActiveMQ](#).
5. Elija Siguiente.
6. En la página Configure settings (Configurar ajustes), sección Details (Detalles), haga lo siguiente:
 - a. Ingrese el nombre del agente.

Important

No agregue información de identificación personal (PII) ni ninguna otra información confidencial en los nombres de agente. Los nombres de agente y los nombres de usuario están disponibles para otros servicios de AWS, incluido CloudWatch Logs. Los nombres de agente no están diseñados para usarse con información privada o confidencial.

- b. Elija el tipo de instancias del agente (por ejemplo, mq.m5.large). Para obtener más información, consulte [Broker instance types](#).
7. En la sección de acceso a la consola web de ActiveMQ, introduzca un nombre de usuario y una contraseña. Las siguientes restricciones se aplican a los nombres de usuario y las contraseñas de los agentes:
 - El nombre de usuario solo puede contener caracteres alfanuméricos, guiones, puntos, guiones bajos y tildes (- . _ ~).
 - Esta contraseña debe tener 12 caracteres como mínimo, debe contener al menos 4 caracteres únicos y no puede incluir comas, dos puntos ni signos de igual (,:=).

 Important

No agregue información de identificación personal (PII) ni ninguna otra información confidencial en los nombres de usuario de agente. Los nombres de usuario de agente están disponibles para otros servicios de AWS, incluido CloudWatch Logs. Los nombres de usuario de agente no están diseñados para usarse con información privada o confidencial.

La barra verde parpadeante en la parte superior de la página confirma que Amazon MQ está creando el agente de réplicas en la región de recuperación. También puede ver el rol de CRDR y el estado de RPO de sus agentes. Para desactivar las columnas Rol de CRDR y Estado de OPR, elija el icono de engranaje de la esquina superior derecha de la tabla Agentes. A continuación, en la página Preferencias, desactive Rol CRDR o Estado de RPO.

Paso 2: Crear una réplica de un agente existente

1. En la página Agentes de la consola de Amazon MQ, elija Crear agente de réplicas.
2. En la página Elegir agente principal, elija un agente existente para utilizarlo como agente principal de CRDR. A continuación, elija Siguiente.
3. En la página Configurar agente de réplicas, use el menú desplegable para elegir la región de réplicas.
4. En la sección Usuario de la consola de ActiveMQ para el agente de réplicas, proporcione un Nombre de usuario y una Contraseña para el usuario de la consola del agente de réplicas.

Las siguientes restricciones se aplican a los nombres de usuario y las contraseñas de los agentes:

- El nombre de usuario solo puede contener caracteres alfanuméricos, guiones, puntos, guiones bajos y tildes (- . _ ~).
- Esta contraseña debe tener 12 caracteres como mínimo, debe contener al menos 4 caracteres únicos y no puede incluir comas, dos puntos ni signos de igual (,:=).

 Important

No agregue información de identificación personal (PII) ni ninguna otra información confidencial en los nombres de usuario de agente. Los nombres de usuario de agente están disponibles para otros servicios de AWS, incluido CloudWatch Logs. Los nombres de usuario de agente no están diseñados para usarse con información privada o confidencial.

5. En la sección Usuario de replicación de datos para puente de acceso entre agentes, proporcione un Nombre de usuario y una Contraseña para el usuario que accederá tanto al agente principal como al agente de réplicas. Las siguientes restricciones se aplican a los nombres de usuario y las contraseñas de los agentes:

- El nombre de usuario solo puede contener caracteres alfanuméricos, guiones, puntos, guiones bajos y tildes (- . _ ~).
- Esta contraseña debe tener 12 caracteres como mínimo, debe contener al menos 4 caracteres únicos y no puede incluir comas, dos puntos ni signos de igual (,:=).

 Important

No agregue información de identificación personal (PII) ni ninguna otra información confidencial en los nombres de usuario de agente. Los nombres de usuario de agente están disponibles para otros servicios de AWS, incluido CloudWatch Logs. Los nombres de usuario de agente no están diseñados para usarse con información privada o confidencial.

Configure los ajustes adicionales. A continuación, elija Siguiente.

6. En la página Revisar y crear, revise los detalles del agente de réplicas. A continuación, elija Crear agente de réplicas.
7. A continuación, reinicie el agente principal. Esto también reiniciará el agente de réplicas. Para obtener instrucciones sobre cómo reiniciar su agente, consulte [Rebooting a Broker](#).

Para obtener más información sobre la configuración de ajustes adicionales para su agente de ActiveMQ, consulte [Introducción: Creación y conexión con un agente de ActiveMQ](#)

Eliminación de un agente de replicación de datos entre regiones de Amazon MQ

Para eliminar un agente de replicación de datos entre regiones (CRDR) principal o de réplicas, primero debe desvincular y, a continuación, reiniciar los agentes. En las instrucciones siguientes se muestra cómo puede desemparejar y reiniciar los agentes utilizando la consola de administración de AWS.

1. En la página Agentes, elija el agente CRDR que desee desemparejar y, a continuación, elija Editar.
2. En la página Editar del agente de la sección Replicación de datos, elija Desemparejar agentes.
3. Introduzca “confirm” en la ventana emergente para confirmar su elección. A continuación, elija Desemparejar agentes.
4. A continuación, reinicie el agente principal desemparejado. Esto también reiniciará el agente de réplicas. Para obtener instrucciones sobre cómo reiniciar su agente, consulte [Rebooting a Broker](#). Una vez que se haya reiniciado el agente principal, ambos se desemparejarán y se pueden eliminar de forma individual. Para eliminar su agente, consulte [Deleting a broker](#).

Inicio de la transición o conmutación por error para ascender al agente de réplicas de Amazon MQ al rol de agente principal

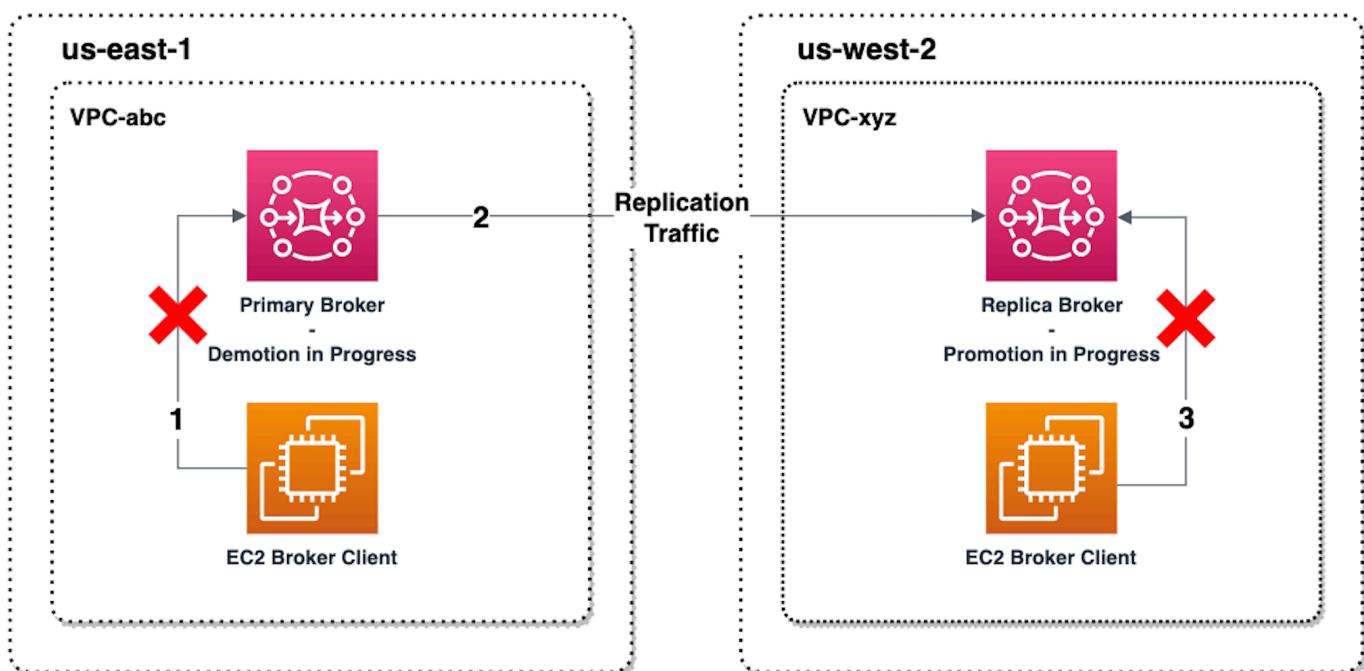
Puede iniciar una conmutación o una conmutación por error cuando desee ascender al agente de réplicas al rol de agente principal. Cuando asciende al agente de réplicas, el agente principal se degrada al rol de agente de réplicas.

Una conmutación prioriza la coherencia sobre la disponibilidad. Se garantiza que los agentes tengan un estado idéntico cuando finalice esta operación de conmutación por error. Con una conmutación, puede haber un período en el que ninguno de los agentes esté disponible para las conexiones de los

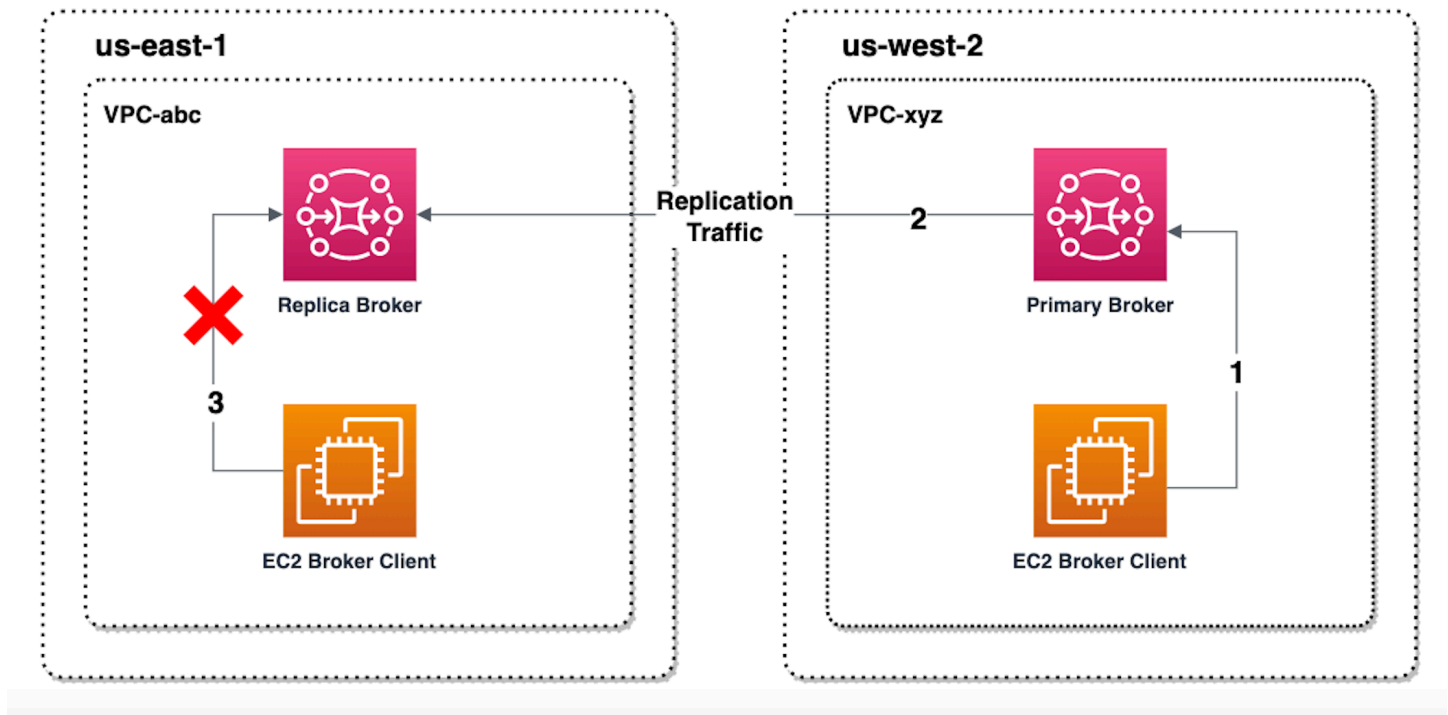
clientes mientras se establece la coherencia entre agentes. Ambos agentes tendrán el mismo estado en el instante en que se ascienda la réplica. El éxito de la conversión depende del estado de ambas regiones y de que la red interregional funcione correctamente.

Una conmutación por error prioriza la disponibilidad sobre la coherencia. No se garantiza que los agentes tengan estados idénticos cuando finalice esta operación. Con una conmutación por error, se garantiza que el agente de réplicas esté disponible inmediatamente para atender el tráfico de clientes, sin esperar a que se sincronicen los datos de réplica ni a que el principal reciba la señal de apagado. La conmutación por error no depende del estado de la región principal original ni de la red interregional para funcionar.

En el siguiente diagrama se ilustra una conmutación en la que ninguno de los agentes acepta conexiones de clientes mientras se vacía la cola de replicación y se sincronizan los estados de los agentes. En este proceso, el cliente de la VPC del agente principal no puede producir más cambios de estado mientras se está produciendo una operación y el agente principal se está degradando a una réplica. Cuando la cola de replicación se vacía y los dos agentes alcanzan un estado idéntico, el cliente de la VPC del agente de réplicas no puede conectarse al agente de réplica hasta que se completa la operación de conmutación por error y el agente de réplicas se asciende a principal.



En el siguiente diagrama se ilustra el estado del agente una vez finalizado el proceso de conmutación. El agente de réplicas original se ascendió al rol de agente principal y está aceptando conexiones de clientes. El cliente puede producir y consumir datos del agente.



Ascender el agente de réplicas mediante la consola

Para ascender el agente de réplicas mediante conmutación o conmutación por error, siga estos pasos en la consola de Amazon MQ.

Note

No puede iniciar la conmutación o la conmutación por error en un agente principal.

1. Cambie a la región del agente de réplicas. En la tabla de agentes, seleccione el agente de réplicas existente que promocionará a principal.
2. En la página de detalles del agente, haga lo siguiente:
 1. Seleccione **Ascender réplica**.
 2. En la ventana emergente, elija **Conmutación** o **Conmutación por error**.
 3. Escriba “confirmar” en el cuadro de texto para confirmar su elección.
 4. Elija **Confirmar**.

Tras iniciar la conmutación por error, el estado del agente cambia a Conmutación por error en curso. La barra de progreso azul de la parte superior de la página Agentes se vuelve verde cuando se completa la conmutación por error.

Note

La configuración solo se replica en el momento en que se crea el agente de réplicas. Las actualizaciones posteriores no se replican.

Métricas de replicación de datos entre regiones en Amazon CloudWatch

La característica de replicación de datos entre regiones de Amazon MQ para ActiveMQ ofrece métricas para mantener la fiabilidad, la disponibilidad y el rendimiento de sus agentes principales y de réplicas. Durante el proceso de replicación, un agente de réplicas de una región secundaria recibe datos replicados de forma asíncrona del agente principal de la región principal. Si el agente principal de la región principal falla, puede ascender a primario al agente de réplicas de la región secundaria iniciando una conmutación o conmutación por error. Para obtener instrucciones sobre la visualización de métricas en Amazon CloudWatch, consulte [Acceso a CloudWatch las métricas de Amazon MQ](#).

Marcas temporales de CRDR

Las siguientes marcas temporales describen cómo se calculan las métricas encontradas en Amazon CloudWatch. Hay cinco marcas temporales en el proceso de replicación de datos:

- Tiempo de observación actual (TCO): el instante actual en el tiempo.
- Hora de creación (TC): el instante en que un evento se creó en la cola de replicación por el agente principal. Disponible tanto en agentes principales como en réplicas.
- Hora de entrega (TD): el instante en el que un evento se entregó correctamente al agente de réplicas. Solo disponible en agentes de réplicas.
- Hora de procesamiento (TP): el instante en el que el agente de la réplica procesó correctamente un evento. Solo disponible en agentes de réplicas.
- Hora de confirmación (TA): el instante en el que el agente principal ha reconocido correctamente un evento. Solo disponible en agentes principales.

Estimar el rendimiento de conmutación o conmutación por error con métricas CRDR CloudWatch

Amazon MQ habilita las métricas para el agente de forma predeterminada. Puede ver las métricas del agente accediendo a la consola de Amazon CloudWatch o mediante la API de CloudWatch. Las siguientes métricas son útiles para comprender el rendimiento de replicación y conmutación por error de sus agentes de CRDR:

Métrica de CloudWatch de Amazon MQ	Motivo del uso de CRDR	
TotalReplicationLag	El tiempo estimado entre TA y TC del último evento no reconocido en el agente principal.	
ReplicationLag	El tiempo estimado entre TP y TC del último evento no reconocido en el agente de réplicas.	
PrimaryWaitTime	El tiempo estimado entre TCO y TC del último evento procesado en el agente principal.	
ReplicaWaitTime	El tiempo estimado entre TCO y TP del último evento procesado en el agente de réplicas.	
QueueSize	El número total de eventos no confirmados en la cola de replicación en el agente principal.	

TotalReplicationLag y ReplicationLag describen la replicación retardada entre los agentes principales y de réplicas. Las dos métricas también se pueden utilizar para estimar el tiempo que transcurrirá hasta que se complete la operación de conmutación o de conmutación por error en curso.

PrimaryWaitTime y ReplicaWaitTime pueden utilizarse para identificar cualquier problema en curso con el proceso de replicación. Si el valor de la métrica crece constantemente, esto puede indicar que el proceso de replicación está degradado o en pausa. La lentitud de la replicación puede deberse a problemas como la partición de la red, los arranques del agente y la larga recuperación.

Tutoriales de ActiveMQ

En los siguientes tutoriales, se muestra cómo puede crear los agentes de ActiveMQ y conectar con ellos. Si quiere usar el código de ejemplo de Java de ActiveMQ, debe instalar [Java Standard Edition Development Kit](#) y efectuar algunos cambios de configuración en el código de ejemplo.

Temas

- [Creación y configuración de una red de agentes de Amazon MQ](#)
- [Conexión de una aplicación Java con el agente Amazon MQ](#)
- [Integración de agentes de ActiveMQ con LDAP](#)
- [Paso 3: \(Opcional\) Conectarse a una AWS Lambda función](#)
- [Creación de un usuario del agente de ActiveMQ](#)
- [Edición de un usuario de agente de ActiveMQ](#)
- [Eliminación de un usuario de agente de ActiveMQ](#)
- [Ejemplos prácticos del uso de Java Message Service \(JMS\) con ActiveMQ](#)

Creación y configuración de una red de agentes de Amazon MQ

Una red de agentes está compuesta por varios [agentes de una sola instancia](#) activos al mismo tiempo o [agentes activos/en espera](#). En este tutorial, obtendrá información acerca de cómo crear una red de dos agentes con una topología de origen y receptor.

Para obtener información general de conceptos e información detallada de la configuración, consulte los siguientes temas:

- [Red de agentes de Amazon MQ](#)

- [Configurar la red de agentes correctamente](#)
- [networkConnector](#)
- [networkConnectionStartAsincrónico](#)
- [Redes de agentes](#) en la documentación de ActiveMQ

Puede utilizar la consola de Amazon MQ para crear una red de agentes de Amazon MQ. Dado que puede comenzar la creación de los dos agentes en paralelo, este proceso tarda unos 15 minutos.

Temas

- [Requisitos previos](#)
- [Paso 1: Permitir tráfico entre agentes](#)
- [Paso 2: Configurar conectores de red para su agente](#)
- [Siguiendo pasos](#)

Requisitos previos

Para crear una red de agentes, debe disponer de lo siguiente:

- Dos o más agentes simultáneamente activos (denominados `MyBroker1` y `MyBroker2` en este tutorial). Para obtener más información sobre la creación de agentes, consulte [Introducción: Creación y conexión con un agente de ActiveMQ](#).
- Los dos corredores deben estar en el mismo VPC o en pares. VPCs Para obtener más información VPCs, consulte [¿Qué es Amazon VPC?](#) en la Guía del usuario de Amazon VPC y [¿Qué es la interconexión de VPC?](#) en la Guía de peering de Amazon VPC.

Important

Si no dispone de una VPC predeterminada, subredes o un grupo de seguridad, debe crearlos en primer lugar. Para obtener más información, consulte los siguientes temas en la Guía del usuario de Amazon VPC.

- [Creación de una VPC predeterminada](#)
- [Creación de una subred predeterminada](#)
- [Creación de un grupo de seguridad](#)

- Dos usuarios con credenciales de inicio de sesión idénticas para ambos agentes. Para obtener más información sobre la creación de usuarios, consulte [Creación de un usuario del agente de ActiveMQ](#).

 Note

Cuando integre la autenticación LDAP con una red de agentes, asegúrese de que el usuario exista como agente de ActiveMQ y como usuario de LDAP.

En el siguiente ejemplo, se usan dos [agentes de una sola instancia](#). Sin embargo, puede crear redes de agentes utilizando [agentes activos/en espera](#) o una combinación de modos de implementación de agente.

Paso 1: Permitir tráfico entre agentes

Una vez que haya creado los agentes, debe permitir el tráfico entre ellos.

1. En la [consola Amazon MQ](#), en la página MyBroker2, en la sección Detalles, en Seguridad y red, elija el nombre de su grupo de seguridad o.



Aparecerá la página Security Groups (Grupos de seguridad) del panel de EC2.

2. En la lista de grupos de seguridad, elija el grupo de seguridad.
3. En la parte inferior de la página, elija Inbound (Entrante) y, a continuación, Edit (Editar).
4. En el cuadro de diálogo Editar reglas de entrada, añada una regla para el OpenWire punto final.
 - a. Seleccione Add Rule (Agregar regla).
 - b. En Type (Tipo), seleccione Custom TCP (TCP personalizada).
 - c. En Rango de puertos, escriba el OpenWire puerto (61617).
 - d. Realice una de las siguientes acciones:
 - Si desea restringir el acceso a una determinada dirección IP, en Source (Origen), deje la opción Custom (Personalizado) seleccionada y escriba la dirección IP de MyBroker1, seguida de /32. (Este convierte la dirección IP en un registro de CIDR válido). Para obtener más información, consulte [Interfaces de redes elásticas](#).

 Tip

Para recuperar la dirección IP de MyBroker1, en la [consola de Amazon MQ](#), elija el nombre del agente y vaya a la sección Details (Detalles).

- Si todos los agentes son privados y pertenecen a la misma VPC, en Source (Origen), deje la opción Custom (Personalizado) seleccionada y escriba el identificador del grupo de seguridad que está editando.

 Note

En el caso de los agentes públicos, debe restringir el acceso mediante direcciones IP.

- e. Seleccione Save.

El agente ahora puede aceptar las conexiones entrantes.

Paso 2: Configurar conectores de red para su agente

Después de permitir el tráfico entre sus agentes, debe configurar conectores de red para uno de ellos.

1. Edite la revisión de configuración para el agente MyBroker1.

- a. En la página MyBroker1, selecciona Editar.
- b. En la página Editar MyBroker 1, en la sección Configuración, elija Ver.

Se muestran el tipo y la versión del motor de agente que utiliza la configuración (por ejemplo, Apache ActiveMQ 5.15.0).

- c. En la pestaña Configuration details, se muestra el número de revisión de la configuración, la descripción y la configuración del agente en formato XML.
- d. Elija Editar configuración.
- e. En la parte inferior del archivo de configuración, anule el comentario de la sección `<networkConnectors>` e incluya la información siguiente:
 - El name del conector de red.

- [La ActiveMQ Web Console username](#) que es común a ambos agentes.
- Habilitar conexiones duplex.
- Realice una de las siguientes acciones:
 - Si va a conectar el bróker a un bróker de instancia única, utilice el `static`: prefijo y el OpenWire punto final `uri` de `MyBroker2`. Por ejemplo:

```
<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>
```

- Si va a conectar el bróker a un bróker activo/en espera, utilice el `static+failover` transporte y el OpenWire punto final `uri` para ambos agentes con los siguientes parámetros de consulta. `?randomize=false&maxReconnectAttempts=0` Por ejemplo:

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

 Note

No incluya las credenciales de inicio de sesión del usuario de ActiveMQ.

- f. Seleccione **Save**.
 - g. En el cuadro de diálogo **Save revision** (Guardar revisión), escriba **Add network of brokers connector for MyBroker2**.
 - h. Seleccione **Save** (Guardar) para guardar la nueva revisión de la configuración.
2. Edite `MyBroker1` para establecer la revisión de configuración más reciente para aplicarla de forma inmediata.

- a. En la página MyBroker1, seleccione Editar.
- b. En la página Editar MyBroker 1, en la sección Configuración, seleccione Programar modificaciones.
- c. En la sección Schedule broker modifications (Programar modificaciones de agente), seleccione aplicar las modificaciones Immediately (Inmediatamente).
- d. Seleccione Aplicar.

MyBroker1 se reinicia y se aplica su revisión de configuración.

Se crea la red de agentes.

Siguientes pasos

Después de configurar su red de agentes, puede probarla produciendo y consumiendo mensajes.

Important

Asegúrese de [habilitar las conexiones entrantes](#) desde su máquina local para el agente MyBroker1 en el puerto 8162 (para la consola web ActiveMQ) y en el puerto 61617 (para el punto final). OpenWire

Es posible que también tenga que ajustar la configuración de los grupos de seguridad para permitir que el productor y el consumidor se conecten a la red de agentes.

1. En la [consola de Amazon MQ](#), acceda a la sección Connections (Conexiones) y anote el punto de enlace de la consola web de ActiveMQ para el agente MyBroker1.
2. Vaya a la ActiveMQ Web Console para el agente MyBroker1.
3. Para verificar que el puente de red está conectado, seleccione Network (Red).

En la sección Network Bridges (Puentes de red), el nombre y la dirección de MyBroker2 se indican en las columnas Remote Broker (Agente remoto) y Remote Address (Dirección remota).

4. Desde cualquier equipo que tenga acceso al agente MyBroker2, cree un consumidor. Por ejemplo:

```
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
```

El consumidor se conecta al OpenWire punto final de la cola MyBroker2 y comienza a consumir los mensajes de la cola. MyQueue

5. Desde cualquier equipo que tenga acceso al agente MyBroker1, cree un productor y envíe algunos mensajes. Por ejemplo:

```
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
```

El productor se conecta al OpenWire punto final de MyBroker1 y comienza a producir mensajes persistentes para ponerlos en colaMyQueue.

Conexión de una aplicación Java con el agente Amazon MQ

Después de crear un agente de ActiveMQ en Amazon MQ, puede conectar su aplicación a él. Los siguientes ejemplos muestran cómo puede utilizar Java Message Service (JMS) para crear una conexión con el agente, crear una cola y enviar un mensaje. Para ver un ejemplo de Java completo y en funcionamiento, consulte [Working Java Example](#).

Puede conectarse a los agentes de ActiveMQ mediante [distintos clientes de ActiveMQ](#). Le recomendamos que utilice el [cliente de ActiveMQ](#).

Temas

- [Requisitos previos](#)
- [Para crear un productor de mensajes y enviar un mensaje](#)
- [Para crear un consumidor de mensajes y recibir el mensaje](#)

Requisitos previos

Habilitar atributos de VPC

Para asegurarse de poder acceder a su agente dentro de su VPC, debe habilitar los atributos `enableDnsHostnames` y `enableDnsSupport` de VPC. Para obtener más información, consulte el tema sobre [soporte de DNS en su VPC](#) en la Guía del usuario de Amazon VPC.

Habilitar las conexiones entrantes

A continuación, habilite las conexiones entrantes para su aplicación.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, elija el nombre de su corredor (por ejemplo, MyBroker).
3. En la **MyBroker** página, en la sección Conexiones, anote las direcciones y los puertos de la URL de la consola web del bróker y de los protocolos a nivel de cable.
4. En la sección Details (Detalles), en Security and network (Seguridad y red), elija el nombre de su grupo de seguridad o



Aparecerá la página Security Groups (Grupos de seguridad) del panel de EC2.

5. En la lista de grupos de seguridad, elija el grupo de seguridad.
6. En la parte inferior de la página, elija Inbound (Entrante) y, a continuación, Edit (Editar).
7. En el cuadro de diálogo Edit inbound rules (Editar reglas entrantes), agregue una regla para cada URL o punto de enlace que desee que sea accesible públicamente (en el ejemplo a continuación se muestra cómo hacerlo para una consola web de agente).
 - a. Seleccione Add Rule (Agregar regla).
 - b. En Type (Tipo), seleccione Custom TCP (TCP personalizada).
 - c. Para Port Range (Rango de puertos), escriba el puerto de la consola web (8162).
 - d. Para Source (Fuente), deje seleccionada la opción Custom (Personalizada) y, luego, escriba la dirección IP del sistema que desea que pueda acceder a la consola web (por ejemplo, 192.0.2.1).
 - e. Seleccione Save.

El agente ahora puede aceptar las conexiones entrantes.

Añadir dependencias de Java

Agregue los paquetes `activemq-client.jar` y `activemq-pool.jar` a la ruta de clases de Java. Los siguientes ejemplos muestran estas dependencias en el archivo `pom.xml` de un proyecto Maven.

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

Para obtener más información acerca de `activemq-client.jar`, consulte el tema sobre la [configuración inicial](#) en la documentación de Apache ActiveMQ.

Important

En el siguiente código de ejemplo, los productores y los consumidores se ejecutan en un único subproceso. Para sistemas de producción (o para probar la conmutación por error de instancias del agente), asegúrese de que sus productores y consumidores se ejecuten en anfitriones o subprocesos independientes.

Para crear un productor de mensajes y enviar un mensaje

Emplee las siguientes instrucciones para crear un productor de mensajes y recibir un mensaje.

1. Cree una fábrica de conexiones JMS agrupadas para el productor de mensajes utilizando el punto de enlace de su agente y, a continuación, ejecute el método `createConnection` contra la fábrica.

 Note

Para un active/standby bróker, Amazon MQ proporciona dos ActiveMQ Web Console URLs, pero solo hay una URL activa a la vez. Del mismo modo, Amazon MQ proporciona dos puntos de enlace para cada protocolo de nivel de cable, pero solo hay un punto de enlace activo en cada par a la vez. Los sufijos -1 y -2 denotan un par redundante. Para obtener más información, consulte [Opciones de implementación de Amazon MQ para agentes de ActiveMQ](#)).

Para los puntos de conexión de protocolo de nivel de cable, debería permitir que la aplicación se conecte a cualquiera de los puntos de conexión mediante el [transporte de conmutación por error](#).

```
// 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

Los productores de mensajes siempre deben usar la clase `PooledConnectionFactory`. Para obtener más información, consulte [Usar siempre el grupo de conexiones](#).

2. Cree una sesión, una cola con el nombre `MyQueue` y un productor de mensajes.

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

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

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

3. Cree la cadena de mensaje `"Hello from Amazon MQ!"` y, a continuación, envíe el mensaje.

```
// 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. Elimine al productor.

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

Para crear un consumidor de mensajes y recibir el mensaje

Emplee las siguientes instrucciones para crear un productor de mensajes y recibir un mensaje.

1. Cree una fábrica de conexiones JMS para el productor de mensajes a través del punto de enlace de su agente y, a continuación, ejecute el método `createConnection` contra la fábrica.

```
// 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

Los consumidores de mensajes nunca usan la clase `PooledConnectionFactory`. Para obtener más información, consulte [Usar siempre el grupo de conexiones](#).

2. Cree una sesión, una cola con el nombre `MyQueue` y un consumidor de mensajes.

```
// 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. Comience a esperar los mensajes y reciba el mensaje cuando llegue.

```
// 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

A diferencia AWS de los servicios de mensajería (como Amazon SQS), el consumidor está constantemente conectado con el intermediario.

4. Cierre el consumidor, la sesión y la conexión.

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

Integración de agentes de ActiveMQ con LDAP

 Important

Amazon MQ no admite el certificado de servidor emitido por una entidad emisora de certificados privada.

Para acceder a sus agentes de ActiveMQ, puede utilizar los siguientes protocolos con TLS habilitado:

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

Amazon MQ ofrece la posibilidad de elegir entre la autenticación nativa de ActiveMQ y la autenticación y autorización de LDAP para administrar los permisos de usuario. Para obtener información sobre las restricciones relacionadas con nombres de usuario y contraseñas de ActiveMQ, consulte [Usuarios](#).

Para permitir que los usuarios y grupos de ActiveMQ trabajen con colas y temas, debe [editar la configuración del agente](#). Amazon MQ utiliza el [complemento de autenticación simple](#) de ActiveMQ

para restringir la lectura y la escritura en los destinos. Para obtener más información y ejemplos, consulte [Configurar siempre una asignación de autorizaciones](#) y [authorizationEntry](#).

 Note

Actualmente, Amazon MQ no admite la autenticación de certificados de cliente.

Temas

- [Integrar LDAP con ActiveMQ](#)
- [Requisitos previos](#)
- [Introducción a LDAP](#)
- [Cómo funciona la integración de LDAP](#)

Integrar LDAP con ActiveMQ

Puede autenticar usuarios de Amazon MQ a través de las credenciales almacenadas en su servidor de protocolo ligero de acceso a directorios (LDAP). También puede agregar, eliminar y modificar usuarios de Amazon MQ, y asignar permisos para temas y colas a través de él. Las operaciones de administración, como la creación, actualización y eliminación de agentes, aún requieren credenciales de IAM y no están integradas con LDAP.

Los clientes que deseen simplificar y centralizar la autenticación y autorización de los agentes de Amazon MQ mediante un servidor LDAP pueden utilizar esta característica. Mantener todas las credenciales de usuario en el servidor LDAP ahorra tiempo y esfuerzo, ya que proporciona una ubicación central donde almacenar y administrar estas credenciales.

Amazon MQ es compatible con LDAP si se utiliza el complemento JAAS de Apache ActiveMQ. Amazon MQ también admite cualquier servidor LDAP, como Microsoft Active Directory u OpenLDAP, que sea compatible con el complemento. Para obtener más información acerca del complemento, consulte la sección [Seguridad](#) de la documentación de ActiveMQ.

Además de los usuarios, puede especificar el acceso a temas y colas para un grupo específico o un usuario a través del servidor LDAP. Para ello, cree entradas que representen temas y colas en el servidor LDAP y, a continuación, asigne permisos a un usuario o grupo de LDAP específico. A continuación, puede configurar el agente para que recupere los datos de autorización del servidor LDAP.

⚠ Important

Cuando se usa LDAP, la autenticación no distingue entre mayúsculas y minúsculas, pero la autorización sí distingue entre mayúsculas y minúsculas para su nombre de usuario.

Requisitos previos

Antes de agregar compatibilidad con LDAP a un agente de Amazon MQ nuevo o existente, debe configurar una cuenta de servicio. Esta cuenta de servicio es necesaria para iniciar una conexión con un servidor LDAP y debe tener los permisos correctos para realizar esta conexión. Esta cuenta de servicio configurará la autenticación LDAP para su agente. Todas las conexiones de cliente posteriores se autenticarán a través de la misma conexión.

La cuenta de servicio es una cuenta del servidor LDAP que tiene acceso para iniciar una conexión. Es un requisito estándar de LDAP, y debe proporcionar las credenciales de la cuenta de servicio una sola vez. Una vez configurada la conexión, todas las conexiones de cliente futuras se autentican a través del servidor LDAP. Las credenciales de su cuenta de servicio se almacenan de forma segura en un formato cifrado, al que solo puede acceder Amazon MQ.

Para integrarse con ActiveMQ, se requiere un árbol de información de directorios (DIT) específico en el servidor LDAP. Para ver un archivo `ldif` de ejemplo que muestra claramente esta estructura, consulte Importar el siguiente archivo LDIF en el servidor LDAP en la sección [Seguridad](#) de la documentación de ActiveMQ.

Introducción a LDAP

Para empezar, vaya a la consola de Amazon MQ y elija LDAP authentication and authorization (Autenticación y autorización LDAP) cuando cree una nueva instancia de agente de Amazon MQ o edite una existente.

Proporcione la siguiente información acerca de la cuenta de servicio:

- Fully qualified domain name (Nombre de dominio completo): la ubicación del servidor LDAP al que se emitirán las solicitudes de autenticación y autorización.

Note

El nombre de dominio completo del servidor LDAP que proporcione no debe incluir el protocolo ni el número de puerto. Amazon MQ antepondrá el protocolo `ldaps` al nombre de dominio completo y agregará el número de puerto `636`.

Por ejemplo, si proporciona el dominio completo `example.com`, Amazon MQ accederá a su servidor LDAP mediante la siguiente URL: `ldaps://example.com:636`.

Para que el anfitrión del agente pueda comunicarse correctamente con el servidor LDAP, el nombre de dominio completo debe poder resolverse públicamente. Para mantener el servidor LDAP privado y seguro, restrinja el tráfico entrante a través de las reglas de entrada del servidor de tal modo que solo permitan el tráfico originado desde la VPC del agente.

- Service account username (Nombre de usuario de la cuenta de servicio): nombre distintivo del usuario que se utilizará para realizar el enlace inicial al servidor LDAP.
- Service account password (Contraseña de la cuenta de servicio): contraseña del usuario que realiza el enlace inicial.

En la siguiente imagen, se resaltan los campos donde se proporcionan estos detalles.

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

En la sección LDAP login configuration (Configuración de inicio de sesión de LDAP), proporcione la siguiente información obligatoria:

- User Base (Base de usuarios): nombre distintivo del nodo del árbol de información de directorios (DIT) en el que se van a buscar los usuarios.
- User Search Matching (Criterios de búsqueda de usuarios): filtro de búsqueda LDAP que se utilizará para buscar usuarios dentro de la userBase. El nombre de usuario del cliente se

sustituye en el marcador de posición {0} del filtro de búsqueda. Para obtener más información, consulte [Autenticación](#) y [Autorización](#).

- **Role Base (Base de roles):** nombre distintivo del nodo del DIT en el que se van a buscar los roles. Los roles se pueden configurar como entradas explícitas de grupos LDAP en el directorio. Una entrada de rol típica puede consistir en un atributo para el nombre del rol, como nombre común (CN), y otro atributo, como `member`, con valores que representan los nombres distintivos o los nombres de usuario de los usuarios que pertenecen al grupo de roles. Por ejemplo, dada la unidad organizativa, que es `group`, puede proporcionar el siguiente nombre distintivo: `ou=group,dc=example,dc=com`.
- **Role Search Matching (Criterios de búsqueda de roles):** filtro de búsqueda de LDAP que se utilizará para buscar roles dentro de la `roleBase`. El nombre distintivo del usuario que coincide con `userSearchMatching` se sustituye en el marcador de posición {0} del filtro de búsqueda. El nombre de usuario del cliente se sustituirá por el marcador de posición {1}. Por ejemplo, si las entradas de roles del directorio incluyen un atributo con el nombre `member`, que contiene los nombres de usuario de todos los usuarios de ese rol, puede proporcionar el siguiente filtro de búsqueda: `(member:=uid={1})`.

En la siguiente imagen, se resaltan los campos donde se especifican estos detalles.

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

En la sección Optional settings (Configuración opcional), puede proporcionar la siguiente información opcional:

- User Role Name (Nombre del rol del usuario): nombre del atributo de LDAP que se indica en la entrada del directorio del usuario para la pertenencia al grupo del usuario. En algunos casos, los roles de usuario pueden identificarse por el valor de un atributo en la entrada del directorio del

usuario. La opción `userRoleName` permite proporcionar el nombre de este atributo. Por ejemplo, consideremos la siguiente entrada de usuario:

```
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
```

Para proporcionar el `userRoleName` correcto para el ejemplo anterior, debería especificar el atributo `memberOf`. Si la autenticación se realiza correctamente, al usuario se le asigna el rol `role1`.

- **Role Name (Nombre del rol):** atributo de nombre de grupo en una entrada de rol cuyo valor es el nombre de ese rol. Por ejemplo, se puede especificar `cn` como el nombre común de una entrada de grupo. Si la autenticación se realiza correctamente, se asigna al usuario el valor del atributo `cn` para cada entrada de rol de la que sea miembro.
- **User Search Subtree (Subárbol de búsqueda de usuarios):** define el ámbito de la consulta de búsqueda de usuario de LDAP. Si es verdadero, el ámbito se configura para que la búsqueda se realice en todo el subárbol bajo el nodo definido por `userBase`.
- **Role Search Subtree (Subárbol de búsqueda de roles):** define el ámbito de la consulta de búsqueda de roles de LDAP. Si es verdadero, el ámbito se configura para que la búsqueda se realice en todo el subárbol bajo el nodo definido por `roleBase`.

En la siguiente imagen, se resalta, los campos donde se especifica esta configuración opcional.

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.

Cómo funciona la integración de LDAP

La integración se puede pensar en dos categorías principales: la estructura para la autenticación y la estructura para la autorización.

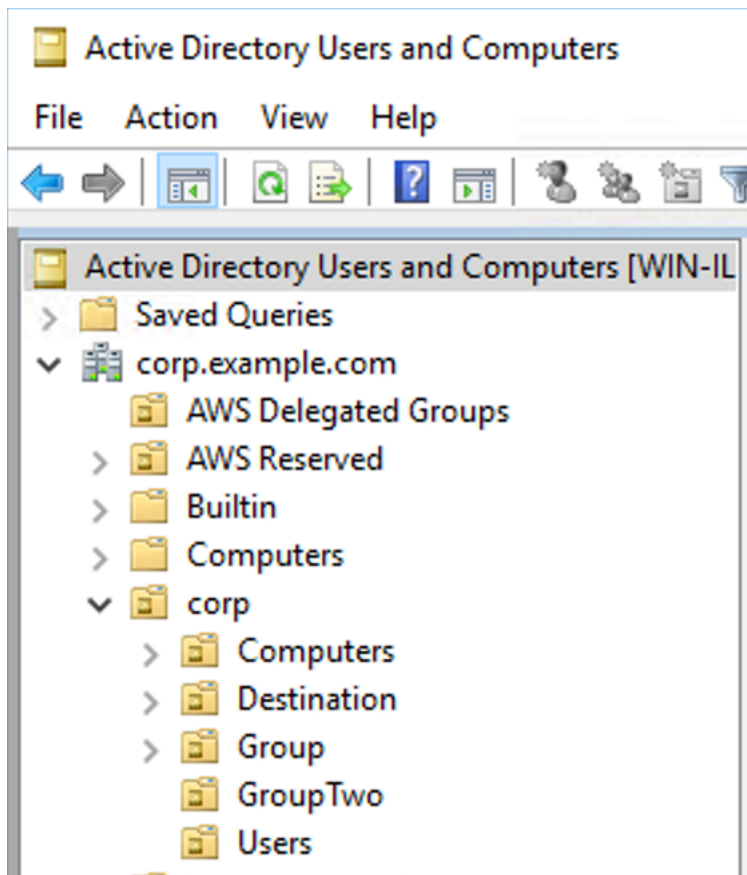
Autenticación

Para la autenticación, las credenciales del cliente deben ser válidas. Estas credenciales se validan contra los usuarios de la base de usuarios del servidor LDAP.

La base de usuarios que se le proporciona al agente de ActiveMQ debe apuntar al nodo del DIT donde se almacenan los usuarios en el servidor LDAP. Por ejemplo, si utilizas y tienes los componentes del dominio corp examplecom, y dentro de ellos tienes unidades corp organizativasUsers, utilizarías lo siguiente como base de usuarios: AWS Managed Microsoft AD

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

El agente de ActiveMQ buscaría usuarios en esta ubicación del DIT para autenticar las solicitudes de conexión del cliente con el agente.



Como el código fuente de ActiveMQ codifica el nombre del atributo de los usuarios como `uid`, debe asegurarse de que cada usuario tenga este atributo configurado. Para simplificar, puede usar el nombre de usuario de la conexión del usuario. Para obtener más información, consulte el código fuente [activemq](#) y la [configuración de mapeos de ID en usuarios y equipos de Active Directory para Windows Server 2016 \(y versiones posteriores\)](#).

Para habilitar el acceso de determinados usuarios a la consola de ActiveMQ, asegúrese de que pertenecen al grupo `amazonmq-console-admins`.

Autorización

Para la autorización, las bases de búsqueda de permisos se especifican en la configuración del agente. La autorización se realiza por destino (o comodín, conjunto de destinos) a través del elemento `cachedLdapAuthorizationMap`, que se encuentra en el archivo de configuración `activemq.xml` del agente. Para obtener más información, consulte el [Módulo de autorización LDAP en caché](#).

 Note

Para poder utilizar el `cachedLDAPAuthorizationMap` elemento en el archivo de `activemq.xml` configuración de su agente, debe elegir la opción de autenticación y autorización LDAP al [crear una configuración mediante la Consola de administración de AWS](#), o establecer la [creación de una configuración mediante la Consola de administración de AWS](#), o establecer la `authenticationStrategy` propiedad LDAP al crear una nueva configuración mediante la API Amazon MQ.

Debe proporcionar los siguientes tres atributos como parte del elemento `cachedLDAPAuthorizationMap`:

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

 Important

Para evitar que la información confidencial se coloque directamente en el archivo de configuración del agente, Amazon MQ bloquea el uso de los siguientes atributos en `cachedLdapAuthorizationMap`:

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

Al crear un intermediario, Amazon MQ sustituye los valores que proporciona a través de la Consola de administración de AWS solicitud de API o que son [ldapServerMetadata](#) propiedad de ella por los atributos anteriores.

En el siguiente ejemplo, se muestra un ejemplo práctico de `cachedLdapAuthorizationMap`.

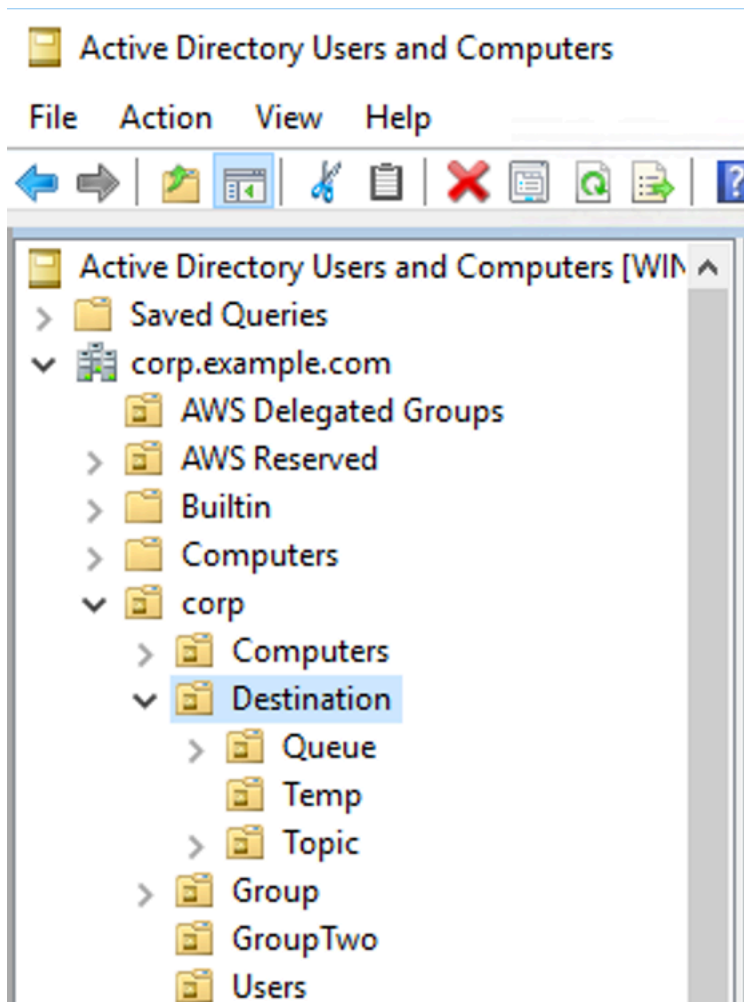
```
<authorizationPlugin>
  <map>
```

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

Estos valores identifican las ubicaciones dentro del DIT donde se especifican los permisos para cada tipo de destino. Así, en el ejemplo anterior AWS Managed Microsoft AD, con los mismos componentes de dominio que, y corp examplecom, especificaría una unidad organizativa con un nombre destination que incluyera todos los tipos de destino. Dentro de esa unidad organizativa, debería crear uno para el destino queues, uno para topics y uno para temp.

Esto significaría que su base de búsqueda de colas, que proporciona información de autorización para destinos de tipo cola, tendría la siguiente ubicación en su DIT:

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



Del mismo modo, las reglas de permisos para temas y destinos temporales se ubicarían en el mismo nivel en el DIT:

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

Dentro de la unidad organizativa de cada tipo de destino (cola, tema, destino temporal), se puede proporcionar un comodín o un nombre de destino específico. Por ejemplo, para proporcionar una regla de autorización para todas las colas que comiencen con el prefijo DEMO.EVENTS.\$, puede crear la siguiente unidad organizativa:

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

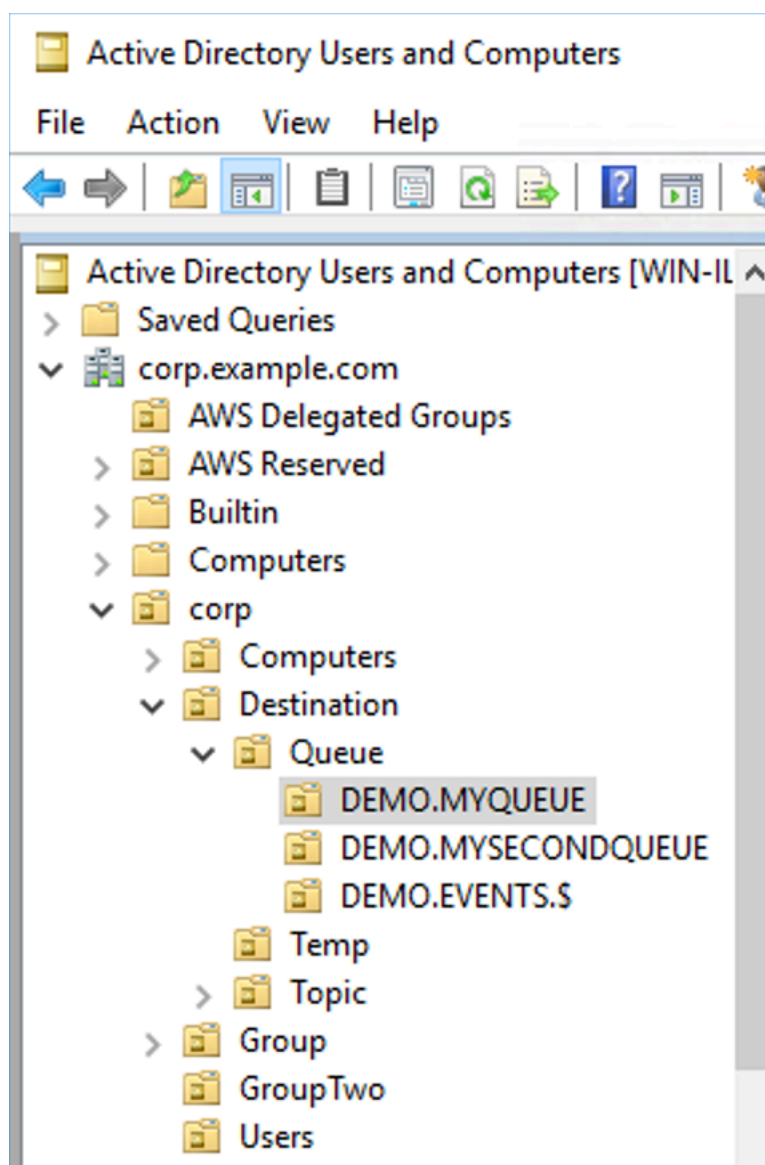
Note

La unidad organizativa DEMO.EVENTS.\$ está dentro de la unidad organizativa Queue.

Para obtener más información acerca de comodines en ActiveMQ, consulte [Caracteres comodín](#)

Para proporcionar reglas de autorización para colas específicas, como DEMO.MYQUEUE, incluya una especificación similar a la siguiente:

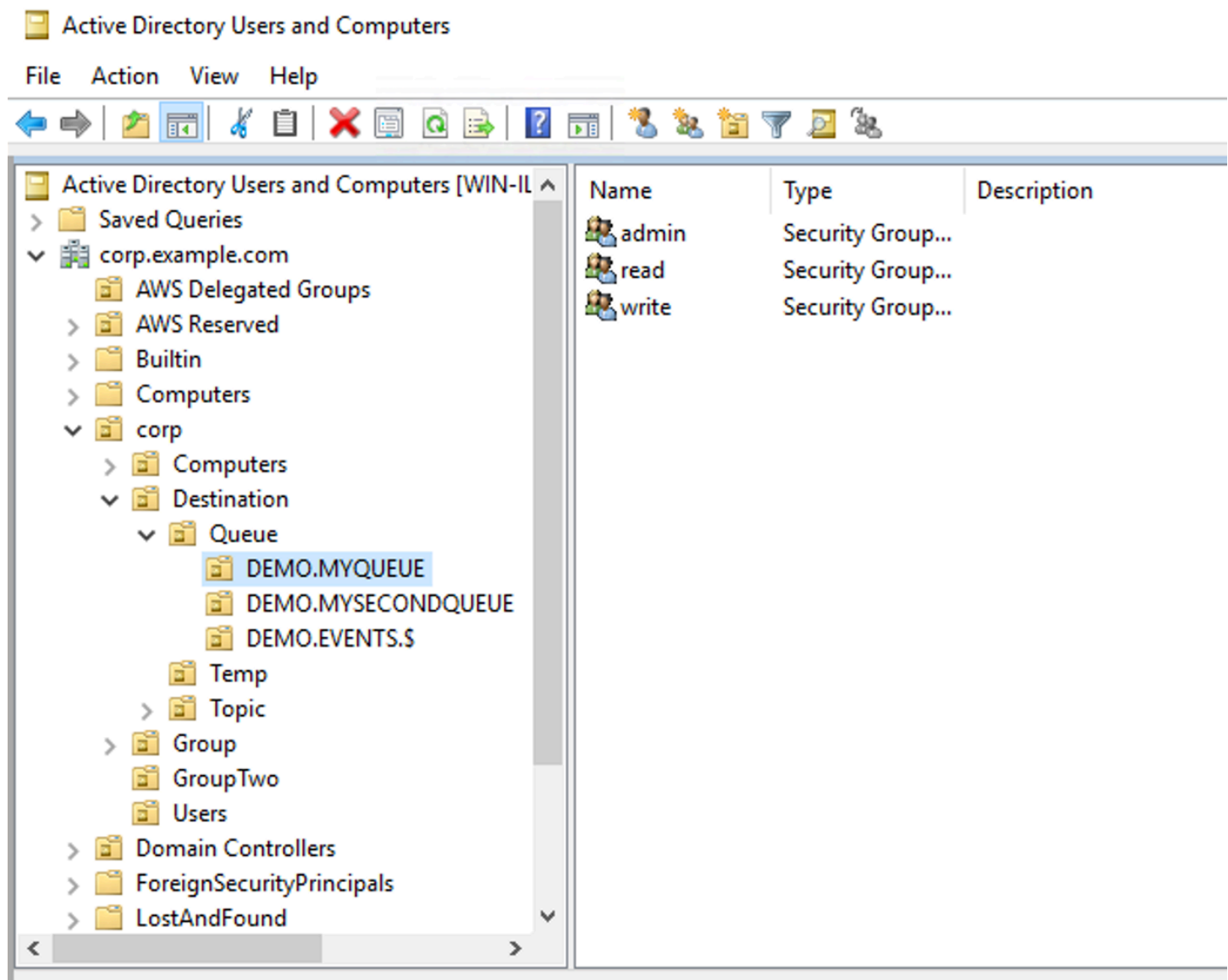
```
OU=DEMO.MYQUEUE,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



Grupos de seguridad

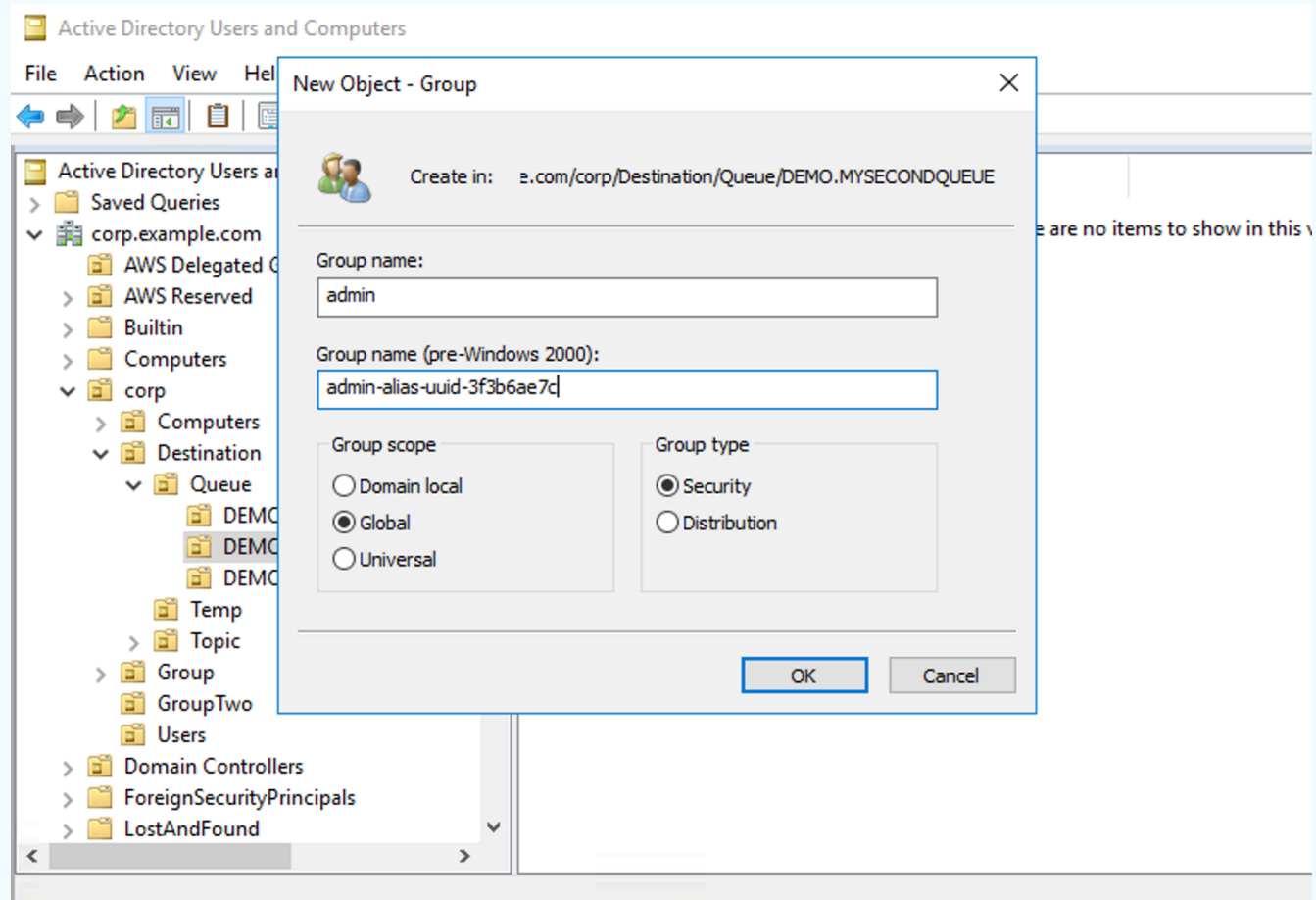
Dentro de cada unidad organizativa que representa un destino o un comodín, debe crear tres grupos de seguridad. Como ocurre con todos los permisos de ActiveMQ, se trata de permisos. `read/write/admin` Para obtener más información acerca de lo que permite a un usuario cada uno de estos permisos, consulte [Seguridad](#) en la documentación de ActiveMQ.

Debe asignar un nombre a estos grupos de seguridad `read`, `write` y `admin`. Dentro de cada uno de estos grupos de seguridad, puede agregar usuarios o grupos, que tendrán permiso para realizar las acciones asociadas. Necesitará estos grupos de seguridad para cada conjunto de destinos comodín o destino individual.



Note

Cuando cree el grupo administrador, se generará un conflicto con el nombre del grupo. Este conflicto se produce porque las reglas heredadas anteriores a Windows 2000 no permiten que los grupos compartan el mismo nombre, aunque se encuentren en diferentes ubicaciones del DIT. El valor que se muestra en el cuadro de texto Pre-Windows 2000 (Anterior a Windows 2000) no tiene ningún impacto sobre la configuración, pero debe ser único global. Para evitar este conflicto, puede agregar un sufijo uuid a cada grupo admin.



Si agrega un usuario al grupo de seguridad `admin` para un destino determinado, el usuario podrá crear y eliminar ese tema. Si lo agrega al grupo de seguridad `read`, podrá leer desde el destino, y si lo agrega al grupo `write`, podrá escribir allí.

Además de agregar usuarios individuales a los permisos de grupos de seguridad, también puede agregar grupos completos. Sin embargo, como ActiveMQ codifica los nombres de los atributos

para los grupos, debe asegurarse de que el grupo que desea agregar tenga la clase de objeto `groupOfNames`, como se muestra en el código fuente [activemq](#).

Para hacerlo, realice el mismo proceso que empleó con `uid` para los usuarios. Consulte [Configuración de mapeos de ID en Usuarios y equipos de Active Directory para Windows Server 2016 \(y versiones posteriores\)](#).

Paso 3: (Opcional) Conectarse a una AWS Lambda función

AWS Lambda puede conectarse a su agente de Amazon MQ y consumirlos. Cuando conecta un agente a Lambda, crea un [mapeo de origen de eventos](#) que lee mensajes de una cola e invoca la función [de forma sincrónica](#). El mapeo de origen de eventos que crea lee los mensajes de su agente en lotes y los convierte en una carga de Lambda en forma de objeto JSON.

Conectar a su agente con una función Lambda

1. Agregue los siguientes permisos de roles de IAM al [rol de ejecución](#) de su función Lambda.
 - [mq: DescribeBroker](#)
 - [ec2: CreateNetworkInterface](#)
 - [ec2: DeleteNetworkInterface](#)
 - [ec2: DescribeNetworkInterfaces](#)
 - [ec2: DescribeSecurityGroups](#)
 - [ec2: DescribeSubnets](#)
 - [ec2: DescribeVpcs](#)
 - [registros: CreateLogGroup](#)
 - [registros: CreateLogStream](#)
 - [registros: PutLogEvents](#)
 - [administrador de secretos: GetSecretValue](#)

Note

Sin los permisos necesarios de IAM, su función no podrá leer correctamente los registros de los recursos de Amazon MQ.

2. (Opcional) Si ha creado un agente sin accesibilidad pública, debe realizar una de las siguientes acciones para permitir que Lambda se conecte con su agente:
 - Puede configurar una gateway NAT para cada subred pública. Para obtener más información, consulte el tema sobre [acceso a Internet y a los servicios para funciones conectadas a la VPC](#) en la AWS Lambda Guía para desarrolladores.
 - Cree una conexión entre Amazon Virtual Private Cloud (Amazon VPC) y Lambda a través de un punto de enlace de la VPC. Su Amazon VPC también debe conectarse a AWS Security Token Service (AWS STS) y a los puntos de conexión de Secrets Manager. Para obtener más información, consulte [Configuración de puntos de conexión de VPC de interfaz para Lambda](#) en la AWS Lambda Guía para desarrolladores.
3. [Configure el agente como origen de eventos](#) para una función Lambda a través de la Consola de administración de AWS. También puede usar el [create-event-source-mapping](#) AWS Command Line Interface comando.
4. Escriba algún código para que su función de Lambda procese los mensajes de lo consumido desde su agente. La carga de Lambda que recupere el mapeo de origen de eventos depende del tipo de motor del agente. A continuación, se muestra un ejemplo de una carga de Lambda para una cola de ActiveMQ en Amazon MQ.

 Note

En el ejemplo, testQueue es el nombre de la cola.

```
{
  "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
  }
]
}
```

Para obtener más información acerca de cómo conectar Amazon MQ a Lambda, las opciones que admite Lambda para un origen de eventos de Amazon MQ y los errores de mapeo de origen de eventos, consulte el tema sobre el [uso de Lambda con Amazon MQ](#) en la AWS Lambda Guía para desarrolladores.

Creación de un usuario del agente de ActiveMQ

Un usuario de ActiveMQ es una persona o una aplicación que pueden acceder a las colas y los temas de un agente de ActiveMQ. Puede configurar a los usuarios para que tengan permisos específicos. Por ejemplo, puede permitir que algunos usuarios accedan a la [consola web de ActiveMQ](#).

Un grupo es una etiqueta semántica. Puede asignar un grupo a un usuario y configurar permisos para que los grupos envíen, reciban y administren colas y temas específicos.

Note

No se pueden configurar grupos independientemente de los usuarios. Se crea una etiqueta de grupo cuando se agrega al menos un usuario y se elimina cuando se los quita a todos.

Note

El grupo `activemq-webconsole` de ActiveMQ en Amazon MQ tiene permisos de administrador en todas las colas y temas. Todos los usuarios de este grupo tendrán acceso de administrador.

En los siguientes ejemplos, se muestra cómo crear, editar y eliminar los usuarios del agente de Amazon MQ mediante la Consola de administración de AWS.

Creación de un nuevo usuario de agente de ActiveMQ

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, elija el nombre de su corredor (por ejemplo MyBroker) y, a continuación, elija Ver detalles.

En la **MyBroker** página, en la sección Usuarios, aparecen todos los usuarios de este corredor.

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

3. Seleccione la opción Crear un usuario.
4. En el cuadro de diálogo Create user (Crear usuario), especifique un nombre de usuario en Username y una contraseña en Password.
5. (Opcional) Escriba los nombres de los grupos a los que pertenece el usuario separados mediante coma (por ejemplo: Devs, Admins).
6. (Opcional) Para permitir que el usuario pueda obtener acceso a [ActiveMQ Web Console](#), seleccione ActiveMQ Web Console.
7. Seleccione la opción Crear un usuario.

⚠ Important

Si realiza cambios en un usuario, los cambios no se aplica al usuario de inmediato. Para aplicar los cambios, debe espere al próximo período de mantenimiento o [reiniciar el agente](#).

Edición de un usuario de agente de ActiveMQ

Para editar un usuario existente, haga lo siguiente:

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, elija el nombre de su corredor (por ejemplo MyBroker) y, a continuación, elija Ver detalles.

En la **MyBroker** página, en la sección Usuarios, aparecen todos los usuarios de este corredor.

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

3. Seleccione sus credenciales de inicio de sesión y elija Editar.

Aparecerá el cuadro de diálogo Edit user (Editar usuario).

4. (Opcional) Escriba una nueva contraseña en Password (Contraseña).
5. (Opcional) Agregue o quite los nombres de los grupos a los que pertenece el usuario separados mediante coma (por ejemplo: Managers, Admins).
6. (Opcional) Para permitir que el usuario pueda obtener acceso a [ActiveMQ Web Console](#), seleccione ActiveMQ Web Console.
7. Para guardar los cambios realizados para el usuario, elija Done (Listo).

⚠ Important

Si realiza cambios en un usuario, los cambios no se aplica al usuario de inmediato. Para aplicar los cambios, debe espere al próximo período de mantenimiento o [reiniciar el agente](#).

Eliminación de un usuario de agente de ActiveMQ

Cuando ya no necesite un usuario, puede eliminarlo.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, elija el nombre de su corredor (por ejemplo MyBroker) y, a continuación, elija Ver detalles.

En la **MyBroker** página, en la sección Usuarios, aparecen todos los usuarios de este corredor.

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

3. Seleccione sus credenciales de inicio de sesión (por ejemplo **MyUser**) y, a continuación, elija Eliminar.
4. Para confirmar la eliminación del usuario, en la sección ¿Eliminar? **MyUser** en el cuadro de diálogo, seleccione Eliminar.

⚠ Important

Si realiza cambios en un usuario, los cambios no se aplica al usuario de inmediato. Para aplicar los cambios, debe espere al próximo período de mantenimiento o [reiniciar el agente](#).

Ejemplos prácticos del uso de Java Message Service (JMS) con ActiveMQ

Los siguientes ejemplos muestran cómo puede trabajar con ActiveMQ mediante programación:

- El código Java de OpenWire ejemplo se conecta a un intermediario, crea una cola y envía y recibe un mensaje. Para obtener un desglose y una explicación más detallados, consulte [Connecting a Java application to your broker](#).
- El ejemplo MQTT de código Java se conecta con un agente, crea un tema y publica y recibe un mensaje.
- El ejemplo STOMP+WSS de código Java se conecta con un agente, crea una cola y publica y recibe un mensaje.

Requisitos previos

Habilitar atributos de VPC

Para asegurarse de poder acceder a su agente dentro de su VPC, debe habilitar los atributos `enableDnsHostnames` y `enableDnsSupport` de VPC. Para obtener más información, consulte el tema sobre [soporte de DNS en su VPC](#) en la Guía del usuario de Amazon VPC.

Habilitar las conexiones entrantes

Para trabajar con Amazon MQ mediante programación, debe utilizar conexiones entrantes.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, elija el nombre de su corredor (por ejemplo, MyBroker).
3. En la **MyBroker** página, en la sección Conexiones, anote las direcciones y los puertos de la URL de la consola web del bróker y de los protocolos a nivel de cable.
4. En la sección Details (Detalles), en Security and network (Seguridad y red), elija el nombre de su grupo de seguridad o



Aparecerá la página Security Groups (Grupos de seguridad) del panel de EC2.

5. En la lista de grupos de seguridad, elija el grupo de seguridad.
6. En la parte inferior de la página, elija Inbound (Entrante) y, a continuación, Edit (Editar).
7. En el cuadro de diálogo Edit inbound rules (Editar reglas entrantes), agregue una regla para cada URL o punto de enlace que desee que sea accesible públicamente (en el ejemplo a continuación se muestra cómo hacerlo para una consola web de agente).
 - a. Seleccione Add Rule (Agregar regla).
 - b. En Type (Tipo), seleccione Custom TCP (TCP personalizada).

- c. Para Port Range (Rango de puertos), escriba el puerto de la consola web (8162).
- d. Para Source (Fuente), deje seleccionada la opción Custom (Personalizada) y, luego, escriba la dirección IP del sistema que desea que pueda acceder a la consola web (por ejemplo, 192.0.2.1).
- e. Seleccione Save.

El agente ahora puede aceptar las conexiones entrantes.

Agregar dependencias de Java

OpenWire

Agregue los paquetes `activemq-client.jar` y `activemq-pool.jar` a la ruta de clases de Java. Los siguientes ejemplos muestran estas dependencias en el archivo `pom.xml` de un proyecto Maven.

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

Para obtener más información acerca de `activemq-client.jar`, consulte el tema sobre la [configuración inicial](#) en la documentación de Apache ActiveMQ.

MQTT

Añada el paquete `org.eclipse.paho.client.mqttv3.jar` en la ruta de clases de Java. En el siguiente ejemplo se muestra esta dependencia en el archivo `pom.xml` de un proyecto Maven.

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

```
        <version>1.2.0</version>
      </dependency>
    </dependencies>
```

Para obtener más información sobre `org.eclipse.paho.client.mqttv3.jar`, consulte [Cliente Java de Eclipse Paho](#).

STOMP+WSS

Añada los paquetes siguientes a la ruta de clases de Java:

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

Los siguientes ejemplos muestran estas dependencias en el archivo `pom.xml` de un proyecto Maven.

```
<dependencies>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-messaging</artifactId>
    <version>5.0.5.RELEASE</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-websocket</artifactId>
    <version>5.0.5.RELEASE</version>
  </dependency>
  <dependency>
    <groupId>javax.websocket</groupId>
    <artifactId>javax.websocket-api</artifactId>
    <version>1.1</version>
  </dependency>
  <dependency>
    <groupId>org.eclipse.jetty.aggregate</groupId>
    <artifactId>jetty-all</artifactId>
    <type>pom</type>
```



```

        <version>9.3.3.v20150827</version>
    </dependency>
    <dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-simple</artifactId>
        <version>1.6.6</version>
    </dependency>
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>2.5.0</version>
    </dependency>
</dependencies>

```

Para obtener más información, consulte el tema sobre la [compatibilidad con STOMP](#) en la documentación de la plataforma Spring.

MQExampleAmazon.java

Important

En el siguiente código de ejemplo, los productores y los consumidores se ejecutan en un único subproceso. Para sistemas de producción (o para probar la conmutación por error de instancias del agente), asegúrese de que sus productores y consumidores se ejecuten en anfitriones o subprocesos independientes.

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);
            }
        }
    }
}

```

Administración de versiones del motor de Amazon MQ para ActiveMQ

Apache ActiveMQ organiza los números de versión según la especificación semántica del control de versiones como X.Y.Z. En las implementaciones de Amazon MQ para ActiveMQ, X indica la versión principal, Y representa el número de versión secundaria y Z hace referencia al número de versión del parche. Amazon MQ considera que un cambio de versión es importante si cambian los números de versión principal. Por ejemplo, actualizar de la versión 5.17 a la 6.0 se considera una actualización

de la versión principal. Un cambio de versión se considera secundario si solo cambia el número de versión secundaria o del parche. Por ejemplo, actualizar desde la versión 5. 18 a 5. 19 se considera una actualización de versión menor. Cuando `autoMinorVersionUpgrade` está activado, Amazon MQ actualiza su agente a la versión del parche más reciente disponible.

Amazon MQ para ActiveMQ recomienda que todos los agentes utilicen la versión secundaria compatible más reciente. Para obtener instrucciones sobre cómo actualizar la versión del motor del agente, consulte [Actualización de una versión del motor del agente de Amazon MQ](#).

Versiones admitidas del motor de Amazon MQ para ActiveMQ

El calendario de soporte de versiones de Amazon MQ indica cuándo finalizará el soporte de una versión del motor del agente. Cuando finaliza el soporte de una versión, Amazon MQ actualiza automáticamente todos los agentes de esta versión a la siguiente versión compatible. Esta actualización se lleva a cabo durante los períodos de mantenimiento programados por su agente, dentro de los 45 días siguientes a la end-of-support fecha.

Amazon MQ avisa con al menos 90 días de antelación antes de que finalice el soporte de una versión. Le recomendamos que actualice su corredor antes de end-of-support esa fecha para evitar cualquier interrupción. Además, no puede crear nuevos agentes en las versiones cuyo fin del soporte está programado en un plazo de 30 días a partir de la fecha de fin del soporte.

Versión de Apache ActiveMQ	Fin del soporte en Amazon MQ
ActiveMQ 5.19 (recomendado)	
ActiveMQ 5.18	
ActiveMQ 5.17	16 de junio de 2025
ActiveMQ 5.16	15 de noviembre de 2024
ActiveMQ 5.15	16 de septiembre de 2024

Cuando crea un nuevo agente de Amazon MQ para ActiveMQ, puede especificar cualquier versión compatible del motor de ActiveMQ. Si no especifica el número de versión del motor al crear un agente, Amazon MQ selecciona automáticamente, de forma predeterminada, el número de versión del motor más reciente.

Actualizaciones de la versión del motor

Puede actualizar manualmente su agente en cualquier momento a la siguiente versión principal o secundaria compatible. Al activar las [actualizaciones automáticas de las versiones secundarias](#), Amazon MQ actualizará su agente a la última versión de parche compatible durante el [período de mantenimiento](#).

Para obtener más información sobre la actualización manual del agente, consulte [the section called “Actualización de la versión del motor”](#).

Lista de versiones del motor compatibles

Puede enumerar todas las versiones de motores principales y secundarias compatibles mediante el comando. [describe-broker-instance-options](#) AWS CLI

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

Para filtrar los resultados por motor y tipo de instancia, utilice las opciones `--engine-type` y `--host-instance-type`, como se muestra a continuación.

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

Por ejemplo, para filtrar los resultados de ActiveMQ y el tipo de instancia, *engine-type* ACTIVEMQ sustitúyalos por `mq.m5.large` y por *instance-type* `mq.m5.large`

Prácticas recomendadas de Amazon MQ para ActiveMQ

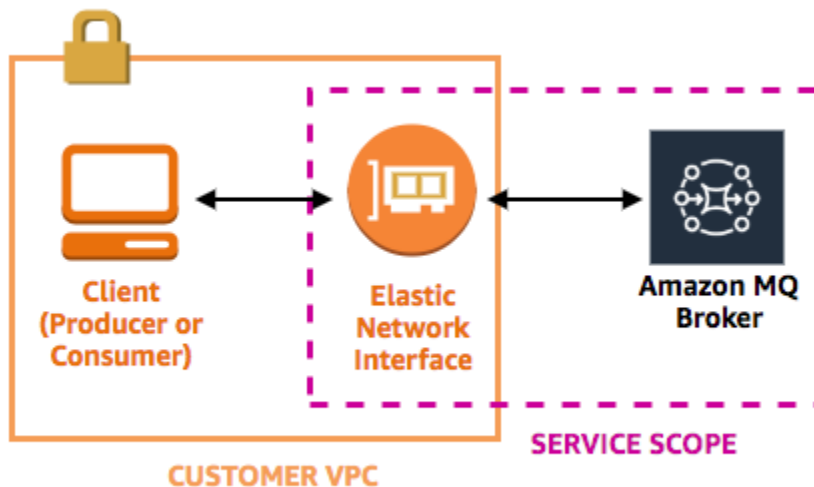
Utilice esta sección como referencia para encontrar rápidamente recomendaciones que le permitan maximizar el rendimiento y minimizar los costos al usar agentes de ActiveMQ en Amazon MQ.

No modifique ni elimine nunca la interfaz de red elástica de Amazon MQ

Cuando se [crea un agente de Amazon MQ](#) por primera vez, Amazon MQ aprovisiona una [interfaz de red elástica](#) en la [nube virtual privada \(VPC\)](#) de la cuenta y, por lo tanto, requiere una serie de [permisos de EC2](#). La interfaz de red permite al cliente (productor o consumidor) comunicarse con el agente de Amazon MQ. Se considera que la interfaz de red está dentro del ámbito del servicio de Amazon MQ, a pesar de que forma parte de la VPC de la cuenta.

⚠ Warning

No se debe modificar ni eliminar esta interfaz de red. Si se modifica o elimina la interfaz de red, se puede provocar una pérdida permanente de la conexión entre la VPC y el agente.



Usar siempre el grupo de conexiones

En un escenario con un único productor y un único consumidor (como el tutorial [Introducción: Creación y conexión con un agente de ActiveMQ](#)), puede usar una sola clase [ActiveMQConnectionFactory](#) para cada productor y consumidor. Por ejemplo:

```
// 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();
```

Sin embargo, en escenarios más realistas con varios productores y consumidores, puede resultar costoso e ineficiente crear un gran número de conexiones para varios productores.

En estos escenarios, debe agrupar varias solicitudes de productor mediante la clase [PooledConnectionFactory](#). Por ejemplo:

 Note

Los consumidores de mensajes nunca usan la clase `PooledConnectionFactory`.

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

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

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

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

Usar siempre el transporte de conmutación por error para conectarse a puntos de enlace de varios agentes

Si necesita que su aplicación se conecte a puntos de enlace de varios clientes, por ejemplo, cuando usa un modo de implementación [activo/de reserva](#) o cuando [migra desde un agente de mensajes local a Amazon MQ](#), utilice el [transporte de conmutación por error](#) para permitir que sus consumidores se conecten con uno de ellos de forma aleatoria. Por ejemplo:

```
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-west-2.amazonaws.com:61617)?randomize=true
```

⚠ Important

Los agentes de zonas de disponibilidad múltiple pueden experimentar conmutaciones por error durante los periodos de mantenimiento y los reinicios del agente. Utilice el transporte de conmutación por error para garantizar la disponibilidad de su agente.

Evitar usar los selectores de mensajes

Es posible usar [selectores JMS](#) para asociar filtros a suscripciones a temas (para dirigir los mensajes a los consumidores en función de su contenido). Sin embargo, el uso de selectores JMS llena el búfer de filtro del agente de Amazon MQ, impidiéndole filtrar mensajes.

En general, no permita a los consumidores dirigir los mensajes, ya que, para que el desacoplamiento de consumidores y productores sea óptimo, tanto el consumidor como el productor deben ser efímeros.

Preferir los destinos virtuales a las suscripciones duraderas

Una [suscripción duradera](#) puede ayudar a garantizar que el consumidor reciba todos los mensajes publicados en un tema, por ejemplo, tras la restauración de una conexión perdida. Sin embargo, el uso de las suscripciones duraderas también impide el uso de la competencia de consumidores y puede tener problemas de desempeño a escala. Considere la posibilidad de utilizar [destinos virtuales](#) en su lugar.

Si utiliza el emparejamiento de Amazon VPC, evite el cliente IPs en el rango CIDR **10.0.0.0/16**

Si está configurando el emparejamiento de Amazon VPC entre la infraestructura local y su agente de Amazon MQ, no debe configurar las conexiones de los clientes dentro del rango de CIDR. IPs 10.0.0.0/16

Desactivar el almacenamiento y el envío simultáneos en colas con consumidores lentos

De forma predeterminada, Amazon MQ optimiza las colas para los consumidores rápidos:

- Se considera que los consumidores son rápidos si pueden mantener el ritmo de los mensajes generados por los productores.

- Se considera que los consumidores son lentos si una cola crea un atasco de mensajes sin confirmar, lo que puede provocar una reducción en el rendimiento del productor.

Para indicar a Amazon MQ que optimice las colas para los consumidores lentos, establezca el atributo `concurrentStoreAndDispatchQueues` en `false`. Para ver una configuración de ejemplo, consulte [concurrentStoreAndDispatchQueues](#).

Elegir el tipo de instancia de agente correcto para obtener el mejor desempeño

La velocidad de los mensajes de un [tipo de instancia de agente](#) depende del caso de uso de su aplicación y de los siguientes factores:

- Uso de ActiveMQ en modo persistente
- Tamaño del mensaje
- El número de productores y consumidores
- El número de destinos

Descripción de la relación entre el tamaño del mensaje, la latencia y el rendimiento

Dependiendo de su caso de uso, es posible que un tipo de instancia de agente más grande no mejore necesariamente el desempeño del sistema. Cuando ActiveMQ escribe mensajes en un almacenamiento duradero, el tamaño de sus mensajes determina el factor limitante de su sistema:

- Si sus mensajes son más pequeños que 100 KB, la latencia de almacenamiento persistente es el factor limitante.
- Si sus mensajes son más grandes que 100 KB, el desempeño de almacenamiento persistente es el factor limitante.

Cuando se utiliza ActiveMQ en modo persistente, la escritura en el almacenamiento se produce normalmente cuando hay pocos consumidores o cuando los consumidores son lentos. En el modo no persistente, la escritura en el almacenamiento también se produce con consumidores lentos si la memoria del montón de la instancia del agente está llena.

Para determinar el mejor tipo de instancia de agente para su aplicación, recomendamos probar diferentes tipos de instancia de agente. Para obtener más información, consulte [Broker instance types](#) y también [Medición del rendimiento para Amazon MQ mediante el punto de referencia de JMS](#).

Casos de uso para tipos de instancias de agentes más grandes

Hay tres casos de uso comunes cuando los tipos de instancia de agente más grandes mejoran el desempeño:

- **Non-persistent mode (Modo no persistente):** cuando la aplicación sea menos sensible a la pérdida de mensajes durante la [conmutación por error de la instancia del agente](#) (por ejemplo, cuando se emiten resultados deportivos), a menudo es posible utilizar el modo no persistente de ActiveMQ. En este modo, ActiveMQ escribe mensajes en el almacenamiento persistente solo si la memoria acumulada de la instancia del agente está llena. Los sistemas que utilizan el modo no persistente pueden beneficiarse de una mayor cantidad de memoria, una CPU más rápida y una red más rápida disponibles en tipos de instancia de agente más grandes.
- **Fast consumers (Consumidores rápidos):** cuando hay consumidores activos disponibles y el indicador [concurrentStoreAndDispatchQueues](#) está habilitado, ActiveMQ permite que los mensajes fluyan directamente del productor al consumidor sin enviar mensajes al almacenamiento (incluso en modo persistente). Si su aplicación puede consumir mensajes rápidamente (o si puede diseñar sus consumidores para que lo hagan), puede beneficiarse de un tipo de instancia de agente más grande. Para permitir que la aplicación consuma mensajes más rápidamente, añada subprocesos de consumo a las instancias de aplicación o escálelas vertical u horizontalmente.
- **Batched transactions (Transacciones por lote):** cuando se utiliza el modo persistente y se envían múltiples mensajes por transacción, se puede obtener un mayor rendimiento general de los mensajes utilizando tipos de instancia de agente más grandes. Para obtener más información, consulte la sección sobre la [necesidad de usar transacciones](#) en la documentación de ActiveMQ.

Elegir el tipo de almacenamiento de agente correcto para obtener el mejor rendimiento

Para aprovechar el alto nivel de durabilidad y la replicación en varias zonas de disponibilidad, utilice Amazon EFS. Para aprovechar la baja latencia y el alto rendimiento, utilice Amazon EBS. Para obtener más información, consulte [Storage](#).

Configurar la red de agentes correctamente

Al crear una [red de agentes](#), configúrela correctamente para su aplicación:

- **Enable persistent mode (Habilitar modo persistente):** dado que cada instancia de agente (en relación con sus pares) actúa como un productor o un consumidor, las redes de agentes no proporcionan replicación distribuida de mensajes. El primer agente que actúa como consumidor recibe un mensaje y lo mantiene en almacenamiento. Este agente envía una confirmación al productor y reenvía el mensaje al siguiente agente. Cuando el segundo agente confirma la persistencia del mensaje, el primer agente elimina el mensaje.

Si se deshabilita el modo persistente, el primer agente confirma al productor sin mantener el mensaje en almacenamiento. Para obtener más información, consulte [Replicated Message Store](#) y [What is the difference between persistent and non-persistent delivery?](#) en la documentación de Apache ActiveMQ.

- **Don't disable advisory messages for broker instances (No desactivar mensajes de aviso para instancias del agente):** para obtener más información, consulte [Mensaje de aviso](#) en la documentación de Apache ActiveMQ.
- **Don't use multicast broker discovery (No utilice detección de agentes mediante multidifusión):** Amazon MQ no admite la detección de agentes mediante multidifusión. Para obtener más información, consulte [What is the difference between discovery, multicast, and zeroconf?](#) en la documentación de Apache ActiveMQ.

Para evitar los reinicios lentos, recupere las transacciones XA preparadas

ActiveMQ admite transacciones distribuidas (XA). Saber cómo procesa ActiveMQ las transacciones XA puede ayudarle a evitar tiempos de recuperación lentos cuando se reinicia el agente y conmutaciones por error en Amazon MQ.

Las transacciones XA preparadas sin resolver se repiten en cada reinicio. Si estas permanecen sin resolver, su número irá creciendo con el tiempo, lo que aumentará considerablemente el tiempo necesario para iniciar el agente. Esto afecta al tiempo de reinicio y de conmutación por error. Debe resolver estas transacciones con una instrucción `commit()` o `rollback()` para que el rendimiento no se degrade con el paso del tiempo.

Para supervisar las transacciones de XA preparadas no resueltas, puede utilizar la `JournalFilesForFastRecovery` métrica de Amazon CloudWatch Logs. Si este número aumenta

o es sistemáticamente mayor que 1, debería recuperar sus transacciones sin resolver con código similar al que se incluye en el siguiente ejemplo. Para obtener más información, consulte [Cuotas en Amazon MQ](#).

El siguiente código de ejemplo recorre las transacciones XA preparadas y las cierra con un `rollback()`.

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

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

public class RecoverXaTransactions {
    private static final ActiveMQXAConnectionFactory ACTIVE_MQ_CONNECTION_FACTORY;
    final static String WIRE_LEVEL_ENDPOINT =
        "tcp://localhost:61616";
    static {
        final String activeMqUsername = "MyUsername123";
        final String activeMqPassword = "MyPassword456";
        ACTIVE_MQ_CONNECTION_FACTORY = new
ActiveMQXAConnectionFactory(activeMqUsername, activeMqPassword, WIRE_LEVEL_ENDPOINT);
        ACTIVE_MQ_CONNECTION_FACTORY.setUsername(activeMqUsername);
        ACTIVE_MQ_CONNECTION_FACTORY.setPassword(activeMqPassword);
    }

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

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

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

En un escenario real, puede comparar sus transacciones XA preparadas con las del administrador de transacciones XA. A continuación, puede decidir si tratar cada transacción preparada con una instrucción `rollback()` o `commit()`.

Uso de Amazon MQ para RabbitMQ

Amazon MQ facilita la creación de un agente de mensajes con los recursos informáticos y de almacenamiento que se adapten a sus necesidades. Puede crear, administrar y eliminar corredores mediante la Consola de administración de AWS API REST de Amazon MQ o la AWS Command Line Interface.

En esta sección, se describen los elementos básicos de un agente de mensajes para los tipos de motor de ActiveMQ y RabbitMQ, se enumeran los tipos de instancias disponibles del agente de Amazon MQ y sus estados, y se proporciona información general de la arquitectura del agente y las opciones de configuración.

Para obtener más información sobre Amazon MQ REST APIs, consulte la referencia de la API REST de [Amazon MQ](#).

¿Qué es un agente de Amazon MQ para RabbitMQ?

Un agente es un entorno de agente de mensajes que se ejecuta en Amazon MQ. Es el componente básico de Amazon MQ. La descripción combinada de la instancia del agente clase (m7g) y tamaño (large, medium) se conoce como tipo de instancias del agente (por ejemplo, mq.m7g.large).

- Un broker de instancia única consiste en un broker en una zona de disponibilidad detrás de un Network Load Balancer (NLB). El agente se comunica con su aplicación y con un volumen de almacenamiento de Amazon EBS.
- Una implementación de clúster es una agrupación lógica de tres nodos de agente de RabbitMQ situados detrás de un balanceador de carga de red, cada uno de los cuales comparte usuarios, colas y un estado distribuido entre varias zonas de disponibilidad (AZ).

Para obtener más información, consulte [Implementación de un](#) bróker de RabbitMQ.

Puertos de agente de escucha

Los corredores RabbitMQ gestionados por Amazon MQ admiten los siguientes puertos de escucha para la conectividad a nivel de aplicación a través de. amqps También puede usar estos puertos para las conexiones de los clientes mediante la consola web RabbitMQ y la API de administración. Todas las conexiones utilizan el cifrado TLS por motivos de seguridad.

- Puerto de escucha 5671: se utiliza para las conexiones AMQP seguras realizadas a través de la URL AMQP segura. Este puerto admite los protocolos AMQP 0-9-1 y AMQP 1.0 en RabbitMQ.
- 4. Por ejemplo, si tenemos un agente con ID de agente `b-c8352341-ec91-4a78-ad9c-a43f23d325bb`, implementado en la región `us-west-2`, la URL de `amqps` completa del agente es: `b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com:5671`.
- Puertos de escucha 443 y 15671: puede utilizar ambos puertos de escucha indistintamente para acceder a un corredor a través de la consola web de RabbitMQ o la API de administración. El puerto 443 proporciona acceso HTTPS estándar, mientras que el puerto 15671 es el puerto de administración tradicional de RabbitMQ con cifrado TLS.

Atributos

Un agente de RabbitMQ tiene varios atributos:

- Un nombre. Por ejemplo, `MyBroker`.
- Un ID. Por ejemplo, `b-1234a5b6-78cd-901e-2fgh-3i45j6k17819`.
- Un nombre de recurso de Amazon (ARN). Por ejemplo, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819`.
- Una URL a la consola web de RabbitMQ. Por ejemplo, `https://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com`.

Para obtener más información, consulte el tema sobre la [consola web de RabbitMQ](#) en la documentación de RabbitMQ.

- Un punto de enlace de AMQP seguro. Por ejemplo, `amqps://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com`.

Para ver una lista completa de los atributos del agente, consulte lo siguiente en la Referencia de la API REST de Amazon MQ:

- [ID de operación de REST: Broker](#)
- [ID de operación de REST: Brokers](#)
- [ID de operación de REST: Reboot](#)

Administración de versiones del motor de Amazon MQ para RabbitMQ

RabbitMQ organiza los números de versión según la especificación semántica del control de versiones como X.Y.Z. En las implementaciones de Amazon MQ para RabbitMQ, X indica la versión principal, Y representa el número de versión secundaria y Z hace referencia al número de versión del parche. Amazon MQ considera que un cambio de versión es importante si cambian los números de versión principal. Por ejemplo, la actualización de la versión 3.13 a la 4.0 se considera una actualización de la versión principal. Un cambio de versión se considera secundario si solo cambia el número de versión secundaria o del parche. Por ejemplo, actualizar desde la versión 3.11.28 a 3.12.13 se considera una actualización de versión menor.

Amazon MQ para RabbitMQ recomienda que todos los corredores utilicen la última versión compatible de RabbitMQ 4.2. Para obtener instrucciones sobre cómo actualizar la versión del motor del agente, consulte [Actualización de una versión del motor del agente de Amazon MQ](#).

Al crear un nuevo bróker de Amazon MQ para RabbitMQ, solo necesita especificar los números de versión principal y secundaria. Por ejemplo, RabbitMQ 4.2. Si no especificas la versión del motor al crear un intermediario, Amazon MQ utilizará automáticamente la última versión del motor de forma predeterminada.

Important

Amazon MQ no admite [transmisiones](#). Crear una transmisión provocará la pérdida de datos. Amazon MQ no admite el uso de registros estructurados en JSON.

Amazon MQ admite dos versiones principales de RabbitMQ:

- [RabbitMQ 4](#)

Amazon MQ admite RabbitMQ 4.2 en la serie de versiones RabbitMQ 4 solo en el tipo de instancia mq.m7g en todos los tamaños de instancia compatibles.

- RabbitMQ 3

Amazon MQ es compatible con RabbitMQ 3.13 en la serie de versiones RabbitMQ 3 en los tipos de instancias mq.t3, mq.m5 y mq.m7g en todos los tamaños de instancia compatibles.

Lista de versiones del motor compatibles

Puede enumerar todas las versiones de motores [describe-broker-instance-options](#) AWS CLI principales y secundarias compatibles mediante el comando.

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

Para filtrar los resultados por motor y tipo de instancia, utilice las opciones `--engine-type` y `--host-instance-type`, como se muestra a continuación.

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

Por ejemplo, para filtrar los resultados de RabbitMQ y el tipo de mq.m7g.large instancia, *engine-type* sustitúyalos por y por. RABBITMQ *instance-type* mq.m7g.large

RabbitMQ 4

Amazon MQ admite RabbitMQ 4.2 en la serie de versiones RabbitMQ 4 solo en el tipo de instancia mq.m7g en todos los tamaños de instancia compatibles.

Important

Solo puede crear nuevos corredores en RabbitMQ 4.2. Actualmente, no se admiten las actualizaciones actuales de RabbitMQ 3.13.

Important

El tipo de cola predeterminado en Amazon MQ para los corredores de RabbitMQ 4.2 será «quórum». Si no se especifica ningún argumento de tipo de cola durante la creación de la cola, se creará una cola de quórum.

Recomendamos encarecidamente utilizar colas de quórum en RabbitMQ 4 por motivos de durabilidad, ya que no se garantiza que las colas clásicas sean duraderas en todos los casos.

Se han introducido los siguientes cambios en RabbitMQ 4 en Amazon MQ

- AMQP 1.0 como protocolo principal: para obtener más información, consulte [Protocolos](#).
- Palas locales: las palas ahora admiten un nuevo protocolo denominado «local», además de los protocolos AMQP 0-9-1 y AMQP 1.0. Las palas locales se basan internamente en el AMQP 1.0, pero en lugar de utilizar conexiones TCP independientes, utilizan conexiones internas entre los nodos del clúster e internas para publicar y consumir mensajes. APIs Esto solo se puede usar para consumir y publicar dentro del mismo clúster y puede ofrecer un mayor rendimiento y utilizar menos recursos que el AMQP 0-9-1 y el AMQP 1.0.
- Las colas de quórum respaldan las prioridades de los mensajes: las prioridades de los mensajes de las colas de quórum están siempre activas y no requieren una política para funcionar. Tan pronto como una cola de quórum reciba un mensaje con una prioridad establecida, se habilitará la priorización. Las colas de quórum internamente solo admiten dos prioridades: alta y normal. Los mensajes sin un conjunto de prioridades se asignarán a la normalidad, al igual que las prioridades 0 a 4. Los mensajes con una prioridad superior a 4 se asignarán a alta. Los mensajes de alta prioridad se preferirán a los mensajes de prioridad normal en una proporción de 2:1, es decir, por cada 2 mensajes de alta prioridad, la cola entregará 1 mensaje de prioridad normal (si está disponible). Por lo tanto, las colas de quórum implementan un tipo de procesamiento prioritario no estricto y de «reparto equitativo». Esto garantiza que siempre se avance en los mensajes prioritarios normales, pero se prefieren los más prioritarios en una proporción de 2:1.
- Khepri: Khepri se utiliza como almacén de metadatos predeterminado para los corredores de RabbitMQ 4
- TLS mutuo (mTLS): Amazon MQ admite el TLS mutuo (mTLS) para los corredores de RabbitMQ, lo que permite a los clientes autenticarse mediante certificados. Para obtener [más](#) información, consulte [Configuración de mTLS](#).
- Complemento de autenticación con certificado SSL: el complemento de autenticación SSL utiliza certificados de cliente de las conexiones mTLS para autenticar a los usuarios, lo que permite la autenticación mediante certificados de cliente X.509 en lugar de credenciales de nombre de usuario y contraseña. Para obtener más información, consulte [Autenticación con certificados SSL](#).
- Complemento de autenticación HTTP: el complemento backend de autenticación HTTP permite delegar la autenticación y la autorización a un servicio HTTP externo. Para obtener más información, consulta [Autenticación y autorización HTTP](#).
- Compatibilidad con JMS: [el broker ahora admite cargas de trabajo de JMS con el complemento de intercambio de temas de JMS habilitado, lo que permite que las aplicaciones de JMS se conecten mediante el cliente JMS de RabbitMQ.](#)

Las siguientes funciones han quedado obsoletas en RabbitMQ 4 en Amazon MQ

- **Duplicación de las colas clásicas:** las colas clásicas siguen siendo compatibles sin cambios importantes en las bibliotecas y aplicaciones de los clientes, pero ahora son del tipo de cola no replicada. Los clientes podrán conectarse a cualquier nodo para publicar y consumir contenido de cualquier cola clásica no replicada. Se recomiendan las colas de quórum para garantizar la replicación y la seguridad de los datos.
- **Eliminación de la QoS global:** se recomienda a los clientes configurar la QoS por consumidor (no global) en lugar de la QoS global, en la que se utiliza una única captura previa compartida para todo el canal.
- **Support para colas transitorias y no exclusivas:** las colas transitorias son colas cuya vida útil está vinculada al tiempo de actividad del nodo en el que se declaran. En un broker de instancia única, se eliminan cuando se reinicia el nodo. En una implementación de clúster, se eliminan cuando se reinicia el nodo en el que están alojados. Recomendamos utilizar el TTL de cola para eliminar automáticamente las colas inactivas y no utilizadas tras un tiempo de inactividad. Las colas exclusivas siguen siendo compatibles y se eliminan una vez que se han eliminado todas las conexiones a la cola.

Los siguientes cambios importantes pueden afectar a sus aplicaciones al actualizar a RabbitMQ 4.2 en Amazon MQ

- **Tipo de cola predeterminado:** el tipo de cola predeterminado en un bróker de RabbitMQ 4 es quórum. Si no se especifica ningún argumento de tipo de cola durante la creación de la cola, se creará una cola de quórum.
- **El límite de reenvío predeterminado en las colas de quórum está establecido en 20:** los mensajes que se reenvíen 20 veces o más se incluirán en letra muerta o se eliminarán (se eliminarán). Si lo habitual en una cola es que haya 20 entregas por mensaje, es necesario configurar un objetivo fijo o un límite superior para dichas colas a fin de evitar la pérdida de datos. La forma recomendada de hacerlo es mediante una política.
- **amqp-lib:** Las versiones amqp-lib del cliente Node JS anteriores a la 0.10.7 o cualquier biblioteca cliente de AMQP que utilice `frame_max < 8192` no podrán conectarse a RabbitMQ
- **[Límites de recursos predeterminados:](#)** Amazon MQ para RabbitMQ ha introducido límites de uso de recursos predeterminados para las conexiones, los canales, los consumidores por canal, las colas, los vhosts, las palas, los intercambios y el tamaño máximo de los mensajes. Sirven como

barreras para proteger la disponibilidad de los corredores y se pueden personalizar mediante configuraciones que se adapten a sus requisitos específicos.

Las siguientes funciones no son compatibles con RabbitMQ 4 en Amazon MQ

- Intercambios aleatorios locales: Amazon MQ no admite los intercambios aleatorios locales, ya que los nodos de Amazon MQ están detrás de un balanceador de carga de red.
- Interceptor de mensajes: los interceptores [de mensajes RabbitMQ](#) no son compatibles con Amazon MQ.
- Métricas por cola: Amazon MQ no venderá las métricas de colas de RabbitMQ para los corredores de RabbitMQ 4 a través de ellas. AWS CloudWatch Amazon MQ seguirá proporcionando métricas a nivel de corredor. AWS CloudWatch Puede consultar las métricas de colas mediante la API de administración de RabbitMQ. Recomendamos consultar las métricas de colas específicas con una frecuencia de un minuto o más.

Compatibilidad con la versión RabbitMQ

El calendario de soporte de versiones de Amazon MQ que aparece a continuación indica cuándo finalizará el soporte de una versión de Broker Engine. Cuando finaliza el soporte de una versión, Amazon MQ actualiza automáticamente todos los agentes de esta versión a la siguiente versión compatible. Esta actualización se lleva a cabo durante los períodos de mantenimiento programados por su agente, dentro de los 45 días siguientes a la end-of-support fecha.

Amazon MQ avisa con al menos 90 días de antelación antes de que finalice el soporte de una versión. Le recomendamos que actualice su bróker antes de end-of-support esa fecha para evitar cualquier interrupción. Además, no puede crear nuevos agentes en las versiones cuyo fin del soporte está programado en un plazo de 30 días a partir de la fecha de fin del soporte.

Versión de RabbitMQ	Fin del soporte en Amazon MQ
4.2 (recomendado)	
3.13	
3.12	17 de marzo de 2025

Actualizaciones de versión

Puede actualizar manualmente su agente en cualquier momento a la siguiente versión principal o secundaria compatible. Para obtener más información sobre la actualización manual de su agente, consulte [Actualización de una versión del motor de agentes de Amazon MQ](#).

Amazon MQ gestiona las actualizaciones a la última versión de parche compatible para todos los corredores de RabbitMQ que utilicen la versión 3.13 o superior. Las actualizaciones manuales y automáticas de las versiones se producen durante el período de mantenimiento programado o después de que reinicie su agente.

Important

RabbitMQ solo permite actualizaciones incrementales de versiones (por ejemplo, de 3.9.x a 3.10.x). No puede omitir las versiones secundarias al actualizar (por ejemplo, de la 3.8.x a la 3.11.x).

Los agentes de instancia única estarán desconectados mientras se reinician. En el caso de los agentes de clúster, las colas reflejadas deben sincronizarse durante el reinicio. Con colas más largas, el proceso de sincronización de colas puede tardar más. Durante el proceso de sincronización de colas, la cola no está disponible para los consumidores ni el productor. Cuando se complete el proceso de sincronización de colas, el agente volverá a estar disponible. Para reducir el impacto, se recomienda actualizar durante momentos de poco tráfico. Para obtener más información sobre las prácticas recomendadas de actualización de versiones, consulte [Prácticas recomendadas de Amazon MQ para RabbitMQ](#).

Opciones de implementación de Amazon MQ para agentes de RabbitMQ

Los agentes de RabbitMQ se pueden crear como agentes de una sola instancia o en una implementación de clúster. Para ambos modos de implementación, Amazon MQ proporciona un alto nivel de durabilidad, ya que almacenan sus datos de forma redundante.

Para acceder a sus agentes de RabbitMQ, utilice [cualquier lenguaje de programación compatible con RabbitMQ](#) y habilite TLS para los siguientes protocolos:

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

Temas

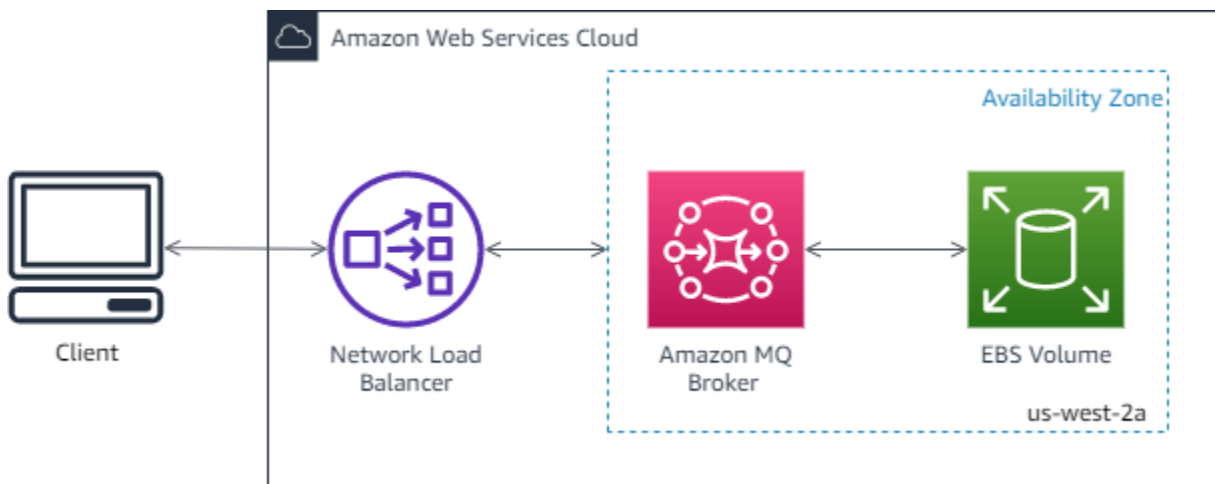
- [Opción 1: Agente de una sola instancia de Amazon MQ para RabbitMQ](#)
- [Opción 2: Implementación de un clúster de Amazon MQ para RabbitMQ](#)

Opción 1: Agente de una sola instancia de Amazon MQ para RabbitMQ

Un agente de una sola instancia está compuesto por un agente en una zona de disponibilidad situada detrás de un balanceador de carga de red (NLB). El agente se comunica con su aplicación y con un volumen de almacenamiento de Amazon EBS. Amazon EBS proporciona almacenamiento a nivel de bloque optimizado para baja latencia y alto rendimiento.

El uso de un balanceador de carga de red garantiza que el punto de enlace del agente de Amazon MQ para RabbitMQ permanezca sin cambios en el caso de que se reemplace la instancia del agente durante un período de mantenimiento o debido a errores de hardware subyacentes de Amazon EC2. Un balanceador de carga de red permite que las aplicaciones y los usuarios continúen utilizando el mismo punto de enlace para conectarse con el agente.

El siguiente diagrama ilustra un agente de una sola instancia de Amazon MQ para RabbitMQ.



Opción 2: Implementación de un clúster de Amazon MQ para RabbitMQ

Una implementación de clúster es una agrupación lógica de tres nodos de agente de RabbitMQ situados detrás de un balanceador de carga de red, cada uno de los cuales comparte usuarios, colas y un estado distribuido entre varias zonas de disponibilidad (AZ).

En una implementación de clúster, Amazon MQ administra automáticamente las políticas de agente para habilitar la replicación clásica en todos los nodos, lo que garantiza una alta disponibilidad. Cada

cola replicada consta de un nodo principal y uno o más espejos. Cada cola tiene su propio nodo principal. Todas las operaciones de una cola en particular se aplican primero en el nodo principal de la cola y, a continuación, se propagan a los espejos. Amazon MQ crea una política predeterminada del sistema que configura `ha-mode` en `all` y `ha-sync-mode` en `automatic`. Esto garantiza que los datos se repliquen en todos los nodos del clúster a través de diferentes zonas de disponibilidad para lograr mayor durabilidad.

 Note

En una implementación de clúster, si se produce una interrupción en la zona de disponibilidad, Amazon MQ intentará reubicar automáticamente los nodos de RabbitMQ afectados en una zona de disponibilidad diferente para mantener el tamaño del clúster. Una vez que se resuelva la interrupción, el clúster se reequilibrará automáticamente en todas las zonas de disponibilidad.

 Note

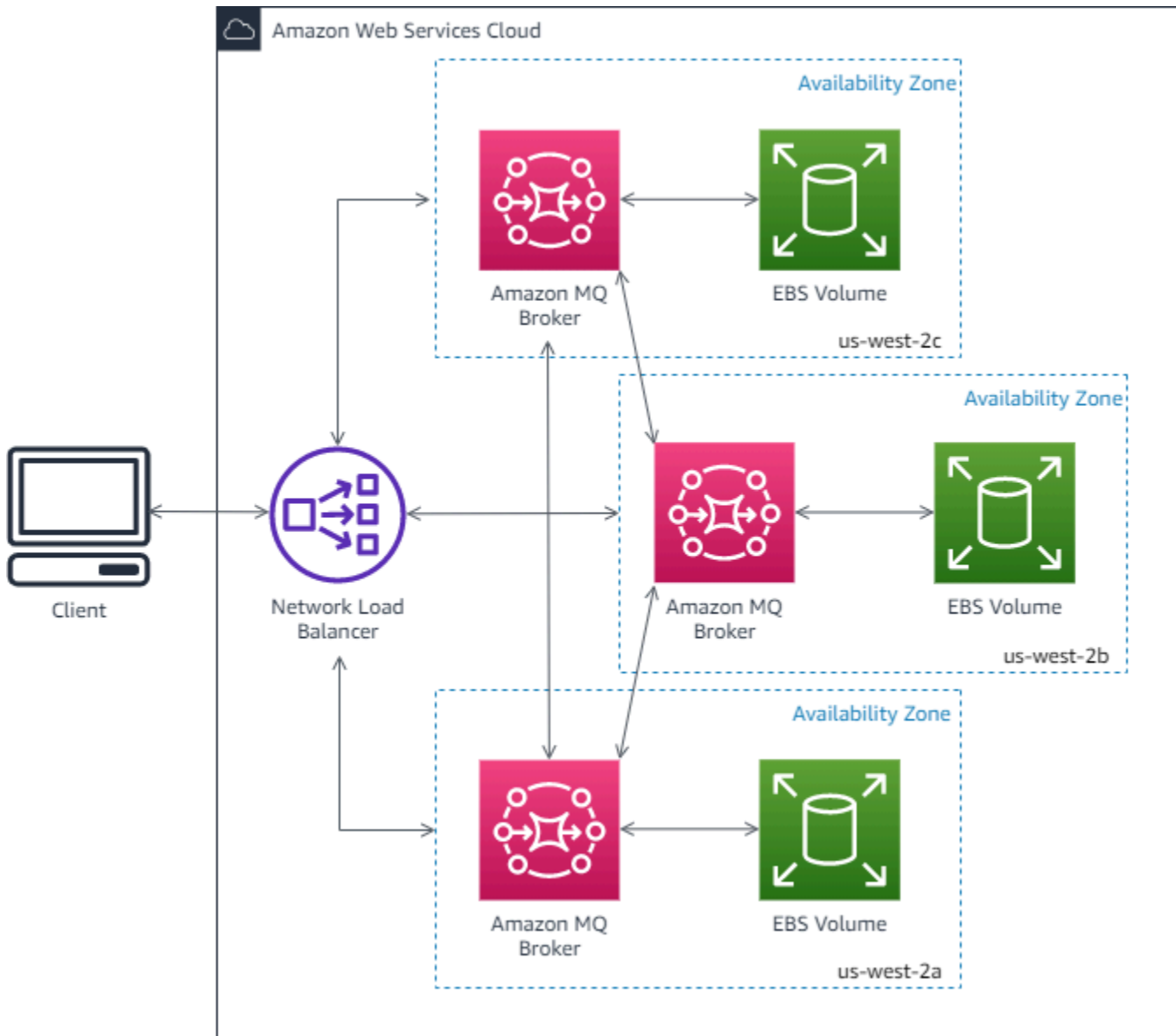
Durante un período de mantenimiento, todo el mantenimiento de un clúster se realiza de a un nodo por vez, con dos nodos como mínimo en funcionamiento en todo momento. Cada vez que se desactiva un nodo, las conexiones de cliente a ese nodo se cortan y deben restablecerse. Debe asegurarse de que el diseño del código de cliente le permita volver a conectarse automáticamente al clúster. Para obtener más información acerca de la recuperación de la conexión, consulte [the section called “Paso 1: Recuperación automática de fallas de red”](#).

Dado que Amazon MQ configura `ha-sync-mode: automatic` durante un período de mantenimiento, las colas se sincronizarán cuando cada nodo vuelva a conectarse al clúster. La sincronización de colas bloquea a todas las demás operaciones de cola. Para mitigar el impacto de la sincronización de colas durante los períodos de mantenimiento, puede mantener las colas cortas.

No se debe eliminar la política predeterminada. Si se elimina, Amazon MQ volverá a crearla automáticamente. Además, Amazon MQ garantizará que las propiedades de alta disponibilidad se apliquen a todas las demás políticas que cree en un agente agrupado. Si agrega una política sin las propiedades de alta disponibilidad, Amazon MQ las agregará por usted. Si agrega una política

con diferentes propiedades de alta disponibilidad, Amazon MQ las reemplazará. Para obtener más información acerca de la replicación clásica, consulte el tema sobre [colas replicadas clásicas](#).

El siguiente diagrama ilustra un agente de RabbitMQ con implementación de clúster con tres nodos en tres zonas de disponibilidad (AZ), cada uno con su propio volumen de Amazon EBS y un estado compartido. Amazon EBS proporciona almacenamiento a nivel de bloque optimizado para baja latencia y alto rendimiento.



Tipos de instancias del agente de Amazon MQ para RabbitMQ

La descripción combinada de la clase de instancia de broker (m7g) y el tamaño (grande, mediano) se denomina tipo de instancia de broker (por ejemplo, mq.m7g.large).

Recomendamos usar los tipos de instancia mq.m7g tanto para las implementaciones en clúster como en las de instancia única.

Amazon MQ avisa con al menos 90 días de antelación antes de que finalice el soporte de un tipo de instancia. Te recomendamos que actualices tu agente a un nuevo tipo de instancia antes de esa fecha para evitar cualquier interrupción. end-of-support

Important

No puede cambiar la categoría de un bróker de un tipo de mq.m5 instancia mq.m7g o a un tipo de mq.t3.micro instancia.

El tipo de mq.t3.micro instancia no admite la implementación de clústeres.

Tipos de instancias para la implementación de clústeres m7g

Recomendamos usar tipos de instancias mq.m7g.x con la implementación en clústeres. En la siguiente tabla, se muestran los tipos de instancias mq.m7g.x disponibles para la implementación de clústeres.

Tipo de instancia	vCPU	Memoria (GiB)	Ancho de banda basal/ráfaga de red (Gbps)	Uso recomendado	Almacenamiento	Tamaño de volumen de disco es por cada nodo (GB)
mq.m7g.medium	1	4	0.52 / 12.5	Evaluación	EBS	5
mq.m7g.large	2	8	0,937/12,5	Producción	EBS	15
mq.m7g.xlarge	4	16	1,876/12,5	Producción	EBS	25
mq.m7g.2xlarge	8	32	3,75/15,0	Producción	EBS	45

Tipo de instancia	vCPU	Memoria (GiB)	Ancho de banda basal/ráfaga de red (Gbps)	Uso recomendado	Almacenamiento	Tamaño de volumen de disco es por cada nodo (GB)
mq.m7g.4xlarge	16	64	7,5/15,0	Producción	EBS	90
mq.m7g.8xlarge	32	128	15 gigabits	Producción	EBS	175
mq.m7g.12xlarge	48	192	22,5 gigabits	Producción	EBS	260
mq.m7g.16xlarge	64	256	30 gigabits	Producción	EBS	345

Tipos de instancias para la implementación de una sola instancia de m7g

En la siguiente tabla, se muestran los tipos de instancias mq.m7g.x disponibles para la implementación de una sola instancia.

Tipo de instancia	vCPU	Memoria (GiB)	Línea base de red/ ancho de banda en ráfaga (Gbps)	Uso recomendado	Almacenamiento	Tamaño de volumen de disco es por cada nodo (GB)
mq.m7g.medium	1	4	0.52 / 12.5	Evaluación	EBS	200
mq.m7g.large	2	8	0,937/12,5	Producción	EBS	200

Tipo de instancia	vCPU	Memoria (GiB)	Línea base de red/ ancho de banda en ráfaga (Gbps)	Uso recomendado	Almacenamiento	Tamaño de volumen de disco es por cada nodo (GB)
mq.m7g.xlarge	4	16	1,876/12,5	Producción	EBS	200
mq.m7g.2xlarge	8	32	3,75/15,0	Producción	EBS	200
mq.m7g.4xlarge	16	64	7,5/15,0	Producción	EBS	200
mq.m7g.8xlarge	32	128	15 gigabits	Producción	EBS	200
mq.m7g.12xlarge	48	192	22,5 gigabits	Producción	EBS	200
mq.m7g.16xlarge	64	256	39 Gigabits	Producción	EBS	200

Tipos de instancias para la implementación de una sola instancia **mq.m5**

En las siguientes tablas, se muestran los tipos de instancias **mq.m5.x** disponibles para la implementación de una sola instancia.

Tipo de instancia	vCPU	Memoria (GiB)	Línea base de red/ ancho de banda en ráfaga (Gbps)	Uso recomendado	Almacenamiento	Tamaño de volumen de disco es por cada nodo (GB)
mq.t3.micro	2	1	0,064/5.0	Evaluación	EBS	20
mq.m5.large	2	8	0,75/10,0	Producción	EBS	200
mq.m5.xlarge	4	16	1.25/10.0	Producción	EBS	200
mq.m5.2xlarge	8	32	2.5/10.0	Producción	EBS	200
mq.m5.4xlarge	16	64	5.0/10.0	Producción	EBS	200

Tipos de instancias para la implementación de clústeres **mq.m5**

En las tablas siguientes se muestran los tipos de instancias **mq.m5.x** disponibles para la implementación en clústeres

Tipo de instancia	vCPU	Memoria (GiB)	Línea base de red/ ancho de banda en ráfaga (Gbps)	Uso recomendado	Almacenamiento	Tamaño de volumen de disco es por cada nodo (GB)
mq.m5.large	2	8	0,75/10,0	Producción	EBS	200

Tipo de instancia	vCPU	Memoria (GiB)	Línea base de red/ ancho de banda en ráfaga (Gbps)	Uso recomendado	Almacenamiento	Tamaño de volumen de disco es por cada nodo (GB)
mq.m5.xlarge	4	16	1.25/10.0	Producción	EBS	200
mq.m5.2xlarge	8	32	2.5/10.0	Producción	EBS	200
mq.m5.4xlarge	16	64	5.0/10.0	Producción	EBS	200

Directrices de dimensionamiento de Amazon MQ para RabbitMQ

Puede elegir el tipo de instancia del agente que mejor se adapte a su aplicación. Al elegir un tipo de instancia, tenga en cuenta los factores que afectarán al rendimiento del bróker:

- el número de clientes y colas
- el volumen de mensajes enviados
- mensajes guardados en la memoria
- mensajes redundantes

m7g.medium Los tipos de instancias de broker más pequeñas solo se recomiendan para probar el rendimiento de las aplicaciones. Recomendamos tipos de instancias de broker más grandes **m7g.large** y superiores o niveles de producción de clientes y colas, alto rendimiento, mensajes en memoria y mensajes redundantes.

Important

No puede degradar a un agente de un tipo de mq.m7g instancia mq.m5 o a un tipo de instancia. mq.t3.micro

Es importante poner a prueba a sus agentes para determinar el tipo y el tamaño de instancia adecuados para sus requisitos de mensajería de carga de trabajo.

Utilice siempre los límites de recursos predeterminados del broker RabbitMQ 4 para determinar el tamaño de instancia adecuado para su aplicación de acuerdo con las prácticas recomendadas de Amazon MQ. Estos límites de recursos predeterminados se basan en los tipos, tipos de m7g instancias y colas de quórum.

- [Límites de recursos predeterminados para la implementación de m7g en una sola instancia](#)
- [Límites de recursos predeterminados para la implementación del clúster m7g](#)

Puede aumentar el valor de cualquier límite hasta los valores máximos definidos por el tipo de instancia y el modo de implementación. Sin embargo, le recomendamos encarecidamente que pruebe el rendimiento del bróker con los valores aumentados antes de utilizarlo en producción.

- [Límites máximos de recursos para la implementación de m7g en una sola instancia](#)
- [Límites máximos de recursos para la implementación de clústeres m7g](#)
- [Límites máximos de recursos para la implementación de una sola instancia de m5](#)
- [Límites máximos de recursos para la implementación de clústeres m5](#)
- [Mensajes de error](#)

Note

Los corredores de RabbitMQ 3.13 no vienen con límites de recursos predeterminados, pero le recomendamos que utilice los valores predeterminados sugeridos.

Límites de recursos predeterminados

Amazon MQ para RabbitMQ admite la configuración de los límites de recursos de los corredores a partir de RabbitMQ 4. Al crear un intermediario, Amazon MQ aplica automáticamente los valores predeterminados a estos límites de recursos. Estos valores predeterminados actúan como barreras para proteger la disponibilidad de su agente y, al mismo tiempo, adaptarse a los patrones de uso habituales de los clientes. Puede personalizar el comportamiento de su corredor cambiando los valores límite de configuración para que se adapten mejor a sus requisitos específicos de carga de trabajo.

Antes de realizar cambios, tenga en cuenta lo siguiente:

 Important

1. Los cambios en la configuración pueden afectar al rendimiento y la disponibilidad del bróker
2. Comprenda el impacto antes de cambiar cualquier opción de configuración predeterminada
3. Pruebe primero los cambios de configuración en entornos que no sean de producción
4. Supervise el estado del corredor después de aplicar los cambios

 Important

Los valores predeterminados y los rangos admitidos para estas configuraciones varían según la versión de RabbitMQ, el tipo de instancia y el modo de despliegue del bróker.

 Important

Nota: Al asociar o crear un bróker con valores de configuración que no estén dentro del rango admitido, se producirá una respuesta de error.

Los límites de recursos predeterminados que se aplican a los corredores de RabbitMQ 4.2 son

- [Límites de recursos predeterminados para la implementación de m7g en una sola instancia](#)
- [Límites de recursos predeterminados para la implementación del clúster m7g](#)

Límites de recursos predeterminados

 Important

Amazon MQ para corredores de RabbitMQ 3, el valor predeterminado está configurado con el límite máximo de recursos y Amazon MQ no permite anular la configuración del límite de recursos.

Valores predeterminados para los corredores de instancia única

Tipo de instancia	Conexiones por nodo	Canales por nodo	Consumidores por canal	Colas	vhosts	Palas	Intercambios	Tamaño del mensaje en bytes
mq.m7g.nano	100	500	10	500	10	30	500	524288
mq.m7g.large	1500	4.500	10	1 000	50	50	1 000	524288
mq.m7g.xlarge	3000	9,000	10	2,000	100	100	2,000	524288
mq.m7g.2xlarge	6000	18 000	10	4.000	150	200	4.000	524288
mq.m7g.4xlarge	12 000	36.000	10	8000	200	400	8000	524288
mq.m7g.8xlarge	24.000	72.000	10	16,000	250	800	16,000	524288
mq.m7g.16xlarge	36.000	108.000	10	24.000	300	1.200	24.000	524288
mq.m7g.32xlarge	48.000	144.000	10	32 000	350	1600	32 000	524288

Valores predeterminados para los corredores de clústeres

Tipo de instancia	Conexiones por nodo	Canales por nodo	Consumidores por canal	Colas	vhosts	Palas	Intercambios	Tamaño del mensaje en bytes
mq.m7g.nano	100	300	10	100	10	10	100	524288
mq.m7g.large	500	1500	10	1 000	50	30	1 000	524288
mq.m7g.xlarge	1 000	3 000	10	2,000	100	60	2,000	524288
mq.m7g.2xlarge	2000	6000	10	4.000	150	120	4.000	524288
mq.m7g.4xlarge	4000	12 000	10	8000	200	240	8000	524288
mq.m7g.8xlarge	8000	24.000	10	16,000	250	480	16,000	524288
mq.m7g.16xlarge	12000	36.000	10	24.000	300	720	24000	524288
mq.m7g.32xlarge	16,000	48.000	10	32 000	350	960	32 000	524288

Límite máximo de recursos de Amazon MQ para RabbitMQ

Directrices de dimensionamiento para m7g con colas de cuórum para la implementación de una única instancia

La siguiente tabla indica los valores de límite máximos para cada tipo de instancia de los agentes de una única instancia.

Tipo de instancia	Conectores	Canales	Consumidores por canal	Colas	Vhosts	Palas	Intercambios	Tamaño del mensaje en bytes
mq.m7g.n dium	300	900	1 000	2.500	10	150	12500	134217728
mq.m7g.l rge	5 000	15.000	1 000	20 000	1500	250	100 000	134217728
mq.m7g.x arge	10 000	30.000	1 000	30.000	1500	500	150.000	134217728
mq.m7g.2 large	20 000	60 000	1 000	40 000	1500	1 000	200.000	134217728
mq.m7g.4 large	40 000	120 000	1 000	60 000	1500	2000	300 000	134217728
mq.m7g.8 large	80 000	240.000	1 000	80 000	1500	4000	400.000	134217728
mq.m7g.1 xlarge	120 000	360,000	1 000	100 000	1500	6000	500.000	134217728
mq.m7g.1 xlarge	160 000	480.000	1 000	120 000	1500	8000	600.000	134217728

Directrices de dimensionamiento para m7g con colas de cuórum para la implementación de clústeres

La siguiente tabla indica los valores de límite máximos para cada tipo de instancia de los agentes de clúster.

Tipo de instancia	Conexiones por nodo	Canales por nodo	Consumidores por canal	Colas	Vhosts	Palas	Intercambios	Tamaño del mensaje en bytes
mq.m7g.n dium	300	900	1 000	500	10	50	500	134217728
mq.m7g.l rge	5 000	15.000	1 000	10 000	1500	150	50 000	134217728
mq.m7g.x arge	10 000	30.000	1 000	15.000	1500	300	75 000	134217728
mq.m7g.2 large	20 000	60 000	1 000	20 000	1500	600	100 000	134217728
mq.m7g.4 large	40 000	120 000	1 000	30.000	1500	1200	150.000	134217728
mq.m7g.8 large	80 000	240.000	1 000	40 000	1500	2400	200.000	134217728
mq.m7g.1 xlarge	120 000	360,000	1 000	50 000	1500	3600	250.000	134217728
mq.m7g.1 xlarge	160 000	480.000	1 000	60 000	1500	4.800	300 000	134217728

Límites máximos de recursos para la implementación de una sola instancia M5

La siguiente tabla indica los valores de límite máximos para cada tipo de instancia de los agentes de una única instancia.

Tipo de instancia	Conectio ns	Canales	Consumido res por canal	Colas	Vhosts	Palas
m5.large	5 000	15.000	1 000	30.000	1500	250
m5.xlarge	10 000	30.000	1 000	60 000	1500	500
m5.2xlarge	20 000	60 000	1 000	120 000	1500	1 000
m5.4xlarge	40 000	120 000	1 000	240 000	1 000	2,000

Límites máximos de recursos para el despliegue de clústeres m5

La siguiente tabla indica los valores de límite máximos para cada tipo de instancia de los agentes de clúster.

Tipo de instancia	Colas	Consumidores por canal	Palas
m5.large	10 000	1 000	150
m5.xlarge	15.000	1 000	300
m5.2xlarge	20 000	1 000	600
m5.4xlarge	30.000	1 000	1200

Los siguientes límites de conexión y canal se aplican por nodo:

Tipo de instancia	Connections	Canales
m5.large	5000	15.000

Tipo de instancia	Connections	Canales
m5.xlarge	10 000	30.000
m5.2xlarge	20 000	60 000
m5.4xlarge	40 000	120 000

Los valores límite exactos para un agente de clústeres pueden ser inferiores al valor indicado en función del número de nodos disponibles y de la forma en que RabbitMQ distribuye los recursos entre los nodos disponibles. Si supera los valores límite, puede crear una nueva conexión a un nodo diferente e intentarlo de nuevo, o bien puede actualizar el tamaño de la instancia para aumentar los límites máximos.

Mensajes de error

Cuando se superan los límites, se devuelven los siguientes mensajes de error. Todos los valores se basan en los límites **m7.large** de una sola instancia.

Note

Los códigos de error de los siguientes mensajes pueden cambiar en función de la biblioteca cliente que utilice.

Conexión

```
ConnectionClosedByBroker 500 "NOT_ALLOWED - connection refused: node
connection limit (5000) is reached"
```

Canal

```
ConnectionClosedByBroker 1500 "NOT_ALLOWED - number of channels opened on
node 'rabbit@ip-10-0-23-173.us-west-2.compute.internal' has reached the
maximum allowed limit of (15,000)"
```

Consumidor

```
ConnectionClosedByBroker: (530, 'NOT_ALLOWED - reached maximum (1,000) of
consumers per channel')
```

Tamaño máximo de mensaje

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

Intercambio

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

Note

Los siguientes mensajes de error utilizan el formato de la API de administración HTTP.

Cola

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

Pala

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

Vhost

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

Valores predeterminados del agente de Amazon MQ para RabbitMQ

Cuando crea un agente de Amazon MQ para RabbitMQ, Amazon MQ aplica un conjunto predeterminado de políticas del agente y límites del vhost para optimizar el rendimiento de su agente. Amazon MQ aplica los límites del vhost únicamente al vhost (/) predeterminado. Amazon MQ no aplicará políticas predeterminadas a los vhost recién creados. Recomendamos mantener estos valores predeterminados para todos los agentes nuevos y existentes. Sin embargo, puede modificar, anular o eliminar estos valores predeterminados en cualquier momento.

Amazon MQ crea diferentes políticas de intermediación y límites de vhost para Amazon MQ para RabbitMQ 3 y RabbitMQ 4. Las diferencias se analizarán en detalle en las siguientes subsecciones.

Amazon MQ crea políticas y límites basados en el tipo de instancias y el modo de implementación del agente que elija al crear el agente. Las directivas predeterminadas se nombran de acuerdo con el modo de implementación, como se indica a continuación:

Amazon MQ para RabbitMQ 3:

- Instancia individual: `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Implementación de clústeres: `&& AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Amazon MQ para RabbitMQ 4:

- Instancia individual: `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Implementación de clústeres: `&& AWS-DEFAULT-POLICY-CLUSTER AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Para los [agentes de una sola instancia](#), Amazon MQ establece el valor de prioridad de la política en 0. Para anular el valor de prioridad predeterminado, puede crear sus propias políticas personalizadas con valores de prioridad más altos. Para [implementaciones de clúster](#), Amazon MQ establece el valor de prioridad en 1 para los valores predeterminados del agente. Para crear su propia política personalizada para clústeres, asigne un valor de prioridad mayor que 1.

Note

En las implementaciones de clúster, se requieren políticas de agente `ha-mode` y `ha-sync-mode` para replicación clásica y alta disponibilidad. Estos ajustes solo se aplican a Amazon MQ para RabbitMQ 3 y no están configurados para RabbitMQ 4.

Si elimina la política `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ` predeterminada, Amazon MQ utiliza la política `ha-all-AWS-OWNED-DO-NOT-DELETE` con un valor de prioridad de 0. Esto garantiza que las políticas `ha-mode` y `ha-sync-mode` aún estén en vigor. Si crea su propia política personalizada, Amazon MQ agrega automáticamente `ha-mode` y `ha-sync-mode` a las definiciones de sus políticas.

Temas

- [Descripciones de políticas y límites](#)
- [Valores predeterminados recomendados](#)

Descripciones de políticas y límites

En la siguiente lista se describen las políticas y los límites predeterminados que Amazon MQ aplica a un agente recién creado. Los valores de `max-length`, `max-queues`, y `max-connections` varían según el tipo de instancias y el modo de implementación de su agente. Estos valores se indican en la sección [Valores predeterminados recomendados](#).

Configuración de los corredores RabbitMQ 3 y RabbitMQ 4

- **queue-mode: lazy** (política): habilita las colas perezosas. De forma predeterminada, las colas almacenan los mensajes en una caché en memoria, lo que permite que el agente entregue mensajes a los consumidores lo más rápido posible. Esto puede ocasionar que el agente se quede sin memoria y que se dispare una alarma de memoria elevada. Las colas perezosas intentan mover los mensajes al disco lo antes posible. Esto significa que se conservan menos mensajes en la memoria en condiciones normales de funcionamiento. Con las colas perezosas, Amazon MQ para RabbitMQ puede admitir cargas de mensajería mucho más grandes y colas más largas. Tenga en cuenta que para ciertos casos de uso, los agentes con colas perezosas pueden realizar tener un rendimiento apenas más lento. Esto se debe a que los mensajes se mueven del disco al agente, en lugar de entregar mensajes desde una caché en memoria.

Modos de implementación

Una sola instancia, clúster

- **max-length: *number-of-messages*** (política): establece un límite para el número de mensajes de una cola. En las implementaciones de clúster, el límite impide que la sincronización de colas se pause en casos como reinicios del agente o después de un período de mantenimiento.

Modos de implementación

Clúster

- **overflow: reject-publish** (policy): hace cumplir las colas con una política `max-length` para rechazar nuevos mensajes después de que el número de mensajes de la cola alcanza el valor `max-length`. A fin de garantizar que los mensajes no se pierdan si una cola está en estado de desbordamiento, las aplicaciones cliente que publiquen mensajes en el agente deben implementar las [confirmaciones del publicador](#). Para obtener información acerca de cómo

implementar las confirmaciones del publicador, consulte [Publisher Confirms](#) (Confirmaciones del publicador) en el sitio web de RabbitMQ.

 Modos de implementación
Clúster

Ajustes específicos de RabbitMQ 3

- **max-queues:** *number-of-queues-per-vhost* (límite del vhost): establece el límite para el número de colas en un agente. Al igual que la definición de la política max-length, al limitar el número de colas en implementaciones de clúster, se evita que la sincronización de colas se pause después de reinicios del agente o períodos de mantenimiento. Limitar las colas también evita el uso excesivo de CPU para mantener las colas.

 Modos de implementación
Una sola instancia, clúster

- **max-connections:** *number-of-connections-per-vhost* (límite el vhost): establece el límite para el número de conexiones de cliente con el agente. Al limitar la cantidad de conexiones de acuerdo con los valores recomendados, se evita el uso excesivo de la memoria del agente, en cuyo caso el agente podría generar una alarma de memoria elevada y se podrían pausar las operaciones.

 Modos de implementación
Una sola instancia, clúster

Valores predeterminados recomendados

 **Important**
max-queues y solo max-connections se aplican a Amazon MQ para RabbitMQ 3.

Note

Los límites `max-length` y `max-queue` predeterminados se prueban y evalúan en función de un tamaño promedio de mensaje de 5 kB. Si sus mensajes pesan mucho más que 5 kB, deberá ajustar y reducir los límites `max-length` y `max-queue`.

En esta tabla se muestran los valores límite predeterminados de un agente recién creado. Amazon MQ aplica estos valores según el tipo de instancias y el modo de implementación del agente.

Tipo de instancia	Modo de implementación	<code>max-length</code>	<code>max-queues</code>	<code>max-connections</code>
mq.m7g.medium	Instancia individual	N/A	2.500	100
	Clúster	500.000	100	100
mq.m7g.large	Instancia individual	N/A	20 000	5 000
	Clúster	8.000.000	10 000	5 000
mq.m7g.xlarge	Instancia individual	N/A	30.000	10 000
	Clúster	9.000.000	15.000	10 000
mq.m7g.2xlarge	Instancia individual	N/A	40 000	20 000
	Clúster	10 000 000	40 000	20 000
mq.m7g.4xlarge	Instancia individual	N/A	60 000	40 000
	Clúster	12.000.000	30.000	40 000

Tipo de instancia	Modo de implementación	max-length	max-queues	max-connections
mq.m7g.8xlarge	Instancia individual	N/A	80 000	80 000
	Clúster	20.000.000	40 000	80 000
mq.m7g.12xlarge	Instancia individual	N/A	100 000	120 000
	Clúster	30 000 000	20 000	120 000
mq.m7g.16xlarge	Instancia individual	N/A	120 000	160 000
	Clúster	40 000 000	50 000	160 000

Tipo de instancia	Modo de implementación	max-length	max-queues	max-connections
t3.micro	Instancia individual	N/A	500	500
m5.large	Instancia individual	N/A	20 000	4.000
m5.large	Clúster	8.000.000	10 000	15.000
m5.xlarge	Instancia individual	N/A	30.000	8000
m5.xlarge	Clúster	9.000.000	10 000	20 000
m5.2xlarge	Instancia individual	N/A	60 000	15.000
m5.2xlarge	Clúster	10 000 000	10 000	40 000

Tipo de instancia	Modo de implementación	max-length	max-queues	max-connections
m5.4xlarge	Instancia individual	N/A	150.000	30.000
m5.4xlarge	Clúster	12.000.000	10 000	100 000

Configuración de un bróker RabbitMQ

Una configuración contiene todos los ajustes de su agente RabbitMQ en formato Cuttlefish. Puede crear una configuración antes de crear los agente. Luego, puede aplicar la configuración a uno o más agentes.

Atributos

La configuración de un agente tiene varios atributos, por ejemplo:

- Un nombre (MyConfiguration)
- Un identificador (c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9)
- Un nombre de recurso de Amazon (ARN) (arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b678cd-901e-2fgh-3i45j6k178l9)

Para ver una lista completa de los atributos de configuración, consulte lo siguiente en la Referencia de la API REST de Amazon MQ:

- [ID de operación de REST: Configuration](#)
- [ID de operación de REST: Configurations](#)

Para ver una lista completa de los atributos de revisión de la configuración, consulte lo siguiente:

- [ID de operación de REST: Configuration Revision](#)
- [ID de operación de REST: Configuration Revisions](#)

Temas

- [Creación y aplicación de configuraciones de bróker de RabbitMQ](#)
- [Edición de una revisión de la configuración de Amazon MQ para RabbitMQ](#)
- [Valores configurables para RabbitMQ en Amazon MQ](#)
- [Soporte de ARN en la configuración de RabbitMQ](#)

Creación y aplicación de configuraciones de agente de ActiveMQ

Una configuración contiene todos los ajustes del agente de RabbitMQ en formato Cuttlefish. Puede crear una configuración antes de crear los agente. A continuación, puede aplicar la configuración a uno o más agentes

En el ejemplo siguiente, se muestra cómo crear y aplicar la configuración de un agente de RabbitMQ mediante la Consola de administración de AWS.

Important

Solo puede eliminar una configuración mediante la API `DeleteConfiguration`. Para obtener más información, consulte [Configuraciones](#) en la Referencia de la API de Amazon MQ.

Crear una nueva configuración

Para aplicar una configuración a su agente, primero debe crear dicha configuración.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En el lado izquierdo, expanda el panel de navegación y elija Configurations (Configuraciones).

Amazon MQ 

Brokers

Configurations

3. En la página Configurations (Configuraciones), seleccione Create configuration (Crear configuración).
4. En la página Create configuration (Crear configuración), en la sección Details (Detalles), escriba el nombre de la configuración en Configuration name (Nombre de configuración) (por ejemplo,

MyConfiguration) y seleccione una versión del motor del agente en Broker engine (Motor del agente).

Para obtener más información acerca de las versiones de motor de RabbitMQ compatibles con Amazon MQ para RabbitMQ, consulte [the section called “Administración de versiones”](#).

5. Seleccione Crear configuración.

Crear una revisión de configuración nueva

Tras crear una configuración, tendrá que editarla mediante una revisión de la configuración.

1. En la lista de configuraciones, elija. **MyConfiguration**

Note

La primera revisión de la configuración siempre se genera automáticamente cuando Amazon MQ crea la configuración.

En la **MyConfiguration** página, se muestran el tipo y la versión del motor de broker que utiliza la nueva revisión de configuración (por ejemplo, RabbitMQ 3.xx.xx).

2. En la pestaña Detalles de configuración, se muestra el número de revisión de la configuración, la descripción y la configuración del agente en formato Cuttlefish.

Note

Cuando se edita la configuración actual, se crea una nueva revisión de configuración.

3. Elija Editar configuración y realice cambios en la configuración de Cuttlefish.
4. Seleccione Save.

Se muestra el cuadro de diálogo Save revision (Guardar revisión).

5. (Opcional) Escriba A description of the changes in this revision.
6. Seleccione Save.

De este modo, guardará la nueva revisión de la configuración.

⚠ Important

Si realiza cambios en una configuración, los cambios no se aplican al agente de inmediato. Para aplicar los cambios, debe esperar al próximo período de mantenimiento o [reiniciar el agente](#).

En la actualidad, no se puede eliminar una configuración.

Aplicar una revisión de configuración al agente

Tras crear la revisión de la configuración, puede aplicarla a su agente.

1. En el lado izquierdo, expanda el panel de navegación y elija Brokers (Agentes).

Amazon MQ ×

Brokers

Configurations

2. En la lista de corredores, seleccione su corredor (por ejemplo) y, a continuación, MyBroker elija Editar.
3. En la **MyBroker** página de edición, en la sección Configuración, seleccione una configuración y una revisión y, a continuación, elija Programar modificaciones.
4. En la sección Schedule broker modifications (Programar modificaciones del agente), elija During the next scheduled maintenance window (Durante el siguiente periodo de mantenimiento programado) si desea aplicar las modificaciones durante el siguiente periodo de mantenimiento programado o Immediately (Inmediatamente) si desea aplicarlas inmediatamente.

⚠ Important

Los agentes de instancia única están desconectados mientras se reinician. En el caso de los agentes de clústeres, solo un nodo está inactivo a la vez mientras el agente se reinicia.

5. Seleccione Aplicar.

La revisión de la configuración se aplicará al broker en el momento especificado.

Edición de una revisión de configuración de Amazon MQ para RabbitMQ

En las siguientes instrucciones se describe cómo editar una revisión de configuración para su agente.

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, seleccione su corredor (por ejemplo MyBroker) y, a continuación, elija Editar.
3. En la **MyBroker** página, elija Editar.
4. En la **MyBroker** página de edición, en la sección Configuración, seleccione una configuración y una revisión y, a continuación, elija Editar.

Note

A menos que seleccione una configuración cuando crea un agente, la primera revisión de la configuración siempre se crea automáticamente cuando Amazon MQ crea el agente.

En la **MyBroker** página, se muestran el tipo y la versión del motor del broker que utiliza la configuración (por ejemplo, RabbitMQ 3.xx.xx).

5. En la pestaña Detalles de configuración, se muestra el número de revisión de la configuración, la descripción y la configuración del agente en formato Cuttlefish.

Note

Cuando se edita la configuración actual, se crea una nueva revisión de configuración.

6. Elija Editar configuración y realice cambios en la configuración de Cuttlefish.
7. Seleccione Save.

Se muestra el cuadro de diálogo Save revision (Guardar revisión).

8. (Opcional) Escriba `A description of the changes in this revision.`
9. Seleccione Save.

De este modo, guardará la nueva revisión de la configuración.

⚠ Important

Si realiza cambios en una configuración, los cambios no se aplican al agente de inmediato. Para aplicar los cambios, debe esperar al próximo período de mantenimiento o [reiniciar el agente](#).

En la actualidad, no se puede eliminar una configuración.

Valores configurables

Puede establecer el valor de las siguientes opciones de configuración del agente modificando su archivo de configuración en la Consola de administración de AWS.

Además de los valores descritos en la tabla siguiente, Amazon MQ admite opciones de configuración de bróker adicionales relacionadas con la autenticación y la autorización, así como con los límites de recursos. Para obtener más información sobre estas opciones de configuración, consulte

- [OAuth Configuración 2.0](#)
- [Configuración LDAP](#)
- [Configuración HTTP](#)
- [Configuración de SSL](#)
- [Configuración mTLS](#)
- [Soporte de ARN](#)
- [Límites de recursos](#)
- [Configuración SSL del cliente AMQP](#)

Configuración	Valor predeterminado	Valores recomendados	Valores	Versiones aplicables	Description (Descripción)
consumer_timeout	1 800 000 ms (30 minutos)	1 800 000 ms (30 minutos)	De 0 a 2.147.483.647 ms. Amazon MQ también	Todas las versiones	Un tiempo de espera en el acuse de recibo de entrega al

Configuración	Valor predeterminado	Valores recomendados	Valores	Versiones aplicables	Description (Descripción)
			admite el valor 0, que significa «infinito».		consumidor para detectar cuándo los consumidores no confirman las entregas.
latido	60 segundos	60 segundos	60 a 3600 segundos	Todas las versiones	Define el tiempo que debe transcurrir hasta que RabbitMQ considere que una conexión no está disponible.

Configuración	Valor predeterminado	Valores recomendados	Valores	Versiones aplicables	Description (Descripción)
management.restrictions.operator_policy_changes.disabled	true	true	true, false	Todas las versiones	Desactiva la realización de cambios en las políticas del operador. Si realiza este cambio, le recomendamos encarecidamente que incluya las propiedades de HA en las políticas de operador propias.
quorum_queue.property_equality_relaxed_checks_on_redeclaration	true	true	true, false	Todas las versiones	Si se establece en TRUE, la aplicación evita una excepción de canal al volver a declarar una cola de cuórum.

Configuración	Valor predeterminado	Valores recomendados	Valores	Versiones aplicables	Description (Descripción)
secure.management.http.headers.enabled	true	true	true, false	Todas las versiones	Activa los encabezados de seguridad HTTP no modificables.

Configuración de acuse de recibo de entrega a los consumidores

Puedes configurar `consumer_timeout` para que detecte cuándo los consumidores no emban las entregas. Si el consumidor no envía un acuse de recibo dentro del tiempo de espera, el canal se cerrará. Por ejemplo, si utiliza el valor predeterminado de 1 800 000 milisegundos, si el consumidor no envía un acuse de recibo de entrega en ese plazo, el canal se cerrará. Amazon MQ también admite el valor 0, que significa «infinito».

Configuración de latidos

Es posible configurar un tiempo de espera de latido para saber si las conexiones se han interrumpido o si han fallado. El valor de latido define el límite de tiempo antes de que se considere que una conexión está inactiva.

Configuración de políticas del operador

La política de operador predeterminada de cada host virtual tiene las siguientes propiedades de HA recomendadas:

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

Los cambios en las políticas del operador a través de la API Consola de administración de AWS o de administración no están disponibles de forma predeterminada. Puede habilitar los cambios agregando la siguiente línea a la configuración del agente:

```
management.restrictions.operator_policy_changes.disabled=false
```

Si realiza este cambio, le recomendamos encarecidamente que incluya las propiedades de HA en las políticas de operador propias.

Configuración de comprobaciones relajadas en la declaración de colas

Si ha migrado sus colas clásicas a colas de quórum pero no ha actualizado su código de cliente, puede evitar una excepción de canal al volver a declarar una cola de quórum configurando `quorum_queue.property_equivalence.relaxed_checks_on_redeclaration` con el valor `true`.

Configuración de encabezados de seguridad HTTP

La configuración `secure.management.http.headers.enabled` habilita los siguientes encabezados de seguridad HTTP:

- [X-Content-Type-Options: nosniff](#): impide que los navegadores rastreen el contenido y que se utilicen algoritmos para deducir el formato de archivo de los sitios web.
- [X-Frame-Options: DENY](#): impide que otros usuarios incrusten el complemento de administración en un marco de su propio sitio web para engañar a otros.
- [Strict-Transport-Security: max-age=47304000; includeSubDomains](#): obliga a los navegadores a utilizar HTTPS al realizar conexiones posteriores al sitio web y sus subdominios durante un período prolongado (1,5 años).

Los corredores de Amazon MQ para RabbitMQ creados en las versiones 3.10 y posteriores tendrán `secure.management.http.headers.enabled` establecido en `true` de forma predeterminada. Puede activar estos encabezados de seguridad HTTP configurando `secure.management.http.headers.enabled` en `true`. Si desea excluirte de estos encabezados de seguridad HTTP, establece `secure.management.http.headers.enabled` en `false`.

Configuración de la autenticación y la autorización 2.0 OAuth

Para obtener información sobre las opciones de configuración OAuth 2.0 y cómo configurar la autenticación OAuth 2.0 para sus corredores, consulte [Configuraciones OAuth 2.0 compatibles](#) y [Uso de la autenticación y autorización OAuth 2.0](#).

Configuración de la autenticación y la autorización de LDAP

Para obtener información sobre las opciones de configuración de LDAP y la configuración de la autenticación LDAP para sus intermediarios, consulte [Configuraciones de LDAP compatibles y. Uso de la autenticación y la autorización de LDAP](#)

Configuración de la autenticación y autorización HTTP

Para obtener información sobre los valores de configuración de la autenticación HTTP y cómo configurar la autenticación HTTP para sus agentes, consulte [Autenticación y autorización HTTP y. Uso de la autenticación y la autorización HTTP](#)

Note

El complemento de autenticación HTTP solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

Configuración de la autenticación con certificado SSL

Para obtener información sobre los valores de configuración de la autenticación con certificados SSL y cómo configurar la autenticación con certificados SSL para sus corredores, consulte [Autenticación con certificados SSL y. Uso de la autenticación con certificado SSL](#)

Note

El complemento de autenticación con certificado SSL solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

Configuración de mTLS

Amazon MQ para RabbitMQ admite el TLS mutuo (mTLS) para conexiones seguras a varios puntos de enlace y servicios externos. El mTLS proporciona una seguridad mejorada al requerir que tanto el cliente como el servidor se autenticuen mediante certificados.

 Note

El uso de autoridades de certificación privadas para los mTLS solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

 Important

Amazon MQ para RabbitMQ impone el uso de archivos de AWS ARNs certificados y claves privadas. Consulte la [compatibilidad con ARN en la configuración de RabbitMQ](#) para obtener más detalles.

En esta página

- [Punto final AMQP](#)
- [Complemento de administración RabbitMQ](#)
- [Complemento RabbitMQ 2.0 OAuth](#)
- [Complemento de autenticación HTTP RabbitMQ](#)
- [Complemento LDAP RabbitMQ](#)
- [Conexiones de cliente AMQP](#)

Punto final AMQP

Configure los mTLS para las conexiones de los clientes al punto final AMQP. Esto se usa con la autenticación con certificado SSL. Para ver las configuraciones compatibles, consulte [Autenticación con certificado SSL](#).

Complemento de administración RabbitMQ

Configure los mTLS para las conexiones a la interfaz de administración de RabbitMQ.

 Note

La API de administración no admite mTLS estrictos.

Configuraciones admitidas

`aws.arns.management.ssl.cacertfile`

Archivo de autoridad de certificación para validar los certificados de los clientes que se conectan a la interfaz de administración.

`management.ssl.verify`

Modo de verificación por pares. Valores admitidos: `verify_none`, `verify_peer`

`management.ssl.depth`

Profundidad máxima de la cadena de certificados para la verificación.

`management.ssl.hostname_verification`

Modo de verificación del nombre de host. Valores admitidos: `wildcard`, `none`

Opciones de SSL no compatibles

No se admiten los siguientes valores de configuración de SSL:

Ver lista completa

- `management.ssl.cert`
- `management.ssl.client_renegotiation`
- `management.ssl.dh`
- `management.ssl.dhfile`
- `management.ssl.fail_if_no_peer_cert`
- `management.ssl.honor_cipher_order`
- `management.ssl.honor_ecc_order`
- `management.ssl.key.RSAPrivateKey`
- `management.ssl.key.DSAPrivateKey`
- `management.ssl.key.PrivateKeyInfo`
- `management.ssl.log_alert`
- `management.ssl.password`
- `management.ssl.psk_identity`

- `management.ssl.reuse_sessions`
- `management.ssl.secure_renegotiate`
- `management.ssl.versions.$version`
- `management.ssl.sni`

Complemento RabbitMQ 2.0 OAuth

Configure los mTLS para las conexiones de Amazon MQ al proveedor OAuth de identidades 2.0. Para ver las configuraciones compatibles, consulte [OAuth Autenticación y autorización 2.0](#)

Complemento de autenticación HTTP RabbitMQ

Configure los mTLS para las conexiones desde Amazon MQ al servidor de autenticación HTTP. Para ver las configuraciones compatibles, consulte [Autenticación y autorización HTTP](#)

Complemento LDAP RabbitMQ

Configure los mTLS para las conexiones desde Amazon MQ al servidor LDAP. Para ver las configuraciones compatibles, consulte [Autenticación y autorización LDAP](#)

Conexiones de cliente AMQP

Configure la verificación por pares de TLS para las conexiones de clientes AMQP utilizadas por federation y shovel. Para obtener más información, consulte Configuración SSL del cliente [AMQP](#).

Important

Amazon MQ no admite actualmente la configuración de certificados de cliente para las conexiones de clientes AMQP. Como resultado, Federation y Shovel no pueden conectarse a intermediarios compatibles con MTLS que requieren la autenticación con certificados de cliente.

Configuración de límite de recursos

Amazon MQ para RabbitMQ admite la configuración de los límites de recursos de los corredores a partir de RabbitMQ 4. Al crear un intermediario, Amazon MQ aplica automáticamente los valores predeterminados a estos límites de recursos. Estos valores predeterminados actúan como barreras para proteger la disponibilidad de su agente y, al mismo tiempo, adaptarse a los patrones de uso

habituales de los clientes. Puede personalizar el comportamiento de su corredor cambiando los valores límite de configuración para que se adapten mejor a sus requisitos específicos de carga de trabajo. Para obtener más información sobre los valores predeterminados y máximos permitidos, consulte [the section called “Directrices de dimensionamiento”](#).

Nombres de recursos y claves de configuración

Nombre del recurso	Clave de configuración
Connection	<code>connection_max</code>
Canal	<code>channel_max_per_node</code>
Cola	<code>cluster_queue_limit</code>
Vhost	<code>vhost_max</code>
Pala	<code>runtime_parameters.limits.shovel</code>
Exchange	<code>cluster_exchange_limit</code>
Consumidor por canal	<code>consumer_max_per_channel</code>
Tamaño máximo de mensaje	<code>max_message_size</code>

¿Cómo anular los límites de recursos

Puede anular los límites de recursos mediante la API de Amazon MQ y la consola Amazon MQ.

El siguiente ejemplo muestra cómo anular el límite predeterminado del recuento de colas mediante: AWS CLI

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

Una invocación correcta crea una revisión de la configuración. Debe asociar la configuración a su corredor RabbitMQ y reiniciar el corredor para aplicar la anulación. Para obtener más información, consulte [RabbitMQ Broker Configurations](#)

Errores de anulación del límite de recursos

Al asociar o crear un corredor con valores de configuración fuera del rango admitido, se produce una respuesta de error similar a la siguiente:

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

Soporte de ARN en la configuración de RabbitMQ

Amazon MQ para RabbitMQ admite AWS ARNs los valores de algunos ajustes de configuración de RabbitMQ. [Esto lo habilita el complemento comunitario rabbitmq-aws de RabbitMQ](#). Amazon MQ ha desarrollado y mantenido este complemento y también se puede utilizar en corredores de RabbitMQ autohospedados que no estén gestionados por Amazon MQ.

Consideraciones importantes

- Los valores de ARN resueltos recuperados por el complemento aws se pasan directamente al proceso RabbitMQ en tiempo de ejecución. No se almacenan en ningún otro lugar del nodo RabbitMQ.
- Amazon MQ para RabbitMQ requiere una función de IAM que Amazon MQ pueda asumir para acceder a la configuración. ARNs Esto se configura mediante un ajuste. `aws.arns.assume_role_arn`
- Los usuarios que CreateBroker utilicen o UpdateBroker APIs tengan una configuración de intermediario que incluya una función de IAM deben tener el `iam:PassRole` permiso para esa función.
- El rol de IAM debe existir en la misma AWS cuenta que el bróker de RabbitMQ. Todos los ARNs elementos de la configuración deben estar presentes en la misma AWS región que el bróker RabbitMQ.
- Amazon MQ añade claves condicionales globales de IAM `aws:SourceAccount` y `aws:SourceArn` cuando asume la función de IAM. [Estos valores deben utilizarse en la política de IAM asociada a la función de protección de diputado confusa](#).

En esta página

- [Claves compatibles](#)
- [Ejemplos de políticas de IAM](#)
- [Validación de acceso](#)
- [Estados de cuarentena relacionados con los corredores](#)
- [Escenario de ejemplo](#)

Claves compatibles

Función de IAM obligatoria

`aws.arns.assume_role_arn`

Rol de IAM (ARN) que Amazon MQ asume para acceder a otros recursos. AWS Se requiere cuando se utiliza cualquier otra configuración de ARN.

Punto final AMQP

Clave de configuración	Description (Descripción)
<code>aws.arns.ssl_options.cacertfile</code>	Archivo de autoridad de certificación para las conexiones de los SSL/TLS clientes. Amazon MQ requiere usar Amazon S3 o almacenar el certificado.

Complemento de administración RabbitMQ

Clave de configuración	Description (Descripción)
<code>aws.arns.management.ssl.cacertfile</code>	Archivo de autoridad de certificación para las conexiones de la interfaz de administración. SSL/TLS Amazon MQ requiere usar Amazon S3 o almacenar el certificado.

Complemento RabbitMQ 2.0 OAuth

Clave de configuración	Description (Descripción)
<code>aws.arns.auth_oauth2.https.cacertfile</code>	Archivo de autoridad de certificación para conexiones HTTPS OAuth 2.0. Amazon MQ requiere usar Amazon S3 o almacenar el certificado.

Complemento de autenticación HTTP RabbitMQ

Clave de configuración	Description (Descripción)
<code>aws.arns.auth_http.ssl_options.cacertfile</code>	Archivo de autoridad de certificación para conexiones de autenticación HTTP. SSL/TLS Amazon MQ requiere usar Amazon S3 o almacenar el certificado.
<code>aws.arns.auth_http.ssl_options.certfile</code>	Archivo de certificado para conexiones TLS mutuas entre Amazon MQ y el servidor de autenticación HTTP. Amazon MQ requiere usar Amazon S3 o almacenar el certificado.
<code>aws.arns.auth_http.ssl_options.keyfile</code>	Archivo de clave privada para conexiones TLS mutuas entre Amazon MQ y el servidor de autenticación HTTP. Amazon MQ requiere usarlo AWS Secrets Manager para almacenar la clave privada.

Complemento LDAP RabbitMQ

Clave de configuración	Description (Descripción)
<code>aws.arns.auth_ldap.ssl_options.cacertfile</code>	Archivo de autoridad de certificación para conexiones LDAP. SSL/TLS Amazon MQ requiere usar Amazon S3 o almacenar el certificado.
<code>aws.arns.auth_ldap.ssl_options.certfile</code>	Archivo de certificado para conexiones TLS mutuas entre Amazon MQ y el servidor LDAP. Amazon MQ requiere usar Amazon S3 o almacenar el certificado.

Clave de configuración	Description (Descripción)
<code>aws.arns.auth_ldap. .ssl_options.keyfile</code>	Archivo de clave privada para conexiones TLS mutuas entre Amazon MQ y el servidor LDAP. Amazon MQ requiere usarlo AWS Secrets Manager para almacenar la clave privada.
<code>aws.arns.auth_ldap. .dn_lookup_bind.password</code>	Contraseña para el enlace de búsqueda de DN de LDAP. Amazon MQ requiere almacenar la contraseña como un valor de texto sin formato. AWS Secrets Manager
<code>aws.arns.auth_ldap. .other_bind.password</code>	Contraseña para otro enlace de LDAP. Amazon MQ requiere almacenar la contraseña como un valor de texto sin formato. AWS Secrets Manager

Ejemplos de políticas de IAM

Para ver ejemplos de políticas de IAM, incluidos documentos de políticas de asunción de roles y documentos de políticas de roles, consulte el ejemplo de implementación del [CDK](#).

Consulte [Uso de la autenticación y la autorización de LDAP](#) los pasos sobre cómo configurar AWS Secrets Manager y los recursos de Amazon S3.

Validación de acceso

Para solucionar problemas en los que no se pueden recuperar los valores del ARN, el complemento `aws` admite [un punto final de la API de administración de RabbitMQ](#) al que se puede llamar para comprobar si Amazon MQ es capaz de asumir la función y resolverla correctamente. AWS ARNs Esto evita la necesidad de actualizar la configuración del broker, actualizar el broker con la nueva revisión de la configuración y reiniciar el broker para probar los cambios de configuración.

Note

El uso de esta API requiere un usuario administrador de RabbitMQ existente. Amazon MQ recomienda crear corredores de prueba con un usuario interno además de otros métodos de acceso. Consulte [Habilitar la autenticación OAuth 2.0 y la autenticación simple \(interna\)](#). A continuación, se puede utilizar este usuario para acceder a la API de validación.

Note

Aunque el complemento aws admite la transferencia de un nuevo rol como entrada a la API de validación, Amazon MQ no admite este parámetro. El rol de IAM utilizado para la validación debe coincidir con el valor de la configuración del `aws.arns.assume_role_arn` bróker.

Estados de cuarentena relacionados con los corredores

Para obtener información sobre los estados de cuarentena de los corredores relacionados con problemas de soporte de ARN, consulte:

- [RABBITMQ_INVALID_ASSUMEROLE](#)
- [RABBITMQ_INVALID_ARN_LDAP](#)
- [RABBITMQ_INVALID_ARN](#)

Escenario de ejemplo

- El corredor `b-f0fc695e-2f9c-486b-845a-988023a3e55b` ha configurado para usar la función de IAM para acceder al secreto `<role>` AWS Secrets Manager `<arn>`
- Si la función proporcionada a Amazon MQ no tiene permiso de lectura sobre el AWS Secrets Manager secreto, aparecerá el siguiente error en los registros de RabbitMQ:

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

Además, el agente entrará en estado de cuarentena. `INVALID_ASSUMEROLE` Para obtener más información, consulte [INVALID_ASSUMEROLE](#).

- Los intentos de autenticación LDAP fallarán y generarán el siguiente error:

```
[error] <0.254.0> LDAP bind failed: invalid_credentials
```

- Corrija la función de IAM con los permisos adecuados
- Llama al punto final de validación para comprobar si RabbitMQ ahora puede acceder al secreto:

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

Configuración SSL del cliente AMQP

Federation y Shovel utilizan AMQP para la comunicación entre intermediarios ascendentes y descendentes. De forma predeterminada, la verificación por pares de TLS está habilitada para los clientes AMQP en Amazon MQ for RabbitMQ 4. Con esta configuración, los clientes AMQP federados y simples que utilicen corredores de Amazon MQ realizarán una verificación por pares al establecer conexiones con un corredor principal.

Los clientes de AMQP que se ejecutan en los corredores de Amazon MQ admiten las mismas autoridades de certificación que Mozilla. Si no utilizas [ACM](#), utiliza un certificado emitido por una CA de la lista de certificados de CA [incluidos en Mozilla](#). Si su agente local utiliza certificados de otras autoridades de certificación, la verificación de SSL fallará. Puede deshabilitar la verificación por pares de TLS para estos casos de uso.

Important

Amazon MQ no admite actualmente la configuración de certificados de cliente para las conexiones de clientes AMQP. Como resultado, Federation y Shovel no pueden conectarse a intermediarios compatibles con MTLS que requieren la autenticación con certificados de cliente.

Important

En Amazon MQ para RabbitMQ 3, las propiedades SSL de los clientes AMQP se configuran con los valores predeterminados de RabbitMQ (`verify_none`). Amazon MQ para RabbitMQ 3 no admite la anulación de estos valores predeterminados.

Note

Con la `verify_peer` configuración predeterminada, puede establecer conexiones de federación y de control entre dos corredores de Amazon MQ, pero esto no permite establecer la conexión entre el corredor de Amazon MQ y los corredores privados o los corredores locales que utilizan certificados de CA que no son de Amazon MQ. Para conectarte con agentes privados o locales, debes deshabilitar la verificación por pares en el agente intermediario de Amazon MQ.

Clave de configuración SSL del cliente AMQP

Configuración	Clave de configuración	Valores admitidos
Verificación por pares SSL del cliente AMQP	<code>amqp_client.ssl_options.verify</code>	<code>verify_none</code> , <code>verify_peer</code>

¿Cómo anular la verificación por pares SSL del cliente AMQP

Puedes anular la verificación por pares SSL del cliente AMQP mediante la API de Amazon MQ y la consola Amazon MQ en los corredores de RabbitMQ 4.

El siguiente ejemplo muestra cómo anular la verificación por pares SSL del cliente AMQP mediante: AWS CLI

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

Una invocación correcta crea una revisión de la configuración. Debe asociar la configuración a su corredor RabbitMQ y reiniciar el corredor para aplicar la anulación. Para obtener más información, consulte [Creating and applying broker configurations](#)

⚠ Important

Cuando se usa `verify_none`, el cifrado SSL sigue activo, pero no se verifica la identidad del par. Use esta configuración solo cuando sea necesario y asegúrese de confiar en la ruta de red al agente de destino.

Autenticación y autorización de Amazon MQ para RabbitMQ

Amazon MQ para RabbitMQ admite los siguientes métodos de autenticación y autorización:

Autenticación y autorización simple

En este método, los usuarios del agente se almacenan internamente en el agente de RabbitMQ y se administran a través de la consola web o la API de administración. Los permisos para vhosts, intercambios, colas y temas se configuran directamente en RabbitMQ. Este es el método predeterminado. Para obtener más información, consulte Autenticación y autorización [sencillas](#).

OAuth Autenticación y autorización 2.0

En este método, los usuarios intermediarios y sus permisos los administra un proveedor de identidades (IdP) OAuth 2.0 externo. La autenticación de los usuarios y los permisos de recursos para vhosts, bolsas, colas y temas se centralizan mediante el sistema Scope del proveedor OAuth 2.0. Esto simplifica la administración de usuarios y permite la integración con los sistemas de identidad existentes. Para obtener más información, consulte [Autenticación y autorización OAuth 2.0](#).

Autenticación y autorización de IAM

En este método, los usuarios del broker se autentican con credenciales de AWS IAM mediante la federación saliente de [IAM](#). Las credenciales de IAM se utilizan para obtener los tokens JWT del AWS Security Token Service (STS), y estos tokens JWT sirven como tokens 2.0 para la autenticación. OAuth Este método aprovecha la compatibilidad OAuth 2.0 existente en Amazon MQ para RabbitMQ, AWS donde actúa como proveedor de identidad 2.0. OAuth AWS IAM gestiona la autenticación de los usuarios, mientras que los permisos de recursos para vhosts, bolsas, colas y temas se gestionan mediante políticas de IAM y alias de ámbito configurados en RabbitMQ. [Para obtener más información, consulte Autenticación y autorización de IAM](#).

Autenticación y autorización de LDAP

En este método, los usuarios intermediarios y sus permisos se administran mediante un servicio de directorio LDAP externo. La autenticación de los usuarios y los permisos de recursos se centralizan a través del servidor LDAP, lo que permite a los usuarios acceder a RabbitMQ con sus credenciales de servicio de directorio existentes. [Para obtener más información, consulte Autenticación y autorización LDAP.](#)

Autenticación y autorización HTTP

En este método, los usuarios del broker y sus permisos se administran mediante un servidor HTTP externo. La autenticación de los usuarios y los permisos de recursos se centralizan a través del servidor HTTP, lo que permite a los usuarios acceder a RabbitMQ mediante su propio proveedor de autenticación y autorización. Para obtener más información sobre este método, consulte [Autenticación y autorización HTTP.](#)

Autenticación con certificado SSL

Amazon MQ admite el TLS mutuo (mTLS) para los corredores de RabbitMQ. El complemento de autenticación SSL utiliza los certificados de cliente de las conexiones mTLS para autenticar a los usuarios. En este método, los usuarios del broker se autentican mediante certificados de cliente X.509 en lugar de credenciales de nombre de usuario y contraseña. El certificado del cliente se valida con una autoridad de certificación (CA) de confianza y el nombre de usuario se extrae de un campo del certificado, como el nombre común (CN) o el nombre alternativo del sujeto (SAN). Este método proporciona una autenticación sólida sin transmitir las credenciales a través de la red. Para obtener más información, consulte [Autenticación con certificado SSL.](#)

Note

RabbitMQ admite varios métodos de autenticación y autorización para su uso simultáneo. Por ejemplo, puede habilitar la autenticación OAuth 2.0 y la autenticación simple (interna). Para obtener más información, consulte la sección del tutorial OAuth 2.0 sobre cómo [habilitar la autenticación OAuth 2.0 y la autenticación simple \(interna\)](#) y la documentación sobre el control de [acceso de RabbitMQ.](#)

Amazon MQ recomienda crear un usuario interno al probar las configuraciones de autenticación. Esto permite validar la configuración de acceso mediante la API de administración de RabbitMQ. [Para obtener más información, consulte Validación de acceso.](#)

Autenticación y autorización simple

Usuarios de agentes de Amazon MQ para RabbitMQ.

Note

En este tema se describe la gestión de los usuarios de los corredores con el mecanismo interno de autenticación y autorización predeterminado de RabbitMQ. Para obtener información sobre todos los métodos de autenticación y autorización compatibles, consulte [Amazon MQ for RabbitMQ Authentication and Authorization](#).

Cada conexión de cliente AMQP 0-9-1 tiene un usuario asociado. Este usuario debe estar autenticado. Cada conexión de cliente también se dirige a un host virtual (vhost). El usuario debe tener un conjunto de permisos para este vhost. Un usuario puede tener permiso de configuración, escritura y lectura de colas e intercambios en un vhost. Las credenciales de usuario y el vhost de destino se especifican cuando se establece la conexión.

Cuando crea por primera vez un agente de Amazon MQ para RabbitMQ, Amazon MQ utiliza las credenciales de inicio de sesión que usted proporciona para crear un usuario de RabbitMQ con la etiqueta `administrator`. Más adelante, puede agregar y administrar usuarios a través de la [API de administración](#) o la consola web de RabbitMQ. También puede utilizar la consola web o la API de administración de RabbitMQ para configurar o modificar permisos y etiquetas de usuario.

Note

Los usuarios de RabbitMQ no se almacenarán ni mostrarán a través de la API de [usuarios](#) de Amazon MQ.

Important

Amazon MQ para RabbitMQ no admite el nombre de usuario “guest” y eliminará la cuenta de invitado predeterminada cuando cree un nuevo agente. Amazon MQ también eliminará periódicamente cualquier cuenta creada por el cliente con dicho nombre.

Para crear un nuevo usuario con la API de administración de RabbitMQ, utilice el siguiente punto de enlace y cuerpo de la solicitud de la API. Sustituya *username* y *password* por sus nuevas credenciales de inicio de sesión.

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

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

Important

- No agregue información de identificación personal (PII) ni ninguna otra información confidencial en los nombres de usuario de agente. Otros AWS servicios, incluidos los registros, pueden acceder a los nombres de usuario de los corredores. CloudWatch Los nombres de usuario de agente no están diseñados para usarse con información privada o confidencial.
- Si pierde el acceso a todas las cuentas de administrador, consulte Cómo [recuperar el acceso de los corredores](#) para utilizar la autenticación de IAM en la recuperación.

La clave tags es obligatoria y consiste en una lista de etiquetas separadas por comas para el usuario. Amazon MQ admite etiquetas de usuario administrator, management, monitoring y policymaker.

Para configurar permisos para un usuario individual, puede usar el siguiente punto de enlace y cuerpo de la solicitud de la API. Reemplace el *vhost* y el *username* con su información. Para el vhost / predeterminado, use %2F.

```
PUT /api/permissions/vhost/username HTTP/1.1

{"configure":".*", "write":".*", "read":".*"}
```

Note

Las claves configure, read, y write son todas obligatorias.

Al utilizar el valor de comodín .*, esta operación otorgará permisos de lectura, escritura y configuración para todas las colas del vhost especificado para el usuario. Para obtener más

información acerca de cómo administrar usuarios a través de la API de administración de RabbitMQ, consulte el tema sobre [API HTTP de administración de RabbitMQ](#).

OAuth Autenticación y autorización 2.0 para Amazon MQ para RabbitMQ

Amazon MQ para RabbitMQ admite varios métodos de autenticación y autorización. Para obtener información sobre todos los métodos admitidos, consulte [Autenticación y autorización de Amazon MQ para corredores de RabbitMQ](#).

En la autenticación y autorización OAuth 2.0, los usuarios intermediarios y sus permisos son administrados por un proveedor de identidad OAuth 2.0 (IdP) externo. La autenticación de los usuarios y los permisos de recursos para los servidores virtuales, las bolsas, las colas y los temas se centralizan mediante el sistema Scope del proveedor OAuth 2.0. Esto simplifica la administración de usuarios y permite la integración con los sistemas de identidad existentes.

Consideraciones importantes

- OAuth Amazon MQ para agentes de ActiveMQ no admite la integración con la versión 2.0.
- Amazon MQ para RabbitMQ no admite el certificado de servidor emitido por una autoridad de certificación privada.
- El complemento RabbitMQ OAuth 2.0 no admite puntos finales de introspección simbólica ni tokens de acceso opacos. Tampoco realiza comprobaciones de revocación de tokens.
- Debe incluir el permiso de IAM para habilitar la versión 2.0 en los corredores `mq:UpdateBrokerAccessConfiguration` existentes. OAuth
- Amazon MQ crea automáticamente un usuario del sistema llamado `monitoring-AWS-OWNED-DO-NOT-DELETE` con permisos de solo supervisión. Este usuario utiliza el sistema de autenticación interno de RabbitMQ incluso en corredores con la versión OAuth 2.0 y está restringido únicamente al acceso a la interfaz de bucle invertido.

Para obtener información sobre cómo configurar OAuth 2.0 para sus corredores de Amazon MQ para RabbitMQ, consulte. [Uso de la autenticación y autorización de OAuth 2.0](#)

En esta página

- [Configuraciones 2.0 compatibles OAuth](#)
- [OAuth Validaciones adicionales para la autenticación 2.0](#)

Configuraciones 2.0 compatibles OAuth

Amazon MQ para RabbitMQ admite todas las [variables configurables](#) del complemento RabbitMQ OAuth 2.0, con las siguientes excepciones:

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

Como Amazon MQ no admite esta clave, no admitimos UAA como IdP.

- `management.oauth_resource_servers.{id/index}.oauth_client_secret`
- `auth_oauth2.signing_keys.{id/index}`

OAuth Validaciones adicionales para la autenticación 2.0

Amazon MQ también aplica las siguientes validaciones adicionales para la autenticación 2.0: OAuth

- Todo tiene que URLs empezar con. `https://`
- Algoritmos de firma compatibles: Ed25519, Ed25519ph, Ed448, Ed448ph, EdDSA, ES256K, ES256, ES384, ES512, HS256, HS384, HS512, PS256, PS384, PS512, RS256, RS384 y RS512.

Autenticación y autorización de IAM para Amazon MQ para RabbitMQ

Amazon MQ para RabbitMQ admite varios métodos de autenticación y autorización. Para obtener información sobre todos los métodos admitidos, consulte [Autenticación y autorización de Amazon MQ para corredores de RabbitMQ](#).

[La autenticación y la autorización de IAM permiten a los usuarios de los corredores autenticarse con credenciales de IAM a través de la federación saliente de AWS IAM](#). En este método, las credenciales de IAM se utilizan para obtener los tokens JWT del AWS Security Token Service (STS). Estos tokens JWT sirven como tokens OAuth 2.0 para la autenticación y aprovechan la compatibilidad OAuth 2.0 existente en Amazon MQ para RabbitMQ, AWS donde actúa como proveedor de identidad 2.0. OAuth AWS IAM gestiona la autenticación de los usuarios, mientras que los permisos de recursos para los hosts virtuales, las bolsas, las colas y los temas se gestionan mediante las políticas de IAM y los alias de ámbito configurados en RabbitMQ.

Consideraciones importantes

- La autenticación de IAM es compatible con las versiones 3.13, 4.2 y posteriores de RabbitMQ. No es compatible con Amazon MQ para agentes de ActiveMQ.
- La autenticación de IAM requiere que la federación saliente de IAM esté configurada y disponible en su cuenta. AWS
- Este método se basa en la infraestructura OAuth 2.0 existente en Amazon MQ para RabbitMQ y AWS actúa como proveedor de identidad 2.0. OAuth
- Amazon MQ crea automáticamente un usuario del sistema llamado `monitoring-AWS-OWNED-DO-NOT-DELETE` con permisos de solo supervisión. Este usuario utiliza el sistema de autenticación interno de RabbitMQ incluso en los corredores con IAM y está restringido únicamente al acceso a la interfaz de bucle invertido.

En esta página

- [Cómo funciona la autenticación de IAM](#)
- [Limitaciones](#)

Cómo funciona la autenticación de IAM

La autenticación de IAM para Amazon MQ para RabbitMQ [utiliza la federación saliente de IAM para AWS permitir que las credenciales de IAM](#) se autenticuen con los agentes de RabbitMQ. Las credenciales de IAM se utilizan para obtener los tokens JWT del AWS Security Token Service (STS), y estos tokens JWT sirven como tokens 2.0 para la autenticación con el bróker RabbitMQ. OAuth

Limitaciones

La autenticación de IAM para Amazon MQ para RabbitMQ tiene la siguiente limitación:

- Configuración de la notificación de alcance: no puede usar una notificación de alcance directamente porque el token JWT de STS está anidado. La clave es `sts.amazonaws.com` que requiere el uso de alias de ámbito en la configuración de RabbitMQ para asignar las funciones de IAM a los permisos de RabbitMQ. Esta limitación también impide utilizar plenamente las políticas de IAM para la autorización y, en su lugar, requiere la configuración de RabbitMQ para la autorización.

Para obtener información sobre cómo configurar la autenticación y la autorización de IAM para sus agentes de Amazon MQ for RabbitMQ, consulte. [Uso de la autenticación y la autorización de IAM](#)

Autenticación y autorización HTTP para Amazon MQ para RabbitMQ

Amazon MQ para RabbitMQ admite la autenticación y la autorización de los usuarios de corredores mediante un servidor HTTP externo. Para ver otros métodos compatibles, consulte [Autenticación y autorización de Amazon MQ para corredores de RabbitMQ](#).

Note

El complemento de autenticación HTTP solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

Consideraciones importantes

- El servidor HTTP debe ser accesible a través de la Internet pública. Amazon MQ para RabbitMQ se puede configurar para autenticarse en el servidor HTTP mediante TLS mutuo.
- Amazon MQ para RabbitMQ impone el uso de AWS ARNs para configuraciones que requieren acceso al sistema de archivos local. Consulte la [compatibilidad con ARN en la configuración de RabbitMQ](#) para obtener más detalles.
- Debe incluir el permiso de IAM para habilitar la autenticación `mq:UpdateBrokerAccessConfiguration` HTTP en los corredores existentes.
- Amazon MQ crea automáticamente un usuario del sistema llamado `monitoring-AWS-OWNED-DO-NOT-DELETE` con permisos de solo supervisión. Este usuario utiliza el sistema de autenticación interno de RabbitMQ incluso en corredores con HTTP y está restringido únicamente al acceso a la interfaz de bucle invertido. Amazon MQ impide la eliminación de este usuario añadiendo la etiqueta de [usuario protegido](#).

Para obtener información sobre cómo configurar la autenticación HTTP para sus agentes de Amazon MQ para RabbitMQ, consulte. [Uso de la autenticación y la autorización HTTP](#)

En esta página

- [Configuraciones HTTP compatibles](#)

- [Validaciones adicionales para configuraciones HTTP en Amazon MQ](#)

Configuraciones HTTP compatibles

Amazon MQ para RabbitMQ admite todas las variables configurables del [complemento de autenticación HTTP de RabbitMQ](#), con las siguientes excepciones que son obligatorias. AWS ARNs
Para obtener más información sobre la compatibilidad con ARN, consulte Compatibilidad con [ARN](#) en la configuración de RabbitMQ.

Configuraciones que requieren ARNs

`auth_http.ssl_options.cacertfile`

En su lugar, use `aws.arns.auth_http.ssl_options.cacertfile`

`auth_http.ssl_options.certfile`

En su lugar, use `aws.arns.auth_http.ssl_options.certfile`

`auth_http.ssl_options.keyfile`

En su lugar, use `aws.arns.auth_http.ssl_options.keyfile`

Opciones de SSL no compatibles

Tampoco se admiten las siguientes opciones de configuración de SSL:

Ver la lista completa

- `auth_http.ssl_options.cert`
- `auth_http.ssl_options.client_renegotiation`
- `auth_http.ssl_options.dh`
- `auth_http.ssl_options.dhfile`
- `auth_http.ssl_options.honor_cipher_order`
- `auth_http.ssl_options.honor_ecc_order`
- `auth_http.ssl_options.key.RSAPrivateKey`
- `auth_http.ssl_options.key.DSAPrivateKey`
- `auth_http.ssl_options.key.PrivateKeyInfo`

- `auth_http.ssl_options.log_alert`
- `auth_http.ssl_options.password`
- `auth_http.ssl_options.psk_identity`
- `auth_http.ssl_options.reuse_sessions`
- `auth_http.ssl_options.secure_renegotiate`
- `auth_http.ssl_options.versions.$version`
- `auth_http.ssl_options.sni`
- `auth_http.ssl_options.crl_check`

Validaciones adicionales para configuraciones HTTP en Amazon MQ

Amazon MQ también aplica las siguientes validaciones adicionales para la autenticación y autorización HTTP:

- `auth_http.http_method` debe ser una `get` o `post`
- Las siguientes configuraciones de ruta deben usar HTTPS URLs:
 - `auth_http.user_path`
 - `auth_http.vhost_path`
 - `auth_http.resource_path`
 - `auth_http.topic_path`
- Si alguna configuración requiere el uso de un AWS ARN, `aws.arns.assume_role_arn` debe proporcionarse.

Autenticación con certificado SSL para Amazon MQ para RabbitMQ

Amazon MQ para RabbitMQ admite la autenticación de usuarios de corredores mediante certificados de cliente X.509. Para ver otros métodos compatibles, consulte [Autenticación y autorización de Amazon MQ para corredores de RabbitMQ](#).

Note

El complemento de autenticación con certificado SSL solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

Consideraciones importantes

- Los certificados de cliente deben estar firmados por una autoridad de certificación (CA) de confianza. Amazon MQ for RabbitMQ valida la cadena de certificados durante la autenticación.
- Amazon MQ for RabbitMQ impone el uso de para configuraciones relacionadas con certificados, como los certificados de CA, y AWS ARNs para configuraciones que requieren acceso al sistema de archivos local. Consulte la [compatibilidad con ARN en la configuración de RabbitMQ](#) para obtener más detalles.
- Amazon MQ crea automáticamente un usuario del sistema llamado `monitoring-AWS-OWNED-DO-NOT-DELETE` con permisos de solo supervisión. Este usuario utiliza el sistema de autenticación interno de RabbitMQ incluso en corredores con certificados SSL y está restringido únicamente al acceso a la interfaz de bucle invertido. Amazon MQ impide la eliminación de este usuario añadiendo la etiqueta de [usuario protegido](#).

Para obtener información sobre cómo configurar la autenticación con certificado SSL para sus corredores de Amazon MQ para RabbitMQ, consulte [Uso de la autenticación con certificado SSL](#).

En esta página

- [Configuraciones SSL compatibles](#)
- [Validaciones adicionales para las configuraciones de SSL en Amazon MQ](#)

Configuraciones SSL compatibles

Amazon MQ para RabbitMQ admite SSL/TLS la configuración de las conexiones de los clientes. Para obtener más información sobre la compatibilidad con ARN, consulte Compatibilidad con [ARN](#) en la configuración de RabbitMQ.

Configuraciones que requieren ARNs

`ssl_options.cacertfile`

En su lugar, use `aws.arns.ssl_options.cacertfile`

Configuraciones de inicio de sesión con certificado SSL

Las siguientes configuraciones controlan la forma en que se extraen los nombres de usuario de los certificados de los clientes:

`ssl_cert_login_from`

Especifica qué campo de certificado se debe utilizar para la extracción del nombre de usuario. Valores admitidos:

- `distinguished_name`- Utilice el nombre distintivo completo
- `common_name`- Utilice el campo Nombre común (CN)
- `subject_alternative_name` o `subject_alt_name` - Utilice el nombre alternativo del asunto

`ssl_cert_login_san_type`

Cuando utilice el nombre alternativo del sujeto, especifique el tipo de SAN. Valores admitidos: `dnsip`, `email`, `uri`, `other_name`

`ssl_cert_login_san_index`

Al utilizar el nombre alternativo del sujeto, especifica el índice de la entrada de SAN que se va a utilizar (de base cero). Debe ser un número entero no negativo.

Opciones de SSL para las conexiones de los clientes

Las siguientes opciones de SSL se aplican a las conexiones de los clientes:

`ssl_options.verify`

Modo de verificación por pares. Valores admitidos: `verify_none`, `verify_peer`

`ssl_options.fail_if_no_peer_cert`

Si se deben rechazar las conexiones si el cliente no proporciona un certificado. Valor booleano.

`ssl_options.depth`

Profundidad máxima de la cadena de certificados para la verificación.

`ssl_options.hostname_verification`

Modo de verificación del nombre de host. Valores admitidos: `wildcard`, `none`

Opciones de SSL no compatibles

No se admiten las siguientes opciones de configuración de SSL:

Ver la lista completa

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

Validaciones adicionales para las configuraciones de SSL en Amazon MQ

Amazon MQ también aplica las siguientes validaciones adicionales para la autenticación de certificados SSL:

- Si alguna configuración requiere el uso de un AWS ARN, `aws.arns.assume_role_arn` debe proporcionarse.

Autenticación y autorización LDAP para Amazon MQ para RabbitMQ

Amazon MQ para RabbitMQ admite la autenticación y la autorización de los usuarios de corredores mediante un servidor LDAP externo. Para ver otros métodos compatibles, consulte [Autenticación y autorización de Amazon MQ para corredores de RabbitMQ](#).

Consideraciones importantes

- El servidor LDAP debe ser accesible a través de la Internet pública. Amazon MQ para RabbitMQ se puede configurar para autenticarse en el servidor LDAP mediante TLS mutuo.
- Amazon MQ para RabbitMQ impone el uso de configuraciones de LDAP confidenciales, como contraseñas, y AWS ARNs para configuraciones que requieren acceso al sistema de archivos local. Consulte la [compatibilidad con ARN en la configuración de RabbitMQ](#) para obtener más detalles.
- Debe incluir el permiso de IAM para habilitar LDAP en `mq:UpdateBrokerAccessConfiguration` los corredores existentes.
- Amazon MQ crea automáticamente un usuario del sistema llamado `monitoring-AWS-OWNED-DO-NOT-DELETE` con permisos de solo supervisión. Este usuario utiliza el sistema de autenticación interno de RabbitMQ incluso en corredores con LDAP y está restringido únicamente al acceso a la interfaz de bucle invertido. Amazon MQ impide la eliminación de este usuario añadiendo la etiqueta de [usuario protegido](#).

Para obtener información sobre cómo configurar LDAP para sus corredores de Amazon MQ para RabbitMQ, consulte. [Uso de la autenticación y la autorización de LDAP](#)

En esta página

- [Configuraciones LDAP compatibles](#)
- [Validaciones adicionales para configuraciones de LDAP en Amazon MQ](#)

Configuraciones LDAP compatibles

Amazon MQ para RabbitMQ admite todas las variables configurables del complemento [LDAP RabbitMQ](#), con las siguientes excepciones que son necesarias. AWS ARNs Para obtener más

información sobre la compatibilidad con ARN, consulte Compatibilidad con [ARN](#) en la configuración de RabbitMQ.

Configuraciones que requieren ARNs

`auth_ldap.dn_lookup_bind.password`

En su lugar, use `aws.arns.auth_ldap.dn_lookup_bind.password`

`auth_ldap.other_bind.password`

En su lugar, use `aws.arns.auth_ldap.other_bind.password`

`auth_ldap.ssl_options.cacertfile`

En su lugar, use `aws.arns.auth_ldap.ssl_options.cacertfile`

`auth_ldap.ssl_options.certfile`

En su lugar, use `aws.arns.auth_ldap.ssl_options.certfile`

`auth_ldap.ssl_options.keyfile`

En su lugar, use `aws.arns.auth_ldap.ssl_options.keyfile`

Opciones de SSL no compatibles

Tampoco se admiten las siguientes opciones de configuración de SSL:

Ver la lista completa

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

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

Validaciones adicionales para configuraciones de LDAP en Amazon MQ

Amazon MQ también aplica las siguientes validaciones adicionales para la autenticación y la autorización de LDAP:

- `auth_ldap.logno` se puede configurar en `network_unsafe`
- El servidor LDAP debe usar LDAPS. `auth_ldap.use_ssl` o bien `auth_ldap.use_starttls` debe estar activado de forma explícita
- Si alguna configuración requiere el uso de un AWS ARN, `aws.arns.assume_role_arn` debe proporcionarse.
- `auth_ldap.servers` debe ser una dirección IP válida o un FQDN válido
- Las siguientes claves deben ser un nombre distintivo LDAP válido:
 - `auth_ldap.dn_lookup_base`
 - `auth_ldap.dn_lookup_bind.user_dn`
 - `auth_ldap.other_bind.user_dn`
 - `auth_ldap.group_lookup_base`

Plugins

Amazon MQ para RabbitMQ también es compatible con los siguientes complementos.

- [Complemento de administración RabbitMQ](#)
- [Plugin Shovel](#)
- [Complemento de federación](#)
- [Complemento de intercambio de hash coherente](#)

- [OAuth Complemento 2](#)
- [Complemento LDAP](#)
- [Complemento HTTP](#)
- [Complemento de certificado SSL](#)
- [complemento aws](#)
- [Complemento JMS Topic Exchange](#)

Complemento de administración RabbitMQ

Amazon MQ para RabbitMQ es compatible con el [complemento de administración RabbitMQ, que proporciona una API de administración](#) basada en HTTP junto con una interfaz de usuario basada en navegador para la consola web de RabbitMQ. Puede usar la consola web y la API de administración para crear y administrar políticas y usuarios del agente.

el complemento Shovel (pala)

Amazon MQ para RabbitMQ es compatible con el [complemento de pala RabbitMQ](#), que permite mover mensajes de colas e intercambios de un bróker a otro. Puede utilizar palas para conectar agentes con acoplamiento débil y distribuir mensajes lejos de los nodos con mayores cargas de mensajes.

Important

No puede configurar la pala entre colas o intercambios si el destino de la pala es un agente privado.

Amazon MQ no admite el uso de palas estáticas.

[Solo se admiten palas dinámicas.](#) Las palas dinámicas se configuran mediante parámetros de tiempo de ejecución y se pueden iniciar y detener en cualquier momento mediante programación mediante una conexión de cliente. Por ejemplo, con la API de administración de RabbitMQ, puede crear una solicitud PUT en el siguiente punto final de la API para configurar una pala dinámica. En el ejemplo, se puede sustituir {vhost} por el nombre del vhost del bróker y {name} por el nombre de la nueva pala dinámica.

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


En el cuerpo de la solicitud, debe especificar una cola o un intercambio, pero no ambos. En el siguiente ejemplo, se configura una distribución dinámica entre una cola local especificada en `src-queue` y una cola remota definida en `dest-queue`. Del mismo modo, puede usar los parámetros `src-exchange` y `dest-exchange` para configurar una pala entre dos intercambios.

```
{
  "value": {
    "src-protocol": "amqp091",
    "src-uri": "amqp://localhost",
    "src-queue": "source-queue-name",
    "dest-protocol": "amqp091",
    "dest-uri": "amqps://b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-
west2.amazonaws.com:5671",
    "dest-queue": "destination-queue-name"
  }
}
```

Complemento Federation (federación)

[Amazon MQ admite las colas y los intercambios federados mediante el complemento de federación RabbitMQ](#). El complemento de federación le permite replicar el flujo de mensajes entre colas, intercambios y consumidores en agentes independientes. Las colas y bolsas federadas utilizan point-to-point enlaces para conectarse con pares de otros corredores. Mientras que los intercambios federados, de forma predeterminada, enrutan mensajes una sola vez, las colas federadas pueden mover mensajes la cantidad de veces que los consumidores lo necesiten.

Puede utilizar el complemento de federación para permitir que un agente descendente consuma un mensaje de un intercambio o cola ascendente. Para habilitar el complemento de federación en agentes descendentes, puede utilizar la consola web o la API de administración de RabbitMQ.

Important

No puede configurar la federación si la cola o el intercambio ascendente se encuentra en un agente privado. Solo puede configurar la federación entre colas o intercambios en agentes públicos, o entre una cola o intercambio ascendente en un agente público y una cola o intercambio descendente en un agente privado.

Por ejemplo, para configurar la federación con la API de administración, puede hacer lo siguiente:

- Configure uno o más flujos ascendentes que definan conexiones de federación con otros nodos. Para definir conexiones de federación, puede utilizar la consola web o la API de administración de RabbitMQ. Con la API de administración, puede crear una solicitud POST a `/api/parameters/federation-upstream/%2f/myupstream` con el siguiente cuerpo de solicitud.

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

- Configure una política para permitir la federación de sus colas o intercambios. Para configurar políticas, puede utilizar la consola web o la API de administración de RabbitMQ. Con la API de administración, puedes crear una solicitud POST para `/api/policies/%2f/federate-me` con el siguiente cuerpo de solicitud.

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

Note

El cuerpo de la solicitud asume que los nombres de los intercambios del servidor comienzan por `amq`. El uso de la expresión regular `^amq\\.` garantizará que la federación esté habilitada para todos los intercambios cuyos nombres comiencen por «`amq`». Los intercambios de su servidor RabbitMQ se pueden nombrar de otra manera.

Complemento de intercambio Hash consistente

Amazon MQ para RabbitMQ es compatible con el complemento RabbitMQ Consistent Hash Exchange [Type](#). Los intercambios Hash consistentes enrutan mensajes a colas en función de un valor hash que se calcula a partir de la clave de enrutamiento de un mensaje. Si se proporciona una clave de enrutamiento razonablemente uniforme, los intercambios Hash consistentes pueden distribuir mensajes entre colas de manera razonablemente pareja.

En el caso de las colas vinculadas a un intercambio de hash coherente, la clave de enlace es la `number-as-a-string` que determina el peso de enlace de cada cola. Las colas con un peso de enlace más alto recibirán una distribución proporcionalmente mayor de mensajes del intercambio Hash consistente al que están enlazadas. En una topología de intercambio Hash consistente, los publicadores pueden simplemente publicar mensajes para el intercambio, pero los consumidores deben estar explícitamente configurados para consumir mensajes de colas específicas.

OAuth Complemento 2.0

[Amazon MQ para RabbitMQ admite el complemento de backend de 2 autenticaciones. OAuth](#)

Este complemento se habilita de forma condicional en función de la configuración de su agente. Cuando está habilitado, este complemento proporciona autenticación y autorización OAuth 2.0 con integración con proveedores de identidad OAuth 2.0 externos para una gestión de usuarios y un control de acceso centralizados. Para obtener más información sobre la autenticación OAuth 2.0, consulte [OAuth Autenticación y autorización 2.0](#).

Complemento LDAP

[Amazon MQ para RabbitMQ es compatible con el complemento de backend de autenticación LDAP.](#)

Este complemento se habilita de forma condicional en función de la configuración de su agente. Cuando está habilitado, este complemento proporciona autenticación y autorización de LDAP con integración a servicios de directorio LDAP externos para una autenticación y autorización de usuarios centralizadas. Para obtener más información sobre la autenticación LDAP, consulte [Autenticación y autorización LDAP](#).

Complemento HTTP

[Amazon MQ para RabbitMQ admite el complemento de backend de autenticación HTTP.](#) Este complemento se habilita de forma condicional en función de la configuración de su agente. Cuando está habilitado, este complemento proporciona autenticación y autorización HTTP con integración en servidores HTTP externos para una autenticación y autorización de usuarios centralizadas. Para obtener más información sobre la autenticación HTTP, consulte [Autenticación y autorización HTTP](#).

Note

El complemento de autenticación HTTP solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

Complemento de certificado SSL

Amazon MQ admite el TLS mutuo (mTLS) para los corredores de RabbitMQ. El [complemento de autenticación SSL](#) utiliza los certificados de cliente de las conexiones mTLS para autenticar a los usuarios. Este complemento se habilita de forma condicional en función de la configuración de su agente. Cuando está habilitado, proporciona una autenticación basada en certificados mediante

certificados de cliente X.509 para una autenticación sólida sin transmitir credenciales a través de la red. Para obtener más información sobre la autenticación con certificados SSL, consulte.

[Autenticación con certificado SSL](#)

Note

El complemento de autenticación con certificado SSL solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

complemento aws

Amazon MQ habilita el [complemento aws](#) para RabbitMQ de forma condicional en función de la configuración de su agente. Este complemento comunitario, desarrollado y mantenido por Amazon MQ, proporciona una recuperación segura de credenciales y certificados de los AWS servicios que se utilizan AWS ARNs en los ajustes de configuración de RabbitMQ. Para obtener más información sobre la compatibilidad con ARN, consulte. [ARN support in RabbitMQ configuration](#)

Complemento JMS Topic Exchange

Amazon MQ para RabbitMQ siempre habilita el [complemento JMS Topic Exchange](#). Funciona con el [cliente JMS RabbitMQ para permitir que las aplicaciones JMS](#) nuevas y existentes se conecten a Amazon MQ for RabbitMQ.

Note

El complemento JMS Topic Exchange solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores. Está activado de forma predeterminada, pero solo se activa cuando el cliente JMS de RabbitMQ se utiliza para ejecutar cargas de trabajo de JMS.

Protocolos admitidos

Puede acceder a sus corredores de RabbitMQ utilizando [cualquier lenguaje de programación compatible con RabbitMQ](#) y habilitando TLS para cualquiera de las siguientes especificaciones de protocolo:

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

- [AMQP 1.0](#)
- [JMS 1.1](#)
- [JMS 2.0](#)
- [JMS 3.1](#)

Compatibilidad con Amazon MQ para RabbitMQ JMS

Ahora puede ejecutar cargas de trabajo JMS 1.1, 2.0 y 3.1 en Amazon MQ para RabbitMQ 4 con el cliente RabbitMQ JMS.

Cliente RabbitMQ JMS

El cliente JMS de RabbitMQ es una biblioteca de clientes de JMS de código abierto que necesita para conectar su aplicación JMS a los corredores de Amazon MQ RabbitMQ. Para obtener más [GitHub información](#), visite el repositorio oficial.

Compatible con JMS 1.1, 2.0 y 3.1 APIs

Desde Amazon MQ para RabbitMQ 4 en adelante, el complemento está siempre activado. `jms-topic-exchange` Por lo tanto, puede usar Amazon MQ para RabbitMQ 4 y el cliente RabbitMQ JMS para su carga de trabajo de JMS. [Se admiten todos los JMS definidos en el JMS 1.1, excepto APIs :](#)

- No se admiten las sesiones APIs de servidor.
- No APIs se admiten transacciones XA.
- No se admite el selector JMS para el destino de la cola JMS.
- No se admite el atributo de `NoLocal` suscripción JMS.

Se admiten todos los nuevos agregados APIs en [JMS 2.0 y JMS 3.1](#), excepto:

- `JMSProducer.setDeliveryDelay` No se admite la API.

Para obtener más información sobre cómo conectar su aplicación JMS a Amazon MQ para el bróker RabbitMQ, consulte el tutorial [sobre Cómo conectar su aplicación JMS a Amazon MQ para](#) el bróker RabbitMQ

Autenticación y autorización

[Se admiten todos los mecanismos de autenticación y autorización que se enumeran en esta sección.](#)

Las credenciales utilizadas para conectarse al corredor mediante el cliente JMS son las mismas que si se conectara al corredor RabbitMQ mediante un cliente Java AMQP.

Interoperabilidad con las colas AMQP de RabbitMQ

Puede utilizar el cliente JMS RabbitMQ para enviar mensajes JMS a un intercambio AMQP y consumir mensajes de una cola AMQP (esta función no admite temas de JMS). Esto le permite interoperar o migrar determinadas cargas de trabajo de JMS a cargas de trabajo de AMQP. [Para obtener más información, consulte la documentación oficial del cliente.](#)

Aplicación de políticas a Amazon MQ para RabbitMQ

Es posible aplicar políticas y límites personalizados con los valores predeterminados recomendados por Amazon MQ. Si ha eliminado las políticas y los límites predeterminados recomendados y desea volver a crearlos, o ha creado vhosts adicionales y desea aplicarles las políticas y los límites predeterminados, puede seguir estos pasos.

Important

En las versiones 3.13 y anteriores del motor Amazon MQ para RabbitMQ, la política de operador predeterminada actual es:

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

En las versiones 4.0 y posteriores, la política de operador predeterminada ha cambiado a:

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

Este cambio es necesario porque RabbitMQ 4 no admite la duplicación de colas clásica y la configuración de las políticas de alta disponibilidad.

No es posible crear una política que se aplique tanto a las colas reflejadas clásicas como a las colas de cuórum. Si desea que su política se aplique exclusivamente a las colas de

cuórum, debe configurar `--apply-to` en `quorum_queues`. Si utiliza las colas reflejadas y las colas de cuórum clásicas, debe crear una política independiente con `--apply-to:classic_queues`, así como una política de colas de cuórum.

Important

Para realizar los pasos que se indican a continuación, debe tener un usuario de agente de Amazon MQ para RabbitMQ con permisos de administrador. Puede utilizar el usuario administrador que generó cuando creó el agente por primera vez, u otro usuario que podría haber creado después. La siguiente tabla proporciona la etiqueta y los permisos de usuario administrador necesarios como patrones de expresiones regulares (regex).

Tags	Regex Read (Lectura)	Regex Configure (Configuración)	Regex Write (Escritura)
administrator	.*	.*	.*

Para obtener más información acerca de cómo crear usuarios de RabbitMQ y administrar etiquetas y permisos de usuario, consulte [Usuarios de agentes de Amazon MQ para RabbitMQ..](#)

Para aplicar políticas y límites de anfitriones virtuales predeterminados mediante la consola web de RabbitMQ

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En el panel de navegación izquierdo, elija Brokers (Agentes).
3. En la lista de agentes, elija el nombre del agente al que desea aplicar la nueva política.
4. En la sección Connections (Conexiones) de la página de detalles del agente, elija la URL de la consola web de RabbitMQ. La consola web de RabbitMQ se abre en una nueva pestaña o ventana del navegador.
5. Inicie sesión en la consola web de RabbitMQ con el nombre y la contraseña del usuario administrador del agente.
6. En la consola web de RabbitMQ, elija Admin (Administrador) en la parte superior de la página.

7. En la página Admin (Administrador), elija Políticas (Políticas) en el panel de navegación derecho.
8. En la página Políticas (Políticas), puede ver una lista de las Políticas de usuario actuales del agente. Debajo de las políticas de usuario, expanda Add/update a policy (Agregar/actualizar una política).
9. Para crear una nueva política de agente, en Add/update a policy (Agregar/actualizar una política), haga lo siguiente:
 - a. En Virtual host (Anfitrión virtual), elija en el menú desplegable el nombre del vhost al que desea asociar las políticas. Para elegir el vhost predeterminado, elija /.

 Note

Si no creó vhosts adicionales, no aparece la opción Virtual host (Anfitrión virtual) en la consola de RabbitMQ y las políticas se aplican únicamente al vhost predeterminado.

- b. En Name (Nombre), escriba un nombre para su política, por ejemplo, **policy-defaults**.
- c. En Pattern (Patrón), ingrese el patrón regexp `.*` para que la política coincida con todas las colas del agente.
- d. En Apply to (Aplicar a), elija Exchanges and queues (Intercambios y colas) en el menú desplegable.
- e. En Priority (Prioridad), ingrese un número entero mayor que todas las demás políticas aplicadas al vhost. Puede aplicar exactamente un conjunto de definiciones de políticas a las colas e intercambios de RabbitMQ en cualquier momento dado. RabbitMQ elige la política de coincidencia con el valor de prioridad más alto. Para obtener más información acerca de las prioridades de las políticas y cómo combinar políticas, consulte el tema sobre [políticas](#) en la documentación del servidor de RabbitMQ.
- f. En Definition (Definición), agregue los siguientes pares clave-valor:
 - **queue-mode=lazy**. Elija String (Cadena) en el menú desplegable.
 - **overflow=reject-publish**. Elija String (Cadena) en el menú desplegable.

 Note

No se aplica a los agentes de una sola instancia.

- **max-length=.** *number-of-messages number-of-messages* Sustitúyalo por el [valor recomendado por Amazon MQ](#) según el tamaño de la instancia y el modo de implementación del bróker, por ejemplo, **8000000** para un `mq.m7g.large` clúster. Seleccione Number (Número) en el menú desplegable.

 Note

No se aplica a los agentes de una sola instancia.

- g. Elija Add/update policy (Agregar/actualizar política).
10. Confirme que la nueva política aparezca en la lista de políticas de usuario.

 Note

Para los agentes de clúster, Amazon MQ aplica automáticamente las definiciones de políticas `ha-mode: all` y `ha-sync-mode: automatic`.

11. Elija Limits (Límites) en el panel de navegación.
12. En la página Limits (Límites), puede ver una lista de los límites de anfitriones virtuales actuales del agente. Debajo de los límites de anfitriones virtuales, expanda Set/update a virtual host limit (Establecer/actualizar un límite de anfitrión virtual).
13. Para crear un nuevo límite de vhost, en Set/update a virtual host limit (Establecer/actualizar un límite de anfitrión virtual), haga lo siguiente:
 - a. En Virtual host (Anfitrión virtual), elija en el menú desplegable el nombre del vhost al que desea asociar las políticas. Para elegir el vhost predeterminado, elija /.
 - b. En Limit (Límite), elija max-connections en las opciones desplegables.
 - c. En Value (Valor), ingrese el [valor recomendado por Amazon MQ](#) de acuerdo con el tamaño de la instancia y el modo de implementación del agente, por ejemplo, **15000** para un clúster `mq.m5.large`.
 - d. Elija Set/update limit (Establecer/actualizar límite).
 - e. Repita los pasos anteriores y, en Limit (Límite), elija max-queues en las opciones desplegables.
14. Confirme que los nuevos límites aparezcan en la lista de límites de anfitriones virtuales.

Para aplicar políticas y límites de anfitriones virtuales predeterminados mediante la API de administración de RabbitMQ

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En el panel de navegación izquierdo, elija Brokers (Agentes).
3. En la lista de agentes, elija el nombre del agente al que desea aplicar la nueva política.
4. En la sección Connections (Conexiones) de la página del agente, anote la URL de la consola web de RabbitMQ. Este es el punto de enlace del agente que se utiliza en una solicitud HTTP.
5. Abra un terminal nuevo o la ventana de línea de comandos que desee.
6. Para crear una nueva política de agente, ingrese el comando `curl`. Este comando asume que hay una cola en el vhost / predeterminado, que está codificado como `%2F`. Para aplicar la política a otro vhost, reemplace `%2F` por el nombre del vhost.

 Note

Sustituya *username* y por *password* sus credenciales de inicio de sesión de administrador. *number-of-messages* Sustitúyalo por el [valor recomendado por Amazon MQ](#) según el tamaño de la instancia y el modo de implementación del bróker. *policy-name* Sustitúyalo por un nombre para la política. *broker-endpoint* Sustitúyala por la URL que anotaste anteriormente.

```
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. Para confirmar que se haya agregado la nueva política a las políticas de usuario de su agente, ingrese el comando `curl` para ver una lista de todas las políticas de agente.

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

8. Para crear un nuevo límite de anfitrión virtual `max-connections`, ingrese el comando `curl`. Este comando asume que hay una cola en el vhost / predeterminado, que está codificado como `%2F`. Para aplicar la política a otro vhost, reemplace `%2F` por el nombre del vhost.

Note

Sustituya *username* y *password* por sus credenciales de inicio de sesión de administrador. *max-connections* Sustitúyalo por el [valor recomendado por Amazon MQ](#) según el tamaño de la instancia y el modo de implementación del bróker. Reemplace el punto de enlace del agente por la URL que anotó anteriormente.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \  
-d '{"value":"number-of-connections"}' \  
broker-endpoint/api/vhost-limits/%2F/max-connections
```

9. Para crear un nuevo límite de anfitrión virtual *max-queues*, repita el paso anterior, pero modifique el comando `curl` como se muestra a continuación.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \  
-d '{"value":"number-of-queues"}' \  
broker-endpoint/api/vhost-limits/%2F/max-queues
```

10. Para confirmar que los nuevos límites se agreguen a los límites de anfitriones virtuales de su agente, ingrese el comando `curl` para ver una lista de todos los límites de anfitriones virtuales del agente.

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

Colas de cuórum para RabbitMQ en Amazon MQ

Las colas de cuórum son un tipo de cola reflejada compuesta por un líder (réplica principal) y seguidores (otras réplicas). Si el líder deja de estar disponible, las colas de cuórum utilizan el algoritmo de consenso [Raft](#) para elegir un nuevo nodo líder por mayoría de votos, y el líder anterior pasa a ser un nodo seguidor del mismo clúster. Los seguidores restantes siguen replicándose como antes. Como cada nodo se encuentra en una zona de disponibilidad diferente, si un nodo no está disponible temporalmente, la entrega de mensajes continúa con la réplica líder recién elegida en otra zona de disponibilidad.

Las colas de cuórum son útiles para gestionar los mensajes dudosos, que se producen cuando un mensaje falla y se vuelve a poner en cola varias veces.

No debe utilizar las colas de cuórum si:

- emplea colas transitorias
- tiene largas demoras de colas
- prioriza latencias bajas

Para declarar una cola de cuórum, defina el encabezado `x-queue-type` en `quorum`.

Temas

- [Migración de colas clásicas a colas de cuórum en Amazon MQ para RabbitMQ](#)
- [Configuraciones de políticas para colas de cuórum de Amazon MQ para RabbitMQ](#)
- [Prácticas recomendadas de las colas de cuórum para Amazon MQ para RabbitMQ](#)

Migración de colas clásicas a colas de cuórum en Amazon MQ para RabbitMQ

Puede migrar sus colas reflejadas clásicas a colas de cuórum en agentes de Amazon MQ de la versión 3.13 o posterior creando un nuevo host virtual en el mismo clúster o llevando a cabo una migración in situ.

Opción 1: Migración de las colas reflejadas clásicas a las colas de cuórum con un nuevo host virtual

Puede migrar sus colas reflejadas clásicas a colas de cuórum en agentes de Amazon MQ de la versión 3.13 o posterior creando un nuevo host virtual en el mismo clúster.

1. En el clúster existente, cree un nuevo host virtual (vhost) con el tipo de cola predeterminado como cuórum.
2. Cree el [Complemento Federation \(federación\)](#) del vhost a partir del nuevo vhost con el URI que apunte al antiguo mediante las colas reflejadas clásicas.
3. Con `rabbitmqadmin`, exporte las definiciones del vhost anterior a un archivo nuevo. Debe realizar cambios en el archivo de esquema para que sea compatible con las colas de cuórum. Para ver la lista completa de los cambios que debe realizar en el archivo, consulte [Moving](#)

[definitions](#) en la documentación sobre las colas de cuórum de RabbitMQ. Tras aplicar los cambios necesarios al archivo, vuelva a importar las definiciones al nuevo vhost.

4. Cree una nueva política en el nuevo vhost. Para obtener recomendaciones sobre las configuraciones de políticas de Amazon MQ para las colas de cuórum, consulte [Configuraciones de políticas para colas de cuórum de Amazon MQ para RabbitMQ](#). A continuación, inicie la federación que creó anteriormente desde el vhost anterior en el nuevo.
5. Dirija a los consumidores y productores al nuevo host virtual.
6. Configure el complemento de pala para mover los mensajes restantes. Cuando la cola esté vacía, elimine la pala.

Migración de las colas reflejadas clásicas en las colas de cuórum existentes

Puede migrar sus colas reflejadas clásicas a colas de cuórum en agentes de Amazon MQ de la versión 3.13 o posterior llevando a cabo una migrando in situ.

1. Detenga a los consumidores y productores.
2. Cree una nueva cola de cuórum temporal.
3. Configure el complemento de pala para mover los mensajes de la antigua cola reflejada clásica a la nueva cola de cuórum temporal. Cuando todos los mensajes se hayan movido a la cola de cuórum temporal, elimine la pala.
4. Elimine la cola reflejada clásica original. A continuación, vuelva a crear una cola de cuórum con el mismo nombre y los mismos enlaces que la cola reflejada clásica original.
5. Cree una pala nueva para mover los mensajes de la cola de cuórum temporal a la nueva cola de cuórum.

Configuraciones de políticas para colas de cuórum de Amazon MQ para RabbitMQ

Puede añadir configuraciones de políticas específicas a las colas de cuórum de su agente de RabbitMQ en Amazon MQ.

Cuando cree una política para colas de cuórum, debe hacer lo siguiente:

- Elimine todos los atributos de la política que comiencen por `ha`, como por ejemplo `ha-mode`, `ha-params`, `ha-sync-mode`, `ha-sync-batch-size`, `ha-promote-on-shutdown` y `ha-promote-on-failure`.

- Remove queue-mode.
- Cambie el desbordamiento cuando esté establecido en reject-publish-dlx.

Important

Amazon MQ para RabbitMQ aplica todos o ninguno de los atributos de una política. No es posible crear una política que se aplique tanto a las colas reflejadas clásicas como a las colas de cuórum. Si desea que su política se aplique exclusivamente a las colas de cuórum, debe configurar `--apply-to` en `quorum_queues`. Si utiliza las colas reflejadas y las colas de cuórum clásicas, debe crear una política independiente con `--apply-to:classic_queues`, así como una política de colas de cuórum.

No necesita modificar las políticas de AWS-DEFAULT porque adoptan automáticamente el nuevo tipo de cola en el parámetro “se aplica a”. Para obtener más información acerca de las directivas predeterminadas de Amazon MQ para RabbitMQ, consulte [Configuración de políticas del operador](#).

Prácticas recomendadas de las colas de cuórum para Amazon MQ para RabbitMQ

Se recomienda utilizar las siguientes prácticas recomendadas para mejorar el rendimiento al trabajar con colas de cuórum.

Gestión de los mensajes dudosos mediante el establecimiento de un límite de entrega

Los mensajes dudosos se producen cuando un mensaje falla y se vuelve a entregar varias veces. Se puede establecer un límite de entrega de mensajes mediante el argumento de política `delivery-limit`, de manera que se descarten los mensajes que se vuelven a entregar varias veces. Si un mensaje se vuelve a entregar más veces de las permitidas según el límite de entrega, RabbitMQ lo descarta y lo elimina. Cuando se establece un límite de entrega, el mensaje se vuelve a colocar cerca del principio de la cola.

Prioridad de mensajes en las colas de cuórum

En las colas de cuórum no hay mensajes con prioridad. Si necesita dar prioridad a los mensajes, debe crear varias colas de cuórum. Para obtener más información sobre cómo priorizar los mensajes con varias colas de cuórum, consulte [Message priority](#) en la documentación de RabbitMQ.

Uso del factor de replicación predeterminado

Amazon MQ para RabbitMQ establece de forma predeterminada un factor de replicación de tres (3) nodos para los agentes de clústeres que utilizan colas de cuórum. Si efectúa cambios en `x-quorum-initial-group-size`, Amazon MQ volverá a utilizar de forma predeterminada el factor de replicación de 3.

Prácticas recomendadas de Amazon MQ para RabbitMQ

Siga estas pautas de preparación para la producción para maximizar el rendimiento de los agentes y optimizar la eficiencia del procesamiento de mensajes cuando trabaje con agentes de Amazon MQ para RabbitMQ.

Important

Actualmente, Amazon MQ no admite [flujos](#), ni utiliza el registro estructurado en JSON, presentado en RabbitMQ 3.9.x.

Temas

- [Prácticas recomendadas para la configuración de agentes y la gestión de conexiones en Amazon MQ para RabbitMQ](#)
- [Prácticas recomendadas de la durabilidad y fiabilidad de mensajes en Amazon MQ para RabbitMQ](#)
- [Prácticas recomendadas para lograr la eficiencia y optimización del rendimiento en Amazon MQ para RabbitMQ](#)
- [Prácticas recomendadas de supervisión y resiliencia de la red en Amazon MQ para RabbitMQ](#)

Prácticas recomendadas para la configuración de agentes y la gestión de conexiones en Amazon MQ para RabbitMQ

La configuración y la gestión de las conexiones del agente son el primer paso para evitar problemas relacionados con el rendimiento de los mensajes del agente, la utilización de los recursos y la capacidad de gestionar las cargas de trabajo de producción. Al [crear y configurar un agente Amazon MQ para RabbitMQ](#), siga las siguientes prácticas recomendadas de selección de los tipos de instancias adecuados, gestionar las conexiones de forma eficaz y configurar la recuperación previa de mensajes para maximizar el rendimiento de su agente.

⚠ Important

Amazon MQ para RabbitMQ no admite el nombre de usuario “guest” y eliminará la cuenta de invitado predeterminada cuando cree un nuevo agente. Amazon MQ también eliminará periódicamente cualquier cuenta creada por el cliente con dicho nombre.

Paso 1: Utilizar las implementaciones de clústeres

Para las cargas de trabajo de producción, recomendamos utilizar implementaciones en clústeres en lugar de agentes de instancia única para garantizar una alta disponibilidad y la resiliencia de los mensajes. Las implementaciones en clúster eliminan los puntos únicos de error y ofrecen una mejor tolerancia a errores.

Las implementaciones en clúster constan de tres nodos de agente de RabbitMQ distribuidos en tres zonas de disponibilidad, lo que proporciona una conmutación automática por error y garantiza que las operaciones continúen incluso si una zona de disponibilidad completa deja de estar disponible. Amazon MQ replica automáticamente los mensajes en todos los nodos para garantizar la disponibilidad durante las averías o el mantenimiento de los nodos.

Las implementaciones en clúster son esenciales en entornos de producción y están cubiertas por el [acuerdo de nivel de servicio de Amazon MQ](#).

Para obtener más información, consulte [Implementación de clústeres en Amazon MQ para RabbitMQ](#).

Paso 2: Cómo elegir el tipo de instancia de agente correcto

El rendimiento de los mensajes de un tipo de instancia de agente depende del caso de uso de su aplicación. Se recomienda usar `M7g.medium` únicamente para probar el rendimiento de una aplicación. El uso de esta instancia menor antes de utilizar instancias mayores en producción puede mejorar el rendimiento de una aplicación. En los tipos de instancias `m7g.large` y superiores, puede usar implementaciones de clústeres para obtener una alta disponibilidad y durabilidad de los mensajes. En cambio, los tipos de instancias de agente más grandes pueden controlar clientes y colas, rendimientos altos, mensajes en memoria y mensajes redundantes a nivel de producción.

Para obtener más información sobre cómo elegir el tipo de instancia correcto, consulte [Directrices de dimensionamiento en Amazon MQ para RabbitMQ](#).

Paso 3: Utilizar las colas de cuórum

Las colas de cuórum, con implementación en clústeres, deberían ser la opción predeterminada para los tipos de colas replicadas en los entornos de producción para los agentes de RabbitMQ a partir de la versión 3.13. Las colas de cuórum son un tipo de cola replicada moderna que proporciona alta fiabilidad, alto rendimiento y latencia estable.

Las colas de cuórum utilizan el algoritmo de consenso Raft para ofrecer una mejor tolerancia a errores. Cuando el nodo líder deja de estar disponible, las colas de cuórum eligen automáticamente a un nuevo líder por mayoría de votos, lo que garantiza que la entrega de mensajes continúe con las mínimas interrupciones. Como cada nodo se encuentra en una zona de disponibilidad diferente, su sistema de mensajería permanece disponible incluso si una zona de disponibilidad completa deja de estar disponible temporalmente.

Para declarar una cola de cuórum, defina el encabezado `x-queue-type` en `quorum` cuando cree las colas.

Para obtener más información sobre las colas de cuórum, incluidas las estrategias de migración y las prácticas recomendadas, consulte [Colas de cuórum en Amazon MQ para RabbitMQ](#).

Paso 4: Uso de múltiples canales

Para evitar la pérdida de conexiones, use varios canales a través de una sola conexión. Las aplicaciones deben evitar una relación de conexión individual con el canal. Recomendamos utilizar una conexión para cada proceso y, a continuación, un canal para cada subprocesso. Evite el uso excesivo de los canales para evitar fugas en ellos.

Prácticas recomendadas de la durabilidad y fiabilidad de mensajes en Amazon MQ para RabbitMQ

Antes de trasladar la aplicación a producción, siga las siguientes prácticas recomendadas para evitar la pérdida de mensajes y la sobreutilización de los recursos.

Paso 1: Uso de mensajes persistentes y colas duraderas

Los mensajes persistentes pueden ayudar a proteger la durabilidad de los datos en situaciones en las que un agente se bloquea o se reinicia. Los mensajes persistentes se escriben en el disco tan pronto como llegan. Sin embargo, a diferencia de las colas perezosas, los mensajes persistentes

se almacenan tanto en la memoria caché como en el disco, a menos que el agente necesite más memoria. En los casos en que se necesita más memoria, los mensajes se eliminan de la memoria mediante el mecanismo del agente de RabbitMQ que administra el almacenamiento de mensajes en el disco, comúnmente conocido como capa de persistencia.

Para habilitar la persistencia de mensajes, puede declarar las colas como `durable` y establecer el modo de entrega de mensajes en `persistent`. En el siguiente ejemplo, se muestra el uso de la [biblioteca de cliente Java de RabbitMQ](#) para declarar una cola duradera. Al trabajar con el AMQP 0-9-1, puede marcar los mensajes como persistentes configurando el modo de entrega como `"2"`.

```
boolean durable = true;
channel.queueDeclare("my_queue", durable, false, false, null);
```

Una vez que haya configurado la cola como duradera, puede enviar un mensaje persistente a la cola estableciendo `MessageProperties` en `PERSISTENT_TEXT_PLAIN`, como se muestra en el siguiente ejemplo.

```
import com.rabbitmq.client.MessageProperties;

channel.basicPublish("", "my_queue",
    MessageProperties.PERSISTENT_TEXT_PLAIN,
    message.getBytes());
```

Paso 2: Configuración de la confirmación del publicador y el acuse de recibo de la entrega al consumidor

Se denomina confirmación del publicador al proceso de confirmar que se ha enviado un mensaje al agente. Las confirmaciones del publicador permiten a la aplicación saber que los mensajes se han almacenado de forma fiable. También pueden ayudar a controlar el ritmo de los mensajes almacenados en el agente. Sin las confirmaciones del publicador, no es posible confirmar que un mensaje se haya procesado correctamente y puede que su agente descarte los mensajes que no pueda procesar.

Del mismo modo, cuando una aplicación cliente envía confirmación de entrega y consumo de mensajes de vuelta al agente, se conoce como acuse de recibo del consumidor. Tanto la confirmación como el acuse de recibo son esenciales para garantizar la seguridad de los datos cuando se trabaja con agentes de RabbitMQ.

El acuse de recibo de entrega del consumidor suele configurarse en la aplicación cliente. Cuando se trabaja con AMQP 0-9-1, el acuse de recibo se puede habilitar configurando el método `basic.consume`. Los clientes de AMQP 0-9-1 también pueden configurar las confirmaciones del publicador mediante el envío del método `confirm.select`.

Normalmente, el acuse de recibo de entrega se habilita en un canal. Por ejemplo, cuando se trabaja con la biblioteca de cliente Java de RabbitMQ, se puede utilizar `Channel#basicAck` para configurar un acuse de recibo positivo `basic.ack`, como se muestra en el siguiente ejemplo.

```
// 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

Los mensajes sin confirmar se deben almacenar en la memoria caché. Para limitar el número de mensajes que un consumidor captura previamente, puede establecer el parámetro [Pre-fetch](#) (Captura previa) para una aplicación cliente.

Puede configurar `consumer_timeout` para detectar cuándo los consumidores no confirman las entregas. Si el consumidor no envía un acuse de recibo dentro del tiempo de espera, el canal se cerrará y recibirá un `PRECONDITION_FAILED`. Para diagnosticar el error, usa la [UpdateConfiguration](#) API para aumentar el `consumer_timeout` valor.

Paso 3: Mantener las colas cortas

En las implementaciones de clúster, las colas con un gran número de mensajes pueden provocar una sobreutilización de recursos. Cuando un agente está sobreutilizado, el reinicio de un agente de Amazon MQ para RabbitMQ puede degradar aún más el rendimiento. Si se reinicia, los agentes sobreutilizados podrían dejar de responder en el estado `REBOOT_IN_PROGRESS`.

Durante los [períodos de mantenimiento](#), Amazon MQ realiza todos los trabajos de mantenimiento de a un nodo por vez para garantizar que el agente permanezca operativo. Como resultado, es posible que las colas deban sincronizarse a medida que cada se vaya reanudando la operación de cada nodo. Durante la sincronización, los mensajes que deben replicarse en los espejos se cargan en la memoria del volumen correspondiente de Amazon Elastic Block Store (Amazon EBS) para procesarlos en lotes. El procesamiento de mensajes en lotes permite agilizar la sincronización de las colas.

Si las colas se mantienen cortas y los mensajes son pequeños, las colas se sincronizan correctamente y reanudan la operación según lo previsto. Sin embargo, si la cantidad de datos de un lote se acerca al límite de memoria del nodo, el nodo genera una alarma de memoria elevada y se pausa la sincronización de colas. Puede confirmar el uso de la memoria comparando las [métricas del RabbitMemUsed nodo RabbitMqMemLimit intermediario con las del nodo intermediario CloudWatch](#). La sincronización no se puede completar hasta que se consuman o eliminen los mensajes, o se reduzca el número de mensajes del lote.

Si la sincronización de colas está en pausa por una implementación de clúster, recomendamos consumir o eliminar mensajes para reducir el número de mensajes en las colas. Una vez que se reduzca la profundidad de la cola y se complete su sincronización, el estado del agente cambiará a `RUNNING`. Para resolver una sincronización de cola en pausa, también puede aplicar una política para [reducir el tamaño del lote de sincronización de colas](#).

También puedes definir políticas de eliminación automática y TTL para reducir de forma proactiva el uso de recursos y NACKs evitar que los consumidores lo hagan al mínimo. Poner los mensajes en cola en el bróker requiere un uso intensivo de la CPU, por lo que un número elevado de ellos puede afectar al rendimiento del bróker. NACKs

Prácticas recomendadas para lograr la eficiencia y optimización del rendimiento en Amazon MQ para RabbitMQ

Puede optimizar el rendimiento de su agente Amazon MQ para RabbitMQ maximizando el rendimiento, minimizando la latencia y garantizando un uso eficiente de los recursos. Siga las siguientes prácticas recomendadas para optimizar el rendimiento de su aplicación.

Paso 1: Mantenga el tamaño de los mensajes por debajo de 1 MB

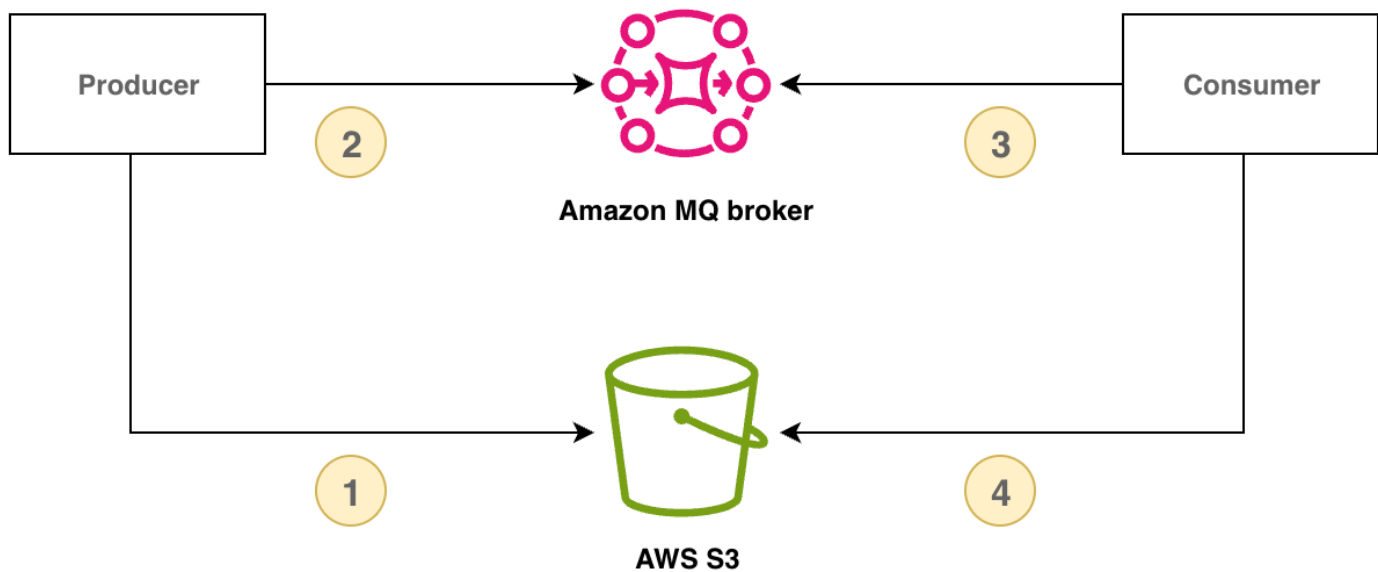
Recomendamos mantener los mensajes en un tamaño inferior a 1 megabyte (MB) para obtener un rendimiento y una fiabilidad óptimos.

De forma predeterminada, RabbitMQ 3.13 admite mensajes de hasta 128 MB, pero los mensajes de gran tamaño pueden activar alarmas de memoria impredecibles que bloquean la publicación y pueden generar una gran presión de memoria al replicar los mensajes en los nodos. Los mensajes sobredimensionados también pueden afectar a los procesos de reinicio y recuperación de los agentes, lo que aumenta los riesgos para la continuidad del servicio y puede provocar una degradación del rendimiento.

Almacene y recupere cargas útiles de gran tamaño mediante el patrón de verificación de notificaciones

Para gestionar mensajes de gran tamaño, puede implementar el patrón de verificación de notificaciones almacenando la carga útil del mensaje en un almacenamiento externo y enviando únicamente el identificador de referencia de la carga útil a través de RabbitMQ. El consumidor utiliza el identificador de referencia de la carga útil para recuperar y procesar el mensaje de gran tamaño.

En el siguiente diagrama se muestra cómo utilizar Amazon MQ para RabbitMQ y Amazon S3 para implementar el patrón de verificación de notificaciones.



En el siguiente ejemplo, se muestra este patrón con Amazon MQ, el [AWS SDK para Java 2.x](#) y [Amazon S3](#):

1. En primer lugar, defina una clase de mensaje que contenga el identificador de referencia de Amazon S3.

```
class Message {  
    // Other data fields of the message...  
  
    public String s3Key;  
    public String s3Bucket;  
}
```

2. Cree un método de publicación que almacene la carga útil en Amazon S3 y envíe un mensaje de referencia a través de RabbitMQ.

```
public void publishPayload() {  
    // Store the payload in S3.  
    String payload = PAYLOAD;  
    String prefix = S3_KEY_PREFIX;  
    String s3Key = prefix + "/" + UUID.randomUUID();  
    s3Client.putObject(PutObjectRequest.builder()  
        .bucket(S3_BUCKET).key(s3Key).build(),  
        RequestBody.fromString(payload));  
  
    // Send the reference through RabbitMQ.
```

```
Message message = new Message();
message.s3Key = s3Key;
message.s3Bucket = S3_BUCKET;
// Assign values to other fields in your message instance.

publishMessage(message);
}
```

3. Implemente un método de consumo que recupere la carga útil de Amazon S3, la procese y elimine el objeto de Amazon S3.

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

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

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

Paso 2: Uso **basic.consume** y consumo de larga duración

Usar `basic.consume` con un consumidor de larga duración es más eficiente que sondear los mensajes individuales utilizando `basic.get`. Para obtener más información, consulte [Sondeo de mensajes individuales](#).

Paso 3: Configurar la captura previa

Puede utilizar el valor de captura previa de RabbitMQ para optimizar la forma en que los consumidores consumen los mensajes. RabbitMQ implementa el mecanismo de captura previa de canales que proporciona AMQP 0-9-1 mediante la aplicación del recuento de captura previa a los consumidores en lugar de a los canales. El valor de captura previa se utiliza para especificar cuántos mensajes se envían al consumidor en un momento dado. De forma predeterminada, RabbitMQ establece un tamaño de búfer ilimitado para las aplicaciones cliente.

Hay varios factores a tener en cuenta al establecer un recuento de captura previa para los consumidores de RabbitMQ. Primero, considere el entorno y la configuración de los consumidores. Debido a que los consumidores necesitan mantener todos los mensajes en la memoria mientras se procesan, un alto valor de captura previa puede tener un impacto negativo en el rendimiento de los consumidores y, en algunos casos, puede provocar el bloqueo de todos los consumidores juntos. Del mismo modo, el propio agente de RabbitMQ guarda todos los mensajes que envía en la memoria caché hasta que recibe el acuse de recibo del consumidor. Un alto valor de captura previa puede hacer que el servidor de RabbitMQ se quede sin memoria rápidamente si el reconocimiento automático no está configurado para los consumidores y si los consumidores tardan un tiempo relativamente largo en procesar mensajes.

Teniendo en cuenta las consideraciones anteriores, recomendamos establecer siempre un valor de captura previa para evitar situaciones en las que un agente de RabbitMQ o sus consumidores se queden sin memoria debido a un gran número de mensajes sin procesar o sin reconocer. Si necesita optimizar sus agentes para que procesen grandes volúmenes de mensajes, puede probarlos junto con los consumidores utilizando un intervalo de recuentos de captura previa para determinar el valor en el que la sobrecarga de red se vuelve en gran medida insignificante en comparación con el tiempo que tarda un consumidor en procesar mensajes.

Note

- Si las aplicaciones cliente se han configurado para confirmar automáticamente la entrega de mensajes a los consumidores, no servirá de nada establecer un valor de captura previa.
- Todos los mensajes que capturados previamente se eliminan de la cola.

En el siguiente ejemplo, se muestra cómo establecer un valor de captura previa de 10 para un solo consumidor utilizando la biblioteca de clientes Java de RabbitMQ.

```
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

En la biblioteca de clientes Java de RabbitMQ, el valor predeterminado para el indicador `global` se establece en `false`, por lo que el ejemplo anterior se puede escribir simplemente como `channel.basicQos(10)`.

Paso 4: Utilice Celery 5.5 o una versión posterior con las colas de cuórum

[Python Celery](#), un sistema de colas de tareas distribuidas, puede generar muchos mensajes no críticos cuando se experimenta una gran carga de tareas. Esta actividad adicional del agente puede provocar [the section called “RABBITMQ_MEMORY_ALARM”](#) y dar lugar a la falta de disponibilidad del agente. Para reducir la posibilidad de que se active una alarma de memoria, haga lo siguiente:

Para todas las versiones de Celery

1. Desactiva [task_create_missing_queues](#) para reducir la pérdida de colas.
2. A continuación, desactiva `worker_enable_remote_control` para detener la creación dinámica de colas `celery@...pidbox`. Esto reducirá la pérdida de colas en el agente.

```
worker_enable_remote_control = false
```

3. Para reducir aún más la actividad de mensajes no críticos, desactive Celery y no los incluya `-E` ni `--task-events` marque [worker-send-task-events](#) al iniciar la aplicación Celery.
4. Inicie la aplicación Celery con los siguientes parámetros:

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

Para las versiones 5.5 y superiores de Celery

1. Actualice [Celery a la versión 5.5](#), la versión mínima que admite las colas de cuórum, o a una versión posterior. Para ver qué versión de Celery está usando, utilice `celery --version`. Para obtener más información sobre los cuórums, consulte [the section called “Colas de cuórum”](#).
2. Después de actualizar a Celery 5.5 o una versión posterior, configure `task_default_queue_type` en modo [“cuórum”](#).
3. A continuación, también debe activar la opción Publicar confirmaciones en [Broker Transport Options](#):

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

Prácticas recomendadas de supervisión y resiliencia de la red en Amazon MQ para RabbitMQ

La resiliencia de la red y las métricas de los agentes de supervisión son esenciales para mantener la fiabilidad de las aplicaciones de mensajería. Siga las siguientes prácticas recomendadas para implementar mecanismos de recuperación automática y estrategias de supervisión de recursos.

Paso 1: Recuperación automática de fallas de red

Se recomienda habilitar siempre la recuperación automática de red para evitar un tiempo de inactividad significativo en caso de falla de las conexiones del cliente con los nodos de RabbitMQ. La biblioteca de cliente Java de RabbitMQ admite la recuperación automática de red de forma predeterminada, a partir de la versión 4.0.0.

[La recuperación automática de la conexión se activa si se produce una excepción no controlada en el I/O bucle de la conexión, si se detecta el tiempo de espera de una operación de lectura del socket o si el servidor pierde un latido.](#)

En caso de falla en la conexión inicial entre un cliente y un nodo de RabbitMQ, no se activará la recuperación automática. Recomendamos escribir el código de la aplicación para tener en cuenta los errores de conexión iniciales al volver a intentar la conexión. En el siguiente ejemplo, se muestran fallas al reintentar iniciar la red mediante la biblioteca de cliente Java de RabbitMQ.

```
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

Si una aplicación cierra una conexión con el método `Connection.Close`, la recuperación automática de red no se activará ni se disparará.

Paso 2: Supervise las métricas y las alarmas de los agentes

Le recomendamos que supervise periódicamente [CloudWatch las métricas](#) y las alarmas de su agente de Amazon MQ for RabbitMQ para identificar y abordar posibles problemas antes de que afecten a su aplicación de mensajería. La supervisión proactiva es esencial para mantener una aplicación de mensajería flexible y garantizar un rendimiento óptimo.

Amazon MQ for RabbitMQ publica métricas CloudWatch que proporcionan información sobre el rendimiento de los corredores, la utilización de los recursos y el flujo de mensajes. Las métricas clave que se deben supervisar incluyen el uso de memoria y el uso del disco. Puede configurar [CloudWatch alarmas](#) para cuando su corredor se acerque a los límites de recursos o experimente una degradación del rendimiento.

Supervise las siguientes métricas esenciales:

RabbitMQMemUsed y RabbitMQMemLimit

Supervise el uso de la memoria para evitar que las alarmas de memoria puedan bloquear la publicación de mensajes.

RabbitMQDiskFree y RabbitMQDiskFreeLimit

Supervise el uso del disco para evitar problemas de espacio en disco que puedan provocar errores en el agente.

En el caso de las implementaciones de clústeres, supervise también las [métricas específicas de los nodos](#) para identificar problemas específicos de los nodos.

Note

Para obtener más información sobre cómo evitar una alarma de memoria excesiva, consulte [Abordar y evitar una alarma de memoria excesiva](#).

Tutoriales de RabbitMQ

En los siguientes tutoriales, se muestra cómo configurar y utilizar RabbitMQ en Amazon MQ. Para obtener más información acerca de cómo trabajar con bibliotecas de cliente compatibles en distintos lenguajes de programación como Node.js, Python, .NET y otros, consulte el tema sobre [tutoriales de RabbitMQ](#) en la Guía de introducción a RabbitMQ.

Temas

- [Edición de las preferencias del agente](#)
- [Uso de Python Pika con Amazon MQ para RabbitMQ](#)
- [Resolver la sincronización de colas en pausa de RabbitMQ](#)
- [Reducir el número de conexiones y canales](#)
- [Paso 2: Conectar al agente una aplicación basada en una JVM](#)
- [Paso 3: \(Opcional\) Conectarse a una AWS Lambda función](#)
- [Uso de la autenticación y autorización de OAuth 2.0 para Amazon MQ para RabbitMQ](#)
- [Uso de la autenticación y autorización de IAM para Amazon MQ for RabbitMQ](#)
- [Uso de la autenticación y autorización LDAP para Amazon MQ for RabbitMQ](#)
- [Uso de la autenticación y autorización HTTP para Amazon MQ for RabbitMQ](#)
- [Uso de la autenticación con certificado SSL para Amazon MQ para RabbitMQ](#)
- [Uso de mTLS para AMQP y puntos finales de administración](#)
- [Conexión de la aplicación JMS](#)

Edición de las preferencias del agente

Puede editar las preferencias de su corredor, como habilitar o deshabilitar CloudWatch los registros, utilizando el. Consola de administración de AWS

Editar opciones del agente de RabbitMQ

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, seleccione su corredor (por ejemplo MyBroker) y, a continuación, elija Editar.
3. En la **MyBroker** página de edición, en la sección Especificaciones, seleccione una versión de Broker Engine o un tipo de instancia de Broker.

4. En la sección CloudWatch Registros, haga clic en el botón de alternancia para activar o desactivar los registros generales. No se requieren más pasos.

 Note

- En el caso de los corredores de RabbitMQ, Amazon MQ utiliza automáticamente un rol vinculado a servicios (SLR) para publicar los registros generales. CloudWatch Para obtener más información, consulte [the section called “Cómo utilizar roles vinculados a servicios”](#)
- Amazon MQ no admite registros de auditoría para los agentes de RabbitMQ.

5. En la sección Maintenance (Mantenimiento), configure el programa de mantenimiento del agente:

Para actualizar el bróker a nuevas versiones a medida que se vayan publicando, seleccione Activar las actualizaciones automáticas de las AWS versiones secundarias. Las actualizaciones automáticas se producen durante el periodo de mantenimiento definido por el día de la semana, la hora del día (en formato de 24 horas) y la zona horaria (UTC de forma predeterminada).

6. Elija Schedule modifications (Programar modificaciones).

 Note

Si elige únicamente Enable automatic minor version upgrades (Habilitar actualizaciones automáticas de versiones secundarias), el botón cambia a Save (Guardar) porque no es necesario reiniciar el agente.

Las preferencias se aplican al agente en el momento especificado.

Uso de Python Pika con Amazon MQ para RabbitMQ

El siguiente tutorial muestra cómo configurar un cliente [Python Pika](#) con TLS configurado para conectarse a un agente de Amazon MQ para RabbitMQ. Pika es una implementación de Python del protocolo AMQP 0-9-1 para RabbitMQ. Este tutorial le guía a través de la instalación de Pika, la declaración de una cola, la configuración de un editor para enviar mensajes a la bolsa predeterminada del corredor y la configuración de un consumidor para que reciba los mensajes de la cola.

Temas

- [Requisitos previos](#)
- [Permisos](#)
- [Paso uno: crear un cliente Python Pika básico](#)
- [Paso dos: crear un editor y enviar un mensaje](#)
- [Paso tres: Crea un consumidor y recibe un mensaje](#)
- [Paso cuatro: \(opcional\) configurar un bucle de eventos y consumir mensajes](#)
- [Sigüientes pasos](#)

Requisitos previos

Para completar los pasos de este tutorial, necesita los siguientes requisitos previos:

- Un agente de Amazon MQ para RabbitMQ. Para obtener más información, consulte [Creación de un agente de Amazon MQ para RabbitMQ](#).
- Instalación de [Python 3](#) para su sistema operativo.
- Instalación de [Pika](#) con Python pip. Para instalar Pika, abra una nueva ventana de terminal y ejecute lo siguiente.

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

Permisos

Para este tutorial, necesitas al menos un usuario del agente de Amazon MQ para RabbitMQ con permiso de escritura y lectura en un anfitrión. La siguiente tabla describe los permisos mínimos necesarios como patrones de expresión regular (regex).

Tags	Regex Configure (Configuración)	Regex Write (Escritura)	Regex Read (Lectura)
none		.*	.*

Los permisos de usuario enumerados solo proporcionan permisos de lectura y escritura al usuario, sin conceder acceso al complemento de administración para realizar operaciones administrativas

en el agente. Puede restringir aún más los permisos proporcionando patrones regexp que limiten el acceso del usuario a las colas especificadas. Por ejemplo, si se cambia el patrón regexp de lectura por `^[hello world].*`, el usuario solo tendrá permiso de lectura desde las colas que empiecen por `hello world`.

Para obtener más información acerca de cómo crear usuarios de RabbitMQ y administrar etiquetas y permisos de usuario, consulte [Usuarios de agentes de Amazon MQ para RabbitMQ](#).

Paso uno: crear un cliente Python Pika básico

Para crear una clase base de cliente Python Pika que defina un constructor y proporcione el contexto SSL necesario para la configuración de TLS al interactuar con un agente de Amazon MQ para RabbitMQ, haga lo siguiente.

1. Abra una nueva ventana de terminal, cree un nuevo directorio para su proyecto y navegue hasta él.

```
$ mkdir pika-tutorial
$ cd pika-tutorial
```

2. Cree un nuevo archivo denominado `basicClient.py`, que contenga el siguiente código Python.

```
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()
```

Ahora puede definir clases adicionales para el editor y consumidor que heredan de `BasicPikaClient`.

Paso dos: crear un editor y enviar un mensaje

Para crear un editor que declare una cola y envíe un solo mensaje, haga lo siguiente.

1. Copie el contenido del siguiente ejemplo de código y guárdelo localmente como `publisher.py` en el mismo directorio creado en el paso anterior.

```
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()
```

La clase `BasicMessageSender` hereda de `BasicPikaClient` e implementa métodos adicionales para delimitar una cola, enviar un mensaje a la cola y cerrar conexiones. El código de ejemplo envía un mensaje al intercambio predeterminado, con una clave de enrutamiento igual al nombre de la cola.

2. En `if __name__ == "__main__":`, sustituya los parámetros pasados a la declaración constructor `BasicMessageSender` con la siguiente información.
 - **<broker-id>**: ID único que genera Amazon MQ para el agente. Puede consultar el ID a partir del ARN de su agente. Por ejemplo, si se le da el ARN `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, el ID del agente sería `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
 - **<username>**: el nombre de usuario de un agente con permisos suficientes para escribir mensajes al agente.
 - **<password>**: la contraseña de usuario de un agente con permisos suficientes para escribir mensajes al agente.
 - **<region>**— La AWS región en la que creó su bróker Amazon MQ para RabbitMQ. Por ejemplo, `us-west-2`.
3. Ejecute el siguiente comando en el mismo directorio que creó `publisher.py`.

```
$ python3 publisher.py
```

Si el código se ejecuta correctamente, aparecerá el siguiente resultado en la ventana de terminal.

```
Trying to declare queue(hello world queue)...
Sent message. Exchange: , Routing Key: hello world queue, Body: b'Hello World!'
```

Paso tres: Crea un consumidor y recibe un mensaje

Para crear un consumidor que reciba un solo mensaje de la cola, haga lo siguiente.

1. Copie el contenido del siguiente ejemplo de código y guárdelo localmente como `consumer.py` en el mismo directorio.

```
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()
```

Al igual que el editor que creó en el paso anterior, `BasicMessageReceiver` hereda `BasicPikaClient` e implementa métodos adicionales para recibir un solo mensaje y cerrar las conexiones.

2. En la declaración `if __name__ == "__main__":`, sustituya los parámetros pasados al constructor `BasicMessageReceiver` por su información.
3. Ejecute el siguiente comando en el directorio de su proyecto.

```
$ python3 consumer.py
```

Si el código se ejecuta correctamente, verá el cuerpo del mensaje y los encabezados, incluida la clave de enrutamiento, en la ventana del terminal.

```
<Basic.GetOk(['delivery_tag=1', 'exchange=', 'message_count=0',  
'redelivered=False', 'routing_key=hello world queue'])> <BasicProperties> b'Hello  
World!'
```

Paso cuatro: (opcional) configurar un bucle de eventos y consumir mensajes

Para consumir múltiples mensajes de una cola, utilice el método [basic_consume](#) de Pika y una función de devolución de llamada como se muestra a continuación

1. En `consumer.py`, agregue la siguiente definición de método a la clase `BasicMessageReceiver`.

```
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. En `consumer.py`, dentro de `if __name__ == "__main__":`, invoque el método `consume_messages` que definió en el paso anterior.

```
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. Ejecute `consumer.py` de nuevo, y si se ejecuta correctamente, los mensajes en cola se mostrarán en la ventana del terminal.

```
[*] Waiting for messages. To exit press CTRL+C
[x] Received b'Hello World!'
[x] Received b'Hello World!'
...
```

Siguientes pasos

- Para obtener más información sobre otras bibliotecas cliente RabbitMQ compatibles, consulte [Documentación del cliente de RabbitMQ](#) en el sitio web de RabbitMQ.

Resolver la sincronización de colas en pausa de RabbitMQ

En una [implementación de clúster](#) de Amazon MQ para RabbitMQ, los mensajes publicados en cada cola se replican en tres nodos de agente. Esta replicación, conocida como espejo, proporciona alta disponibilidad para los agentes de RabbitMQ. En una implementación de clúster, las colas constan de un nodo principal y una o más réplicas. Cada operación aplicada a una cola replicada,

incluidos los mensajes en cola, se aplica primero a la cola principal y, a continuación, se replica en sus espejos.

Por ejemplo, considere una cola replicada en tres nodos: el nodo principal (`main`) y dos espejos (`mirror-1` y `mirror-2`). Si todos los mensajes de esta cola replicada se propagan correctamente a todos los espejos, la cola se sincroniza. Si un nodo (`mirror-1`) deja de estar disponible durante un intervalo de tiempo, la cola sigue funcionando y puede continuar poniendo mensajes en cola. Sin embargo, para que la cola se sincronice, los mensajes publicados en `main` mientras `mirror-1` no está disponible deben replicarse en `mirror-1`.

Para obtener más información acerca de la replicación, consulte el tema sobre [Colas replicadas clásicas](#) en el sitio web de RabbitMQ.

Mantenimiento y sincronización de colas

Durante los [periodos de mantenimiento](#), Amazon MQ realiza todos los trabajos de mantenimiento de a un nodo por vez para garantizar que el agente permanezca operativo. Como resultado, es posible que las colas deban sincronizarse a medida que cada se vaya reanudando la operación de cada nodo. Durante la sincronización, los mensajes que deben replicarse en los espejos se cargan en la memoria del volumen correspondiente de Amazon Elastic Block Store (Amazon EBS) para procesarlos en lotes. El procesamiento de mensajes en lotes permite agilizar la sincronización de las colas.

Si las colas se mantienen cortas y los mensajes son pequeños, las colas se sincronizan correctamente y reanudan la operación según lo previsto. Sin embargo, si la cantidad de datos de un lote se acerca al límite de memoria del nodo, el nodo genera una alarma de memoria elevada y se pausa la sincronización de colas. Puede confirmar el uso de la memoria comparando las [métricas del `RabbitMemUsed` nodo `RabbitMqMemLimit` intermediario con las del nodo intermediario `CloudWatch`](#). La sincronización no se puede completar hasta que se consuman o eliminen los mensajes, o se reduzca el número de mensajes del lote.

Note

La reducción del tamaño del lote de sincronización de colas puede generar mayor número de transacciones de replicación.

Para resolver una sincronización de colas en pausa, siga los pasos que se indican en este tutorial, que muestra la aplicación de una política `ha-sync-batch-size` y reinicio de la sincronización de colas.

Temas

- [Requisitos previos](#)
- [Paso 1: Aplicar una política `ha-sync-batch-size`](#)
- [Paso 2: Reiniciar la sincronización de colas](#)
- [Siguiendo pasos](#)
- [Recursos relacionados](#)

Requisitos previos

Para este tutorial, debe tener un usuario de agente de Amazon MQ para RabbitMQ con permisos de administrador. Puede utilizar el usuario administrador que generó cuando creó el agente por primera vez, u otro usuario que podría haber creado después. La siguiente tabla proporciona la etiqueta y los permisos de usuario administrador necesarios como patrones de expresiones regulares (regex).

Tags	Regex Read (Lectura)	Regex Configure (Configuración)	Regex Write (Escritura)
administrator	.*	.*	.*

Para obtener más información acerca de cómo crear usuarios de RabbitMQ y administrar etiquetas y permisos de usuario, consulte [Usuarios de agentes de Amazon MQ para RabbitMQ](#).

Paso 1: Aplicar una política **ha-sync-batch-size**

Estos procedimientos se utilizan para agregar una política que se aplica a todas las colas creadas en el agente. Puede utilizar la consola web o la API de administración de RabbitMQ. Para obtener más información, consulte [Complemento de administración](#) en el sitio web de RabbitMQ.

Para aplicar una política **ha-sync-batch-size** a través de la consola web de RabbitMQ

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En el panel de navegación izquierdo, elija Brokers (Agentes).

3. En la lista de agentes, elija el nombre del agente al que desea aplicar la nueva política.
4. En la sección Connections (Conexiones) de la página del agente, elija la URL de la consola web de RabbitMQ. La consola web de RabbitMQ se abre en una nueva pestaña o ventana del navegador.
5. Inicie sesión en la consola web de RabbitMQ con las credenciales del usuario administrador del agente.
6. En la consola web de RabbitMQ, elija Admin (Administrador) en la parte superior de la página.
7. En la página Admin (Administrador), elija Policies (Políticas) en el panel de navegación derecho.
8. En la página Policies (Políticas), puede ver una lista de las Políticas de usuario actuales del agente. Debajo de las políticas de usuario, expanda Add/update a policy (Agregar/actualizar una política).

 Note

De forma predeterminada, los clústeres de Amazon MQ para RabbitMQ se crean con una política de agente inicial con el nombre `ha-a11-AWS-OWNED-DO-NOT-DELETE`. Amazon MQ administra esta política para garantizar que todas las colas del agente se repliquen en los tres nodos y que las colas se sincronicen automáticamente.

9. Para crear una nueva política de agente, en Add/update a policy (Agregar/actualizar una política), haga lo siguiente:
 - a. En Name (Nombre), escriba un nombre para su política, por ejemplo, **batch-size-policy**.
 - b. En Pattern (Patrón), ingrese el patrón regexp `.*` para que la política coincida con todas las colas del agente.
 - c. En Apply to (Aplicar a), elija Exchanges and queues (Intercambios y colas) en el menú desplegable.
 - d. En Priority (Prioridad), ingrese un número entero mayor que todas las demás políticas aplicadas al vhost. Puede aplicar exactamente un conjunto de definiciones de políticas a las colas e intercambios de RabbitMQ en cualquier momento dado. RabbitMQ elige la política de coincidencia con el valor de prioridad más alto. Para obtener más información acerca de las prioridades de las políticas y cómo combinar políticas, consulte el tema sobre [políticas](#) en la documentación del servidor de RabbitMQ.

e. En Definition (Definición), agregue los siguientes pares clave-valor:

- **ha-sync-batch-size=100**. Elija Número en la lista desplegable.

 Note

Es posible que deba ajustar y calibrar el valor de `ha-sync-batch-size` en función del número y el tamaño de los mensajes sin sincronizar de las colas.

- **ha-mode=all**. Elija String (Cadena) en el menú desplegable.

 Important

Se requiere la definición `ha-mode` para todas las políticas relacionadas con la alta disponibilidad. Si se omite, se produce un error de validación.

- **ha-sync-mode=automatic**. Elija String (Cadena) en el menú desplegable.

 Note

Se requiere la definición `ha-sync-mode` para todas las políticas personalizadas. Si se omite, Amazon MQ la agrega automáticamente.

f. Elija Add/update policy (Agregar/actualizar política).

10. Confirme que la nueva política aparezca en la lista de políticas de usuario.

Para aplicar una política **ha-sync-batch-size** a través de la API de administración de RabbitMQ

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En el panel de navegación izquierdo, elija Brokers (Agentes).
3. En la lista de agentes, elija el nombre del agente al que desea aplicar la nueva política.
4. En la sección Connections (Conexiones) de la página del agente, anote la URL de la consola web de RabbitMQ. Este es el punto de enlace del agente que se utiliza en una solicitud HTTP.
5. Abra un terminal nuevo o la ventana de línea de comandos que desee.
6. Para crear una nueva política de agente, ingrese el comando `curl`. Este comando asume que hay una cola en el vhost / predeterminado, que está codificado como `%2F`.

 Note

Sustituya *username* y por *password* las credenciales de inicio de sesión de administrador de su corredor. Puede que tengas que ajustar y calibrar el valor de `ha-sync-batch-size` (*100*) en función del número y el tamaño de los mensajes no sincronizados de tus colas. Reemplace el punto de enlace del agente por la URL que anotó anteriormente.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \  
-d '{"pattern":".*", "priority":1, "definition":{"ha-sync-batch-size":100, "ha-  
mode":"all", "ha-sync-mode":"automatic"}}' \  
https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-west-2.amazonaws.com/api/  
policies/%2Fbatch-size-policy
```

7. Para confirmar que se haya agregado la nueva política a las políticas de usuario de su agente, ingrese el comando `curl` para ver una lista de todas las políticas de agente.

```
curl -i -u username:password https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-  
west-2.amazonaws.com/api/policies
```

Paso 2: Reiniciar la sincronización de colas

Después de aplicar una nueva política `ha-sync-batch-size` a su agente, reinicie la sincronización de colas.

Para reiniciar la sincronización de colas a través de la consola web de RabbitMQ

 Note

Para abrir la consola web de RabbitMQ, consulte las instrucciones anteriores en el paso 1 de este tutorial.

1. En la consola web de RabbitMQ, en la parte superior de la página, elija **Queues (Colas)**.

2. En la página **Queues (Colas)**, en **All queues (Todas las colas)**, busque la cola en pausa. En la fila **Policy (Política)**, la cola debe mostrar el nombre de la nueva política que ha creado (por ejemplo, `batch-size-policy`).
3. Para reiniciar el proceso de sincronización con un tamaño de lote reducido, primero cancele la sincronización de la cola. A continuación, reinicie la sincronización de colas.

Note

Si la sincronización se detiene y no se completa correctamente, intente reducir el valor de `ha-sync-batch-size` y volver a reiniciar la sincronización de colas.

Siguientes pasos

- Una vez que la cola se sincronice correctamente, puede controlar la cantidad de memoria que utilizan sus nodos de RabbitMQ consultando la métrica de Amazon. CloudWatch `RabbitMQMemUsed`. También puede consultar la métrica `RabbitMQMemLimit` para monitorear el límite de memoria de un nodo. Para obtener más información, consulte [Acceso a CloudWatch las métricas de Amazon MQ](#) y [CloudWatch Métricas disponibles de Amazon MQ para corredores de RabbitMQ](#).
- Para evitar la sincronización de colas en pausa, recomendamos mantener las colas cortas y procesar mensajes. Para cargas de trabajo con tamaños de mensaje más grandes, también recomendamos actualizar el tipo de instancias del agente a un tamaño de instancia más grande con más memoria. Para obtener más información sobre los tipos de instancias y la edición de preferencias del agente, consulte [Edición de las preferencias del agente](#).
- Cuando crea un nuevo agente de Amazon MQ para RabbitMQ, Amazon MQ aplica un conjunto de políticas y límites de anfitriones virtuales predeterminados para optimizar el rendimiento del agente. Si su agente no tiene las políticas y los límites predeterminados recomendados, le recomendamos que los cree usted mismo. Para obtener más información acerca de cómo crear políticas y límites de vhost predeterminados, consulte <https://docs.aws.amazon.com/amazon-mq/latest/developer-guide/rabbitmq-defaults.html>.

Recursos relacionados

- [UpdateBrokerInput](#)— Utilice esta propiedad de agente para actualizar un tipo de instancia de agente mediante la API Amazon MQ.

- [Parámetros y políticas](#)(Documentación del servidor de RabbitMQ): obtenga más información acerca de los parámetros y las políticas de RabbitMQ en el sitio web de RabbitMQ.
- [API HTTP de administración de RabbitMQ](#): obtenga más información acerca de la API de administración de RabbitMQ.

Reducir el número de conexiones y canales

Las conexiones al agente de RabbitMQ para Amazon MQ las pueden cerrar las aplicaciones cliente, o bien manualmente mediante la consola web de RabbitMQ. Para cerrar una conexión mediante la consola web de RabbitMQ, haga lo siguiente:

1. Inicie sesión en la consola Consola de administración de AWS web RabbitMQ de su bróker y ábrala.
2. En la consola de RabbitMQ, elija la pestaña Connections (Conexiones).
3. En la página Connections (Conexiones), en All connections (Todas las conexiones), elija el nombre de la conexión que desea cerrar en la lista.
4. En la página de detalles de la conexión, elija Close this connection (Cerrar esta conexión) para ampliar la sección y, a continuación, elija Force Close (Forzar cierre). Opcionalmente, puede reemplazar el texto predeterminado de Reason (Razón) con una descripción propia. RabbitMQ de Amazon MQ devolverá la razón que especifique al cliente cuando cierre la conexión.
5. Elija OK (Aceptar) en el cuadro de diálogo para confirmar y cerrar la conexión.

Cuando se cierra una conexión, los canales asociados a la conexión cerrada también se cierran.

Note

Es posible que las aplicaciones cliente estén configuradas para restablecer automáticamente las conexiones con el agente después de cerrarlas. En tal caso, cerrar las conexiones desde la consola web del agente no será suficiente para reducir el recuento de conexiones o canales.

En el caso de agentes sin acceso público, se pueden bloquear temporalmente las conexiones denegando el tráfico entrante en el puerto del protocolo de mensajes adecuado; por ejemplo, el puerto 5671 para las conexiones AMQP. Se puede bloquear el puerto del grupo de seguridad que

se haya proporcionado a Amazon MQ al crear el agente. Para obtener más información sobre la modificación de un grupo de seguridad de VPC, consulte [Agregar reglas a un grupo de seguridad](#) en la Guía del usuario de Amazon VPC.

Paso 2: Conectar al agente una aplicación basada en una JVM

Después de crear un agente de RabbitMQ, puede conectar su aplicación con él. Los siguientes ejemplos muestran cómo puede utilizar la [biblioteca de cliente de Java de RabbitMQ](#) para crear una conexión con su agente, crear una cola y enviar un mensaje. Puede conectarse con agentes de RabbitMQ mediante bibliotecas de cliente de RabbitMQ compatibles para varios idiomas. Para obtener más información de las bibliotecas de cliente de RabbitMQ compatibles, consulte el tema sobre [bibliotecas de cliente de RabbitMQ y herramientas para desarrolladores](#).

Requisitos previos

Note

Los siguientes requisitos previos solo se aplican a los agentes de RabbitMQ que se crean sin accesibilidad pública. Si está creando un agente con accesibilidad pública, puede omitirlos.

Habilitar atributos de VPC

Para asegurarse de poder acceder a su agente dentro de su VPC, debe habilitar los atributos `enableDnsHostnames` y `enableDnsSupport` de VPC. Para obtener más información, consulte el tema sobre [soporte de DNS en su VPC](#) en la Guía del usuario de Amazon VPC.

Habilitar las conexiones entrantes

1. Inicie sesión en la [consola de Amazon MQ](#).
2. En la lista de corredores, elija el nombre de su corredor (por ejemplo,). `MyBroker`
3. En la ***MyBroker*** página, en la sección Conexiones, anote las direcciones y los puertos de la URL de la consola web del bróker y de los protocolos a nivel de cable.
4. En la sección Details (Detalles), en Security and network (Seguridad y red), elija el nombre de su grupo de seguridad o



Aparecerá la página Security Groups (Grupos de seguridad) del panel de EC2.

5. En la lista de grupos de seguridad, elija el grupo de seguridad.
6. En la parte inferior de la página, elija Inbound (Entrante) y, a continuación, Edit (Editar).
7. En el cuadro de diálogo Edit inbound rules (Editar reglas entrantes), agregue una regla para cada URL o punto de enlace que desee que sea accesible públicamente (en el ejemplo a continuación se muestra cómo hacerlo para una consola web de agente).
 - a. Seleccione Add Rule (Agregar regla).
 - b. En Type (Tipo), seleccione Custom TCP (TCP personalizada).
 - c. Para Source (Fuente), deje seleccionada la opción Custom (Personalizada) y, luego, escriba la dirección IP del sistema que desea que pueda acceder a la consola web (por ejemplo, 192.0.2.1).
 - d. Seleccione Save.

El agente ahora puede aceptar las conexiones entrantes.

Agregar dependencias de Java

Si está utilizando Apache Maven para automatizar compilaciones, agregue la siguiente dependencia a su archivo `pom.xml`. Para obtener más información de los archivos Project Object Model en Apache Maven, consulte el tema sobre [introducción a POM](#).

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

Si utiliza [Gradle](#) para automatizar compilaciones, declare la siguiente dependencia.

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

Importar clases `Connection` y `Channel`

El cliente Java de RabbitMQ utiliza `com.rabbitmq.client` como su paquete de nivel superior, y las clases de API `Connection` y `Channel` representan una conexión AMQP 0-9-1 y un canal,

respectivamente. Importe las clases `Connection` y `Channel` antes de usarlas, como se muestra en el ejemplo a continuación.

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

Crear un **ConnectionFactory** y conectarlo con su agente

Utilice el siguiente ejemplo para crear una instancia de la clase `ConnectionFactory` con los parámetros dados. Use el método `setHost` para configurar el punto de enlace del agente que anotó antes. Para conexiones de nivel de cable AMQPS, utilice el puerto 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();
```

Publicar un mensaje en un intercambio

Puede usar `Channel.basicPublish` para publicar mensajes en un intercambio. El siguiente ejemplo utiliza la clase `Builder` de AMQP para compilar un objeto de propiedades de mensaje con tipo de contenido `plain/text`.

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

```
messageBodyBytes);
```

Note

Tenga en cuenta que `BasicProperties` es una clase interna de la clase titular generada automáticamente, que es AMQP.

Suscribirse a una cola y recibir un mensaje

Para recibir un mensaje, puede suscribirse a una cola a través de la interfaz `Consumer`. Una vez suscrito, los mensajes se entregarán automáticamente a medida que lleguen.

La forma más sencilla de implementar un `Consumer` es usar la subclase `DefaultConsumer`. Se puede pasar un objeto `DefaultConsumer` como parte de una ejecución `basicConsume` para configurar la suscripción como se muestra en el ejemplo a continuación.

```
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

Dado que especificamos `autoAck = false`, es necesario reconocer los mensajes que se entregaron al `Consumer`, preferentemente a través del método `handleDelivery`, como se muestra en el ejemplo.

Cerrar la conexión y desconectarse del agente

Para desconectarse del agente de RabbitMQ, cierre el canal y la conexión como se muestra a continuación.

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

Note

Para obtener más información del uso de la biblioteca de cliente de Java de RabbitMQ, consulte la [Guía de API de cliente de Java de RabbitMQ](#).

Paso 3: (Opcional) Conectarse a una AWS Lambda función

AWS Lambda puede conectarse a su agente de Amazon MQ y consumirlos. Cuando conecta un agente a Lambda, crea un [mapeo de origen de eventos](#) que lee mensajes de una cola e invoca la función [de forma sincrónica](#). El mapeo de origen de eventos que crea lee los mensajes de su agente en lotes y los convierte en una carga de Lambda en forma de objeto JSON.

Conectar a su agente con una función Lambda

1. Agregue los siguientes permisos de roles de IAM al [rol de ejecución](#) de su función Lambda.

- [mq: DescribeBroker](#)
- [ec2: CreateNetworkInterface](#)
- [ec2: DeleteNetworkInterface](#)
- [ec2: DescribeNetworkInterfaces](#)
- [ec2: DescribeSecurityGroups](#)
- [ec2: DescribeSubnets](#)
- [ec2: DescribeVpcs](#)
- [registros: CreateLogGroup](#)
- [registros: CreateLogStream](#)
- [registros: PutLogEvents](#)
- [administrador de secretos: GetSecretValue](#)

Note

Sin los permisos necesarios de IAM, su función no podrá leer correctamente los registros de los recursos de Amazon MQ.

2. (Opcional) Si ha creado un agente sin accesibilidad pública, debe realizar una de las siguientes acciones para permitir que Lambda se conecte con su agente:
 - Puede configurar una gateway NAT para cada subred pública. Para obtener más información, consulte el tema sobre [acceso a Internet y a los servicios para funciones conectadas a la VPC](#) en la AWS Lambda Guía para desarrolladores.
 - Cree una conexión entre Amazon Virtual Private Cloud (Amazon VPC) y Lambda a través de un punto de enlace de la VPC. Su Amazon VPC también debe conectarse a AWS Security Token Service (AWS STS) y a los puntos de conexión de Secrets Manager. Para obtener más información, consulte [Configuración de puntos de conexión de VPC de interfaz para Lambda](#) en la AWS Lambda Guía para desarrolladores.
3. [Configure el agente como origen de eventos](#) para una función Lambda a través de la Consola de administración de AWS. También puede usar el [create-event-source-mapping](#) AWS Command Line Interface comando.
4. Escriba algún código para que su función Lambda procese los mensajes de lo consumido desde su agente. La carga de Lambda que recupere el mapeo de origen de eventos depende del tipo de motor del agente. A continuación, se muestra un ejemplo de una carga de Lambda para una cola de RabbitMQ en Amazon MQ.

Note

En el ejemplo, test es el nombre de la cola y / es el nombre del anfitrión virtual predeterminado. Al recibir mensajes, el origen de eventos muestra los mensajes entest::/.

```
{
  "eventSource": "aws:rmq",
  "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
  "rmqMessagesByQueue": {
```

```

"test::/": [
  {
    "basicProperties": {
      "contentType": "text/plain",
      "contentEncoding": null,
      "headers": {
        "header1": {
          "bytes": [
            118,
            97,
            108,
            117,
            101,
            49
          ]
        },
        "header2": {
          "bytes": [
            118,
            97,
            108,
            117,
            101,
            50
          ]
        },
        "numberInHeader": 10
      }
    },
    "deliveryMode": 1,
    "priority": 34,
    "correlationId": null,
    "replyTo": null,
    "expiration": "60000",
    "messageId": null,
    "timestamp": "Jan 1, 1970, 12:33:41 AM",
    "type": null,
    "userId": "AIDACKCEVSQ6C2EXAMPLE",
    "appId": null,
    "clusterId": null,
    "bodySize": 80
  },
  "redelivered": false,
  "data": "eyJ0aW1lbnV0IjowLCJkYXRhIjoiQ1pybWYwR3c4T3Y0YnFMUXhENEUifQ=="
}

```

```
]
}
}
```

Para obtener más información de cómo conectar Amazon MQ a Lambda, las opciones que admite Lambda para un origen de eventos de Amazon MQ y los errores de asignación de orígenes de eventos, consulte el tema sobre el [uso de Lambda con Amazon MQ](#) en la AWS Lambda Guía para desarrolladores.

Uso de la autenticación y autorización de OAuth 2.0 para Amazon MQ para RabbitMQ

En este tutorial se describe cómo configurar la [autenticación de OAuth 2.0](#) para sus agentes de Amazon MQ para RabbitMQ que utilizan Amazon Cognito como proveedor de OAuth 2.0.

Note

Amazon Cognito no está disponible en China (Pekín) y China (Ningxia).

Important

Este tutorial es específico de Amazon Cognito, pero puede utilizar otros proveedores de identidad (IdP). Para obtener más información, consulte [Ejemplos de autenticación de OAuth 2.0](#).

En esta página

- [Requisitos previos para configurar la autenticación OAuth 2.0](#)
- [Configuración de la autenticación OAuth 2.0 con Amazon Cognito mediante AWS CLI](#)
- [Configuración de OAuth 2.0 y autenticación simple con Amazon Cognito](#)

Requisitos previos para configurar la autenticación OAuth 2.0

Puede configurar los recursos de Amazon Cognito necesarios en este tutorial mediante la implementación de la pila AWS CDK, [Pila de Amazon Cognito para el complemento OAuth 2 de](#)

[RabbitMQ](#). Si va a configurar Amazon Cognito manualmente, asegúrese de cumplir los siguientes requisitos previos antes de configurar OAuth 2.0 en su Amazon MQ para agentes de RabbitMQ:

Requisitos previos para configurar Amazon Cognito

- Configure un punto de conexión de Amazon Cognito mediante la creación de un grupo de usuarios. Para ello, consulte el blog [How to use OAuth 2.0 in Amazon Cognito: Learn about the different OAuth 2.0 grants](#).
- Cree un servidor de recursos llamado `rabbitmq` en el grupo de usuarios con los siguientes ámbitos definidos: `read:all`, `write:all`, `configure:all` y `tag:administrator`. Estos ámbitos se asociarán a los permisos de RabbitMQ.

Para obtener información sobre la creación de un servidor de recursos, consulte [Definir un servidor de recursos para su grupo de usuarios \(Consola de administración de AWS\)](#) en la Guía para desarrolladores de Amazon Cognito.

- Cree los siguientes clientes de aplicaciones.
 - Cliente de aplicación para el grupo de usuarios de tipo `Machine-to-Machine application`. Se trata de un cliente confidencial con un secreto de cliente que se utilizará para los clientes AMQP de RabbitMQ. Para obtener más información sobre los clientes de aplicaciones y cómo crear uno, consulte [Tipos de clientes de aplicaciones](#) y [Creación de un cliente de aplicaciones](#).
 - Cliente de aplicación para el grupo de usuarios de tipo `Single-page application`. Se trata de un cliente público que se utilizará para iniciar sesión en la consola de administración de RabbitMQ. Debe actualizar este cliente de aplicación para incluir el punto de conexión del agente Amazon MQ para RabbitMQ que creará en el siguiente procedimiento como URL de devolución de llamada permitida. Para obtener más información, consulte [Configuración del inicio de sesión administrado con la consola de Amazon Cognito](#).

Requisito previo para configurar Amazon MQ

- Una instalación de [Docker](#) que funcione para ejecutar un script de bash que compruebe si la configuración de OAuth 2.0 se ha realizado correctamente o no.
- versión AWS CLI `>= 2.28.23` para que añadir un nombre de usuario y una contraseña sea opcional durante la creación del agente.

Configuración de la autenticación OAuth 2.0 con Amazon Cognito mediante AWS CLI

El siguiente procedimiento muestra cómo configurar la autenticación OAuth 2.0 para sus agentes de Amazon MQ for RabbitMQ que utilizan Amazon Cognito como IdP. Este procedimiento utiliza AWS CLI para crear y configurar los recursos necesarios.

En el siguiente procedimiento, asegúrese de reemplazar los valores de los marcadores de posición, como el ID de configuración y la revisión, `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` y `<2>` por sus valores reales.

1. Cree una configuración nueva mediante el comando AWS CLI [create-configuration](#) tal y como se muestra en el siguiente ejemplo.

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

Este comando devuelve una respuesta de JSON similar a la siguiente.

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

2. Cree un archivo de configuración llamado **rabbitmq.conf** para utilizar OAuth 2.0 como método de autenticación y autorización, como se muestra en el siguiente ejemplo.

```
auth_backends.1 = oauth2
```

```
# FIXME: Update this value with the token signing key URL of your Amazon Cognito
  user pool.
# If you used the AWS CDK stack to deploy Amazon Cognito, this is one of the stack
  outputs.
auth_oauth2.jwks_url = ${RabbitMq0Auth2TestStack.JwksUri}
auth_oauth2.resource_server_id = rabbitmq
# Amazon Cognito does not include an audience field in access tokens
auth_oauth2.verify_aud = false

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

# Allow OAuth 2.0 login for RabbitMQ management console
management.oauth_enabled = true
# FIXME: Update this value with the client ID of your public application client
management.oauth_client_id
  = ${RabbitMq0Auth2TestStack.ManagementConsoleAppClientId}
# FIXME: Update this value with the base JWKS URI (without /.well-known/jwks.json)
auth_oauth2.issuer = ${RabbitMq0Auth2TestStack.Issuer}
management.oauth_scopes = rabbitmq/tag:administrator
```

Esta configuración utiliza [alias de ámbito](#) para asignar los ámbitos definidos en Amazon Cognito a ámbitos compatibles con RabbitMQ.

3. Actualice la configuración mediante el comando AWS CLI [update-configuration](#), como se muestra en el siguiente ejemplo. En este comando, añada el identificador de configuración que recibió en la respuesta al paso 1 de este procedimiento. Por ejemplo, **c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca**.

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

Este comando devuelve una respuesta de JSON similar a la siguiente.

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

4. Cree un agente con la configuración de OAuth 2.0 creada en el paso 2 de este procedimiento. Para ello, utilice el comando [create-broker](#) de AWS CLI como se muestra en el siguiente ejemplo. En este comando, indique el identificador de configuración y el número de revisión que obtuvo en las respuestas de los pasos 1 y 2, respectivamente. Por ejemplo, **c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca** y **2**.

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

Este comando devuelve una respuesta de JSON similar a la siguiente.

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

5. Compruebe que el estado del agente pasa de CREATION_IN_PROGRESS a RUNNING mediante el comando [describe-broker](#) de AWS CLI, tal como se muestra en el siguiente ejemplo. En este

comando, proporcione el ID de agente que obtuvo en el resultado del paso anterior. Por ejemplo, **b-2a1b5133-a10c-49d2-879b-8c176c34cf73**.

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

Este comando devuelve una respuesta de JSON similar a la siguiente. La siguiente respuesta es una versión abreviada del resultado completo que devuelve el comando `describe-broker`. Esta respuesta muestra el estado del agente y la estrategia de autenticación utilizada para proteger al agente. En este caso, la estrategia de autenticación `config_managed` indica que el agente utiliza el método de autenticación OAuth 2.

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

Para iniciar sesión en la consola de administración de RabbitMQ mediante OAuth2, es necesario añadir el punto de conexión del agente como una URL de devolución de llamada válida en el cliente de la aplicación Amazon Cognito correspondiente. Para obtener más información, consulte el paso 5 de la configuración de nuestro ejemplo de [pila de CDK de Amazon Cognito](#).

6. Verifique la autenticación y la autorización de OAuth 2.0 con el siguiente script de `perf-test.sh`.

Utilice este script de bash para probar la conectividad con su agente de Amazon MQ para RabbitMQ. Este script obtiene un token de Amazon Cognito y verifica si la conexión se configuró correctamente. Si se ha configurado correctamente, verá que su agente publica y consume mensajes.

Si recibe un error `ACCESS_REFUSED`, puede solucionar los problemas de configuración mediante los registros de CloudWatch de su agente. Encontrará el enlace al grupo de registros de CloudWatch de su agente en la consola de Amazon MQ.

En este script, deberá proporcionar los siguientes valores:

- `CLIENT_ID` y `CLIENT_SECRET`: puede encontrar estos valores en la App clients (Página de clientes de aplicación) de la consola Amazon Cognito.

- Dominio de Cognito: puede encontrarlo en la consola de Amazon Cognito. En Branding (Marca), seleccione Domain (Dominio). En la página Domain (Dominio), encontrará este valor en la sección Resource servers (Servidores de recursos).
- Punto de conexión del agente de Amazon MQ: puede encontrar este valor en Connections (Conexiones), en la página de detalles del agente de la consola de Amazon MQ.

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

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
application client
CLIENT_ID=${RabbitMq0Auth2TestStack.AmqpAppClientId}
CLIENT_SECRET=${RabbitMq0Auth2TestStack.AmqpAppClientSecret}

# FIXME: Update this value with the domain of your Amazon Cognito user pool
RESPONSE=$(curl -X POST ${RabbitMq0Auth2TestStack.TokenEndpoint} \
-H "Content-Type: application/x-www-form-urlencoded" \
-d
"grant_type=client_credentials&client_id=${CLIENT_ID}&client_secret=${CLIENT_SECRET}&scope=
configure:all rabbitmq/read:all rabbitmq/tag:administrator rabbitmq/write:all")

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

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

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

docker run -it --rm --ulimit nofile=40960:40960 pivotalrabbitmq/perf-test:latest \
```

```
--queue-pattern 'test-queue-%d' --queue-pattern-from 1 --queue-pattern-to
$QUEUES_COUNT \
--producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
--id "test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
--uri ${CONNECTION_STRING} \
--flag persistent --rate $PRODUCER_RATE
```

Configuración de OAuth 2.0 y autenticación simple con Amazon Cognito

Al crear un agente con autenticación OAuth 2.0, puede especificar uno de los siguientes métodos de autenticación:

- Solo con OAuth 2.0: para usar este método, no proporcione un nombre de usuario ni una contraseña al crear el agente. El [procedimiento anterior](#) muestra cómo utilizar únicamente el método de autenticación OAuth 2.0.
- Tanto la autenticación OAuth 2.0 como la autenticación simple: para utilizar este método, proporcione un nombre de usuario y una contraseña al crear el agente. Además, añada `auth_backends.2 = internal` a la configuración de su agente, como se muestra en el siguiente procedimiento.

En el siguiente procedimiento, asegúrese de reemplazar los valores de los marcadores de posición, como *<ConfigurationId>* y *<Revision>*, por sus valores reales.

1. Para utilizar ambos métodos de autenticación, cree la configuración de agente, como se muestra en el siguiente ejemplo.

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

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

auth_oauth2.scope_prefix = rabbitmq/
auth_oauth2.scope_aliases.1.alias = rabbitmq/read:all
auth_oauth2.scope_aliases.1.scope = rabbitmq/read:*/*
```

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

Esta configuración utiliza [alias de ámbito](#) para asignar los ámbitos definidos en Amazon Cognito a ámbitos compatibles con RabbitMQ.

2. Cree un agente que utilice los dos métodos de autenticación, como se muestra en el siguiente ejemplo.

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

3. Compruebe que el estado del agente y la configuración para establecer el método de autenticación se hayan realizado correctamente, tal como se describe en los pasos 5 y 6 del procedimiento [Configuración de la autenticación OAuth 2.0 con Amazon Cognito](#).

Uso de la autenticación y autorización de IAM para Amazon MQ for RabbitMQ

El siguiente procedimiento muestra cómo habilitar la autenticación y la autorización de AWS IAM para un bróker de Amazon MQ for RabbitMQ. Tras habilitar la IAM, los usuarios pueden autenticarse con las credenciales de IAM para acceder a la API de administración de AWS RabbitMQ y conectarse a través de AMQP. Para obtener más información sobre cómo funciona la autenticación de IAM con Amazon MQ para RabbitMQ, consulte [the section called “Autenticación y autorización de IAM”](#)

Requisitos previos

- AWS credenciales de administrador de la AWS cuenta propietaria del bróker Amazon MQ for RabbitMQ

- Un entorno de shell configurado con estas credenciales de administrador (mediante perfiles AWS CLI o variables de entorno)
- AWS CLI instalada y configurada
- jqProcesador JSON de línea de comandos instalado
- curlherramienta de línea de comandos instalada

Configurar la autenticación y la autorización de IAM mediante AWS CLI

1. Defina las variables de entorno

Establezca las variables de entorno necesarias para su corredor:

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

2. Habilite los tokens JWT salientes

Habilite la federación de identidades web salientes para su cuenta: AWS

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

El resultado muestra una URL de identificador de emisor única para su cuenta en este formato.
`https://<id>.tokens.sts.global.api.aws`

3. Cree el documento de política de IAM

Cree un documento de política que conceda permisos para obtener tokens de identidad web:

```
cat > policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

        "Sid": "VisualEditor0",
        "Effect": "Allow",
        "Action": [
            "sts:GetWebIdentityToken",
            "sts:TagGetWebIdentityToken"
        ],
        "Resource": "*"
    }
]
}
EOF

```

4. Cree la política de confianza

Recupera la identidad de la persona que llama y crea un documento de política de confianza:

```

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

```

5. Cree el rol de IAM

Cree el rol de IAM y adjunte la política:

```

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

```

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

6. Configure los ajustes de RabbitMQ OAuth2

Cree un archivo de configuración de RabbitMQ con los ajustes de autenticación y autorización: OAuth2

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

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

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

management.oauth_enabled = true
EOF
```

7. Actualice la configuración del corredor

Aplique la nueva configuración a su corredor:

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

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

# Apply the configuration to the broker
```

```
aws mq update-broker --broker-id $BROKER_ID --configuration Id=$CONFIG_ID,Revision=$REVISION

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

Espere a que vuelva al estado de corredor RUNNING antes de continuar con el siguiente paso.

8. Obtenga un token JWT

Asuma el rol de IAM y obtenga un token de identidad web:

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

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

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

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

9. Acceda a la API de administración de RabbitMQ

Utilice el token JWT para acceder a la API de administración de RabbitMQ:

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

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

Una respuesta correcta confirma que la autenticación de IAM funciona correctamente. La respuesta contiene información general sobre el corredor en formato JSON.

10. Conéctese a través de AMQP utilizando el token JWT

Pruebe la conectividad AMQP con el token JWT con la herramienta de prueba de rendimiento:

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

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

Si recibe un ACCESS_REFUSED error, puede solucionar los problemas de configuración utilizando los registros de su agente. CloudWatch Puede encontrar el enlace al grupo de CloudWatch registros de registros de su agente en la consola de Amazon MQ.

Uso de la autenticación y autorización LDAP para Amazon MQ for RabbitMQ

En este tutorial se describe cómo configurar la autenticación y la autorización de LDAP para su Amazon MQ para corredores de RabbitMQ mediante AWS Managed Microsoft AD

En esta página

- [Requisitos previos para configurar la autenticación y la autorización de LDAP](#)
- [Configuración de LDAP en RabbitMQ mediante CLI AWS](#)

Requisitos previos para configurar la autenticación y la autorización de LDAP

Puede configurar AWS los recursos necesarios en este tutorial mediante la implementación de la [pila de AWS CDK para la integración de LDAP de Amazon MQ para RabbitMQ](#) con. AWS Managed Microsoft AD

Esta pila de CDK crea automáticamente todos los AWS recursos necesarios AWS Managed Microsoft AD, incluidos los usuarios y grupos de LDAP, Network Load Balancer, los certificados y las funciones de IAM. Consulte el paquete README para obtener una lista completa de los recursos creados por la pila.

Si va a configurar los recursos manualmente en lugar de utilizar la pila de CDK, asegúrese de disponer de la infraestructura equivalente antes de configurar LDAP en su Amazon MQ para corredores de RabbitMQ.

Requisito previo para configurar Amazon MQ

AWS Versión CLI >= 2.28.23 para que la adición de un nombre de usuario y una contraseña sea opcional durante la creación del corredor.

Configuración de LDAP en RabbitMQ mediante CLI AWS

Este procedimiento utiliza la AWS CLI para crear y configurar los recursos necesarios. En el siguiente procedimiento, asegúrese de reemplazar los valores de los marcadores de posición, como el ID de configuración y la revisión, <c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca> por sus valores <2> reales.

1. Cree una nueva configuración mediante el comando `create-configuration` AWS CLI, como se muestra en el siguiente ejemplo.

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

Este comando devuelve una respuesta de JSON similar a la siguiente.

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

2. Cree un archivo de configuración llamado `rabbitmq.conf` a utilizar LDAP como método de autenticación y autorización, como se muestra en el siguiente ejemplo. Sustituya todos los valores de marcador de posición de la plantilla (marcados con `${RabbitMqLdapTestStack.*}`) por valores reales de las salidas de la pila de requisitos AWS CDK previos implementada o de una infraestructura equivalente.

```
auth_backends.1 = ldap

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

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

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

```

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

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

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

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

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

# FIXME: This provides all authenticated users access to all topics
# - update to restrict access as required
auth_ldap.queries.topic_access = ''

```

```
{for, [{permission, write, {constant, true}},
      {permission, read, {constant, true}}
    ]
}
...
```

3. Actualice la configuración mediante el comando `update-configuration` AWS CLI, como se muestra en el siguiente ejemplo. En este comando, añada el identificador de configuración que recibió en la respuesta al paso 1 de este procedimiento. Por ejemplo, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

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

Este comando devuelve una respuesta de JSON similar a la siguiente.

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

4. Cree un broker con la configuración de LDAP que creó en el paso 2 de este procedimiento. Para ello, utilice el comando `create-broker` AWS CLI como se muestra en el siguiente ejemplo. En este comando, indique el identificador de configuración y el número de revisión que obtuvo en las respuestas de los pasos 1 y 2, respectivamente. Por ejemplo, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` y 2.

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

Este comando devuelve una respuesta de JSON similar a la siguiente.

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

Restricción de nombres de corredores

Para empezar, el rol de IAM creado por la pila de CDK necesaria restringe los nombres de los corredores. `rabbitmq-ldap-test` Asegúrese de que el nombre de su corredor siga este patrón o el rol de IAM no tendrá permiso para asumir el rol para la resolución de ARN.

5. Compruebe que el estado del bróker pasa de `CREATION_IN_PROGRESS` a `RUNNING`, mediante el comando `describe-broker` AWS CLI, como se muestra en el siguiente ejemplo. En este comando, proporcione el ID de agente que obtuvo en el resultado del paso anterior, por ejemplo, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

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

Este comando devuelve una respuesta de JSON similar a la siguiente. La siguiente respuesta es una versión abreviada del resultado completo que devuelve el comando `describe-broker`. Esta respuesta muestra el estado del agente y la estrategia de autenticación utilizada para proteger al agente. En este caso, la estrategia de `config_managed` autenticación indica que el agente utiliza el método de autenticación LDAP.

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

6. Valide el acceso a RabbitMQ con uno de los usuarios de prueba creados con la pila de CDK necesaria

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

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

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

Uso de la autenticación y autorización HTTP para Amazon MQ for RabbitMQ

En este tutorial se describe cómo configurar la autenticación y la autorización HTTP para sus agentes de Amazon MQ for RabbitMQ mediante un servidor HTTP externo.

Note

El complemento de autenticación HTTP solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

En esta página

- [Requisitos previos para configurar la autenticación y la autorización HTTP](#)
- [Configuración de la autenticación HTTP en RabbitMQ mediante CLI AWS](#)

Requisitos previos para configurar la autenticación y la autorización HTTP

Puede configurar AWS los recursos necesarios en este tutorial mediante la implementación de la [pila de AWS CDK para la integración de autenticación HTTP de Amazon MQ para RabbitMQ](#).

Esta pila de CDK crea automáticamente todos los AWS recursos necesarios, incluidos el servidor de autenticación HTTP, los certificados y las funciones de IAM. Consulte el paquete README para obtener una lista completa de los recursos creados por la pila.

Si va a configurar los recursos manualmente en lugar de utilizar la pila de CDK, asegúrese de disponer de la infraestructura equivalente antes de configurar la autenticación HTTP en su Amazon MQ para agentes de RabbitMQ.

Requisito previo para configurar Amazon MQ

AWS Versión CLI $\geq$ 2.28.23 para que agregar un nombre de usuario y una contraseña sea opcional durante la creación del corredor.

Configuración de la autenticación HTTP en RabbitMQ mediante CLI AWS

Este procedimiento utiliza la AWS CLI para crear y configurar los recursos necesarios. En el siguiente procedimiento, asegúrese de reemplazar los valores de los marcadores de posición por sus valores reales.

1. Cree una nueva configuración mediante el comando `create-configuration` AWS CLI, como se muestra en el siguiente ejemplo.

```
aws mq create-configuration \  
  --name "rabbitmq-http-config" \  
  --engine-type "RABBITMQ" \  
  --engine-version "4.2"
```

Este comando devuelve una respuesta de JSON similar a la siguiente.

```
{  
  "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-  
ae0c-eb15b38b22ca",  
  "AuthenticationStrategy": "simple",  
  "Created": "2025-07-17T16:03:01.759943+00:00",  
  "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",  
  "LatestRevision": {  
    "Created": "2025-07-17T16:03:01.759000+00:00",  
    "Description": "Auto-generated default for rabbitmq-http-config on RabbitMQ  
4.2",  
    "Revision": 1  
  },  
  "Name": "rabbitmq-http-config"  
}
```

2. Cree un archivo de configuración llamado `rabbitmq.conf` a utilizar HTTP como método de autenticación y autorización, como se muestra en el siguiente ejemplo. Sustituya todos los valores de marcador de posición de la plantilla (marcados con `${...}`) por valores reales de las salidas de la pila de AWS CDK requisitos previos implementada o de una infraestructura equivalente.

```
auth_backends.1 = cache  
auth_backends.2 = http  
auth_cache.cached_backend = http  
  
# HTTP authentication settings
```



```
# For more information, see https://github.com/rabbitmq/rabbitmq-auth-backend-http

# FIXME: Replace the ${...} placeholders with actual values
# from your deployed prerequisite CDK stack outputs.
auth_http.http_method = post
auth_http.user_path = ${HttpServerUserPath}
auth_http.vhost_path = ${HttpServerVhostPath}
auth_http.resource_path = ${HttpServerResourcePath}
auth_http.topic_path = ${HttpServerTopicPath}

# TLS/HTTPS configuration
auth_http.ssl_options.verify = verify_peer
auth_http.ssl_options.sni = test.amazonaws.com

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.auth_http.ssl_options.cacertfile = ${CaCertArn}
```

3. Actualice la configuración mediante el comando `update-configuration` AWS CLI. Utilice el ID de configuración del paso 3.

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

Este comando devuelve una respuesta de JSON similar a la siguiente.

```
{
  "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
  "Created": "2025-07-17T16:57:04.520931+00:00",
  "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
  "LatestRevision": {
    "Created": "2025-07-17T16:57:39.172000+00:00",
    "Revision": 2
  }
}
```

```
    },  
    "Name": "rabbitmq-http-config",  
    "Warnings": []  
  }  
}
```

4. Cree un intermediario con la configuración HTTP. Utilice el identificador de configuración y el número de revisión de los pasos anteriores.

```
aws mq create-broker \  
  --broker-name "rabbitmq-http-test-1" \  
  --engine-type "RABBITMQ" \  
  --engine-version "4.2" \  
  --host-instance-type "mq.m7g.large" \  
  --deployment-mode "SINGLE_INSTANCE" \  
  --logs '{"General": true}' \  
  --publicly-accessible \  
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>", "Revision":  
<2>}'
```

Este comando devuelve una respuesta de JSON similar a la siguiente.

```
{  
  "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-http-  
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",  
  "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"  
}
```

5. Compruebe que el estado del bróker pasa de `CREATION_IN_PROGRESS` a `RUNNING`, mediante el comando `describe-broker` AWS CLI.

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

Este comando devuelve una respuesta de JSON similar a la siguiente. La estrategia de `config_managed` autenticación indica que el agente utiliza el método de autenticación HTTP.

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

6. Valide el acceso a RabbitMQ con uno de los usuarios de prueba creados con la pila de CDK necesaria

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

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

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

Uso de la autenticación con certificado SSL para Amazon MQ para RabbitMQ

En este tutorial se describe cómo configurar la autenticación con certificado SSL para sus corredores de Amazon MQ for RabbitMQ mediante una autoridad de certificación privada.

Note

El complemento de autenticación con certificado SSL solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

En esta página

- [Requisitos previos para configurar la autenticación con certificado SSL](#)
- [Configuración de la autenticación con certificado SSL en RabbitMQ mediante CLI AWS](#)

Requisitos previos para configurar la autenticación con certificado SSL

La autenticación con certificados SSL utiliza el TLS mutuo (mTLS) para autenticar a los clientes mediante certificados X.509. Puede configurar AWS los recursos necesarios en este tutorial mediante la implementación de la [pila de AWS CDK para la integración de mTLS de Amazon MQ para RabbitMQ](#).

Esta pila de CDK crea automáticamente todos los AWS recursos necesarios, incluidos la autoridad de certificación, los certificados de cliente y las funciones de IAM. Consulte el paquete README para obtener una lista completa de los recursos creados por la pila.

Note

Antes de implementar la pila de CDK, defina la variable de `RABBITMQ_TEST_USER_NAME` entorno. Este valor se utilizará como nombre común (CN) en el certificado del cliente y debe coincidir con el nombre de usuario que utilice en los pasos del tutorial. Por ejemplo: `export RABBITMQ_TEST_USER_NAME="myuser"`

Si va a configurar los recursos manualmente en lugar de utilizar la pila de CDK, asegúrese de disponer de la infraestructura equivalente antes de configurar la autenticación con certificado SSL en su Amazon MQ para agentes de RabbitMQ.

Requisito previo para configurar Amazon MQ

AWS Versión CLI $\geq$ 2.28.23 para que la adición de un nombre de usuario y una contraseña sea opcional durante la creación del corredor.

Configuración de la autenticación con certificado SSL en RabbitMQ mediante CLI AWS

Este procedimiento utiliza la AWS CLI para crear y configurar los recursos necesarios. En el siguiente procedimiento, asegúrese de reemplazar los valores de los marcadores de posición, como el ID de configuración y la revisión, <c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca> por sus valores <2> reales.

1. Cree una nueva configuración mediante el comando `create-configuration` AWS CLI, como se muestra en el siguiente ejemplo.

```
aws mq create-configuration \
  --name "rabbitmq-ssl-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

Este comando devuelve una respuesta de JSON similar a la siguiente.

```
{
  "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
  "AuthenticationStrategy": "simple",
  "Created": "2025-07-17T16:03:01.759943+00:00",
  "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
  "LatestRevision": {
    "Created": "2025-07-17T16:03:01.759000+00:00",
    "Description": "Auto-generated default for rabbitmq-ssl-config on RabbitMQ 4.2",
    "Revision": 1
  },
  "Name": "rabbitmq-ssl-config"
}
```

2. Cree un archivo de configuración denominado `rabbitmq.conf` para utilizar la autenticación con certificado SSL, como se muestra en el siguiente ejemplo. Sustituya todos los valores de marcador de posición de la plantilla (marcados con `${...}`) por valores reales de las salidas de la pila de AWS CDK requisitos previos implementada o de una infraestructura equivalente.

```

auth_mechanisms.1 = EXTERNAL
ssl_cert_login_from = common_name

auth_backends.1 = internal

# Reject if no client cert
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# FIXME: Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.ssl_options.cacertfile = ${CaCertArn}

```

3. Actualice la configuración mediante el comando `update-configuration` AWS CLI, como se muestra en el siguiente ejemplo. En este comando, añada el identificador de configuración que recibió en la respuesta al paso 1 de este procedimiento. Por ejemplo, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```

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

```

Este comando devuelve una respuesta de JSON similar a la siguiente.

```

{
  "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
  "Created": "2025-07-17T16:57:04.520931+00:00",
  "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
  "LatestRevision": {
    "Created": "2025-07-17T16:57:39.172000+00:00",
    "Revision": 2
  }
}

```

```

    },
    "Name": "rabbitmq-ssl-config",
    "Warnings": []
  }

```

4. Cree un agente con la configuración de autenticación del certificado SSL que creó en el paso 2 de este procedimiento. Para ello, utilice el comando `create-broker` AWS CLI tal y como se muestra en el siguiente ejemplo. En este comando, indique el identificador de configuración y el número de revisión que obtuvo en las respuestas de los pasos 1 y 2, respectivamente. Por ejemplo, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` y 2.

```

aws mq create-broker \
  --broker-name "rabbitmq-ssl-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>", "Revision": <2>}' \
  --users '[{"Username": "testuser", "Password": "testpassword"}]'

```

Este comando devuelve una respuesta de JSON similar a la siguiente.

```

{
  "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-ssl-test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
  "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}

```

5. Compruebe que el estado del bróker pasa de `CREATION_IN_PROGRESS` a `RUNNING`, mediante el comando `describe-broker` AWS CLI, como se muestra en el siguiente ejemplo. En este comando, proporcione el ID de corredor que obtuvo en el resultado del paso anterior. Por ejemplo, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

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

Este comando devuelve una respuesta de JSON similar a la siguiente. La siguiente respuesta es una versión abreviada del resultado completo que devuelve el comando `describe-broker`. Esta respuesta muestra el estado del agente y la estrategia de autenticación utilizada para proteger al agente. En este caso, la estrategia de `config_managed` autenticación indica que el agente utiliza el método de autenticación con certificado SSL.

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

6. Verifique la autenticación del certificado SSL con el siguiente `ssl.sh` script.

Utilice este script de bash para probar la conectividad con su agente de Amazon MQ para RabbitMQ. Este script usa su certificado de cliente para la autenticación y verifica si la conexión se configuró correctamente. Si se ha configurado correctamente, verá que su agente publica y consume mensajes.

Si recibe un `ACCESS_REFUSED` error, puede solucionar los problemas de configuración utilizando los CloudWatch registros de su agente. Encontrará el enlace al grupo de CloudWatch registros de su agente en la consola de Amazon MQ.

En este script, deberá proporcionar los siguientes valores:

- **USERNAME:** El nombre común (CN) de su certificado de cliente.
- **CLIENT_KEYSTORE:** Ruta al archivo del almacén de claves del cliente (PKCS12 formato). Si ha utilizado la pila de CDK necesaria, la ruta por defecto es. `$(pwd)/certs/client-keystore.p12`
- **KEYSTORE_PASSWORD:** contraseña para el almacén de claves de su cliente. Si ha utilizado la pila de CDK necesaria, la contraseña predeterminada es. `changeit`

- **BROKER_DNS:** Puedes encontrar este valor en Conexiones, en la página de detalles del bróker de la consola Amazon MQ.

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

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
application client
USERNAME=<client_cert_common_name>
CLIENT_KEYSTORE=$(pwd)/certs/client-keystore.p12
KEYSTORE_PASSWORD=changeit

BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://${BROKER_DNS}:5671

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

finch run --rm --ulimit nofile=40960:40960 \
  -v ${CLIENT_KEYSTORE}:/certs/client-keystore.p12:ro \
  -e JAVA_TOOL_OPTIONS="-Djavax.net.ssl.keyStore=/certs/client-
keystore.p12 -Djavax.net.ssl.keyStorePassword=${KEYSTORE_PASSWORD} -
Djavax.net.ssl.keyStoreType=PKCS12" \
  pivotalrabbitmq/perf-test:latest \
  --queue-pattern 'test-queue-cert-%d' --queue-pattern-from 1 --queue-pattern-to
$QUEUES_COUNT \
  --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
  --id "cert-test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
  --uri ${CONNECTION_STRING} \
  --sasl-external \
  --use-default-ssl-context \
  --flag persistent --rate $PRODUCER_RATE
```

Uso de mTLS para AMQP y puntos finales de administración

Este tutorial describe cómo configurar el TLS mutuo (mTLS) para las conexiones de los clientes AMQP y la interfaz de administración de RabbitMQ mediante una autoridad de certificación privada.

Note

El uso de autoridades de certificación privadas para los mTL solo está disponible para Amazon MQ para RabbitMQ versión 4 y versiones posteriores.

En esta página

- [Requisitos previos para configurar los mTLS](#)
- [Configuración de mTLS en RabbitMQ mediante CLI AWS](#)

Requisitos previos para configurar los mTLS

Puede configurar AWS los recursos necesarios en este tutorial implementando la [pila de AWS CDK para la integración de Amazon MQ para RabbitMQ mTLS](#) con.

Esta pila de CDK crea automáticamente todos los AWS recursos necesarios, incluidos la autoridad de certificación, los certificados de los clientes y las funciones de IAM. Consulte el paquete README para obtener una lista completa de los recursos creados por la pila.

Si va a configurar los recursos manualmente en lugar de utilizar la pila de CDK, asegúrese de disponer de la infraestructura equivalente antes de configurar mTLS en su Amazon MQ para corredores de RabbitMQ.

Requisito previo para configurar Amazon MQ

AWS Versión CLI $\geq$ 2.28.23 para que la adición de un nombre de usuario y una contraseña sea opcional durante la creación del corredor.

Configuración de mTLS en RabbitMQ mediante CLI AWS

Este procedimiento utiliza la AWS CLI para crear y configurar los recursos necesarios. En el siguiente procedimiento, asegúrese de reemplazar los valores de los marcadores de posición, como

el ID de configuración y la revisión, <c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca> por sus valores <2> reales.

1. Cree una nueva configuración mediante el comando `create-configuration` AWS CLI, como se muestra en el siguiente ejemplo.

```
aws mq create-configuration \  
  --name "rabbitmq-mtls-config" \  
  --engine-type "RABBITMQ" \  
  --engine-version "4.2"
```

Este comando devuelve una respuesta de JSON similar a la siguiente.

```
{  
  "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-  
ae0c-eb15b38b22ca",  
  "AuthenticationStrategy": "simple",  
  "Created": "2025-07-17T16:03:01.759943+00:00",  
  "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",  
  "LatestRevision": {  
    "Created": "2025-07-17T16:03:01.759000+00:00",  
    "Description": "Auto-generated default for rabbitmq-mtls-config on RabbitMQ  
4.2",  
    "Revision": 1  
  },  
  "Name": "rabbitmq-mtls-config"  
}
```

2. Cree un archivo de configuración denominado `rabbitmq.conf` para configurar los mTLS para el AMQP y los puntos finales de administración, como se muestra en el siguiente ejemplo. Sustituya todos los valores de marcador de posición de la plantilla (marcados con `${...}`) por valores reales de las salidas de la pila de AWS CDK requisitos previos implementada o de una infraestructura equivalente.

```
auth_backends.1 = internal
```

```
# TLS configuration
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true
management.ssl.verify = verify_peer

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# FIXME: Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.ssl_options.cacertfile = ${CaCertArn}
aws.arns.management.ssl.cacertfile = ${CaCertArn}
```

3. Actualice la configuración mediante el comando `update-configuration` AWS CLI, como se muestra en el siguiente ejemplo. En este comando, añada el identificador de configuración que recibió en la respuesta al paso 1 de este procedimiento. Por ejemplo, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

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

Este comando devuelve una respuesta de JSON similar a la siguiente.

```
{
  "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
  "Created": "2025-07-17T16:57:04.520931+00:00",
  "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
  "LatestRevision": {
    "Created": "2025-07-17T16:57:39.172000+00:00",
    "Revision": 2
  },
  "Name": "rabbitmq-mtls-config",
  "Warnings": []
}
```

4. Cree un corredor con la configuración mTLS que creó en el paso 2 de este procedimiento. Para ello, utilice el comando `create-broker` AWS CLI tal y como se muestra en el siguiente ejemplo. En este comando, indique el identificador de configuración y el número de revisión que obtuvo en las respuestas de los pasos 1 y 2, respectivamente. Por ejemplo, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` y 2.

```
aws mq create-broker \
  --broker-name "rabbitmq-mtls-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>", "Revision": <2>}' \
  --users '[{"Username": "testuser", "Password": "testpassword"}]'
```

Este comando devuelve una respuesta de JSON similar a la siguiente.

```
{
  "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-mtls-test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
  "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Compruebe que el estado del bróker pasa de `CREATION_IN_PROGRESS` a `RUNNING`, mediante el comando `describe-broker` AWS CLI, como se muestra en el siguiente ejemplo. En este comando, proporcione el ID de corredor que obtuvo en el resultado del paso anterior. Por ejemplo, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

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

Este comando devuelve una respuesta de JSON similar a la siguiente. La siguiente respuesta es una versión abreviada del resultado completo que devuelve el comando `describe-broker`.

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

6. Verifique la autenticación de mTLS con el siguiente `mtls.sh` script.

Utilice este script de bash para probar la conectividad con su agente de Amazon MQ para RabbitMQ. Este script usa su certificado de cliente para autenticarse y verificar si la conexión se configuró correctamente. Si se ha configurado correctamente, verá que su agente publica y consume mensajes.

Si recibe un `ACCESS_REFUSED` error, puede solucionar los problemas de configuración utilizando los CloudWatch registros de su agente. Encontrará el enlace al grupo de CloudWatch registros de su agente en la consola de Amazon MQ.

En este script, deberá proporcionar los siguientes valores:

- `USERNAMEyPASSWORD`: las credenciales de usuario de RabbitMQ que creó con el bróker.
- `CLIENT_KEYSTORE`: Ruta al archivo del almacén de claves de su cliente (formato). PKCS12. Si ha utilizado la pila de CDK necesaria, la ruta por defecto es. `$(pwd)/certs/client-keystore.p12`
- `KEYSTORE_PASSWORD`: contraseña para el almacén de claves de su cliente. Si ha utilizado la pila de CDK necesaria, la contraseña predeterminada es. `changeit`
- `BROKER_DNS`: Puedes encontrar este valor en Conexiones, en la página de detalles del bróker de la consola Amazon MQ.

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

# Client information
```

```
## FIXME: Update this value with the client ID and secret of your confidential
application client
USERNAME=<testuser>
PASSWORD=<testpassword>
CLIENT_KEYSTORE=$(pwd)/certs/client-keystore.p12
KEYSTORE_PASSWORD=changeit

BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://${USERNAME}:${PASSWORD}@${BROKER_DNS}:5671

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

finch run --rm --ulimit nofile=40960:40960 \
    -v ${CLIENT_KEYSTORE}:/certs/client-keystore.p12:ro \
    -e JAVA_TOOL_OPTIONS="-Djavax.net.ssl.keyStore=/certs/client-
keystore.p12 -Djavax.net.ssl.keyStorePassword=${KEYSTORE_PASSWORD} -
Djavax.net.ssl.keyStoreType=PKCS12" \
    pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-cert-%d' --queue-pattern-from 1 --queue-pattern-to
$QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "cert-test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --use-default-ssl-context \
    --flag persistent --rate $PRODUCER_RATE
```

Conexión de la aplicación JMS

En este tutorial se muestra cómo conectar la aplicación JMS al bróker Amazon MQ for RabbitMQ mediante el cliente JMS RabbitMQ. Aprenderá a crear un productor para enviar mensajes y un consumidor para recibir los mensajes de las colas de RabbitMQ.

Antes de empezar, añada la dependencia JMS de RabbitMQ adecuada a tu proyecto de Maven:

Para JMS 1.1 y 2.0:

```
<dependencies>
```

```
<dependency>
  <groupId>com.rabbitmq.jms</groupId>
  <artifactId>rabbitmq-jms</artifactId>
  <version>2.12.0</version>
</dependency>

</dependencies>
```

Para JMS 3.1:

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>3.5.0</version>
  </dependency>

</dependencies>
```

Cree un productor

El siguiente ejemplo de código muestra cómo escribir en una cola de RabbitMQ mediante JMS:

```
import jakarta.jms.*;
import com.rabbitmq.jms.admin.*;

// Setting the connection factory
RMQConnectionFactory factory = new RMQConnectionFactory();
factory.setHost(envProps.getProperty("RABBITMQ_HOST", "localhost"));
factory.setPort(Integer.parseInt(envProps.getProperty("RABBITMQ_PORT", "5672")));
factory.setUsername(envProps.getProperty("RABBITMQ_USERNAME", "guest"));
factory.setPassword(envProps.getProperty("RABBITMQ_PASSWORD", "guest"));
factory.setVirtualHost(envProps.getProperty("RABBITMQ_VIRTUAL_HOST", "/"));
factory.useSslProtocol();

connection = factory.createConnection();
connection.start();

String queueName = "test-queue-jms";
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
```



```
RMQDestination destination = new RMQDestination(queueName, true, false);

// Send the message to the queue
MessageProducer producer = session.createProducer(destination);
producer.setDeliveryMode(DeliveryMode.PERSISTENT);

String msg_content = "Hello World!!";
TextMessage textMessage = session.createTextMessage(msg_content);
producer.send(textMessage);

System.out.printf("Published to AMQP queue '%s': %s", queueName, msg_content);
```

Crea un consumidor

El siguiente ejemplo de código muestra cómo leer una cola de RabbitMQ mediante JMS:

```
import jakarta.jms.*;
import com.rabbitmq.jms.admin.*;

// Setting the connection factory
RMQConnectionFactory factory = new RMQConnectionFactory();
factory.setHost(envProps.getProperty("RABBITMQ_HOST", "localhost"));
factory.setPort(Integer.parseInt(envProps.getProperty("RABBITMQ_PORT", "5672")));
factory.setUsername(envProps.getProperty("RABBITMQ_USERNAME", "guest"));
factory.setPassword(envProps.getProperty("RABBITMQ_PASSWORD", "guest"));
factory.setVirtualHost(envProps.getProperty("RABBITMQ_VIRTUAL_HOST", "/"));
factory.useSslProtocol();

// Establish the connection and session
jakarta.jms.Connection connection = factory.createConnection();

String queueName = "test-queue-jms";
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

RMQDestination destination = new RMQDestination();
destination.setDestinationName(queueName);
destination.setAmqp(true);
destination.setAmqpQueueName(queueName);

// Initialize consumer
MessageConsumer consumer = session.createConsumer(destination);
consumer.setMessageListener(message -> {
    try {
```

```
        if (message instanceof TextMessage) {
            TextMessage textMessage = (TextMessage) message;
            System.out.printf("Message: %s%n", textMessage.getText());
        } else if (message instanceof BytesMessage) {
            BytesMessage bytesMessage = (BytesMessage) message;
            byte[] bytes = new byte[(int) bytesMessage.getBodyLength()];
            bytesMessage.readBytes(bytes);
            String content = new String(bytes);
            System.out.printf("Message: %s%n", content);
        } else {
            System.out.printf("Message: [%s]%n", message.getClass().getSimpleName());
        }
    } catch (JMSEException e) {
        System.err.printf("Error processing message: %s%n", e.getMessage());
    }
});

connection.start();
```

Seguridad en Amazon MQ

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de los centros de datos y las arquitecturas de red diseñados para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre AWS usted y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS programas](#) de de . Para obtener más información sobre los programas de conformidad que se aplican a Amazon MQ, consulte [AWS Servicios incluidos en el ámbito del programa de conformidad AWS Servicios incluidos en el ámbito del programa](#) .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza Amazon MQ. En los siguientes temas, se le mostrará cómo configurar Amazon MQ para satisfacer sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de Amazon MQ.

Temas

- [Protección de los datos en Amazon MQ](#)
- [Identity and Access Management para Amazon MQ](#)
- [Validación de la conformidad para Amazon MQ](#)
- [Resiliencia en Amazon MQ](#)
- [Seguridad de la infraestructura en Amazon MQ](#)
- [Prácticas recomendadas de seguridad para Amazon MQ](#)

Protección de los datos en Amazon MQ

El [modelo de](#) se aplica a protección de datos en Amazon MQ. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte [Cómo trabajar con CloudTrail senderos](#) en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger la información confidencial almacenada en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-3](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con Amazon MQ u otro dispositivo Servicios de AWS mediante la consola, la API o. AWS CLI AWS SDKs Cualquier dato que introduzca

en etiquetas o campos de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Tanto para los agentes de Amazon MQ for ActiveMQ como de Amazon MQ para RabbitMQ, no utilice ninguna información de identificación personal (PII) ni ninguna otra información confidencial para nombres de agente o nombres de usuario al crear recursos a través de la consola web del agente o la API de Amazon MQ. Otros AWS servicios, incluidos los registros, pueden acceder a los nombres de los corredores y a los nombres de usuario. CloudWatch Los nombres de usuario de agente no están diseñados para usarse con información privada o confidencial.

Important

TLS 1.3 no está disponible para los agentes de RabbitMQ.

Cifrado

Los datos de usuario que almacena Amazon MQ se cifran en reposo. El cifrado en reposo de Amazon MQ proporciona mayor seguridad porque cifra los datos mediante las claves de cifrado almacenadas en AWS Key Management Service (KMS). Este servicio ayuda a reducir la carga y la complejidad operativas que conlleva la protección de información confidencial. Con el cifrado en reposo, puede crear aplicaciones sensibles a la seguridad que cumplen los requisitos de cifrado y normativos.

Todas las conexiones entre los agentes de Amazon MQ usan Transport Layer Security (TLS) para proporcionar el cifrado en tránsito.

Amazon MQ cifra los mensajes en reposo y en tránsito mediante claves de cifrado que administra y almacena de forma segura. Para obtener más información, consulte la Guía para desarrolladores de [AWS Encryption SDK](#).

Cifrado en reposo

Amazon MQ se integra con AWS Key Management Service (KMS) para ofrecer un cifrado transparente del lado del servidor. Amazon MQ siempre cifra sus datos en reposo.

Al crear un bróker de Amazon MQ para ActiveMQ o un bróker de Amazon MQ para RabbitMQ, puede especificar lo que AWS KMS key quiere que Amazon MQ utilice para cifrar los datos en reposo. Si

no especifica una clave de KMS, Amazon MQ crea una clave de AWS KMS propia para usted y la utiliza en su nombre. Amazon MQ admite actualmente las claves de KMS simétricas. Para obtener más información acerca de las claves KMS, consulte [AWS KMS keys](#).

Cuando cree un agente, para configurar lo que Amazon MQ utilizará para la clave de cifrado, seleccione una de estas opciones:

- Amazon MQ owned KMS key (default) (Clave KMS de Amazon MQ [predeterminada]): la clave es propiedad de y está administrada por Amazon MQ y no está en su cuenta.
- AWS clave de KMS AWS gestionada: la clave de KMS gestionada (aws/mq) es una clave de KMS de su cuenta que Amazon MQ crea, gestiona y utiliza en su nombre.
- Select existing customer managed CMK (Seleccionar clave de KMS existente administrada por el cliente): usted crea y administra en AWS Key Management Service (KMS) claves de KMS administradas por el cliente.

Important

- La revocación de una concesión no se puede deshacer. Elimine el agente para revocar los derechos de acceso.
- Para Agentes de Amazon MQ para ActiveMQ que utilizan Amazon Elastic File System (EFS) para almacenar los datos de mensajes, la revocación de los permisos para utilizar las claves KMS de su cuenta puede tardar varias horas después de realizar las acciones necesarias.
- Para agentes de Amazon MQ para RabbitMQ y Amazon MQ para ActiveMQ que utilizan EBS para almacenar los datos de los mensajes; si se desactiva, se programa la eliminación o se revoca la adjudicación que permite a Amazon EBS utilizar las claves KMS en la cuenta, Amazon MQ no podrá mantener al agente y podría cambiar a un estado degradado.
- Si ha desactivado la clave o ha programado su eliminación, puede volver a activarla o cancelar la eliminación de la clave y mantener al agente en buen estado.
- Una vez realizadas las acciones necesarias, la desactivación de una clave o la revocación de una concesión pueden tardar varias horas.
- Para cifrar o descifrar CloudWatch registros, no puede configurar lo que Amazon MQ utiliza para su clave de cifrado. CloudWatch Los registros protegen los datos en reposo mediante el cifrado, y los grupos de registros se cifran. El servicio CloudWatch de registros

administra el cifrado del lado del servidor de forma predeterminada. Para obtener más información sobre cómo se cifran los grupos de [CloudWatch registros, consulta la Guía del usuario de Amazon Logs](#).

Al crear un [agente de instancia única](#) con una clave de KMS para RabbitMQ, verá dos eventos de CreateGrant registrados en AWS CloudTrail. El primer evento es que Amazon MQ cree una concesión para la clave de KMS. El segundo evento consiste en que EBS cree una concesión para que la utilice EBS.

CreateGrant AWS CloudTrail entrada de registro: agente de instancia única

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

Verá un evento para la creación de la concesión de EBS.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "mq.amazonaws.com"
  },
  "eventTime": "2023-02-23T19:09:40Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "mq.amazonaws.com",
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",
  "requestParameters": {
    "granteePrincipal": "mq.amazonaws.com",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "constraints": {
      "encryptionContextSubset": {
        "aws:ebs:id": "vol-0b670f00f7d5417c0"
      }
    },
    "operations": [
      "Decrypt"
    ],
    "retiringPrincipal": "ec2.us-east-1.amazonaws.com"
  },
  "responseElements": {
    "grantId":
      "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
  },
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",

```

```

      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventCategory": "Management"
}

```

Al crear una [implementación de clúster](#) con una clave KMS para RabbitMQ, verá cinco eventos CreateGrant registrados en AWS CloudTrail. Los dos primeros eventos son creaciones de concesiones para Amazon MQ. Los siguientes tres eventos son concesiones creadas por EBS para que las utilice EBS.

CreateGrant AWS CloudTrail entrada de registro: despliegue de clústeres

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

Verá tres eventos para la creación de concesiones de EBS.

```

    {
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "AWSService",
        "invokedBy": "mq.amazonaws.com"
      },
      "eventTime": "2023-02-23T19:09:40Z",
      "eventSource": "kms.amazonaws.com",

```

```

    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "mq.amazonaws.com",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "constraints": {
            "encryptionContextSubset": {
                "aws:ebs:id": "vol-0b670f00f7d5417c0"
            }
        },
        "operations": [
            "Decrypt"
        ],
        "retiringPrincipal": "ec2.us-east-1.amazonaws.com"
    },
    "responseElements": {
        "grantId":
        "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventCategory": "Management"
}

```

Para obtener más información acerca de las claves de KMS, consulte [AWS KMS keys](#) en la Guía para desarrolladores de AWS Key Management Service .

Cifrado en tránsito

Amazon MQ para ActiveMQ: Amazon MQ para ActiveMQ requiere seguridad de la capa de transporte (TLS) sólida y cifra los datos en tránsito entre los agentes de la implementación de Amazon MQ. Todos los datos que pasan entre los agentes de Amazon MQ se cifran mediante seguridad de la capa de transporte (TLS) sólida. Esto se aplica a todos los protocolos disponibles.

Amazon MQ para RabbitMQ: Amazon MQ para RabbitMQ requiere un cifrado de seguridad de la capa de transporte (TLS) sólido para todas las conexiones de los clientes. El tráfico de replicación de clústeres de RabbitMQ solo transita por la VPC de su agente y todo el tráfico de red entre los centros de AWS datos se cifra de forma transparente en la capa física. Los agentes agrupados en clústeres de Amazon MQ para RabbitMQ actualmente no admiten el [cifrado entre nodos](#) para la replicación de clústeres. [Para obtener más información, consulte Cifrado y en tránsito data-in-transit. Data-at-Rest](#)

Protocolos de Amazon MQ para ActiveMQ

Para acceder a sus agentes de ActiveMQ, puede utilizar los siguientes protocolos con TLS habilitado:

- [AMQP](#)
- [MQTT](#)
- MQTT ha terminado [WebSocket](#)
- [OpenWire](#)
- [STOMP](#)
- STOMP WebSocket

Conjuntos de cifrado TLS compatibles con ActiveMQ

ActiveMQ en Amazon MQ admite los siguientes conjuntos de cifrado:

- TLS_ECDHE_RSA_CON_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_CON_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_CON_AES_256_GCM_SHA384

- TLS_DHE_RSA_CON_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_CON_AES_256_GCM_SHA384
- TLS_RSA_CON_AES_256_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_CON_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_CON_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_CON_AES_128_GCM_SHA256
- TLS_DHE_RSA_CON_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_CON_AES_128_GCM_SHA256
- TLS_RSA_CON_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA

Protocolos de Amazon MQ para RabbitMQ

Para acceder a sus agentes de RabbitMQ, puede utilizar los siguientes protocolos con TLS habilitado:

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

Conjuntos de cifrado TLS compatibles con RabbitMQ

RabbitMQ en Amazon MQ admite los siguientes conjuntos de cifrado:

- TLS_ECDHE_RSA_CON_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_CON_AES_128_GCM_SHA256

Identity and Access Management para Amazon MQ

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. Los administradores de IAM controlan a qué personas se las puede autenticar (pueden iniciar sesión) y autorizar (tiene permisos) para utilizar

recursos de Amazon MQ. La IAM es una Servicio de AWS herramienta que puede utilizar sin coste adicional.

Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración del acceso con políticas](#)
- [Cómo funciona Amazon MQ con IAM](#)
- [Ejemplos de políticas basadas en identidades de Amazon MQ](#)
- [Autenticación y autorización de las API para Amazon MQ](#)
- [Autenticación y autorización del agente](#)
- [AWS políticas gestionadas para Amazon MQ](#)
- [Uso de roles vinculados a servicios para Amazon MQ](#)
- [Solución de problemas de identidad y acceso de Amazon MQ](#)

Público

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicite permisos al administrador si no puede acceder a las características (consulte [Solución de problemas de identidad y acceso de Amazon MQ](#)).
- Administrador del servicio: determine el acceso de los usuarios y envíe las solicitudes de permiso (consulte [Cómo funciona Amazon MQ con IAM](#)).
- Administrador de IAM: escribe las políticas para administrar el acceso (consulte [Ejemplos de políticas basadas en identidades de Amazon MQ](#)).

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las

credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte [Cómo iniciar sesión en la Cuenta de AWS](#) en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte [AWS Signature Version 4 para solicitudes de API](#) en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear un Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz que tiene acceso completo a todos Servicios de AWS los recursos. Se recomienda encarecidamente que no utilice el usuario raíz para las tareas diarias. Para ver las tareas que requieren credenciales de usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

Usuarios y grupos

Un [usuario de IAM](#) es una identidad con permisos específicos para una sola persona o aplicación. Recomendamos el uso de credenciales temporales en lugar de usuarios de IAM con credenciales de larga duración. Para obtener más información, consulte [Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) especifica un conjunto de usuarios de IAM y facilita la administración de los permisos para grupos grandes de usuarios. Para obtener más información, consulte [Casos de uso para usuarios de IAM](#) en la Guía del usuario de IAM.

Roles de IAM

Un [Rol de IAM](#) es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol [cambiando de un rol de usuario a uno de IAM \(consola\)](#) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte [Métodos para asumir un rol](#) en la Guía del usuario de IAM.

Los roles de IAM son útiles para el acceso de usuario federado, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon EC2. Para obtener más información, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Administración del acceso con políticas

AWS Para controlar el acceso, puede crear políticas y adjuntarlas a AWS identidades o recursos. Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de políticas de JSON, consulte [Información general de políticas de JSON](#) en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de política de permisos JSON que asocia a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte [Definición de permisos de IAM personalizados con políticas administradas por el cliente](#) en la Guía del usuario de IAM.

Las políticas basadas en identidad pueden ser políticas insertadas (incrustadas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener información sobre cómo elegir entre políticas administradas e insertadas, consulte [Selección entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de políticas JSON que se asocian a un recurso. Los ejemplos incluyen las Políticas de confianza de roles de IAM y las Políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Debe [especificar una entidad principal](#) en una política basada en recursos.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios compatibles. AWS WAF ACLs Para obtener más información ACLs, consulte la [descripción general de la lista de control de acceso \(ACL\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos otorgados por los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte [Políticas de control de servicios](#) en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCPs): establece los permisos máximos disponibles para los recursos de tus cuentas. Para obtener más información, consulte [Políticas de control de recursos \(RCPs\)](#) en la Guía del AWS Organizations usuario.
- Políticas de sesión: políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

Cómo funciona Amazon MQ con IAM

Antes de utilizar IAM para administrar el acceso a Amazon MQ, debe conocer qué características de IAM están disponibles con Amazon MQ. Para obtener una visión general de cómo Amazon MQ y otros AWS servicios funcionan con IAM, consulte [AWS Servicios que funcionan con IAM en la Guía del usuario de IAM](#).

Amazon MQ utiliza IAM para las operaciones de la API de Amazon MQ a fin de crear, actualizar, eliminar y publicar agentes. Para que los agentes puedan publicar mensajes y suscribirse a ellos, Amazon MQ para ActiveMQ admite la autenticación ActiveMQ nativa y LDAP, mientras que Amazon MQ para RabbitMQ admite la autenticación de IAM y otros métodos. Para obtener más información, consulte [the section called “Autenticación y autorización del agente”](#).

Temas

- [Políticas de basadas en identidades de Amazon MQ](#)
- [Políticas de basadas en recursos de Amazon MQ](#)
- [Autorización basada en etiquetas de Amazon MQ](#)
- [Roles de IAM de Amazon MQ](#)

Políticas de basadas en identidades de Amazon MQ

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Amazon MQ admite acciones, claves de condiciones y recursos específicos. Para obtener información sobre todos los elementos que utiliza en una política JSON, consulte [Referencia de los elementos de las políticas JSON de IAM](#) en la Guía del usuario de IAM.

Acciones

Los administradores pueden usar las políticas de JSON para especificar quién tiene acceso a qué. AWS Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Las acciones de políticas de Amazon MQ utilizan el siguiente prefijo antes de la acción: `mq:`.

Por ejemplo, para conceder a alguien permiso para ejecutar una instancia de Amazon MQ con la

operación `CreateBroker` de la API de Amazon MQ, debe incluir la acción `mq:CreateBroker` en la política. Las instrucciones de la política deben incluir un elemento `Action` o un elemento `NotAction`. Amazon MQ define su propio conjunto de acciones que describen las tareas que se pueden realizar con este servicio.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

```
"Action": [  
    "mq:action1",  
    "mq:action2"
```

Puede utilizar caracteres comodín para especificar varias acciones (*). Por ejemplo, para especificar todas las acciones que comiencen con la palabra `Describe`, incluya la siguiente acción:

```
"Action": "mq:Describe*"
```

Para ver una lista de las acciones de Amazon MQ, consulte [Acciones definidas por Amazon MQ](#) en la Guía del usuario de IAM.

Recursos

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). En el caso de las acciones que no admiten permisos por recurso, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

En Amazon MQ, los AWS recursos principales son un agente de mensajes de Amazon MQ y su configuración. Cada uno de los agentes y configuraciones de Amazon MQ tiene asociados nombres de recursos de Amazon (ARNs) exclusivos, como se muestra en la siguiente tabla.

Tipos de recursos	ARN	Claves de condición
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}

Para obtener más información sobre el formato de ARNs, consulte [Amazon Resource Names \(ARNs\) y AWS Service Namespaces](#).

Por ejemplo, para especificar el agente denominado MyBroker con brokerId b-1234a5b6-78cd-901e-2fgh-3i45j6k17819 en la instrucción, utilice el siguiente ARN:

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

Para especificar todos los agentes y configuraciones que pertenecen a una cuenta específica, utilice el carácter comodín (*):

```
"Resource": "arn:aws:mq:us-east-1:123456789012:*"
```

Algunas acciones de Amazon MQ, como las que se utilizan para crear recursos, no se pueden llevar a cabo en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

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

La acción CreateTags de la API requiere tanto un agente como una configuración. Para especificar varios recursos en una sola sentencia, sepárelos ARNs con comas.

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

Para ver una lista de los tipos de recursos de Amazon MQ y sus tipos ARNs, consulte [Recursos definidos por Amazon MQ](#) en la Guía del usuario de IAM. Para obtener información acerca de las

acciones con las que puede especificar el ARN de cada recurso, consulte [Acciones definidas por Amazon MQ](#).

Claves de condición

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede crear expresiones condicionales que utilizan [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales](#) en la Guía del usuario de IAM.

Amazon MQ no define ninguna clave de condición específica del servicio, pero admite el uso de algunas claves de condición globales. Para ver una lista de claves de condición de Amazon MQ, consulte la tabla que se incluye a continuación o [Claves de condición de Amazon MQ](#) en la Guía del usuario de IAM. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte [acciones definidas por Amazon MQ](#).

Claves de condición	Descripción	Tipo
aws: RequestTag /\$ {} TagKey	Filtra acciones basadas en las etiquetas que se pasan en la solicitud.	Cadena
aws: ResourceTag /\$ {} TagKey	Filtra acciones en función de las etiquetas asociadas con el recurso.	Cadena
AWS: TagKeys	Filtra acciones basadas en las claves de etiqueta que se pasan en la solicitud.	Cadena

Ejemplos

Para ver ejemplos de políticas basadas en identidad de Amazon MQ, consulte [Ejemplos de políticas basadas en identidades de Amazon MQ](#).

Políticas de basadas en recursos de Amazon MQ

Actualmente, Amazon MQ no admite la autenticación de IAM mediante permisos basados en recursos o políticas basadas en recursos.

Autorización basada en etiquetas de Amazon MQ

Puede asociar etiquetas a los recursos de Amazon MQ o transferirlas en una solicitud a Amazon MQ. Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el [elemento de condición](#) de una política utilizando las claves de condición `mq:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Amazon MQ admite políticas basadas en etiquetas. Por ejemplo, puede denegar el acceso a todos los recursos de Amazon MQ que contengan una etiqueta con la clave `environment` y el valor `production`:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "mq:DeleteBroker",
        "mq:RebootBroker",
        "mq:DeleteTags"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/environment": "production"
        }
      }
    }
  ]
}
```

Esta política denegaría (Deny) la posibilidad de eliminar o reiniciar un agente de Amazon MQ con la etiqueta `environment/production`.

Para obtener más información acerca del etiquetado, consulte:

- [Adición de etiquetas a los recursos de Amazon MQ](#)
- [Control del acceso mediante etiquetas de IAM](#)

Roles de IAM de Amazon MQ

Un [rol de IAM](#) es una entidad de tu AWS cuenta que tiene permisos específicos.

Uso de credenciales temporales con Amazon MQ

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Para obtener credenciales de seguridad temporales, puede llamar a operaciones de AWS STS API como [AssumeRole](#) o [GetFederationToken](#).

Amazon MQ admite el uso de credenciales temporales.

Roles de servicio

Esta característica permite que un servicio asuma un [rol de servicio](#) en su nombre. Este rol permite que el servicio obtenga acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles de servicio aparecen en su cuenta de IAM y son propiedad de la cuenta. Esto significa que un administrador de IAM puede cambiar los permisos de este rol. Sin embargo, hacerlo podría deteriorar la funcionalidad del servicio.

Amazon MQ admite roles de servicio.

Ejemplos de políticas basadas en identidades de Amazon MQ

De forma predeterminada, los usuarios y roles no tienen permiso para crear ni modificar los recursos de Amazon MQ. Tampoco pueden realizar tareas mediante la AWS API Consola de administración de AWS AWS CLI, o. Un administrador de IAM debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos especificados que necesiten. El administrador debe adjuntar esas políticas a los usuarios o grupos de IAM que necesiten esos permisos.

Para obtener información acerca de cómo crear una política basada en identidad de IAM con estos documentos de políticas JSON de ejemplo, consulte [Creación de políticas en la pestaña JSON](#) en la Guía del usuario de IAM.

Temas

- [Prácticas recomendadas relativas a políticas](#)
- [Uso de la consola de Amazon MQ](#)
- [Permitir a los usuarios consultar sus propios permisos](#)

Prácticas recomendadas relativas a políticas

Las políticas basadas en identidades determinan si alguien puede crear, eliminar o acceder a los recursos de Amazon MQ de su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos en muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las [políticas administradas por AWS](#) o las [políticas administradas por AWS para funciones de tarea](#) en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte [Políticas y permisos en IAM](#) en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo CloudFormation. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de

políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte [Validación de políticas con el Analizador de acceso de IAM](#) en la Guía del usuario de IAM.

- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte [Acceso seguro a la API con MFA](#) en la Guía del usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

Uso de la consola de Amazon MQ

Para acceder a la consola de Amazon MQ, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirte enumerar y ver detalles sobre los recursos de Amazon MQ de tu AWS cuenta. Si crea una política basada en identidad que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles de IAM) que tengan esa política.

Para garantizar que esas entidades puedan seguir utilizando la consola de Amazon MQ, adjunte también la siguiente política de AWS gestión a las entidades. Para obtener más información, consulte [Agregar de permisos a un usuario](#) en la Guía del usuario de IAM.

```
AmazonMQReadOnlyAccess
```

No es necesario conceder permisos mínimos de consola a los usuarios que solo realicen llamadas a la AWS API AWS CLI o a la misma. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intenta realizar.

Permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "ViewOwnUserInfo",
    "Effect": "Allow",
    "Action": [
      "iam:GetUserPolicy",
      "iam:ListGroupsForUser",
      "iam:ListAttachedUserPolicies",
      "iam:ListUserPolicies",
      "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
  },
  {
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
      "iam:GetGroupPolicy",
      "iam:GetPolicyVersion",
      "iam:GetPolicy",
      "iam:ListAttachedGroupPolicies",
      "iam:ListGroupPolicies",
      "iam:ListPolicyVersions",
      "iam:ListPolicies",
      "iam:ListUsers"
    ],
    "Resource": "*"
  }
]
}

```

Autenticación y autorización de las API para Amazon MQ

Amazon MQ utiliza la firma de AWS solicitudes estándar para la autenticación de la API. Para obtener más información, consulte [Firma de solicitudes de la API de AWS](#) en la Referencia general de AWS.

Note

Actualmente, Amazon MQ no admite la autenticación de IAM mediante permisos basados en recursos o políticas basadas en recursos.

Para autorizar a AWS los usuarios a trabajar con intermediarios, configuraciones y usuarios, debe editar los permisos de su política de IAM.

Temas

- [Permisos de IAM requeridos para crear un agente de Amazon MQ](#)
- [Referencia sobre los permisos de la API REST de Amazon MQ](#)
- [Referencia de permisos adicionales de Amazon MQ](#)
- [Permisos de nivel de recurso para las acciones de la API de Amazon MQ](#)

Permisos de IAM requeridos para crear un agente de Amazon MQ

Para crear un agente, debe utilizar la política `AmazonMQFullAccess` de IAM o incluir los siguientes permisos de EC2 en la política de IAM.

La siguiente política personalizada consta de dos declaraciones (una condicional), que concede permisos para manipular los recursos que necesita Amazon MQ para crear un agente de ActiveMQ.

Important

- La acción `ec2:CreateNetworkInterface` es necesaria para que Amazon MQ pueda crear una interfaz de red elástica (ENI) en su cuenta en su nombre.
- La acción `ec2:CreateNetworkInterfacePermission` autoriza a Amazon MQ a adjuntar la ENI a un agente de ActiveMQ.
- La clave de condición `ec2:AuthorizedService` garantiza que los permisos de ENI solo se puedan conceder a las cuentas de servicio de Amazon MQ.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "mq:*",
      "ec2:CreateNetworkInterface",
      "ec2>DeleteNetworkInterface",
      "ec2:DetachNetworkInterface",
```

```

        "ec2:DescribeInternetGateways",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeRouteTables",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs"
    ],
    "Effect": "Allow",
    "Resource": "*"
  },{
    "Action": [
        "ec2:CreateNetworkInterfacePermission",
        "ec2>DeleteNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfacePermissions"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "ec2:AuthorizedService": "mq.amazonaws.com"
        }
    }
  }
}]
}
```

Para obtener más información, consulte [Paso 2: crea un usuario y obtén tus credenciales AWS](#) y [No modifique ni elimine nunca la interfaz de red elástica de Amazon MQ](#).

Referencia sobre los permisos de la API REST de Amazon MQ

En la siguiente tabla se enumeran Amazon MQ REST APIs y los permisos de IAM correspondientes.

Amazon MQ REST APIs y permisos necesarios

Amazon MQ REST APIs	Permisos necesarios
CreateBroker	mq:CreateBroker
CreateConfiguration	mq:CreateConfiguration
CreateTags	mq:CreateTags

Amazon MQ REST APIs	Permisos necesarios
CreateUser	mq:CreateUser
DeleteBroker	mq:DeleteBroker
DeleteUser	mq:DeleteUser
DescribeBroker	mq:DescribeBroker
DescribeConfiguration	mq:DescribeConfiguration
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

Referencia de permisos adicionales de Amazon MQ

En la siguiente tabla se enumeran la API de Amazon MQ y el permiso de IAM adicional necesario para funciones específicas, como OAuth la autenticación 2.0.

API de REST de Amazon MQ	Permiso	Description (Descripción)
UpdateBroker	<code>mq:UpdateBrokerAccessConfiguration</code>	Necesita este permiso para actualizar las opciones de autenticación y autorización en la configuración del agente asociada. Para obtener más información, consulte OAuth Autenticación y autorización 2.0 para Amazon MQ para RabbitMQ .

Permisos de nivel de recurso para las acciones de la API de Amazon MQ

Los permisos de nivel de recurso hacen referencia a la capacidad de especificar en qué recursos los usuarios tienen permitido realizar acciones. Amazon MQ admite parcialmente los permisos de nivel de recurso. Esto significa que, para algunas acciones de Amazon MQ, usted puede determinar cuándo se permite utilizarlas a los usuarios en función de si se cumplen una serie de condiciones o de los recursos concretos que estos pueden utilizar.

En la siguiente tabla se describen las acciones de la API Amazon MQ que actualmente admiten permisos a nivel de recursos, así como los recursos, los recursos y las claves de condición compatibles para cada acción. ARNs

Important

Si una acción de la API de Amazon MQ no aparece en la tabla, significa que no admite los permisos de nivel de recurso. Si una acción de la API de Amazon MQ no admite este tipo de permisos de nivel de recurso, usted puede conceder permisos a los usuarios para que la utilicen, pero tendrá que usar un carácter comodín * para el elemento de recurso de la instrucción de la política.

Acción API	Tipos de recursos (*necesarios)
CreateConfiguration	Configuraciones*

Acción API	Tipos de recursos (*necesarios)
<u>CreateTags</u>	<u>agentes,configuraciones</u>
<u>CreateUser</u>	<u>Agentes*</u>
<u>DeleteBroker</u>	<u>Agentes*</u>
<u>DeleteUser</u>	<u>Agentes*</u>
<u>DescribeBroker</u>	<u>Agentes*</u>
<u>DescribeConfiguration</u>	<u>Configuraciones*</u>
<u>DescribeConfigurat ionRevision</u>	<u>Configuraciones*</u>
<u>DescribeUser</u>	<u>Agentes*</u>
<u>ListConfigurationR evisions</u>	<u>Configuraciones*</u>
<u>ListConfigurationR evisions</u>	<u>Configuraciones*</u>
<u>ListTags</u>	<u>agentes,configuraciones</u>
<u>ListUsers</u>	<u>Agentes*</u>
<u>RebootBroker</u>	<u>Agentes*</u>
<u>UpdateBroker</u>	<u>Agentes*</u>
<u>UpdateConfiguration</u>	<u>Configuraciones*</u>
<u>UpdateUser</u>	<u>Agentes*</u>

Autenticación y autorización del agente

Amazon MQ ofrece distintos métodos de autenticación y autorización en función del tipo de motor de intermediación.

Autenticación y autorización de Amazon MQ para ActiveMQ

Amazon MQ para ActiveMQ admite los siguientes métodos de autenticación y autorización:

Autenticación y autorización simple

En este método, los usuarios de los corredores se crean y administran a través de la consola o API de Amazon MQ. Los usuarios se pueden configurar con permisos específicos para acceder a las colas, los temas y la consola web ActiveMQ. Para obtener más información sobre este método, consulte [Creación de un usuario de ActiveMQ broker](#).

Autenticación y autorización LDAP

En este método, los usuarios intermediarios se autentican mediante las credenciales almacenadas en el servidor LDAP. Puede agregar, eliminar y modificar usuarios y asignar permisos a temas y colas a través del servidor LDAP, lo que proporciona autenticación y autorización centralizadas. Para obtener más información sobre este método, consulte [Integración de agentes ActiveMQ con LDAP](#).

Autenticación y autorización de Amazon MQ para RabbitMQ

Amazon MQ para RabbitMQ admite los siguientes métodos de autenticación y autorización:

Autenticación y autorización simple

En este método, los usuarios del agente se almacenan internamente en el agente de RabbitMQ y se administran a través de la consola web o la API de administración. Los permisos para vhosts, intercambios, colas y temas se configuran directamente en RabbitMQ. Este es el método predeterminado. Para obtener más información, consulte Autenticación y autorización [sencillas](#).

OAuth Autenticación y autorización 2.0

En este método, los usuarios intermediarios y sus permisos los administra un proveedor de identidades (IdP) OAuth 2.0 externo. La autenticación de los usuarios y los permisos de recursos para vhosts, bolsas, colas y temas se centralizan mediante el sistema Scope del proveedor OAuth

2.0. Esto simplifica la administración de usuarios y permite la integración con los sistemas de identidad existentes. Para obtener más información, consulte [Autenticación y autorización OAuth 2.0](#).

Autenticación y autorización de IAM

En este método, los usuarios del broker se autentican con credenciales de AWS IAM mediante la federación saliente de [IAM](#). Las credenciales de IAM se utilizan para obtener los tokens JWT del AWS Security Token Service (STS), y estos tokens JWT sirven como tokens 2.0 para la autenticación. OAuth Este método aprovecha la compatibilidad OAuth 2.0 existente en Amazon MQ para RabbitMQ, AWS donde actúa como proveedor de identidad 2.0. OAuth AWS IAM gestiona la autenticación de los usuarios, mientras que los permisos de recursos para vhosts, bolsas, colas y temas se gestionan mediante políticas de IAM y alias de ámbito configurados en RabbitMQ. [Para obtener más información, consulte Autenticación y autorización de IAM](#).

Autenticación y autorización de LDAP

En este método, los usuarios intermediarios y sus permisos se administran mediante un servicio de directorio LDAP externo. La autenticación de los usuarios y los permisos de recursos se centralizan a través del servidor LDAP, lo que permite a los usuarios acceder a RabbitMQ con sus credenciales de servicio de directorio existentes. [Para obtener más información, consulte Autenticación y autorización LDAP](#).

Autenticación y autorización HTTP

En este método, los usuarios del broker y sus permisos se administran mediante un servidor HTTP externo. La autenticación de los usuarios y los permisos de recursos se centralizan a través del servidor HTTP, lo que permite a los usuarios acceder a RabbitMQ con su propio proveedor de autenticación y autorización. Para obtener más información sobre este método, consulte [Autenticación y autorización HTTP](#).

Autenticación con certificado SSL

Amazon MQ admite el TLS mutuo (mTLS) para los corredores de RabbitMQ. El complemento de autenticación SSL utiliza los certificados de cliente de las conexiones mTLS para autenticar a los usuarios. En este método, los usuarios del broker se autentican mediante certificados de cliente X.509 en lugar de credenciales de nombre de usuario y contraseña. El certificado del cliente se valida con una autoridad de certificación (CA) de confianza y el nombre de usuario se extrae de un campo del certificado, como el nombre común (CN) o el nombre alternativo del sujeto (SAN). Este método proporciona una autenticación sólida sin transmitir las credenciales a través de la red. Para obtener más información, consulte [Autenticación con certificado SSL](#).

 Note

RabbitMQ admite varios métodos de autenticación y autorización para su uso simultáneo. Por ejemplo, puede habilitar la autenticación OAuth 2.0 y la autenticación simple (interna). Para obtener más información, consulte la sección del tutorial OAuth 2.0 sobre cómo [habilitar la autenticación OAuth 2.0 y la autenticación simple \(interna\)](#) y la documentación sobre el control de [acceso de RabbitMQ](#).

Amazon MQ recomienda crear un usuario interno al probar las configuraciones de autenticación. Esto permite validar la configuración de acceso mediante la API de administración de RabbitMQ. [Para obtener más información, consulte Validación de acceso.](#)

AWS políticas gestionadas para Amazon MQ

Una política AWS administrada es una política independiente creada y administrada por AWS. Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir [políticas administradas por el cliente](#) específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando estén disponibles nuevas operaciones de API para los servicios existentes.

Para obtener más información, consulte [Políticas administradas por AWS](#) en la Guía del usuario de IAM.

Amazon MQ admite las siguientes políticas AWS gestionadas:

- [AmazonMQApiFullAccess](#)
- [AmazonMQApiReadOnlyAccess](#)
- [Amazon MQFull Access](#)
- [AmazonMQReadOnlyAccess](#)

- [AmazonMQServiceRolePolicy](#)

AWS política gestionada: Amazon MQService RolePolicy

No puede asociar AmazonMQServiceRolePolicy a sus entidades de IAM. Esta política está asociada a un rol vinculado a un servicio que permite que Amazon MQ realice acciones en su nombre. Para obtener más información acerca de esta política de permisos y las acciones que le permite realizar a Amazon MQ, consulte [the section called “Permisos de roles vinculados a servicios para Amazon MQ”](#).

Amazon MQ actualiza las políticas gestionadas AWS

Consulta los detalles sobre las actualizaciones de las políticas AWS gestionadas de Amazon MQ desde que este servicio comenzó a realizar el seguimiento de estos cambios. Para obtener alertas automáticas sobre cambios en esta página, suscríbese a la fuente RSS en la página de [historial de documentos](#) de la API de Amazon MQ.

Cambio	Descripción	Fecha
Amazon MQ comenzó a realizar el seguimiento de los cambios	Amazon MQ comenzó a realizar un seguimiento de los cambios en sus políticas AWS gestionadas.	5 de mayo de 2021

Uso de roles vinculados a servicios para Amazon MQ

[Amazon MQ utiliza funciones vinculadas a AWS Identity and Access Management servicios \(IAM\).](#)

Un rol vinculado a un servicio es un tipo único de rol de IAM que está vinculado directamente a Amazon MQ. Amazon MQ predefine las funciones vinculadas al servicio e incluyen todos los permisos que el servicio requiere para llamar a AWS otros servicios en tu nombre.

Un rol vinculado a un servicio simplifica la configuración de Amazon MQ porque ya no tendrá que agregar manualmente los permisos requeridos. Amazon MQ define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo Amazon MQ puede asumir

sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de Amazon MQ, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#) y busque los servicios que muestran Yes (Sí) en la columna Service-Linked Role (Rol vinculado a servicio). Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

Permisos de roles vinculados a servicios para Amazon MQ

Amazon MQ usa el rol vinculado al servicio denominado MQ. Amazon MQ AWSService RoleForAmazon usa este rol vinculado al servicio para llamar a los servicios en tu nombre. AWS

El rol vinculado al servicio de AWSService RoleForAmazon MQ confía en los siguientes servicios para que asuman el rol:

- `mq.amazonaws.com`

Amazon MQ utiliza la política de permisos [AmazonMQServiceRolePolicy](#), que se adjunta a la función vinculada al servicio de AWSService RoleForAmazon MQ, para realizar las siguientes acciones en los recursos especificados:

- Acción: `ec2:CreateVpcEndpoint` en el recurso `vpc`.
- Acción: `ec2:CreateVpcEndpoint` en el recurso `subnet`.
- Acción: `ec2:CreateVpcEndpoint` en el recurso `security-group`.
- Acción: `ec2:CreateVpcEndpoint` en el recurso `vpc-endpoint`.
- Acción: `ec2:DescribeVpcEndpoints` en el recurso `vpc`.
- Acción: `ec2:DescribeVpcEndpoints` en el recurso `subnet`.
- Acción: `ec2:CreateTags` en el recurso `vpc-endpoint`.

- Acción: `logs:PutLogEvents` en el recurso `log-group`.
- Acción: `logs:DescribeLogStreams` en el recurso `log-group`.
- Acción: `logs:DescribeLogGroups` en el recurso `log-group`.
- Acción: `CreateLogStream` en el recurso `log-group`.
- Acción: `CreateLogGroup` en el recurso `log-group`.

Cuando crea un agente de Amazon MQ para RabbitMQ, la política de permisos `AmazonMQServiceRolePolicy` permite que Amazon MQ realice las siguientes tareas en su nombre.

- Cree un punto de enlace de la Amazon VPC para el agente utilizando la VPC, la subred y el grupo de seguridad de Amazon que proporcione. Puede utilizar el punto de enlace que creó para su agente para conectarse con el agente a través de la consola de administración de RabbitMQ, la API de administración o mediante programación.
- Cree grupos de registros y publique registros de corredores en Amazon CloudWatch Logs.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcEndpoints"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVpcEndpoint"
      ],
      "Resource": [
```



```

        "arn:aws:ec2:*:*:vpc/*",
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
    ],
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateVpcEndpoint"
    ],
    "Resource": [
        "arn:aws:ec2:*:*:vpc-endpoint/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/AMQManaged": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "CreateVpcEndpoint"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:DeleteVpcEndpoints"
    ],
    "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/AMQManaged": "true"
        }
    }
},
{

```

```

    "Effect": "Allow",
    "Action": [
        "logs:PutLogEvents",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups",
        "logs:CreateLogStream",
        "logs:CreateLogGroup"
    ],
    "Resource": [
        "arn:aws:logs:*:*:log-group:/aws/amazonmq/*"
    ]
}

```

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte [Permisos de roles vinculados a servicios](#) en la Guía del usuario de IAM.

Creación de un rol vinculado a un servicio para Amazon MQ

No necesita crear manualmente un rol vinculado a servicios. Cuando creas un agente por primera vez, Amazon MQ crea un rol vinculado a un servicio para llamar a los AWS servicios en tu nombre. Todos los agentes que cree posteriormente utilizarán el mismo rol y no se creará ningún nuevo rol.

Important

Este rol vinculado a servicios puede aparecer en su cuenta si se ha completado una acción en otro servicio que utilice las características compatibles con este rol. Para obtener más información, consulte [Un nuevo rol ha aparecido en mi cuenta de IAM](#).

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta.

También puede utilizar la consola de IAM para crear un rol vinculado al servicio con el caso de uso de Amazon MQ. En la API AWS CLI o en la AWS API, cree una función vinculada al servicio con el nombre del servicio. `mq.amazonaws.com` Para obtener más información, consulte [Crear un rol vinculado a un servicio](#) en la Guía del usuario de IAM. Si elimina este rol vinculado al servicio, puede utilizar este mismo proceso para volver a crear el rol.

 Important

Los roles vinculados a servicios solo se crean para Amazon MQ para RabbitMQ.

Edición de un rol vinculado a un servicio para Amazon MQ

Amazon MQ no le permite editar el rol vinculado al servicio de AWSService RoleForAmazon MQ. Sin embargo, puede editar la descripción del rol mediante IAM. Para obtener más información, consulte [Editar un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Eliminación de un rol vinculado a un servicio para Amazon MQ

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Así no tendrá una entidad no utilizada que no se supervise ni mantenga de forma activa. Sin embargo, debe limpiar los recursos de su rol vinculado al servicio antes de eliminarlo manualmente.

 Note

Si el servicio de Amazon MQ está utilizando el rol cuando usted intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de Amazon MQ utilizados por el MQ AWSService RoleForAmazon

- Elimine sus agentes de Amazon MQ mediante la Consola de administración de AWS CLI de Amazon MQ o la API de Amazon MQ. Para obtener más información acerca de cómo eliminar agentes, consulte [???](#).

Para eliminar manualmente el rol vinculado a servicios mediante IAM

Utilice la consola de IAM AWS CLI, la o la AWS API para eliminar la función vinculada al AWSService RoleForAmazon servicio MQ. Para obtener más información, consulte [Eliminación de un rol vinculado a servicios](#) en la Guía del usuario de IAM.

Regiones admitidas para los roles vinculados a servicios de Amazon MQ

Amazon MQ admite el uso de roles vinculados a servicios en todas las regiones en las que el servicio está disponible. Para obtener más información, consulte [AWS Regiones y puntos de conexión](#).

Nombre de la región	Identidad de la región	Compatibilidad en Amazon MQ
Este de EE. UU. (Norte de Virginia)	us-east-1	Sí
Este de EE. UU. (Ohio)	us-east-2	Sí
Oeste de EE. UU. (Norte de California)	us-west-1	Sí
Oeste de EE. UU. (Oregón)	us-west-2	Sí
Asia-Pacífico (Mumbai)	ap-south-1	Sí
Asia-Pacífico (Osaka)	ap-northeast-3	Sí
Asia-Pacífico (Seúl)	ap-northeast-2	Sí
Asia-Pacífico (Singapur)	ap-southeast-1	Sí
Asia-Pacífico (Sídney)	ap-southeast-2	Sí
Asia-Pacífico (Tokio)	ap-northeast-1	Sí
Canadá (centro)	ca-central-1	Sí
Europa (Fráncfort)	eu-central-1	Sí
Europa (Irlanda)	eu-west-1	Sí
Europa (Londres)	eu-west-2	Sí
Europa (París)	eu-west-3	Sí
América del Sur (São Paulo)	sa-east-1	Sí
AWS GovCloud (US)	us-gov-west-1	No

Solución de problemas de identidad y acceso de Amazon MQ

Utilice la siguiente información para diagnosticar y solucionar los problemas comunes que es posible que surjan cuando se trabaja con Amazon MQ e IAM.

Temas

- [No tengo autorización para realizar una acción en Amazon MQ](#)
- [No estoy autorizado a realizar iam: PassRole](#)
- [Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de Amazon MQ](#)

No tengo autorización para realizar una acción en Amazon MQ

Si Consola de administración de AWS le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

El siguiente ejemplo de error se produce cuando el mateojackson usuario intenta usar la consola para ver los detalles de una *widget*, pero no tiene *mq:GetWidget* permisos.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
mq:GetWidget on resource: my-example-widget
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *my-example-widget* mediante la acción *mq:GetWidget*.

No estoy autorizado a realizar iam: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción *iam:PassRole*, sus políticas deben actualizarse para permitirle pasar un rol a Amazon MQ.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir la función al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado *marymajor* intenta utilizar la consola para realizar una acción en Amazon MQ. Sin embargo, la acción requiere que el servicio cuente con permisos que otorga un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis recursos de Amazon MQ

Se puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Se puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admiten políticas basadas en recursos o listas de control de acceso (ACLs), puede usar esas políticas para permitir que las personas accedan a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si Amazon MQ admite estas características, consulte [Cómo funciona Amazon MQ con IAM](#).
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte [Proporcionar acceso a un usuario de IAM en otro de su propiedad en la Cuenta de AWS Guía del usuario](#) de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta [Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros](#) en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte [Proporcionar acceso a usuarios autenticados externamente \(identidad federada\)](#) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte [Acceso a recursos entre cuentas en IAM](#) en la Guía del usuario de IAM.

Validación de la conformidad para Amazon MQ

Los auditores externos evalúan la seguridad y el cumplimiento de Amazon MQ como parte de varios programas de AWS cumplimiento. Esto incluye SOC, PCI, HIPAA y otros.

Para saber si uno Servicio de AWS está dentro del ámbito de aplicación de programas de conformidad específicos, consulte [Servicios de AWS Alcance por programa de conformidad Servicios de AWS](#) y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de [AWS cumplimiento > Programas AWS](#).

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#).

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. Para obtener más información sobre su responsabilidad de conformidad al utilizarlos Servicios de AWS, consulte [AWS la documentación de seguridad](#).

Resiliencia en Amazon MQ

La infraestructura global de AWS se compone de regiones y zonas de disponibilidad de AWS. AWS Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre las regiones y zonas de disponibilidad de AWS, consulte [Infraestructura global de AWS](#).

Seguridad de la infraestructura en Amazon MQ

Como servicio gestionado, Amazon MQ está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte [Seguridad AWS en la nube](#). Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte [Protección de infraestructuras en un marco](#) de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las llamadas a la API AWS publicadas para acceder a Amazon MQ a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Prácticas recomendadas de seguridad para Amazon MQ

Los siguientes patrones de diseño pueden mejorar la seguridad de su agente de Amazon MQ.

Temas

- [Preferir agentes sin accesibilidad pública](#)
- [Configurar siempre una asignación de autorizaciones](#)
- [Bloqueo de protocolos innecesarios con grupos de seguridad de VPC](#)

Para obtener más información acerca de cómo Amazon MQ cifra sus datos, así como una lista de protocolos compatibles, consulte el tema sobre [protección de datos](#).

Preferir agentes sin accesibilidad pública

A los agentes creados sin acceso público no se puede obtener acceso desde fuera de la [VPC](#). Esto reduce en gran medida la susceptibilidad de su corredor a los ataques de denegación de servicio distribuido desde la Internet pública. DDo Para obtener más información, consulte [Cómo prepararse para los ataques DDo S reduciendo su superficie de ataque](#) en el blog sobre AWS seguridad.

Configurar siempre una asignación de autorizaciones

Debido a que ActiveMQ no tiene configurada ninguna asignación de autorizaciones, cualquier usuario autenticado puede llevar a cabo cualquier acción en el agente. Por lo tanto, la práctica recomendada consiste en restringir los permisos por grupo. Para obtener más información, consulte [authorizationEntry](#).

 Important

Si especifica un mapa de autorizaciones que no incluya al grupo `activemq-webconsole`, no puede utilizar la consola web de ActiveMQ porque el grupo no está autorizado a enviar mensajes al agente de Amazon MQ ni para recibir mensajes de este.

Bloqueo de protocolos innecesarios con grupos de seguridad de VPC

Para mejorar la seguridad de los agentes privados, debe restringir las conexiones de los protocolos y puertos innecesarios configurando correctamente el grupo de seguridad de Amazon VPC. Por ejemplo, para restringir el acceso a la mayoría de los protocolos y, al mismo tiempo, permitir el acceso a la consola web OpenWire y a ella, solo puede permitir el acceso a los protocolos 61617 y 8162. Esto limita tu exposición al bloquear los protocolos que no utilizas OpenWire y, al mismo tiempo, permitir que la consola web funcione normalmente.

Permita únicamente los puertos de los protocolos que esté utilizando.

- AMQP: 5671
- MQTT: 8883
- OpenWire: 61617
- STOMP: 61614
- WebSocket: 61619

Para obtener más información, consulte:

- [Grupos de seguridad de su VPC](#)
- [Grupo de seguridad predeterminado para su VPC](#)
- [Uso de grupos de seguridad](#)

Registro y monitoreo de agentes de Amazon MQ

El monitoreo es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de sus AWS soluciones. Debe recopilar los datos de supervisión de todas las partes de la AWS solución para poder depurar más fácilmente una falla multipunto en caso de que se produzca. AWS proporciona varias herramientas para supervisar los recursos de Amazon MQ y responder a posibles incidentes:

Puede utilizarla CloudWatch para ver y analizar las métricas de su agente de Amazon MQ. Puede ver y analizar las métricas de su corredor desde la CloudWatch consola AWS CLI, el o el CloudWatch AWS CLI. CloudWatch Las métricas de Amazon MQ las consulta automáticamente el bróker y, a continuación, se transfieren a CloudWatch cada minuto. En el caso de los corredores ActiveMQ CloudWatch , monitorea solo los primeros 1000 destinos. En el caso de los corredores RabbitMQ, CloudWatch monitorea solo los primeros 500 destinos, ordenados por número de consumidores.

Para obtener una lista completa de las métricas de Amazon MQ, consulte [CloudWatch Métricas disponibles Amazon MQ para corredores de ActiveMQ](#).

Para obtener información sobre cómo crear una CloudWatch alarma para una métrica, consulta [Crear o editar una CloudWatch alarma](#) en la Guía del CloudWatch usuario de Amazon.

Acceso a CloudWatch las métricas de Amazon MQ

Puedes acceder a CloudWatch las métricas mediante la Consola de administración de AWS API y. AWS CLI

Es posible que desee acceder a CloudWatch las métricas sin utilizar la Consola de administración de AWS.

Para acceder a las métricas de Amazon MQ mediante el AWS CLI, utilice el [get-metric-statistics](#) comando. Para obtener más información, consulta [Obtener estadísticas de una métrica](#) en la Guía del CloudWatch usuario de Amazon.

Para acceder a las métricas de Amazon MQ mediante la CloudWatch API, usa la [GetMetricStatistics](#) acción. Para obtener más información, consulta [Obtener estadísticas de una métrica](#) en la Guía del CloudWatch usuario de Amazon.

Acceder a las CloudWatch métricas mediante el Consola de administración de AWS

En el siguiente ejemplo, se muestra cómo acceder a CloudWatch las métricas de Amazon MQ mediante la opción Consola de administración de AWS. Si ya ha iniciado sesión en la consola de Amazon MQ, en la página de detalles del agente, seleccione Acciones, Ver métricas. CloudWatch

1. Inicie sesión en la [consola de CloudWatch](#).
2. En el panel de navegación, elija Metrics.
3. Seleccione el espacio de nombres de métrica AmazonMQ.
4. Seleccione una de las siguientes dimensiones de métricas:
 - Broker Metrics
 - Queue Metrics by Broker
 - Topic Metrics by Broker

En este ejemplo, se selecciona Broker Metrics (Métricas del agente).

5. Ahora puede examinar las métricas de Amazon MQ:
 - Para ordenar las métricas, utilice el encabezado de columna.
 - Para representar gráficamente una métrica, active la casilla de verificación situada junto a ella.
 - Para filtrar por métrica, elija el nombre de la métrica y, a continuación, seleccione Add to search (Añadir a búsqueda).

CloudWatch Métricas disponibles Amazon MQ para corredores de ActiveMQ

Métricas de Amazon MQ para ActiveMQ

Métrica	Unidad	Description (Descripción)
AmqpMaximumConnections	Recuento	Número máximo de clientes que se pueden conectar al

Métrica	Unidad	Description (Descripción)
		agente mediante AMQP. Para obtener más información sobre cuotas de conexión, consulte Quotas in Amazon MQ .
BurstBalance	Porcentaje	Porcentaje de créditos de ráfaga que queda en el volumen de Amazon EBS que se utiliza para conservar los datos de mensajes para los agentes de rendimiento optimizado. Si este valor llega a cero, las IOPS proporcionadas por el volumen de Amazon EBS disminuirán hasta que se vuelva a llenar la cuota de ráfagas. Para obtener más información sobre cómo funcionan las cuotas de ráfagas de Amazon EBS, consulte: Créditos de E/S y rendimiento por ráfagas .

Métrica	Unidad	Description (Descripción)
CpuCreditBalance	Créditos (vCPU/minutos)	 Important Esta métrica solo está disponible para el tipo de instancias <code>mq.t2.micro</code> del agente. Las métricas de créditos de CPU solo están disponibles en intervalos de cinco minutos. Cantidad de créditos de CPU obtenidos que una instancia ha acumulado desde que se lanzó o se inició (incluido el número de créditos del lanzamiento). El saldo de créditos está disponible para que la instancia del agente lo gaste en ráfagas que aumenten el uso de la CPU por encima de la referencia. Los créditos se acumulan en el saldo de créditos cuando se consiguen y se eliminan del saldo de créditos cuando se gastan. El saldo de créditos tiene un límite máximo. Una vez que se alcanza dicho límite, los nuevos créditos que

Métrica	Unidad	Description (Descripción)
		se consiguen no se tienen en cuenta.
CpuUtilization	Porcentaje	Porcentaje de unidades informáticas Amazon EC2 asignadas que el agente usa actualmente.
CurrentConnectionsCount	Recuento	Número actual de conexiones activas en el agente actual.
EstablishedConnectionsCount	Recuento	El número total de conexiones, activas e inactivas, que se han establecido en el agente.
HeapUsage	Porcentaje	Porcentaje del límite de memoria JVM de ActiveMQ que el agente usa actualmente.
InactiveDurableTopicSubscribersCount	Recuento	El número de suscriptores inactivos a temas duraderos, hasta un máximo de 2000.
JobSchedulerStorePercentUsage	Porcentaje	Porcentaje de espacio en disco utilizado por el almacén del programador de trabajos.
JournalFilesForFastRecovery	Recuento	Número de archivos de diario que volverán a reproducirse después de un cierre correcto.
JournalFilesForFullRecovery	Recuento	Número de archivos de diario que volverán a reproducirse después de un cierre incorrecto.

Métrica	Unidad	Description (Descripción)
MqttMaximumConnections	Recuento	Número máximo de clientes que se pueden conectar al agente mediante MQTT. Para obtener más información sobre cuotas de conexión, consulte Quotas in Amazon MQ .
NetworkConnectorConnectionCount	Recuento	El número de nodos conectados al bróker en una red de brókeres que lo utilizan NetworkConnector.
NetworkIn	Bytes	El volumen de tráfico entrante para el agente.
NetworkOut	Bytes	El volumen de tráfico saliente para el agente.
OpenTransactionCount	Recuento	Número total de transacciones en curso.
OpenwireMaximumConnections	Recuento	El número máximo de clientes que puede utilizar para conectarse a su bróker OpenWire. Para obtener más información sobre cuotas de conexión, consulte Quotas in Amazon MQ .

Métrica	Unidad	Description (Descripción)
StompMaximumConnections	Recuento	Número máximo de clientes que se pueden conectar al agente mediante STOMP. Para obtener más información sobre cuotas de conexión, consulte Quotas in Amazon MQ .
StorePercentUsage	Porcentaje	Porcentaje utilizado por el límite de almacenamiento. Si llega a 100, el agente rechazará los mensajes.
TempPercentUsage	Porcentaje	Porcentaje de almacenamiento temporal disponible utilizado por mensajes no persistentes.
TotalConsumerCount	Recuento	Número de consumidores de mensajes suscritos en los destinos del agente actual.
TotalMessageCount	Recuento	El número de mensajes almacenados en el agente.
TotalProducerCount	Recuento	Número de productores de mensajes activos en los destinos del agente actual.
VolumeReadOps	Recuento	Número de operaciones de lectura realizadas en el volumen de Amazon EBS.
VolumeWriteOps	Recuento	Número de operaciones de escritura realizadas en el volumen de Amazon EBS.

Métrica	Unidad	Description (Descripción)
WsMaximumConnections	Recuento	El número máximo de clientes que puede utilizar para conectarse a su bróker WebSocket. Para obtener más información sobre cuotas de conexión, consulte Quotas in Amazon MQ .

Dimensiones para las métricas de agentes de ActiveMQ

Dimensión	Description (Descripción)
Broker	Nombre del agente  Note Los agentes con una sola instancia tienen el sufijo -1. Un active/standby bróker de alta disponibilidad tiene los sufijos -1 y -2 para su par redundante.

Métricas de destino (cola y tema) de ActiveMQ

Important

Las siguientes métricas incluyen los recuentos por minuto durante el CloudWatch período de votación.

- EnqueueCount
- ExpiredCount
- DequeueCount
- DispatchCount

- InFlightCount

Por ejemplo, en un [período de CloudWatch](#) de cinco minutos EnqueueCount tiene cinco valores de recuento, cada uno para una porción de un minuto del período. Las estadísticas Minimum y Maximum proporcionan el valor por minuto más alto y más bajo durante el período especificado.

Métrica	Unidad	Description (Descripción)
ConsumerCount	Recuento	El número de consumidores suscritos al destino.
EnqueueCount	Recuento	Número de mensajes enviados al destino (por minuto).
EnqueueTime	Tiempo (milisegundos)	La end-to-end latencia desde que un mensaje llega a un intermediario hasta que se entrega a un consumidor.

 **Note**

EnqueueTime no mide la end-to-end latencia desde que un productor envía un mensaje hasta que llega al intermediario, ni la latencia desde que un intermediario recibe un mensaje hasta que el intermediario lo confirma. Por el contrario, EnqueueTime es el número de

Métrica	Unidad	Description (Descripción)
		milisegundos desde el momento en que el agente recibe un mensaje hasta que se entrega correctamente a un consumidor.
ExpiredCount	Recuento	Número de mensajes que no se pudieron entregar por haber caducado (por minuto).
DispatchCount	Recuento	Número de mensajes enviados a los consumidores (por minuto).
DequeueCount	Recuento	Número de mensajes confirmados por el consumidor (por minuto).
InFlightCount	Recuento	Número de mensajes enviados a los consumidores que no se han confirmado.
ReceiveCount	Recuento	El número de mensajes que se han recibido del agente remoto para un conector de red dúplex.
MemoryUsage	Porcentaje	El porcentaje del límite de memoria que el destino utiliza actualmente.
ProducerCount	Recuento	El número de productores para el destino.

Métrica	Unidad	Description (Descripción)
QueueSize	Recuento	Número de mensajes de la cola.  Important Esta métrica se aplica únicamente a las colas.
TotalEnqueueCount	Recuento	Número total de mensajes que se han enviado al agente.
TotalDequeueCount	Recuento	Número total de mensajes que han consumido los clientes.

Note

Las métricas TotalEnqueueCount y TotalDequeueCount las incluyen mensajes para temas de asesoramiento. Para obtener más información acerca de los mensajes de temas de asesoramiento, consulte la [documentación de ActiveMQ](#).

Dimensiones de las métricas de destino (cola y tema) de ActiveMQ

Dimensión	Description (Descripción)
Broker	El nombre del agente.  Note Los agentes con una sola instancia tienen el sufijo -1. Un active/st andby intermediario de alta disponibi

Dimensión	Description (Descripción)
	lidad tiene los sufijos -1 y su -2 par redundante.
Topic o Queue	Nombre del tema o la cola.
NetworkConnector	El nombre del conector de red.

CloudWatch Métricas disponibles de Amazon MQ para corredores de RabbitMQ

Métricas de agentes de RabbitMQ

Métrica	Unidad	Description (Descripción)
ExchangeCount	Recuento	Número total de intercambios configurados en el agente.
QueueCount	Recuento	Número total de colas configuradas en el agente.
ConnectionCount	Recuento	Número total de conexiones establecidas en el agente.
ChannelCount	Recuento	Número total de canales establecidos en el agente.
ConsumerCount	Recuento	Número total de consumidores conectados al agente.
MessageCount	Recuento	Número total de mensajes en las colas.

Métrica	Unidad	Description (Descripción)
		 Note El número que se obtiene correspon de a la suma total de mensajes listos y no reconocidos en el agente.
MessageReadyCount	Recuento	Número total de mensajes listos en las colas.
MessageUnacknowledgedCount	Recuento	Número total de mensajes sin confirmar en las colas.
PublishRate	Recuento	Velocidad a la que se publican los mensajes para el agente. El número que se muestra representa el número de mensajes por segundo en el momento del muestreo.
ConfirmRate	Recuento	Velocidad a la que el servidor de RabbitMQ confirma los mensajes publicados. Puede comparar esta métrica con PublishRate para entender mejor el rendimiento de su agente. El número que se muestra representa el número de mensajes por segundo en el momento del muestreo.

Métrica	Unidad	Description (Descripción)
AckRate	Recuento	Velocidad a la que los consumidores confirman los mensajes. El número que se muestra representa el número de mensajes por segundo en el momento del muestreo.
SystemCpuUtilization	Porcentaje	Porcentaje de unidades informáticas Amazon EC2 asignadas que el agente usa actualmente. Para las implementaciones de clúster, este valor represent a el agregado de todos los valores de métrica correspondientes de los tres nodos de RabbitMQ.
RabbitMQMemLimit	Bytes	Límite de RAM para un agente de RabbitMQ. Para las implementaciones de clúster, este valor represent a el agregado de todos los valores de métrica correspondientes de los tres nodos de RabbitMQ.

Métrica	Unidad	Description (Descripción)
RabbitMQMemUsed	Bytes	Volumen de RAM utilizado por un agente de RabbitMQ. Para las implementaciones de clúster, este valor represent a el agregado de todos los valores de métrica correspondientes de los tres nodos de RabbitMQ.
RabbitMQDiskFreeLimit	Bytes	Límite de disco para un agente de RabbitMQ. Para las implementaciones de clúster, este valor represent a el agregado de todos los valores de métrica correspondientes de los tres nodos de RabbitMQ. Esta métrica varía según el tamaño de la instancia.
RabbitMQDiskFree	Bytes	Volumen total de espacio libre en disco disponible en un agente de RabbitMQ. Cuando el uso del disco supera su límite, el clúster bloqueará todas las conexiones del productor. Para las implementaciones de clúster, este valor representa el agregado de todos los valores de métrica correspondientes de los tres nodos de RabbitMQ.

Métrica	Unidad	Description (Descripción)
RabbitMQFdUsed	Recuento	Número de descriptores de archivo usados. Para las implementaciones de clúster, este valor represent a el agregado de todos los valores de métrica correspondientes de los tres nodos de RabbitMQ.
RabbitMQIOReadAverageTime	Recuento	El tiempo medio (en milisegundos) que RabbitMQ tarda en realizar una operación de lectura. El valor es proporcional al tamaño del mensaje.
RabbitMQIOWriteAverageTime	Recuento	El tiempo medio (en milisegundos) que RabbitMQ tarda en realizar una operación de escritura. El valor es proporcional al tamaño del mensaje.

Dimensiones para las métricas de agentes de RabbitMQ

Dimensión	Description (Descripción)
Broker	El nombre del agente.

Métricas de nodos de RabbitMQ

Métrica	Unidad	Description (Descripción)
SystemCpuUtilization	Porcentaje	Porcentaje de unidades informáticas Amazon EC2

Métrica	Unidad	Description (Descripción)
		asignadas que el agente usa actualmente.
RabbitMQMemLimit	Bytes	Límite de RAM para un nodo de RabbitMQ.
RabbitMQMemUsed	Bytes	Volumen de RAM utilizado por un nodo de RabbitMQ. Cuando el uso de memoria supera el límite, el clúster bloqueará todas las conexiones del productor.
RabbitMQDiskFreeLimit	Bytes	Límite de disco para un nodo de RabbitMQ. Esta métrica varía según el tamaño de la instancia.
RabbitMQDiskFree	Bytes	Volumen total de espacio libre en disco disponible en un nodo de RabbitMQ. Cuando el uso del disco supera su límite, el clúster bloqueará todas las conexiones del productor.
RabbitMQFdUsed	Recuento	Número de descriptores de archivo usados.

Dimensiones para métricas de nodos de RabbitMQ

Dimensión	Description (Descripción)
Node	Nombre del nodo.

Dimensión	Description (Descripción)
	 Note El nombre de un nodo consta de dos partes: un prefijo (normalmente <code>rabbit</code>) y un nombre de anfitrión. Por ejemplo, <code>rabbit@ip-10-0-0-230.us-west-2.compute.internal</code> es un nombre de nodo con el prefijo <code>rabbit</code> y el nombre de anfitrión <code>ip-10-0-0-230.us-west-2.compute.internal</code>.
Broker	El nombre del agente.

Métricas de colas de RabbitMQ

Métrica	Unidad	Description (Descripción)
ConsumerCount	Recuento	Número de consumidores suscritos a la cola.
MessageReadyCount	Recuento	Número de mensajes actualmente disponibles para entregarse.
MessageUnacknowledgedCount	Recuento	Número de mensajes para los que el servidor está a la espera de confirmación.
MessageCount	Recuento	Número total de <code>MessageReadyCount</code> y <code>MessageUnacknowledgedCount</code> (también conocido como profundidad de la cola).

Dimensiones para métricas de colas de RabbitMQ

Note

Amazon MQ para RabbitMQ no publicará métricas para anfitriones virtuales ni colas con nombres que contengan espacios en blanco, pestañas u otros caracteres que no son ASCII. Para obtener más información sobre los nombres de las dimensiones, consulta [Dimension](#) en la Amazon CloudWatch API Reference.

Dimensión	Description (Descripción)
Queue	El nombre de la cola .
VirtualHost	El nombre del host virtual.
Broker	El nombre del agente.

Métricas de redes de RabbitMQ

Métrica	Unidad	Description (Descripción)
NetworkOut	Bytes	El número de bytes enviados por la instancia en todas las interfaces de red. Esta métrica identifica el volumen de tráfico de red saliente de una sola instancia. El número registrado es el número de bytes enviados durante el periodo. Si utiliza el monitoreo básico (5 minutos) y la estadística es Suma, puede dividir este número por 300 para conocer los bytes por segundo. Si utiliza el monitoreo detallado (1 minuto) y la estadística es Suma, divídalo por 60. También puede utilizar la función matemática CloudWatch métrica DIFF_TIME para encontrar los bytes por segundo. Por ejemplo, si ha representado gráficamente NetworkOut CloudWatch como m1, la fórmula matemática a métrica $m1 / (DIFF\_TIME(m1))$ devuelve la métrica en bytes/segundo. Para obtener más información sobre

Métrica	Unidad	Description (Descripción)
		DIFF_TIME y otras funciones matemáticas de métricas, consulte Uso de matemáticas de métricas. Estadísticas significativas: suma, promedio, mínimo, máximo
NetworkIn	Bytes	El número de bytes recibidos por la instancia en todas las interfaces de red. Esta métrica identifica el volumen de tráfico de red entrante para una sola instancia. El número registrado es el número de bytes recibidos durante el periodo. Si utiliza el monitoreo básico (5 minutos) y la estadística es Suma, puede dividir este número por 300 para conocer los bytes por segundo. Si utiliza el monitoreo detallado (1 minuto) y la estadística es Suma, divídalo por 60. También puede usar la función matemática CloudWatch métrica DIFF_TIME para encontrar los bytes por segundo. Por ejemplo, si ha representado gráficamente NetworkIn CloudWatch como m1, la fórmula matemática métrica $m1 / (DIFF\_TIME(m1))$ devuelve la métrica en bytes/segundo. Para obtener más información sobre DIFF_TIME y otras funciones matemáticas de métricas, consulte Uso de matemáticas de métricas. Estadísticas significativas: suma, promedio, mínimo, máximo

Dimensiones los agentes de RabbitMQ

Dimensión	Description (Descripción)
BrokerId	ID del agente

Configuración de registros de Amazon MQ para RabbitMQ

Cuando habilitas el CloudWatch registro para tus agentes de RabbitMQ, Amazon MQ utiliza una función vinculada a un servicio para publicar los registros generales. CloudWatch Si no existe un rol vinculado a servicios de Amazon MQ cuando cree un agente por primera vez, Amazon MQ creará uno automáticamente. Todos los corredores de RabbitMQ posteriores utilizarán la misma función vinculada al servicio para publicar los registros. CloudWatch

Para obtener más información acerca roles vinculados a servicios, consulte [Uso de roles vinculados a servicios](#) en la Guía del usuario de AWS Identity and Access Management . Para obtener más información acerca de cómo utiliza Amazon MQ los roles vinculados a servicios, consulte [the section called “Cómo utilizar roles vinculados a servicios”](#).

Registro de llamadas a la API de Amazon MQ mediante AWS CloudTrail

Amazon MQ está integrado con AWS CloudTrail un servicio que proporciona un registro de las llamadas de Amazon MQ que realiza un usuario, un rol o un servicio. AWS CloudTrail captura las llamadas a la API relacionadas con los corredores y las configuraciones de Amazon MQ como eventos, incluidas las llamadas desde la consola de Amazon MQ y las llamadas en código desde Amazon MQ. APIs [Para obtener más información al respecto CloudTrail, consulte la Guía del AWS CloudTrail usuario.](#)

Note

CloudTrail no registra las llamadas a la API relacionadas con las operaciones de ActiveMQ (por ejemplo, el envío y la recepción de mensajes) ni con la consola web ActiveMQ. Para registrar la información relacionada con las operaciones de ActiveMQ, puede configurar [Amazon MQ para que publique registros generales y de auditoría en Amazon](#) Logs. CloudWatch

Con la información CloudTrail recopilada, puede identificar una solicitud específica a una API de Amazon MQ, la dirección IP del solicitante, la identidad del solicitante, la fecha y hora de la solicitud, etc. Si configura una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3. Si no configura una ruta, puede ver los eventos más recientes en el historial de eventos

de la CloudTrail consola. Para obtener más información, consulte [Información general acerca de la creación de registros de seguimiento](#) en la [Guía del usuario de AWS CloudTrail](#).

Información de Amazon MQ en CloudTrail

Cuando crea su AWS cuenta, CloudTrail está habilitada. Cuando se produce una actividad de eventos de Amazon MQ compatible, se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los últimos eventos de su cuenta de AWS . Para obtener más información, consulte [Visualización de eventos con el historial de CloudTrail eventos](#) en la Guía del AWS CloudTrail usuario.

Un rastro permite CloudTrail entregar los archivos de registro a un bucket de Amazon S3. Puede crear un registro para mantener un registro continuo de los eventos en su AWS cuenta. De forma predeterminada, cuando creas una ruta con la Consola de administración de AWS, la ruta se aplica a todas AWS las regiones. La ruta registra los eventos de todas AWS las regiones y entrega los archivos de registro al bucket de Amazon S3 especificado. También puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para obtener más información, consulte los siguientes temas en la Guía del usuario de AWS CloudTrail :

- [CloudTrail Servicios e integraciones compatibles](#)
- [Configuración de las notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir archivos de CloudTrail registro de varias regiones](#)
- [Recibir archivos de CloudTrail registro de varias cuentas](#)

Amazon MQ admite el registro de los parámetros de la solicitud y las respuestas de lo siguiente APIs como eventos en los archivos de CloudTrail registro:

- [CreateConfiguration](#)
- [DeleteBroker](#)
- [DeleteUser](#)
- [RebootBroker](#)
- [UpdateBroker](#)

 Note

RebootBroker los archivos de registro se registran al reiniciar el bróker. Durante el período de mantenimiento, el servicio se reinicia automáticamente y los archivos de RebootBroker registro no se registran.

 Important

Para los siguientes GET métodos APIs, se registran los parámetros de la solicitud, pero las respuestas se redactan:

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

En los siguientes casos APIs, los parámetros data y de la password solicitud se ocultan con asteriscos (): ***

- [CreateBroker](#) (POST)
- [CreateUser](#) (POST)
- [UpdateConfiguration](#) (PUT)
- [UpdateUser](#) (PUT)

Cada entrada de evento o registro contiene información acerca del solicitante. Esta información le ayuda a determinar lo siguiente:

- ¿La solicitud se realizó con las credenciales del nodo raíz o del usuario de ?

- ¿La solicitud se realizó con credenciales de seguridad temporales de una función o un usuario federado?
- ¿La solicitud la realizó otro servicio? AWS

Para obtener más información, consulte el elemento [CloudTrailUserIdentity en la Guía del usuario](#).AWS CloudTrail

Ejemplo de entrada del archivo de registro de Amazon MQ

Un rastro es una configuración que permite la entrega de eventos como archivos de registro al bucket de Amazon S3 especificado. CloudTrail Los archivos de registro contienen una o más entradas de registro.

Un evento representa una única solicitud de cualquier origen e incluye información sobre la solicitud a una API de Amazon MQ, la dirección IP del solicitante, la identidad del solicitante, la fecha y la hora de la solicitud, etcétera.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro para una llamada a la [CreateBroker](#)API.

Note

Como los archivos de CloudTrail registro no son una pila ordenada de archivos públicos APIs, no muestran la información en ningún orden específico.

```
{
  "eventVersion": "1.06",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
    "accountId": "111122223333",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "AmazonMqConsole"
  },
  "eventTime": "2018-06-28T22:23:46Z",
  "eventSource": "amazonmq.amazonaws.com",
  "eventName": "CreateBroker",
  "awsRegion": "us-west-2",
```

```
"sourceIpAddress": "203.0.113.0",
"userAgent": "PostmanRuntime/7.1.5",
"requestParameters": {
  "engineVersion": "5.15.9",
  "deploymentMode": "ACTIVE_STANDBY_MULTI_AZ",
  "maintenanceWindowStartTime": {
    "dayOfWeek": "THURSDAY",
    "timeOfDay": "22:45",
    "timeZone": "America/Los_Angeles"
  },
  "engineType": "ActiveMQ",
  "hostInstanceType": "mq.m5.large",
  "users": [
    {
      "username": "MyUsername123",
      "password": "****",
      "consoleAccess": true,
      "groups": [
        "admins",
        "support"
      ]
    },
    {
      "username": "MyUsername456",
      "password": "****",
      "groups": [
        "admins"
      ]
    }
  ],
  "creatorRequestId": "1",
  "publiclyAccessible": true,
  "securityGroups": [
    "sg-a1b234cd"
  ],
  "brokerName": "MyBroker",
  "autoMinorVersionUpgrade": false,
  "subnetIds": [
    "subnet-12a3b45c",
    "subnet-67d8e90f"
  ]
},
"responseElements": {
  "brokerId": "b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9",
```

```
    "brokerArn": "arn:aws:mq:us-  
east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9"  
  },  
  "requestID": "a1b2c345-6d78-90e1-f2g3-4hi56jk7l890",  
  "eventID": "a12bcd3e-fg45-67h8-ij90-12k34d5l16mn",  
  "readOnly": false,  
  "eventType": "AwsApiCall",  
  "recipientAccountId": "111122223333"  
}
```

Configuración de registros de Amazon MQ para ActiveMQ

Para permitir que Amazon MQ publique registros en Logs, debe [añadir un permiso a su usuario de Amazon MQ](#) y [configurar una política basada en recursos para Amazon MQ](#) antes de crear o reiniciar el agente. CloudWatch

Note

Al activar los registros y publicar mensajes desde la consola web ActiveMQ, el contenido del mensaje se envía CloudWatch a los registros y se muestra en ellos.

A continuación, se describen los pasos para configurar CloudWatch los registros de sus agentes de ActiveMQ.

Temas

- [Comprensión de la estructura del inicio de sesión en los registros CloudWatch](#)
- [Agregar el permiso CreateLogGroup a su usuario de Amazon MQ](#)
- [Configurar una política basada en recursos para Amazon MQ.](#)
- [Prevención de la sustitución confusa entre servicios](#)

Comprensión de la estructura del inicio de sesión en los registros CloudWatch

Puede habilitar registros generales y de auditoría cuando defina las opciones de configuración avanzada del agente o cuando edite un agente.

El registro general habilita el nivel de INFO registro predeterminado (no se admite el DEBUG registro) y se publica `activemq.log` en un grupo de registros de tu CloudWatch cuenta. El grupo de registros tiene un formato similar al siguiente:

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

El [registro de auditoría](#) permite registrar las acciones de administración realizadas mediante JMX o la consola web ActiveMQ y las `audit.log` publica en un grupo de registros de su cuenta. CloudWatch El grupo de registros tiene un formato similar al siguiente:

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

Dependiendo de si tiene un [agente de una sola instancia](#) o un [agente activo/en espera](#), Amazon MQ crea uno o dos flujos de registro en cada grupo de registros. Los flujos de registros tienen un formato similar al siguiente.

```
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.log  
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-2.log
```

Los sufijos -1 y -2 indican instancias de agentes individuales. Para obtener más información, consulte [Trabajar con grupos de registros y transmisiones](#) de [CloudWatch registros en la Guía del usuario de Amazon Logs](#).

Agregar el permiso **CreateLogGroup** a su usuario de Amazon MQ

Para permitir que Amazon MQ cree un grupo de CloudWatch registros, debe asegurarse de que el usuario que crea o reinicia el corredor tiene el permiso. `logs:CreateLogGroup`

Important

Si no agrega el permiso `CreateLogGroup` a su usuario de Amazon MQ antes de que el usuario cree o reinicie el agente, Amazon MQ no crea el grupo de registros.

El siguiente ejemplo de [política basada en IAM](#) concede permiso para `logs:CreateLogGroup` a los usuarios a los que se les asocia esta política.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*"
        }
    ]
}
```

Note

Aquí, el término usuario se refiere a usuarios, no a usuarios de Amazon MQ, que se crean cuando se configura un nuevo agente. Para obtener más información acerca de la configuración de usuarios y de políticas de IAM, consulte la sección [Información general sobre administración de identidades](#) de la Guía del usuario de IAM.

Para obtener más información, consulta [CreateLogGroup](#) la referencia de la API CloudWatch de Amazon Logs.

Configurar una política basada en recursos para Amazon MQ.

Important

Si no configuras una política basada en recursos para Amazon MQ, el agente no podrá publicar los registros en Logs. CloudWatch

Para permitir que Amazon MQ publique registros en su grupo de registros de CloudWatch Logs, configure una política basada en recursos para dar a Amazon MQ acceso a las siguientes acciones de la API de Logs: CloudWatch

- [CreateLogStream](#)— Crea un flujo de CloudWatch registros para el grupo de registros especificado.
- [PutLogEvents](#)— Envía los eventos al flujo de registro de CloudWatch registros especificado.

La siguiente política basada en recursos otorga permisos para `logs:CreateLogStream` y `logs:PutLogEvents` para. AWS

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": { "Service":
"mq.amazonaws.com" },
            "Action": [ "logs:CreateLogStream",
"logs:PutLogEvents" ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*"
        }
    ]
}
```

Esta política basada en recursos debe configurarse mediante el AWS CLI siguiente comando. En el ejemplo, reemplace `us-east-1` por su propia información.

```
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

Como en este ejemplo se usa el `/aws/amazonmq/` prefijo, debe configurar la política basada en recursos solo una vez por cuenta y por región. AWS

Prevención de la sustitución confusa entre servicios

El problema de la sustitución confusa es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación de identidad entre servicios puede provocar el confuso problema de un diputado. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que le ayudan a proteger los datos de todos los servicios cuyos directores de servicio tengan acceso a los recursos de su cuenta.

Te recomendamos que utilices las claves de contexto de condición [aws:SourceAccount](#) global [aws:SourceArn](#) y las claves de contexto de condición global en tu política basada en recursos de Amazon MQ para limitar el acceso a CloudWatch los registros a uno o más corredores específicos.

Note

Si se utilizan ambas claves contextuales de condición global, el valor `aws:SourceAccount` y la cuenta del valor `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utilicen en la misma declaración de política.

El siguiente ejemplo muestra una política basada en recursos que limita el acceso a CloudWatch Logs a un único agente de Amazon MQ.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "mq.amazonaws.com"
            },
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ]
        }
    ]
}
```

```

    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "123456789012",
        "aws:SourceArn": "arn:aws:mq:us-
west-1:123456789012:broker:my-broker:123456789012"
      }
    }
  }
}

```

También puede configurar su política basada en recursos para limitar el acceso a CloudWatch los registros a todos los corredores de una cuenta, como se muestra a continuación.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "mq.amazonaws.com"
        ]
      },
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn":
            "arn:aws:mq:*:123456789012:broker:*"
        },
        "StringEquals": {

```



```
        "aws:SourceAccount": "123456789012"  
      }  
    }  
  }  
]  
}
```

Para obtener más información sobre el problema de seguridad del suplente confuso, consulte [El problema del suplente confuso](#) en la Guía del usuario.

Solución de problemas CloudWatch de configuración de registros con Amazon MQ

En algunos casos, es posible que CloudWatch los registros no siempre se comporten como se esperaba. En esta sección se proporciona información general sobre los problemas comunes y se muestra cómo resolverlos.

Los grupos de registros no aparecen en CloudWatch

[Agregue el permiso `CreateLogGroup` a su usuario de Amazon MQ](#) y reinicie el agente. Esto permite que Amazon MQ crear el grupo de registros.

Los flujos de registro no aparecen en los grupos de CloudWatch registros

[Configure una política basada en recursos para Amazon MQ](#). Esto permite que el agente publique sus registros.

Cuotas en Amazon MQ

En este tema, se muestran los límites de Amazon MQ. Muchos de los siguientes límites pueden cambiarse para cuentas de AWS específicas. Para solicitar un aumento de cuota, consulte [Cuotas de servicio de AWS](#) en la Referencia general de Amazon Web Services. Los límites actualizados no estarán visibles incluso después de que se haya aplicado el aumento del límite. Para obtener más información sobre cómo ver los límites de conexión actuales en Amazon CloudWatch, consulte [Monitoreo de los agentes de Amazon MQ mediante Amazon CloudWatch](#).

Temas

- [Agentes](#)
- [Configuraciones](#)
- [Usuarios](#)
- [Almacenamiento de datos](#)
- [Limitación de API](#)

Agentes

En la siguiente tabla, se muestran las cuotas relacionadas con los agentes de Amazon MQ.

Límite	Descripción
Nombre del agente	• Debe ser exclusivo en la cuenta de AWS. • Debe tener de 1 a 50 caracteres. • Solo debe contener caracteres del conjunto de caracteres ASCII imprimibles. • Solo puede contener caracteres alfanuméricos, guiones, puntos, guiones bajos y tildes. (- . _ ~).
Número de agentes, por región	50

Límite	Descripción
Conexiones de nivel de cable por protocolo para agente más pequeño	 Important No se aplica a los agentes de RabbitMQ. 300 para agentes de tipo de instancias <code>mq.*.micro</code> .
Conexiones de nivel de cable por protocolo para agente más grande	 Important No se aplica a los agentes de RabbitMQ. 2000 para agentes de tipo de instancias <code>mq.*.*large</code> .
Grupos de seguridad por agente	5
Destinos de ActiveMQ (colas y temas) monitoreados en CloudWatch	CloudWatch monitorea solo los primeros 1000 destinos.
Destinos de RabbitMQ (colas) monitoreados en CloudWatch	CloudWatch monitorea solo los primeros 500 destinos, ordenados por número de consumidores.
Etiquetas por agente	50

Configuraciones

En la siguiente tabla, se muestran las cuotas relacionadas con las configuraciones de Amazon MQ.

Límite	Descripción
Nombre de la configuración	• Debe tener de 1 a 150 caracteres. • Solo debe contener caracteres del conjunto de caracteres ASCII imprimibles. • Solo puede contener caracteres alfanuméricos, guiones, puntos, guiones bajos y tildes. (- . _ ~).
Revisiones por configuración	300

Usuarios

En la siguiente tabla, se muestran las cuotas relacionadas con los usuarios del agente de Amazon MQ para ActiveMQ.

Límite	Descripción
Nombre de usuario	• Debe tener de 1 a 100 caracteres. • Solo debe contener caracteres del conjunto de caracteres ASCII imprimibles. • Solo puede contener caracteres alfanuméricos, guiones, puntos, guiones bajos y tildes. (- . _ ~). • No debe contener comas (,).
Contraseña	• Debe tener de 12 a 250 caracteres. • Solo debe contener caracteres del conjunto de caracteres ASCII imprimibles.

Límite	Descripción
	• Debe contener al menos 4 caracteres únicos. • No debe contener comas (,).
Usuarios por agente (autenticación simple)	250
Grupos por usuario (autenticación simple)	20

Almacenamiento de datos

En la siguiente tabla, se muestran las cuotas relacionadas con el almacenamiento de datos de Amazon MQ.

Límite	Descripción
Capacidad de almacenamiento por cada agente más pequeño	20 GB para agentes de tipo de instancias <code>mq.*.micro</code> . Para obtener más información acerca de los tipos de instancias de Amazon MQ, consulte Broker instance types .
Capacidad de almacenamiento por cada agente más grande	200 GB para agentes de tipo de instancias <code>mq.m5.*</code> . Para obtener más información acerca de los tipos de instancias de Amazon MQ, consulte Broker instance types .
Límite de uso del programador de trabajos por agente respaldado por Amazon EBS	 Important No se aplica a los agentes de RabbitMQ. 50 GB. Para obtener más información acerca del uso de programadores de trabajos, consulte

Límite	Descripción
	JobSchedulerUsage en la Documentación de la API de Apache ActiveMQ.
Capacidad de almacenamiento temporal por agente más pequeño.	 Important No se aplica a los agentes de RabbitMQ. 5 GB para agentes de tipo de instancias <code>mq.*.micro</code> .
Capacidad de almacenamiento temporal por agente más grande.	 Important No se aplica a los agentes de RabbitMQ. 50 GB para agentes de tipo de instancias <code>mq.m5.*</code> .

Limitación de API

Las siguientes cuotas de limitaciones controladas se han agregado por cuenta de AWS en todas las API de Amazon MQ para preservar la banda ancha del servicio. Para obtener más información acerca de las API de Amazon MQ, consulte la [Referencia de la API REST de Amazon MQ](#).

Important

Estas cuotas no se aplican a las API de mensajería de agente de Amazon MQ para ActiveMQ o Amazon MQ para RabbitMQ. Por ejemplo, Amazon MQ no limita de forma controlada el envío ni la recepción de mensajes.

Límite de ráfaga de la API	Límite de tasa de la API
100	15

Solución de problemas de Amazon MQ

En esta sección, se describen los problemas comunes que se le podrían plantear cuando usa agentes de Amazon MQ y las medidas que puede adoptar para resolverlos. Para obtener información general sobre la solución de problemas, consulte [the section called “Solución de problemas generales de Amazon MQ”](#). Para resolver los problemas de una versión de motor específica, consulte las siguientes secciones.

Solución de problemas de ActiveMQ en Amazon MQ

Tema sobre solución de problemas	Description (Descripción)
Solución de problemas generales	Utilice la información que se indica en esta sección para diagnosticar y resolver los problemas comunes que se le planteen cuando trabaje con agentes de ActiveMQ en Amazon MQ.
BROKER_ENI_DELETED	ActiveMQ en Amazon MQ emitirá una alarma BROKER_ENI_DELETED al eliminar la interfaz de red elástica (ENI) de un agente.
BROKER_OOM	Amazon MQ en ActiveMQ emitirá una alarma BROKER_OOM cuando el agente se someta a un ciclo de reinicio debido a una capacidad de memoria insuficiente

Solución de problemas de RabbitMQ en Amazon MQ

Tema sobre solución de problemas	Description (Descripción)
Solución de problemas generales	Diagnostique problemas comunes que pueden surgir cuando trabaje con agentes RabbitMQ.

Tema sobre solución de problemas	Description (Descripción)
<u>RABBITMQ_MEMORY_ALARM</u>	RabbitMQ emitirá una alarma de memoria alta cuando el uso de memoria del bróker, identificado mediante una CloudWatch métrica, supere el límite de memoriaRabbitMQMemUsed , identificado por. RabbitMQMemLimit
<u>RABBITMQ_INVALID_KMS_KEY</u>	RabbitMQ en Amazon MQ generará un código INVALID_KMS_KEY que requiere una acción crítica cuando un agente creado con una gestión por el cliente AWS KMS key(CMK) detecte que la clave (KMS) está deshabilitada. AWS Key Management Service
<u>RABBITMQ_INVALID_ASSUME_ROLE</u>	RabbitMQ en Amazon MQ generará un código INVALID_ASSUME_ROLE que requiere una acción crítica cuando Amazon MQ no pueda asumir el ARN de la función de IAM especificado en. aws.arns.assume_role_arn

Tema sobre solución de problemas	Description (Descripción)
<u>RABBITMQ_INVALID_ARN_LDAP</u>	RabbitMQ en Amazon MQ generará un código de acción crítica INVALID_ARN_LDAP cuando la contraseña de la cuenta de servicio LDAP (ARN) no sea válida o no esté accesible.
<u>RABBITMQ_INVALID_ARN_HTTP</u>	RabbitMQ en Amazon MQ generará un código de acción crítica INVALID_ARN_HTTP cuando uno o más de los certificados SSL o el archivo de claves ARNs de HTTP auth_backend no sean válidos o estén inaccesibles.
<u>RABBITMQ_INVALID_ARN_SSL</u>	RabbitMQ en Amazon MQ generará un código de acción crítica INVALID_ARN_SSL cuando uno o más de los almacenes de confianza de certificados de CA para EXTERNAL auth_mechanism no sean válidos o no ARNs estén accesibles.
<u>RABBITMQ_INVALID_ARN</u>	RabbitMQ en Amazon MQ generará un código INVALID_ARN que requiere una acción crítica cuando uno o varios elementos de la configuración del bróker no sean ARNs válidos o estén inaccesibles.

Tema sobre solución de problemas	Description (Descripción)
RABBITMQ_DISK_ALARM	La alarma de límite de disco indica que el volumen del disco utilizado por un nodo de RabbitMQ ha disminuido debido a la gran cantidad de mensajes que no se han consumido mientras se agregaban nuevos mensajes.

Solución de problemas generales de Amazon MQ

Utilice la información que se proporciona en esta sección para diagnosticar los problemas comunes que se le planteen cuando se trabaje con agentes de Amazon MQ, como los problemas de conexión con su agente y reinicios de los agentes.

Contenido

- [No puedo conectarme con la consola web o los puntos de enlace de mi agente.](#)
- [Mi agente se está ejecutando y puedo verificar la conectividad usando telnet, pero mis clientes no se pueden conectar y están devolviendo excepciones SSL.](#)
- [Creé un agente pero me dio error.](#)
- [Mi agente se reinició y no estoy seguro de porqué.](#)

No puedo conectarme con la consola web o los puntos de enlace de mi agente.

Si tiene problemas para conectarse con su agente a través de la consola web o los punto de enlace de nivel de cable, le recomendamos que siga estos pasos.

1. Verifique que no está intentando conectarse con su agente desde detrás de un firewall. Es posible que tenga que configurar el firewall para poder acceder al agente.
2. Compruebe si está intentando conectarse al agente mediante un punto de conexión [FIPS](#). Amazon MQ solo es compatible con los puntos de conexión FIPS al utilizar las operaciones de la API, pero no para las conexiones a nivel de cable con la propia instancia del agente.

3. Verifique si la opción Public accessibility (Accesibilidad pública) está configurada en Yes (Sí) para su agente. Si está configurada en No, verifique las reglas de la [lista de control de acceso \(ACL\)](#) de la red de su subred. Si ha creado una red personalizada ACLs, es posible que deba cambiar las reglas de ACL de la red para proporcionar acceso a su agente. Para obtener más información de las redes de Amazon VPC, consulte el tema sobre la [habilitación del acceso a Internet](#) en la Guía del usuario de Amazon VPC
4. Verifique las reglas del grupo de seguridad de su agente. Asegúrese de permitir las conexiones a los siguientes puertos:

 Note

Los siguientes puertos se agrupan según los tipos de motor porque ActiveMQ en Amazon MQ y RabbitMQ en Amazon MQ para utilizan diferentes puertos para las conexiones.

ActiveMQ en Amazon MQ

- Consola web: puerto 8162
- OpenWire — Puerto 61617
- AMQP: puerto 5671
- STOMP: puerto 61614
- MQTT: puerto 8883
- WSS: puerto 61619

RabbitMQ en Amazon MQ

- Consola web y API de administración: puerto 443 y 15671
- AMQP: puerto 5671

5. Ejecute las siguientes pruebas de conectividad de red para su tipo de motor de agente.

 Note

Para los agentes sin accesibilidad pública, ejecute las pruebas desde una instancia de Amazon EC2 dentro de la misma Amazon VPC que su agente de Amazon MQ y evalúe las respuestas.

ActiveMQ on Amazon MQ

Para probar la conectividad de red de su agente ActiveMQ en Amazon MQ

1. Abra un nuevo terminal o una ventana de líneas de comandos.
2. Ejecute el comando `nslookup` para consultar el registro DNS del agente. Para implementaciones en modo [activo/en espera](#), pruebe los puntos de enlace activos y en espera. Los active/standby puntos finales se identifican con un sufijo `-1` o se `-2` añaden al identificador único del corredor. Reemplace el punto de enlace por la información de su cuenta.

```
$ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Si la consulta se realiza correctamente, se mostrará un resultado similar al que se incluye a continuación.

```
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
```

La dirección IP resuelta debe coincidir con las direcciones IP proporcionadas en la consola de Amazon MQ. Esto indica que el nombre de dominio se está resolviendo correctamente en el servidor DNS y puede pasar al paso siguiente.

3. Ejecute el comando `telnet` para probar la ruta de red para su agente. Reemplace el punto de enlace por la información de su cuenta. `port` Sustitúyalo 8162 por el número de puerto de la consola web u otros puertos a nivel de cable para probar protocolos adicionales según sea necesario.

Note

En el caso de active/standby las implementaciones, recibirá un mensaje de `Connect failed error` si se ejecuta `telnet` con el punto final en espera. Se espera que esto suceda, ya que se está ejecutando la propia instancia en espera,

no el proceso de ActiveMQ, que no tiene acceso al volumen de almacenamiento de Amazon EFS del agente. Ejecute el comando para ambos puertos de enlace, -1 y -2, y asegúrese de probar tanto las instancias activas como las de espera.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com port
```

Para la instancia activa, se mostrará un resultado similar al que se incluye a continuación.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com.  
Escape character is '^['.
```

4. Aplique alguna de las siguientes acciones.

- Si el comando `telnet` se ejecuta correctamente, verifique la métrica [EstablishedConnectionsCount](#) y confirme que el agente no haya alcanzado el [límite de conexiones de nivel de cable](#) máximo. Para confirmar si se ha alcanzado el límite, también puede revisar los registros General del agente. Si esta métrica es mayor que cero, hay al menos un cliente conectado actualmente con el agente. Si la métrica muestra cero conexiones, vuelva a realizar la prueba de la ruta `telnet` y espere al menos un minuto antes de desconectarse, porque las métricas del agente se publican cada un minuto.
- Si el comando `telnet` da error, verifique el estado de la [interfaz de red elástica](#) del agente y confirme que el estado sea `in-use`. [Cree un registro de flujo de Amazon VPC](#) para la interfaz de red de cada instancia y revise los registros de flujo generados. Busque las direcciones IP que utilizó el agente cuando ejecutó el comando `telnet` y confirme que los paquetes de conexión, incluso un paquete de retorno, estén `ACCEPTED`. Para obtener más información y ver un ejemplo de registro de flujo, consulte [Ejemplos de entradas del registro de flujo](#) en la Guía para desarrolladores de Amazon VPC.

5. Ejecute el comando `curl` para verificar la conectividad con la consola web del administrador de ActiveMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com:8162/index.html
```

Si el comando se ejecuta correctamente, se debe obtener un documento HTML similar al que se incluye a continuación.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://
www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
    <title>Apache ActiveMQ</title>
    ...
```

RabbitMQ on Amazon MQ

Para probar la conectividad de red de su agente de RabbitMQ en Amazon MQ

1. Abra un nuevo terminal o una ventana de líneas de comandos.
2. Ejecute el comando `nslookup` para consultar el registro DNS del agente. Reemplace el punto de enlace por la información de su cuenta.

```
$ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Si la consulta se realiza correctamente, se mostrará un resultado similar al que se incluye a continuación.

```
Non-authoritative answer:
Server:  dns-resolver-corp-sfo-1.sfo.corp.amazon.com
Address:  172.10.123.456

Name:     rabbit-broker-1c23e456ca78-b9000123b4ebbab5.elb.us-
west-2.amazonaws.com
Addresses: 52.12.345.678
           52.23.234.56
           41.234.567.890
           54.123.45.678
Aliases:  b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

3. Ejecute el comando `telnet` para probar la ruta de red para su agente. Reemplace el punto de enlace por la información de su cuenta. Puede sustituirlo por *port* un puerto 443 para la consola web y probar la conexión 5671 AMQP a nivel de cable.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com port
```

Si el comando se ejecuta correctamente, se mostrará un resultado similar al que se incluye a continuación.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com.  
Escape character is '^['.
```

Note

La conexión telnet se cerrará automáticamente después de unos segundos.

4. Aplique alguna de las siguientes acciones.
 - Si el comando `telnet` se ejecuta correctamente, verifique la métrica [ConnectionCount](#) y confirme que el agente no haya alcanzado el valor configurado en la política [max-connections](#) predeterminada. Para confirmar si se ha alcanzado el límite, también puede revisar el grupo de registros `Connection.log` del agente. Si esta métrica es mayor que cero, hay al menos un cliente conectado actualmente con el agente. Si la métrica muestra cero conexiones, vuelva a realizar la prueba de la ruta `telnet`. Es posible que tenga que repetir este proceso si la conexión se cierra antes de que su agente haya publicado nuevas métricas de conexión. CloudWatch Las métricas se publican cada un minuto.
 - Para los agentes sin accesibilidad pública, si el comando `telnet` da error, verifique el estado de las [interfaces de red elásticas](#) del agente y confirme que el estado sea `in-use`. [Cree un registro de flujo de Amazon VPC](#) para cada interfaz de red y revise los registros de flujo generados. Busque las direcciones IP privadas que utilizó el agente cuando invocó el comando `telnet` y confirme que los paquetes de conexión, incluso el paquete de retorno, estén `ACCEPTED`. Para obtener más información y ver un ejemplo de registro de flujo, consulte [Ejemplos de entradas del registro de flujo](#) en la Guía para desarrolladores de Amazon VPC.

 Note

Este paso no se aplica a los agentes de RabbitMQ en Amazon MQ con accesibilidad pública.

5. Ejecute el comando `curl` para verificar la conectividad con la consola web del administrador de RabbitMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com:443/index.html
```

Si el comando se ejecuta correctamente, se debe obtener un documento HTML similar al que se incluye a continuación.

```
<!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>
    ...
```

Mi agente se está ejecutando y puedo verificar la conectividad usando **telnet**, pero mis clientes no se pueden conectar y están devolviendo excepciones SSL.

Es posible que su certificado de punto de conexión del agente se haya actualizado durante el [período de mantenimiento](#) del agente. Los certificados de agente de Amazon MQ se rotan periódicamente para garantizar la disponibilidad y la seguridad continuas de los agentes.

Recomendamos utilizar la entidad de certificación (CA) raíz de Amazon en [Amazon Trust Services](#) para autenticar en el almacén de confianza de los clientes. Todos los certificados de agente de Amazon MQ están firmados con esta CA raíz. Al utilizar una CA raíz de Amazon, ya no tendrá que descargar el nuevo certificado de agente de Amazon MQ cada vez que haya una actualización de certificado en el agente.

Creé un agente pero me dio error.

Si el estado del agente está es `CREATION_FAILED`, haga lo siguiente.

- Verifique sus permisos de IAM. Para crear un bróker, debe utilizar la política de IAM AWS gestionada `AmazonMQFullAccess` o tener el conjunto correcto de permisos de Amazon EC2 en su política de IAM personalizada. Para obtener más información sobre los permisos de Amazon EC2 requeridos, consulte [Permisos del IAM requeridos para crear un agente de Amazon MQ](#).
- Verifique si la subred que está eligiendo para su agente está en una Amazon Virtual Private Cloud (VPC) compartida. Para crear un agente de Amazon MQ en una Amazon VPC compartida, debe crearlo en la cuenta propietaria de Amazon VPC.

Mi agente se reinició y no estoy seguro de porqué.

Si su agente se reinició automáticamente, podría deberse a una de las siguientes razones.

- Es posible que su agente se haya reiniciado debido a un período de mantenimiento semanal programado. Amazon MQ realiza de forma periódica el mantenimiento del hardware, el sistema operativo o el software del motor de un agente de mensajes. La duración del mantenimiento varía, pero puede durar hasta dos horas, dependiendo de las operaciones programadas para el agente de mensajes. Los agentes pueden reiniciarse en cualquier momento durante el período de mantenimiento de dos horas. Para obtener más información de los periodos de mantenimiento del agente, consulte [the section called “Programación del mantenimiento del agente”](#).
- Es posible que el tipo de instancias del agente no sea adecuado para la carga de trabajo de la aplicación. Por ejemplo, ejecutar una carga de trabajo de producción en una `mq.t3.micro` podría hacer que el agente se quede sin recursos. Un uso elevado de la CPU o de la memoria del agente pueden provocar que el agente se reinicie imprevistamente. Para ver la cantidad de CPU y memoria que utiliza su agente, utilice las siguientes CloudWatch métricas para su tipo de motor.
 - ActiveMQ en Amazon MQ: verifique `CpuUtilization` para obtener el porcentaje de unidades de computación de Amazon EC2 asignadas que utiliza actualmente el agente. Verifique `HeapUsage` para obtener el porcentaje del límite de memoria JVM de ActiveMQ que el agente usa actualmente.
 - RabbitMQ en Amazon MQ: verifique `SystemCpuUtilization` para obtener el porcentaje de unidades de computación de Amazon EC2 asignadas que utiliza actualmente el agente. Verifique `RabbitMQMemUsed` para obtener el volumen de RAM utilizado en bytes, y divídale por `RabbitMQMemLimit` para obtener el porcentaje de memoria que utiliza el nodo de RabbitMQ.

Para obtener información de los tipos de instancias del agente y cómo elegir el tipo de instancias adecuado para su carga de trabajo, consulte [Broker instance types](#).

Solución de problemas de ActiveMQ en Amazon MQ

Utilice la información que se indica en esta sección para diagnosticar y resolver los problemas comunes que se le planteen cuando trabaje con agentes de ActiveMQ en Amazon MQ.

Contenido

- [No puedo ver los registros generales o de auditoría de mi corredor en CloudWatch los registros aunque haya activado el registro.](#)
- [Después del reinicio del agente o la ventana de mantenimiento, no puedo conectarme a mi agente aunque el estado sea RUNNING ¿Por qué?](#)
- [Veo que algunos de mis clientes se conectan con el agente, mientras que otros no pueden conectarse.](#)
- [Estoy viendo una excepción org.apache.jasper.JasperException: An exception occurred processing JSP page en la consola de ActiveMQ al realizar operaciones.](#)

No puedo ver los registros generales o de auditoría de mi corredor en CloudWatch los registros aunque haya activado el registro.

Si no puede ver los registros de su corredor en CloudWatch Logs, haga lo siguiente.

1. Verifique si el usuario que crea o reinicia el agente tiene el permiso `logs:CreateLogGroup`. Si no agrega el permiso `CreateLogGroup` a un usuario antes de que el usuario cree o reinicie el agente, Amazon MQ no creará el grupo de registros.
2. Compruebe si ha configurado una política basada en recursos que permita a Amazon MQ publicar registros en Logs. CloudWatch Para permitir que Amazon MQ publique registros en su grupo de registros de CloudWatch Logs, configure una política basada en recursos para dar a Amazon MQ acceso a las siguientes acciones de la API de Logs: CloudWatch
 - [CreateLogStream](#)— Crea un flujo de CloudWatch registros para el grupo de registros especificado.
 - [PutLogEvents](#)— Envía los eventos al flujo de registro de CloudWatch registros especificado.

[Para obtener más información sobre la configuración de ActiveMQ en Amazon MQ para publicar registros en Logs, consulte Configuración de registros CloudWatch .](#)

Después del reinicio del agente o la ventana de mantenimiento, no puedo conectarme a mi agente aunque el estado sea **RUNNING** ¿Por qué?

Es posible que se produzcan problemas de conexión tras el reinicio de un agente que inició, después de que se haya completado un período de mantenimiento programado o en un evento de error, en el que se activa la instancia en espera. En cualquier caso, los problemas de conexión tras el reinicio de un agente se deben a que persiste una cantidad inusualmente elevada de mensajes en el volumen de almacenamiento de Amazon EFS o Amazon EBS de su agente. Durante un reinicio, Amazon MQ mueve los mensajes persistentes del almacenamiento a la memoria del agente. Para confirmar este diagnóstico, puedes monitorizar las siguientes métricas CloudWatch para tu bróker de Amazon MQ for ActiveMQ:

- **StoragePercentUsage**: grandes porcentajes al 100 por ciento o cerca del 100 por ciento pueden provocar que el agente rechace las conexiones.
- **JournalFilesForFullRecovery**: indica la cantidad de archivos de diario que volverán a reproducirse después de un cierre y reinicio incorrectos. Un valor en aumento o siempre superior a uno indica transacciones no resueltas que pueden provocar problemas de conexión después del reinicio.
- **OpenTransactionCount**: un número superior a cero tras un reinicio indica que el agente intentará almacenar mensajes consumidos anteriormente, lo que provoca problemas de conexión.

Para resolver este problema, recomendamos resolver las transacciones XA con un `rollback()` o `commit()`. Para obtener más información y ver un ejemplo de código de resolución de transacciones de XA mediante `rollback()`, consulte la información sobre la [recuperación de transacciones de XA](#).

Veo que algunos de mis clientes se conectan con el agente, mientras que otros no pueden conectarse.

Si su agente está en el estado **RUNNING** y algunos clientes pueden conectarse con el agente correctamente, mientras que otros no pueden hacerlo, es posible que haya alcanzado el límite de [conexiones de nivel de cable](#) para el agente. Para comprobar que haya alcanzado el límite de conexiones a nivel de cable, haga lo siguiente:

- Consulta los registros generales de los corredores de ActiveMQ en Amazon MQ en Logs. CloudWatch Si se ha alcanzado el límite, verá Reached Maximum Connections en los registros de agente. Para obtener más información sobre CloudWatch los registros de ActiveMQ en los corredores de Amazon MQ, consulte [the section called “Comprensión de la estructura del inicio de sesión en los registros CloudWatch”](#)

Una vez que se alcance el límite de conexiones a nivel de cable, el agente rechazará activamente las conexiones entrantes adicionales. Para resolver este problema, recomendamos actualizar el tipo de instancias del agente. Si quiere obtener más información para elegir el mejor tipo de instancias para su carga de trabajo, consulte [Broker instance types](#).

Si ha confirmado que la cantidad de conexiones de nivel de cable es inferior al límite de conexión del agente, el problema podría estar relacionado con el reinicio de los clientes. Consulte los registros de su agente para ver si hay entradas numerosas y frecuentes de ... Inactive for longer than 600000 ms - removing La entrada de registro indica el reinicio de clientes o problemas de conectividad. Este efecto es más evidente cuando los clientes se conectan al agente a través de un Network Load Balancer (NLB) con clientes que con frecuencia se desconectan y reconectan al agente. Esto se observa con mayor frecuencia en clientes basados en contenedores.

Verifique los registros del lado del cliente para obtener más información. El agente limpiará las conexiones TCP inactivas después de 600 000 ms y liberará el socket de conexión.

Estoy viendo una excepción **org.apache.jasper.JasperException: An exception occurred processing JSP page** en la consola de ActiveMQ al realizar operaciones.

Si utiliza una autenticación simple y configura AuthorizationPlugin para la autorización de colas y temas, asegúrese de utilizar el elemento AuthorizationEntries en el archivo de configuración XML y otorgar el permiso de grupo activemq-webconsole para todas las colas y temas. Esto garantiza que la consola web de ActiveMQ pueda comunicarse con el agente de ActiveMQ.

El siguiente ejemplo AuthorizationEntry otorga permisos de lectura y escritura para todas las colas y temas del grupo activemq-webconsole.

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

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

Del mismo modo, al integrar su agente con LDAP, asegúrese de otorgar permiso para el grupo `amazonmq-console-admins`. Para obtener más información sobre la integración LDAP, consulte [the section called “Cómo funciona la integración de LDAP”](#).

Solución de problemas de RabbitMQ en Amazon MQ

Utilice la información que se indica en esta sección para diagnosticar y resolver los problemas comunes que se le planteen cuando trabaje con agentes de RabbitMQ en Amazon MQ.

Contenido

- [No puedo ver las métricas de mis colas o hosts virtuales en CloudWatch.](#)
- [¿Cómo habilito los complementos de RabbitMQ en Amazon MQ?](#)
- [No puedo cambiar la configuración de Amazon VPC para el agente.](#)
- [Las implementaciones de clúster han pausado mis sincronizaciones de colas.](#)
- [El agente de una sola instancia de Amazon MQ para RabbitMQ se encuentra en un bucle de reinicio.](#)
- [He perdido el acceso a todas las cuentas de administrador de mi agente.](#)

No puedo ver las métricas de mis colas o hosts virtuales en CloudWatch.

Si no puede ver las métricas de las colas o los hosts virtuales CloudWatch, compruebe si los nombres de las colas o los hosts virtuales contienen espacios en blanco, pestañas u otros caracteres que no sean ASCII.

Amazon MQ no puede publicar métricas de anfitriones virtuales y colas con nombres que contengan espacios en blanco, tabulaciones u otros caracteres que no son ASCII.

Para obtener más información sobre los nombres de las dimensiones, consulta [Dimension](#) en la referencia de las CloudWatch API de Amazon.

¿Cómo habilito los complementos de RabbitMQ en Amazon MQ?

RabbitMQ en Amazon MQ actualmente solo admite el complemento de administración, pala, federación e intercambio de hash consistente de RabbitMQ, que están habilitados de forma

predeterminada. Para obtener más información sobre el uso de complementos admitidos, consulte [the section called “Plugins”](#).

No puedo cambiar la configuración de Amazon VPC para el agente.

Amazon MQ no admite cambiar la configuración de Amazon VPC después de crear su agente. Tenga en cuenta que deberá crear un nuevo agente con la nueva configuración de Amazon VPC y actualizar la URL de conexión del cliente con la nueva URL de conexión del agente.

Las implementaciones de clúster han pausado mis sincronizaciones de colas.

Cuando se solucionan alarmas de memoria elevada de RabbitMQ, es posible que se descubra que no se pueden consumir los mensajes de una o varias colas. Es posible que estas colas estén sincronizando mensajes entre nodos, y durante ese proceso las colas respectivas no están disponibles para publicación y consumo. Las sincronizaciones de colas pueden quedar en pausa debido a la alarma de memoria elevada, e incluso contribuir a que se produzca esa alarma de memoria.

Para obtener información sobre la detención y el reintento de sincronizaciones de colas en pausa, consulte [the section called “Resolver la sincronización de colas en pausa”](#).

El agente de una sola instancia de Amazon MQ para RabbitMQ se encuentra en un bucle de reinicio.

Un agente de instancia única de Amazon MQ para RabbitMQ que genere una alarma de memoria elevada corre el riesgo de dejar de estar disponible si se reinicia y no tiene suficiente memoria para iniciarse. Esto puede provocar que RabbitMQ entre en un bucle de reinicio, y evitar que se puedan realizar más interacciones con el agente hasta que se resuelva el problema. Si el agente se encuentra en un bucle de reinicio, no se podrán aplicar las [prácticas recomendadas](#) de Amazon MQ para resolver la alarma de memoria elevada.

Para recuperar el agente, recomendamos actualizar a un tipo de instancia de mayor tamaño con más memoria. A diferencia de las implementaciones de clúster, se puede actualizar un agente de instancia única mientras experimenta una alarma de memoria elevada, porque no hay sincronizaciones de colas que realizar entre nodos durante un reinicio.

He perdido el acceso a todas las cuentas de administrador de mi agente.

Puede recuperar el acceso mediante la autenticación de IAM. Habilite la federación de identidades web salientes para su AWS cuenta, cree un rol de IAM con permisos para obtener tokens de identidad web, configure su agente para que acepte la autenticación de IAM mediante la OAuth versión 2.0 y, a continuación, utilice las credenciales de IAM para obtener un token de JWT y cree un nuevo usuario administrador. Para obtener instrucciones detalladas, consulte [the section called “Uso de la autenticación y la autorización de IAM”](#).

Alarma de interfaz de red elástica eliminada para ActiveMQ en Amazon MQ

ActiveMQ en Amazon MQ emitirá una alarma `BROKER_ENI_DELETED` al eliminar la interfaz de red elástica (ENI) de un agente. Cuando se [crea un agente de Amazon MQ](#) por primera vez, Amazon MQ aprovisiona una [interfaz de red elástica](#) en la [nube virtual privada \(VPC\)](#) de la cuenta y, por lo tanto, requiere una serie de [permisos de EC2](#).

No se debe modificar ni eliminar esta interfaz de red. Si se modifica o elimina la interfaz de red, se puede provocar una pérdida permanente de la conexión entre la VPC y el agente. Si desea eliminar la interfaz de red, debe eliminar primero el agente.

ActiveMQ en Amazon MQ:: alarma de falta de memoria de un agente

ActiveMQ en Amazon MQ emitirá una alarma `BROKER_OOM` cuando el agente se someta a un ciclo de reinicio debido a una capacidad de memoria insuficiente. Cuando un agente se encuentra en un ciclo de reinicio, también denominado ciclo de rebote, el agente inicia repetidos intentos de recuperación en un breve período de tiempo. Los agentes que no puedan completar el inicio debido a la capacidad de memoria insuficiente pueden ingresar un ciclo de reinicio, durante el cual las interacciones con el agente son limitadas.

Amazon MQ habilita las métricas para el agente de forma predeterminada. Puedes ver las estadísticas de tu bróker accediendo a la CloudWatch consola de Amazon o mediante la CloudWatch API. Las siguientes métricas son útiles para diagnosticar la alarma `BROKER_OOM` de ActiveMQ:

Métrica Amazon MQ CloudWatch	Razón del uso elevado de memoria	
TotalMessageCount	Los mensajes se almacenan en la memoria hasta que se consumen o descartan. Un recuento elevado de mensajes puede indicar una sobreutilización de los recursos, y puede provocar una alarma de memoria elevada.	
HeapUsage	Porcentaje del límite de memoria JVM de ActiveMQ que el agente usa actualmente. Un porcentaje más alto indica que el agente está usando importantes recursos y puede provocar una alarma de OOM.	
ConnectionCount	Las conexiones de cliente utilizan memoria, y demasiadas conexiones simultáneas pueden provocar una alarma de memoria elevada.	
CpuUtilization	El porcentaje de unidades informáticas EC2 asignadas que el agente usa actualmente.	
TotalConsumerCount	Para cada consumidor conectado al agente, se cargan un número determinado de mensajes desde el almacenamiento en la	

Métrica Amazon MQ CloudWatch	Razón del uso elevado de memoria	
	memoria antes de entregarlos al consumidor. Un número elevado de conexiones de consumidores puede suponer un uso elevado de memoria y provocar una alarma de memoria elevada.	

Para evitar los ciclos de reinicio y evitar la alarma `BROKER_OOM`, asegúrese de que los mensajes se consuman rápidamente. Puede hacerlo eligiendo el tipo de instancia de agente más eficaz y también limpiando la [cola de mensajes fallidos](#) para descartar los mensajes caducados o que no se puedan entregar. Puede obtener más información sobre cómo garantizar un rendimiento eficaz en [ActiveMQ en Amazon MQ para las prácticas recomendadas](#).

Amazon MQ para RabbitMQ: alarma de memoria alta

Amazon MQ para RabbitMQ emitirá una alarma de memoria alta cuando el uso de memoria del agente, identificado mediante una CloudWatch métrica, supere el límite de `memoriaRabbitMQMemUsed`, identificado por `RabbitMQMemLimit`.

Un agente de RabbitMQ que haya generado una alarma de memoria elevada bloqueará todos los clientes que estén publicando mensajes. Es posible que el agente entre en un [bucle de reinicio](#), experimente una [pausa en la sincronización de colas](#) o encuentre otros problemas que compliquen el diagnóstico y la resolución de la alarma.

Para diagnosticar y resolver una alarma de memoria insuficiente, primero siga todas las [prácticas recomendadas](#) para RabbitMQ y, a continuación, complete los siguientes pasos.

Important

- Amazon MQ establece `RabbitMQMemLimit` y se ajusta específicamente teniendo en cuenta la memoria disponible para cada tipo de instancia de host.

- Amazon MQ no reiniciará un agente que experimente una alarma de memoria elevada, y devolverá una excepción para operaciones de la API [RebootBroker](#) mientras el agente continúe generando la alarma.

Paso 1: Diagnostique una alarma de memoria elevada

Existen dos formas de diagnosticar las alarmas de memoria elevada en su agente Amazon MQ para RabbitMQ. Le recomendamos que consulte las métricas de la consola web de RabbitMQ y de Amazon MQ. CloudWatch

Diagnosticar una alarma de memoria elevada mediante la consola web de RabbitMQ

La consola web de RabbitMQ puede generar y mostrar información detallada sobre el uso de memoria de cada nodo. Puede encontrar esta información haciendo lo siguiente:

1. Inicie sesión en la consola web Consola de administración de AWS RabbitMQ de su bróker y ábrala.
2. En la consola de RabbitMQ, en la página Overview (Información general), elija el nombre de un nodo de la lista Nodes (Nodos).
3. En la página de detalles del nodo, elija Memory details (Detalles de memoria) para ampliar la sección y ver la información de uso de memoria del nodo.

La información de uso de memoria que proporciona RabbitMQ en la consola web puede ayudar a determinar qué recursos pueden estar consumiendo demasiada memoria y contribuyendo a provocar la alarma de memoria elevada. Para obtener más información los detalles de uso de memoria disponibles a través de la consola web de RabbitMQ, consulte [Razonamiento sobre el uso de memoria](#) en el sitio web de la documentación de RabbitMQ Server.

Diagnosticar una alarma de memoria elevada mediante métricas de Amazon MQ

Amazon MQ habilita las métricas para el agente de forma predeterminada. Puede [ver las métricas de su corredor](#) accediendo a la CloudWatch consola o mediante la API. CloudWatch Las siguientes métricas son útiles para diagnosticar la alarma de memoria elevada de RabbitMQ.

Métrica Amazon MQ CloudWatch	Razón del uso elevado de memoria	
MessageCount	Los mensajes se almacenan en la memoria hasta que se consumen o descartan. Un recuento elevado de mensajes puede indicar una sobreutilización de los recursos, y puede provocar una alarma de memoria elevada.	
QueueCount	Las colas se almacenan en la memoria, y un gran número de colas puede provocar una alarma de memoria elevada.	
ConnectionCount	Las conexiones de cliente utilizan memoria, y demasiadas conexiones simultáneas pueden provocar una alarma de memoria elevada.	
ChannelCount	De forma similar a las conexiones, los canales establecidos con cada conexión también se almacenan en la memoria del nodo, y un número elevado de canales puede provocar una alarma de memoria elevada.	
ConsumerCount	Para cada consumidor conectado al agente, se cargan un número determinado de mensajes desde el almacenamiento en la	

Métrica Amazon MQ CloudWatch	Razón del uso elevado de memoria	
	memoria antes de entregarlos al consumidor. Un número elevado de conexiones de consumidores puede suponer un uso elevado de memoria y provocar una alarma de memoria elevada.	
PublishRate	La publicación de mensajes utiliza la memoria del agente. Si la velocidad a la que se publican los mensajes con destino al agente es demasiado alta y supera significativamente la velocidad con la que el agente envía mensajes a los consumidores, es posible que el agente genere una alarma de memoria elevada.	

Paso 2: Abordar y prevenir una alarma de memoria elevada

Note

El estado de RABBITMQ_MEMORY_ALARM puede tardar varias horas en borrarse después de realizar las acciones necesarias.

Siga todas las [prácticas recomendadas](#) de RabbitMQ como método general de prevención. Para cada colaborador específico que se identifique, recomendamos el siguiente conjunto de acciones con objeto de solucionar y prevenir las alarmas de memoria elevada de RabbitMQ.

Razón del uso de memoria elevada	Recomendación de Amazon MQ para el direccionamiento	Recomendación de Amazon MQ para la prevención
Número de mensajes	Consumir los mensajes publicados en las colas, purgar los mensajes de las colas o eliminar las colas de su agente.	Habilite lazy queues y establezca o reduzca el límite de profundidad de las colas .
Número de colas	Reduzca el número de colas.	Establezca o reduzca el límite de recuento de colas .
Número de conexiones	Reduzca el número de conexiones .	Establezca o reduzca el límite de recuento de conexiones .
Número de canales	Reduzca la cantidad de canales .	Establezca un número máximo de canales por conexión en las aplicaciones cliente.
Número de consumidores	Reduzca el número de consumidores conectados al agente.	Establezca un límite de captura previa de consumidores que sea pequeño.
Frecuencia de publicación de mensajes	Reduzca la velocidad a la que los publicadores envían mensajes al agente.	Active las confirmaciones del publicador .
Tasa de intentos de conexión del cliente	Reduzca la frecuencia con la que los clientes intentan conectarse al agente para publicar o consumir mensajes, o configurar el agente.	Utilice conexiones de mayor duración para reducir el número y la frecuencia de los intentos de conexión.

Una vez que se haya resuelto la alarma de memoria del agente, se puede actualizar el tipo de instancia de host a una instancia con recursos adicionales. Para obtener información sobre cómo

actualizar el tipo de instancia del agente, consulte [UpdateBrokerInput](#) en la Referencia de la API de REST de Amazon MQ.

 Note

No puede cambiar a un agente de un tipo de instancia `mq.m5.x` a un tipo de instancia `mq.t3.micro`. Para cambiar a una versión anterior, debe eliminar su agente y crear uno nuevo.

RabbitMQ en Amazon MQ: clave no válida AWS Key Management Service

RabbitMQ en Amazon MQ generará un código `INVALID_KMS_KEY` que requiere una acción crítica cuando un agente creado con una gestión por el cliente AWS KMS key(CMK) detecte que la clave (KMS) está deshabilitada. AWS Key Management Service Un agente de RabbitMQ con una CMK verifica periódicamente que la clave KMS esté habilitada y que el agente disponga de todas las concesiones necesarias. Si RabbitMQ no puede verificar que la clave está habilitada, el agente se pondrá en cuarentena y RabbitMQ devolverá `INVALID_KMS_KEY`.

Sin una clave de KMS activa, el agente no dispone de permisos básicos para las claves de KMS administradas por el cliente. El agente no podrá realizar operaciones criptográficas mediante su clave hasta que vuelva a activarla y se reinicie el agente. Un agente de RabbitMQ con una clave de KMS desactivada se pone en cuarentena para evitar su deterioro. Después de que RabbitMQ determine que la clave de KMS está activa de nuevo, su agente se retirará de la cuarentena. Amazon MQ no reinicia un agente con una clave de KMS no válida y devuelve una excepción para las operaciones de la API `RebootBroker` mientras el agente siga teniendo una clave de KMS no válida.

Diagnóstico y solución de `INVALID_KMS_KEY`

Para diagnosticar y abordar el código de acción requerida `INVALID_KMS_KEY`, debe usar la interfaz de AWS línea de comandos (CLI) y la consola. AWS Key Management Service

Para volver a activar su clave de KMS

1. Llame al método `kmsKeyId` para recuperar `DescribeBroker` para su agente de CMK.
2. Inicie sesión en la consola. AWS Key Management Service

3. En la página Claves administradas por el cliente, Localice el ID de la clave de KMS del agente problemático y compruebe que el estado es Activado.
4. Si su clave de KMS se ha desactivado, vuelva a activarla mediante la selección de Acciones de clave y, a continuación, elija Activar. Una vez reactivada la clave, deberá esperar a que RabbitMQ retire el agente de la cuarentena.

Para comprobar que las concesiones necesarias siguen asociadas a la clave KMS del bróker, llama al `ListGrant` método para comprobar que `mq_rabbit_grant` `mq_grant` están presentes. Si se ha eliminado la concesión o la clave de KMS, deberá eliminar el agente y crear uno nuevo con todas las concesiones necesarias. Para conocer los pasos para eliminar un agente, consulte [Eliminación de un agente](#).

Para evitar el código requerido de acción crítica `INVALID_KMS_KEY`, no elimine ni desactive manualmente una clave de KMS ni una concesión de CMK. Si desea eliminar la clave, elimine primero el agente.

RabbitMQ en Amazon MQ: alarma de límite de disco

La alarma de límite de disco indica que el volumen del disco utilizado por un nodo de RabbitMQ ha disminuido debido a la gran cantidad de mensajes que no se han consumido mientras se agregaban nuevos mensajes. RabbitMQ emitirá una alarma de límite de disco cuando el espacio libre en disco del bróker, identificado por la CloudWatch métrica de `AmazonRabbitMQDiskFree`, alcance el límite de disco, identificado por `RabbitMQDiskFreeLimit`. `RabbitMQDiskFreeLimit` establece Amazon MQ y se ha definido teniendo en cuenta el espacio en disco disponible para cada tipo de instancia de bróker.

Un agente de Amazon MQ para RabbitMQ que haya generado una alarma de límite de disco no estará disponible para la publicación de mensajes nuevos. Si tiene un publicador y un consumidor en la misma conexión, el consumidor tampoco estará disponible para recibir mensajes. Cuando se ejecuta RabbitMQ en un clúster, la alarma de disco se extiende a todo el clúster. Si un nodo no supera el límite, todos los demás nodos bloquearán los mensajes entrantes. Debido a la falta de espacio en el disco, es posible que el agente experimente otros problemas que compliquen el diagnóstico y la resolución de la alarma.

Amazon MQ no reiniciará un agente que experimente una alarma de disco y devolverá una excepción para operaciones de la API `RebootBroker` mientras el agente continúe generando la alarma.

 Note

No puede cambiar a un agente de un tipo de instancia `mq.m5` a un tipo de instancia `mq.t3.micro`. Si desea cambiar a una versión anterior, debe eliminar su agente y crear uno nuevo.

Diagnóstico y direccionamiento de la alarma de límite de disco

Amazon MQ habilita las métricas para el agente de forma predeterminada. Puedes [ver las estadísticas de tu bróker](#) accediendo a la CloudWatch consola de Amazon o mediante la CloudWatch API. `MessageCountes` una métrica útil para diagnosticar la alarma de límite de disco de RabbitMQ. Los mensajes se almacenan en la memoria hasta que se consumen o descartan. Un recuento elevado de mensajes indica una sobreutilización del almacenamiento en disco y puede provocar una alarma de disco.

Para diagnosticar la alarma de límite de disco, utilice la consola de administración de Amazon MQ para:

- Cree una nueva conexión para consumir mensajes publicados en las colas.
- Purgue mensajes de las colas.
- Elimine las colas del agente.

 Note

El estado de `RABBITMQ_DISK_ALARM` puede tardar varias horas en borrarse después de realizar las acciones necesarias.

Para evitar que se vuelva a generar la alarma de límite de disco, puede actualizar el [tipo de instancia](#) de host a una instancia con recursos adicionales. Para obtener información sobre cómo actualizar el tipo de instancia del agente, consulte `UpdateBrokerInput` en la referencia de la API de REST de Amazon MQ. También recomendamos mantener a los publicadores y a los consumidores conectados de forma diferente.

Amazon MQ para RabbitMQ: alarma de cambio de tipo de instancia

`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` indica que el cambio de tipo de instancia de agente solicitado no puede continuar debido al elevado uso del disco en el nodo RabbitMQ actual. Amazon MQ para RabbitMQ emitirá esta alarma cuando el uso actual del disco supere lo que estaría disponible en el tipo de instancia solicitado, según lo identificado por la métrica. `CloudWatch RabbitMQDiskFree`

Los agentes de RabbitMQ que entren en el estado

`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` seguirán estando disponibles para sus aplicaciones, pero el cambio de tipo de instancia solicitado no se realizará. Amazon MQ permite los reinicios del agente en este estado, pero no puede cambiar el tipo de instancia mientras el uso del disco se mantenga por encima del umbral del tipo de instancia solicitado. El agente devolverá una excepción para las operaciones de API `ModifyBroker` que intenten cambiar el tipo de instancia mientras se encuentre en este estado.

Diagnosticar y abordar la alarma de cambio de tipo de instancia

Amazon MQ habilita las métricas para el agente de forma predeterminada. Puede ver las métricas de su agente accediendo a la CloudWatch consola o mediante la API.

`CloudWatch MessageCounty RabbitMQDiskFree` las métricas se pueden usar para diagnosticar `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`.

Para resolver el estado de cuarentena y permitir que el cambio de tipo de instancia continúe, utilice la consola de administración de Amazon MQ para:

- Cree una nueva conexión para consumir mensajes publicados en las colas.
- Purgue mensajes de las colas.
- Elimine las colas del agente.

Note

El estado de `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` puede tardar varias horas en borrarse después de realizar las acciones necesarias.

RabbitMQ en Amazon MQ: rol de IAM no válido

RabbitMQ en Amazon MQ generará un código `INVALID_ASSUMEROLE` que requiere una acción crítica cuando el ARN del rol de IAM especificado no sea válido o Amazon MQ no pueda asumirlo. `aws.arns.assume_role_arn` Esto puede ocurrir cuando el rol no existe, se encuentra en una AWS cuenta diferente a la del intermediario o no tiene la relación de confianza necesaria con `mq.amazonaws.com`.

Un agente en cuarentena con `RABBITMQ_INVALID_ASSUMEROLE` no puede recuperar las credenciales o los certificados necesarios para la autenticación LDAP, lo que hace que la autenticación LDAP no esté disponible. Si LDAP es el único método de autenticación configurado, los usuarios no podrán conectarse al agente. Amazon MQ requiere la función de IAM para AWS acceder a los recursos a los que se hace referencia ARNs en la configuración del bróker, AWS Secrets Manager como los secretos o los objetos de Amazon S3 que se utilizan para la autenticación LDAP.

Diagnóstico y tratamiento de `RABBITMQ_INVALID_ASSUMEROLE`

Para diagnosticar y abordar el código de acción requerida por `RABBITMQ_INVALID_ASSUMEROLE`, debe usar Amazon Logs y la consola. CloudWatch AWS Identity and Access Management

Para resolver el problema de asumir el rol no válido

1. Navegue hasta Amazon CloudWatch Logs Insights y ejecute la siguiente consulta en el grupo de registros de su agente/`aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Busque mensajes de error similares a los siguientes:

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

3. Compruebe la configuración de las funciones de IAM y solucione cualquier problema, como:
 - Asegúrese de que el rol esté en la misma AWS cuenta que el corredor
 - Compruebe que la política de confianza permita a mq.amazonaws.com asumir el rol
 - Confirme que el rol tiene los permisos adecuados para acceder a los recursos necesarios AWS
4. Valide la solución mediante el punto final de la API de [validación de acceso del ARN](#) antes de actualizar la configuración del broker.
5. Actualice la configuración del corredor y reinícielo.

RabbitMQ en Amazon MQ: ARN de LDAP no válido

RabbitMQ en Amazon MQ generará un código de acción crítica `INVALID_ARN_LDAP` cuando el ARN configurado para la contraseña de la cuenta de servicio LDAP no sea válido o sea inaccesible. Esto se aplica a los datos especificados en o, que deben hacer referencia a los secretos que contienen contraseñas de texto simple. ARNs `aws.arns.auth_ldap.dn_lookup_bind.password`
`aws.arns.auth_ldap.other_bind.password` AWS Secrets Manager

Un agente en cuarentena en `RABBITMQ_INVALID_ARN_LDAP` no puede autenticarse con la cuenta de servicio LDAP, lo que hace que la autenticación LDAP no esté disponible. Si LDAP es el único método de autenticación configurado, los usuarios no podrán conectarse al agente. La invalidez ARNs puede deberse a un formato incorrecto de la sintaxis del ARN, a referencias a secretos inexistentes, a secretos ubicados en una región AWS diferente a la del agente o a permisos insuficientes de `secretsmanager:GetSecretValue` : en la función de IAM.

Diagnóstico y tratamiento de `RABBITMQ_INVALID_ARN_LDAP`

Para diagnosticar y abordar el código de acción requerida por `RABBITMQ_INVALID_ARN_LDAP`, debe usar Amazon Logs y la consola. CloudWatch

Para resolver el problema del ARN de LDAP no válido

1. Navegue hasta Amazon CloudWatch Logs Insights y ejecute la siguiente consulta en el grupo de registros de su agente/`aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
```

```
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Busque mensajes de error similares a los siguientes:

```
[error] <0.254.0> aws_arn_config: {<<"could not resolve
ARN 'arn:aws:secretsmanager:xxx' for configuration
'aws.arns.auth_ldap.dn_lookup_bind.password', error: \"AWS service is unavailable
\">>,{error,\"AWS service is unavailable\"}}
```

3. Comprueba el secreto de Secrets Manager y soluciona cualquier problema como:
 - Compruebe que el secreto existe en la misma AWS región que el corredor
 - Confirme que la sintaxis del ARN es correcta
 - Asegúrese de que el rol de IAM tenga permisos `secretsmanager: GetSecretValue`
4. Valide la solución mediante el punto final de la API de [validación de acceso del ARN](#) antes de actualizar la configuración del broker.
5. Actualice la configuración del corredor y reinícielo.

RabbitMQ en Amazon MQ: ARN HTTP no válido

RabbitMQ en Amazon MQ generará un código de acción crítica `INVALID_ARN_HTTP` cuando uno o más de los certificados SSL o el archivo de claves ARNs de `HTTP auth_backend` no sean válidos o estén inaccesibles. Esto se aplica a los ARNs especificados en `aws.arns.auth_http.ssl_options.certfile` o `aws.arns.auth_http.ssl_options.cacertfile` `aws.arns.auth_http.ssl_options.keyfile`, que deben hacer referencia a los objetos y AWS Secrets Manager secretos de Amazon S3 que contienen certificados y claves privadas.

Un agente que se encuentre en cuarentena con `RABBITMQ_INVALID_ARN_HTTP` no puede autenticarse a través del servidor HTTP. Si HTTP es el único método de autenticación configurado, los usuarios no podrán conectarse al agente. La invalidez ARNs puede deberse a un formato incorrecto de la sintaxis del ARN, a referencias a secretos inexistentes, a secretos ubicados en una región AWS diferente a la del agente o a permisos de `s3GetObject` / `secretsmanagerGetSecretValue` : insuficientes en la función de IAM.

Diagnóstico y tratamiento de RABBITMQ_INVALID_ARN_HTTP

Para diagnosticar y abordar el código de acción requerida por RABBITMQ_INVALID_ARN_HTTP, debe usar Amazon Logs y la consola. CloudWatch

Para resolver el problema del ARN HTTP no válido

1. Navegue hasta Amazon CloudWatch Logs Insights y ejecute la siguiente consulta en el grupo de registros de su agente/aws/amazonmq/broker/<broker-id>/general:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Busque mensajes de error similares a los siguientes:

```
[error] <0.209.0> aws_arn_config: {<<"could not resolve ARN 'arn:aws:s3:::xxxx' for
configuration 'aws.arns.auth_http.ssl_options.certfile', error: \"AWS service is
unavailable\">>,{error,\"AWS service is unavailable\"}}
```

3. Compruebe el secreto del S3 Object/Secrets Manager y solucione cualquier problema, como:
 - Compruebe que el recurso existe en la misma AWS región que el bróker
 - Confirme que la sintaxis del ARN es correcta
 - Asegúrese de que el rol de IAM tenga los permisos s3: GetObject y secretsmanager: GetSecretValue
4. Valide la solución mediante el punto final de la API de [validación de acceso del ARN](#) antes de actualizar la configuración del broker.
5. Actualice la configuración del corredor y reinícielo.

RabbitMQ en Amazon MQ: ARN SSL no válido

RabbitMQ en Amazon MQ generará un código de acción crítica INVALID_ARN_SSL cuando uno o más de los almacenes de confianza de certificados de CA para EXTERNAL auth_mechanism

no sean válidos o no ARNs estén accesibles. Esto se aplica a los ARNs especificados en `aws.arns.ssl_options.cacertfile` o `aws.arns.management.ssl.cacertfile`, que deben hacer referencia al objeto PCA de Amazon S3 o ACM que contiene el certificado.

Un bróker que se encuentre en cuarentena con `RABBITMQ_INVALID_ARN_SSL` no puede autenticar los certificados de los clientes durante los intercambios mutuos de TLS porque no hay configurado un almacén de confianza válido. Si el único método de autenticación configurado es un mecanismo de autenticación externo, los usuarios no podrán conectarse al intermediario. La invalidez ARNs puede deberse a un formato incorrecto de la sintaxis del ARN, a referencias a objetos S3 inexistentes, a objetos S3 ubicados en una región AWS diferente a la del intermediario o a permisos `s3GetObject: GetCertificateAuthorityCertificate /acm-pca:` insuficientes en la función de IAM.

Diagnóstico y tratamiento de `RABBITMQ_INVALID_ARN_SSL`

Para diagnosticar y abordar el código de acción requerida por `RABBITMQ_INVALID_ARN_SSL`, debe usar Amazon Logs y la consola. CloudWatch

Para resolver el problema del ARN de SSL no válido

1. Navegue hasta Amazon CloudWatch Logs Insights y ejecute la siguiente consulta en el grupo de registros de su agente/`aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Busque mensajes de error similares a los siguientes:

```
[error] <0.209.0> aws_arn_config: {<<"could not resolve ARN 'arn:aws:acm-pca:xxxx'
for configuration 'aws.arns.ssl_options.cacertfile', error: \"AWS service is
unavailable\">>,{error,"AWS service is unavailable"}}
```

3. Compruebe el objeto S3/ACM-PCA y solucione cualquier problema, como:
 - Compruebe que el secreto existe en la misma región que el bróker AWS
 - Confirme que la sintaxis del ARN es correcta

- Asegúrese de que el rol de IAM tenga los permisos s3: GetObject /acm-pca: GetCertificateAuthorityCertificate
4. Valide la solución mediante el punto final de la API de [validación de acceso del ARN](#) antes de actualizar la configuración del broker.
 5. Actualice la configuración del corredor y reinícielo.

RabbitMQ en Amazon MQ: ARN no válido

RabbitMQ en Amazon MQ generará un código INVALID_ARN que requiere una acción crítica cuando una o más ARNs configuraciones en el corredor no sean válidas o no estén accesibles. Esto se aplica a los ARNs certificados SSL, AWS Secrets Manager secretos, objetos de Amazon S3 u otras referencias a AWS recursos que no estén cubiertas por códigos de cuarentena más específicos, como RABBITMQ_INVALID_ARN_LDAP o RABBITMQ_INVALID_ASSUME_ROLE.

Un corredor que se encuentre en cuarentena con ARNs RABBITMQ_INVALID_ARN puede ver degradadas sus funciones en función de cuáles no sean válidas. Las funciones que dependen de los recursos inaccesibles no estarán disponibles y el corredor registrará los errores que indiquen qué ARN no se pudo resolver. El impacto en la disponibilidad del corredor depende de si se requiere el ARN no válido para las operaciones críticas del corredor.

Diagnóstico y tratamiento de RABBITMQ_INVALID_ARN

Para diagnosticar y abordar el código de acción requerida por RABBITMQ_INVALID_ARN, debe usar Amazon CloudWatch Logs y la consola de servicio adecuada para el recurso afectado. AWS

Para resolver el problema de un ARN no válido

1. Navegue hasta Amazon CloudWatch Logs Insights y ejecute la siguiente consulta en el grupo de registros de su agente/aws/amazonmq/broker/<broker-id>/general:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Busque mensajes de error similares a los siguientes:


```
[error] <0.254.0> aws_arn_config: {<<"could not resolve ARN  
'arn:aws:s3:::bucket-name/certificate.pem' for configuration  
'aws.arns.auth_ldap.ssl_options.cacertfile', error: \"AWS service is unavailable  
\">>,{error,\"AWS service is unavailable\"}}
```

3. Compruebe el AWS recurso y solucione cualquier problema, como:
 - Compruebe que el recurso existe en la misma AWS región que el corredor
 - Confirme que la sintaxis del ARN es correcta
 - Asegúrese de que el rol de IAM tenga los permisos adecuados para acceder al recurso
4. Valide la solución mediante el punto final de la API de [validación de acceso del ARN](#) antes de actualizar la configuración del broker.
5. Actualice la configuración del corredor y reinícielo.

Recursos relacionados

Recursos de Amazon MQ

En la tabla siguiente, se muestran los recursos útiles para trabajar con Amazon MQ.

Recurso	Descripción
Referencia de la API de REST de Amazon MQ	Descripciones de recursos de REST, solicitudes de ejemplo, métodos HTTP, esquemas, parámetros y los errores devueltos por el servicio.
Amazon MQ en la Referencia de los comandos de la AWS CLI	Descripciones de los comandos de la AWS CLI que puede usar para trabajar con los agentes de mensajes.
Amazon MQ en la Guía del usuario de AWS CloudFormation	El recurso AWS::Amazon MQ::Broker le permite crear agentes de Amazon MQ, agregar cambios de configuración o modificar los usuarios del agente especificado, devolver información sobre el agente especificado y eliminar el agente especificado. El recurso AWS::Amazon MQ::Configuration le permite crear configuraciones de Amazon MQ, agregar cambios de configuración o modificar usuarios, y devolver información sobre la configuración especificada.
Regiones y puntos de conexión de	Información acerca de las regiones y los puntos de enlace de Amazon MQ
Página del producto	Página web principal con información acerca de Amazon MQ.

Recurso	Descripción
Foro de debate	Foro de la comunidad donde los desarrolladores tratan aspectos técnicos relacionados con Amazon MQ.
AWS Información sobre Premium Support	Página web principal para obtener información acerca de AWS Premium Support, un canal de soporte individualizado y de respuesta rápida que lo ayudará a crear y ejecutar aplicaciones en servicios de infraestructura de AWS

Recursos de Amazon MQ para ActiveMQ

En la tabla siguiente se listan recursos útiles para trabajar con Apache ActiveMQ.

Recurso	Descripción
Guía de introducción a Apache ActiveMQ	La documentación oficial de Apache ActiveMQ.
ActiveMQ en acción	Una guía de Apache ActiveMQ que abarca la anatomía de los mensajes JMS, conectores, persistencia de mensajes, autenticación y autorización.
Clientes multilinguaje	Lista de lenguajes de programación y de las correspondientes bibliotecas de Apache ActiveMQ. Consulte también ActiveMQ Client y QpidJMS Client .

Recursos de Amazon MQ para RabbitMQ

En la siguiente tabla, se listan recursos útiles para trabajar con RabbitMQ.

Recurso	Descripción
Guía de introducción a RabbitMQ	La documentación oficial de RabbitMQ.
Bibliotecas de clientes y herramientas para desarrolladores de RabbitMQ	Una guía para las bibliotecas de cliente y las herramientas para desarrolladores oficialmente compatibles para trabajar con RabbitMQ utilizando diversos lenguajes y plataformas de programación.
Prácticas recomendadas de RabbitMQ	Guía de CloudAMQP sobre recomendaciones y prácticas recomendadas para trabajar con RabbitMQ.

Notas de la versión de Amazon MQ

En la siguiente tabla, se muestran las versiones y mejoras de las características de Amazon MQ.

Date	Actualización de documentación
19 de febrero de 2026	Amazon MQ ahora es compatible con ActiveMQ 5.19, una nueva versión secundaria del motor. Para obtener más información, consulte • Página de lanzamiento de ActiveMQ 5.19 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Actualización de una versión del motor del agente de Amazon MQ • Uso de archivos de configuración XML Spring
22 de enero de 2026	Amazon MQ ahora admite el complemento de intercambio de temas JMS para corredores de RabbitMQ 4.2 y versiones posteriores. Puede utilizar el cliente JMS oficial de RabbitMQ para ejecutar cargas de trabajo de JMS en Amazon MQ para el bróker RabbitMQ. Es compatible con JMS 1.1, 2.0 y 3.1. Para obtener más información, consulte • Especificaciones oficiales del JMS 2.0 (compatible con versiones anteriores y versiones ampliadas del JMS 1.1) • Especificaciones oficiales del JMS 3.1 • Limitación del cliente RabbitMQ JMS • Conexión de su aplicación JMS a Amazon MQ para el bróker RabbitMQ
8 de enero de 2026	Amazon MQ ahora admite la autenticación con certificados SSL para los corredores de RabbitMQ 4.2 y versiones posteriores mediante certificados de cliente X.509 y una configuración de TLS mutua (mTLS). Puede configurar la autenticación con certificados SSL y los mTLS a través de Consola de administración de AWS, AWS CloudFormation AWS CLI, o AWS CDK en todos los Regiones de AWS lugares donde Amazon MQ esté disponible.

Date	Actualización de documentación
	Para obtener más información, consulte Autenticación con certificados SSL y Configuración de mTLS.
6 de enero de 2026	Amazon MQ ahora admite la autenticación y autorización HTTP para los corredores de RabbitMQ 4.2 y versiones posteriores con servidores HTTP externos. Puede configurar la autenticación HTTP mediante Consola de administración de AWS, AWS CloudFormation AWS CLI, o AWS CDK en todos los Regiones de AWS lugares donde Amazon MQ esté disponible. Para obtener más información, consulte Autenticación y autorización HTTP.
20 de noviembre de 2025	Amazon MQ ahora es compatible con RabbitMQ 4.2, una nueva versión principal que incluye soporte nativo para el protocolo AMQP 1.0, un nuevo almacén de metadatos basado en Raft, Khepri, palas locales y prioridad es de mensajes para las colas de quórum. RabbitMQ 4.2 también incluye varias correcciones de errores y mejoras de rendimiento para la gestión del rendimiento y la memoria. Si bien esta versión presenta nuevas funciones, hay algunos cambios importantes. Para obtener más información, consulte • RabbitMQ 4 • Notas de lanzamiento de RabbitMQ de código abierto • Configuración de los límites de recursos • Protocolos compatibles • Actualizaciones de la versión de Amazon MQ
18 de noviembre de 2024	Amazon MQ ahora admite instancias m7g con tecnología Graviton3 para RabbitMQ en una variedad de tamaños, desde medianos hasta 16 veces grandes en África (Ciudad del Cabo). Para obtener más información, consulte Tipos de instancias del agente de Amazon MQ para RabbitMQ.

Date	Actualización de documentación
17 de noviembre de 2025	Amazon MQ ahora admite la autenticación y autorización de LDAP para los corredores de RabbitMQ con servicios de directorio LDAP externos. Puede configurar LDAP a través de Consola de administración de AWS, AWS CloudFormation AWS CLI, o AWS CDK en todos los Regiones de AWS lugares donde Amazon MQ esté disponible. Para obtener más información, consulte Autenticación y autorización LDAP para Amazon MQ para RabbitMQ.
22 de octubre de 2025	Amazon MQ ya está disponible en la región Asia-Pacífico (Nueva Zelanda). Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .
3 de septiembre de 2025	Amazon MQ ahora admite la autenticación y la autorización OAuth 2.0 para los corredores de RabbitMQ con proveedores de identidad públicos (). IdPs Puede configurar desde OAuth 2.0 hasta Consola de administración de AWS AWS CloudFormation AWS CLI, o AWS CDK en todos los Regiones de AWS lugares donde Amazon MQ esté disponible. Para obtener más información, consulte OAuth Autenticación y autorización 2.0 para Amazon MQ para RabbitMQ.
22 de julio de 2025	Amazon MQ ahora admite instancias de m7g con tecnología Graviton3 para RabbitMQ en una variedad de tamaños, desde medianos hasta 16xlarge. Los clústeres de RabbitMQ que se ejecutan en instancias de m7g ofrecen hasta un 50 % más de capacidad de carga de trabajo y hasta un 85 % de mejoras de rendimiento en comparación con los clústeres comparables de Amazon MQ para RabbitMQ que se ejecutan en instancias de m5. Las instancias de M7g también tienen tamaños de volumen de disco optimizados que varían según el tamaño de la instancia. Para obtener más información, consulte Broker instance types. Las instancias de M7g de Amazon MQ ya están disponibles en todas las regiones de Amazon MQ, excepto en las de África (Ciudad del Cabo), Oeste de Canadá (Calgary) y Europa (Milán).

Date	Actualización de documentación
8 de julio de 2025	Amazon MQ ya está disponible en la región de Asia-Pacífico (Taipéi). Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .
22 de abril de 2025	Ahora puede eliminar las configuraciones de los agentes de Amazon MQ mediante la API <code>DeleteConfiguration</code> . Para obtener más información, consulte Configuraciones en la Referencia de la API de Amazon MQ.
16 de abril de 2025	Amazon MQ para RabbitMQ ahora admite el uso de puntos de enlace de doble pila (IPv4 y IPv6) para conectarse con agentes públicos y privados. Para obtener más información, consulte Connecting to Amazon MQ y Configuring a private Amazon MQ broker.
7 de abril de 2025	Amazon MQ ya se encuentra disponible en las regiones Asia-Pacífico (Tailandia) y México (centro). Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .
13 de febrero de 2025	Los puntos de conexión FIPS de la API de Amazon MQ ya están disponibles en las regiones de Canadá (Centro) y de Oeste de Canadá (Calgary). Para obtener más información sobre el uso de puntos de conexión FIPS con la API de Amazon MQ, consulte Connecting to Amazon MQ. Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .

Date	Actualización de documentación
12 de febrero de 2025	Amazon MQ anuncia las siguientes fechas de finalización del soporte de los tipos de instancias: Broker instance types • <code>mq.t2.micro</code> ActiveMQ: 12 de mayo de 2025 • <code>mq.m4.large</code> ActiveMQ: 12 de mayo de 2025 No puede crear agentes en <code>mq.t2.micro</code> o <code>mq.m4.large</code> a partir del 17 de marzo de 2025.
10 de diciembre de 2024	Amazon MQ ahora admite la conexión entre sus nubes privadas virtuales (VPCs) y la API de Amazon MQ sin exponer su tráfico a la Internet pública. AWS PrivateLink Para obtener más información, consulte the section called “Conexión a Amazon MQ mediante PrivateLink de AWS”.
18 de noviembre de 2024	Amazon MQ está ya disponible en la región de Asia-Pacífico (Malasia). Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .
14 de noviembre de 2024	Amazon MQ anuncia las siguientes fechas de finalización del soporte de las versiones de motor: Administración de versiones del motor de Amazon MQ para ActiveMQ • ActiveMQ 5.17: 16 de junio de 2025 Administración de versiones del motor de Amazon MQ para RabbitMQ • RabbitMQ 3.11: 17 de febrero de 2025 • RabbitMQ 3.12: 17 de marzo de 2025 Para obtener más información sobre cómo actualizar a la última versión, consulte Actualización de una versión del motor del agente de Amazon MQ.

Date	Actualización de documentación
13 de noviembre de 2024	Amazon MQ ahora admite puntos de enlace de servicio de doble pila a los que puede conectarse mediante una u otra. IPv4 IPv6 Los puntos de conexión del servicio regional de doble pila de Amazon MQ se pueden resolver con los registros DNS A y AAAA. Para obtener más información, consulte ??? .
25 de julio de 2024	Amazon MQ ahora es compatible con ActiveMQ 5.18, un nuevo lanzamiento de versión secundaria del motor. Para obtener más información, consulte los siguientes temas: • Página de la versión de ActiveMQ 5.18 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Actualización de una versión del motor del agente de Amazon MQ • Uso de archivos de configuración XML Spring
22 de julio de 2024	Ahora, Amazon MQ admite colas de cuórum solo en agentes que utilicen la versión 3.13 o superior. Las colas de cuórum son un tipo de cola FIFO reflejada que utiliza el algoritmo de consenso Raft para mantener la coherencia de datos. Las colas de cuórum gestionan los mensajes dudosos, lo que puede ayudarle a administrar los mensajes sin procesar. Para empezar a trabajar rápidamente con colas de cuórum, consulte Colas de cuórum para RabbitMQ en Amazon MQ.

Date	Actualización de documentación
2 de julio de 2024	Amazon MQ para RabbitMQ ahora es compatible RabbitMQ versión 3.13, una versión secundaria. En el caso de los agentes que utilizan la versión 3.13 o superior del motor, Amazon MQ administra las actualizaciones a la última versión de parche compatible durante el período de mantenimiento. Para obtener más información, consulte Actualización de una versión del motor del agente de Amazon MQ. Directrices de dimensionamiento de Amazon MQ para RabbitMQ se ha actualizado para incluir nuevos límites de colas, consumidores por canal y palas para los agentes que utilizan la versión 3.13 del motor. Para obtener más información sobre las correcciones y funciones de esta versión, consulte las notas de la versión 3.13 de RabbitMQ en el repositorio del servidor de RabbitMQ. GitHub Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
10 de junio de 2024	Amazon MQ ya está disponible en la región de Oeste de Canadá (Calgary). Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .

Date	Actualización de documentación
10 de mayo de 2024	El calendario de soporte de versiones de Amazon MQ indica cuándo finaliza el soporte de una versión del motor del agente. Cuando finaliza el soporte de una versión de motor, Amazon MQ actualiza automáticamente todos los agentes de esta versión secundaria a la siguiente compatible. Amazon MQ avisa con al menos 90 días de antelación antes de que finalice el soporte de una versión de motor. Para consultar el calendario de soporte de versiones y el fin del soporte, consulte lo siguiente: • Administración de versiones del motor de Amazon MQ para ActiveMQ • Administración de versiones del motor de Amazon MQ para RabbitMQ También puede activar las actualizaciones automáticas de las versiones secundarias para que su agente se actualice a la siguiente versión del parche durante un período de mantenimiento. Para obtener más información, consulte Actualización de una versión del motor del agente de Amazon MQ

Date	Actualización de documentación
9 de mayo de 2024	Amazon MQ para RabbitMQ ahora es compatible RabbitMQ versión 3.12, una versión secundaria. Todos los corredores del 3.12.13 y versiones posteriores utilizan Classic Queues versión 2 (CQv2), y todas las colas del 3.12.13 y versiones posteriores se comportan como colas perezosas. Recomendamos a los corredores que utilicen versiones anteriores a la 3.12.13 que activen las colas CQv2 y perezosas, o que actualicen a la versión más reciente de Amazon MQ para RabbitMQ. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.12 de RabbitMQ en el repositorio del servidor RabbitMQ. GitHub Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
4 de marzo de 2024	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ 3.11.28. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.11.28 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.

Date	Actualización de documentación
19 de enero de 2024	Amazon MQ para RabbitMQ no admite el nombre de usuario “guest” y eliminará la cuenta de invitado predeterminada cuando cree un nuevo agente. Amazon MQ también eliminará periódicamente cualquier cuenta creada por el cliente con dicho nombre.
15 de diciembre de 2023	Amazon MQ ya está disponible en la región de Israel (Tel Aviv). Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .
11 de diciembre de 2023	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ 3.10.25. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.10.25 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
26 de octubre de 2023	Amazon MQ ha publicado las versiones secundarias más recientes de ActiveMQ 5.15.16, 5.16.7 y 5.17.6 con una actualización crítica. Hemos dejado de utilizar las versiones secundarias más antiguas de ActiveMQ y actualizaremos todos los agentes a cualquier versión de 5.15 a 5.15.16 o de 5.16 a 5.16.7 y de 5.17 a 5.17.6. Para obtener más información sobre la actualización del agente de ActiveMQ, consulte Administración de versiones del motor de Amazon MQ para ActiveMQ.

Date	Actualización de documentación
27 de septiembre de 2023	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ 3.11.20. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.11.20 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
27 de julio de 2023	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ 3.11.16. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.11.16 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
27 de julio de 2023	Amazon MQ para RabbitMQ ahora permite crear y aplicar configuraciones al agente de RabbitMQ. Para obtener más información sobre cómo agregar configuraciones al agente, consulte RabbitMQ Broker Configurations. Para obtener más información acerca de esta característica, consulte: • Políticas del operador • Cambios en las políticas del operador

Date	Actualización de documentación
23 de junio de 2023	Amazon MQ ahora es compatible con ActiveMQ 5.17.3, un nuevo lanzamiento de versión secundaria del motor. Esta versión es compatible con la nueva característica de replicación de datos entre regiones (CRDR) de Amazon MQ. Para obtener más información, consulte los siguientes temas: • Para empezar a utilizar CRDR, consulte Replicación de datos entre regiones para Amazon MQ para ActiveMQ en la Guía del desarrollador. • Página de la versión de ActiveMQ 5.17.3 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Actualización de una versión del motor del agente de Amazon MQ • Uso de archivos de configuración XML Spring
21 de junio de 2023	Amazon MQ para ActiveMQ ahora ofrece una función de replicación de datos entre regiones (CRDR) que permite la replicación asíncrona de mensajes desde el agente principal de una región principal al agente de réplicas de una región de réplicas. AWS Si el agente principal de la región principal falla, puede ascender a primario al agente de réplicas de la región secundaria iniciando una conmutación o conmutación por error. Para empezar a utilizar CRDR, consulte Replicación de datos entre regiones para Amazon MQ para ActiveMQ en la Guía del desarrollador.
18 de mayo de 2023	Amazon MQ ya está disponible en las siguientes regiones: • Asia-Pacífico (Melbourne) • Asia-Pacífico (Hyderabad) • Europa (España) • Europa (Zúrich) Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .

Date	Actualización de documentación
14 de abril de 2023	Amazon MQ para RabbitMQ ahora es compatible con la versión 3.9.27 de RabbitMQ. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.9.27 de RabbitMQ sobre el repositorio de servidores de RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
14 de abril de 2023	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ versión 3.10.20. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.10.20 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.

Date	Actualización de documentación
31 de marzo de 2023	Amazon MQ para RabbitMQ ha desactivado el motor de RabbitMQ versión 3.10.17. El equipo de Amazon MQ para RabbitMQ y los mantenedores de código abierto de RabbitMQ han identificado un problema con la consola de administración de RabbitMQ en la versión 3.10.17. Amazon MQ ha retirado esta versión. Para mitigar los impactos de este problema, cree nuevos agentes con la versión 3.10.10 mientras trabajamos para admitir una nueva versión de revisión de RabbitMQ. Recomendamos activar la opción de actualización de la versión para obtener automáticamente las últimas correcciones de errores, actualizaciones de seguridad y mejoras de rendimiento. Para obtener más información sobre las versiones disponibles de Amazon MQ para RabbitMQ, consulte Versiones del motor de Amazon MQ para RabbitMQ.
1 de marzo de 2023	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ versión 3.10.17. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.10.17 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.

Date	Actualización de documentación
21 de febrero de 2023	Amazon MQ para RabbitMQ ahora se integra con AWS Key Management Service (KMS) para ofrecer cifrado del lado del servidor. Ahora puede seleccionar su propia CMK gestionada por el cliente o utilizar una AWS clave de KMS gestionada en su cuenta. AWS KMS Para obtener más información, consulte Cifrado en reposo. Amazon MQ admite el uso de AWS KMS claves de las siguientes maneras. • Amazon MQ owned KMS key (default) (Clave KMS de Amazon MQ [predeterminada]): la clave es propiedad de y está administrada por Amazon MQ y no está en su cuenta. • AWS clave de KMS AWS gestionada: la clave de KMS gestionada (aws/mq) es una clave de KMS de su cuenta que Amazon MQ crea, gestiona y utiliza en su nombre. • Select existing customer managed CMK (Seleccionar clave de KMS existente administrada por el cliente): usted crea y administra en AWS Key Management Service (KMS) claves de KMS administradas por el cliente.
13 de enero de 2023	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ versión 3.8.34. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.8.34 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.

Date	Actualización de documentación
15 de diciembre de 2022	Amazon MQ para RabbitMQ ahora es compatible con la versión 3.9.24 de RabbitMQ. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.9.24 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
13 de diciembre de 2022	Amazon MQ está disponible ahora en la región de Medio Oriente (EAU). Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .
14 de noviembre de 2022	Amazon MQ para RabbitMQ ahora admite 3.10, una versión principal de motor. Ahora puede habilitar la versión 2 () de Classic Queues en sus colas de RabbitMQ. CQv2 No se admiten las actualizaciones directas de 3.8 a 3.10. Para obtener más información, consulte los siguientes temas: • Notas de lanzamiento de RabbitMQ 3.10.10 • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.

Date	Actualización de documentación
9 de noviembre de 2022	Amazon MQ ahora es compatible con ActiveMQ 5.17.2, una nueva versión secundaria de motor. Para obtener más información, consulte los siguientes temas: • Página de la versión de ActiveMQ 5.17.2 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Actualización de una versión del motor del agente de Amazon MQ • Uso de archivos de configuración XML Spring
17 de agosto de 2022	Amazon MQ ahora es compatible con ActiveMQ 5.17.1, una nueva versión de motor principal. Para obtener más información, consulte los siguientes temas: • Página de la versión de ActiveMQ 5.17.1 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Actualización de una versión del motor del agente de Amazon MQ • Uso de archivos de configuración XML Spring
14 de julio de 2022	Amazon MQ ahora es compatible con ActiveMQ 5.16.5, una versión secundaria de motor. Para obtener más información, consulte los siguientes temas: • Página de la versión de ActiveMQ 5.16.5 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Uso de archivos de configuración XML Spring • Actualización de una versión del motor del agente de Amazon MQ
4 de mayo de 2022	Amazon MQ agrega un lenguaje inclusivo para el elemento <code>networkConnector</code> en la configuración de agente. • Creación y configuración de una red de agentes de Amazon MQ

Date	Actualización de documentación
25 de abril de 2022	Amazon MQ Esta versión agrega el estado de agente <code>CRITICAL_ACTION_REQUIRED</code> y la propiedad de API <code>ActionRequired</code>. <code>CRITICAL_ACTION_REQUIRED</code> le informa cuando su agente se reduce. <code>ActionRequired</code> le proporciona un código que puede utilizar para encontrar instrucciones en la Guía del desarrollador sobre cómo resolver el problema. • Resolución de problemas • Documentación de ActionRequired en la referencia de la API de Amazon MQ.
20 de abril de 2022	Amazon MQ ahora es compatible con ActiveMQ 5.16.4, una versión secundaria de motor. Para obtener más información, consulte los siguientes temas: • Página de la versión de ActiveMQ 5.16.4 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Uso de archivos de configuración XML Spring • Actualización de una versión del motor del agente de Amazon MQ
1 de marzo de 2022	Amazon MQ ahora está disponible en la región Asia-Pacífico (Yakarta). Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS.

Date	Actualización de documentación
25 de febrero de 2022	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ versión 3.8.27. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.8.27 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
16 de febrero de 2022	Amazon MQ ahora está disponible en la región de África (Ciudad del Cabo). Para obtener más información sobre las regiones disponibles, consulte AWS Regiones y puntos de conexión en la guía Referencia general de AWS .

Date	Actualización de documentación
14 de febrero de 2022	Amazon MQ para RabbitMQ ahora es compatible con la versión 3.9.13 de RabbitMQ. No pueden utilizarse las actualizaciones automáticas de versiones secundarias para actualizar de Rabbit 3.8 a 3.9. Para hacerlo, actualice manualmente su agente. Para obtener más información sobre las nuevas funciones introducidas en RabbitMQ 3.9, consulte la página de notas de la versión 3.9.0 en el sitio web. GitHub  Note Actualmente, Amazon MQ no admite transmisiones, o el uso del registro estructurado en JSON, introducido en RabbitMQ 3.9. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: - Notas de la versión 3.9.13 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
7 de febrero de 2022	Amazon MQ para RabbitMQ presenta nuevas métricas de agente que le permiten supervisar la utilización media de recursos en los tres nodos de una implementación de clúster. Para obtener más información, consulte los siguientes temas: the section called “Métricas para RabbitMQ”

Date	Actualización de documentación
18 de enero de 2022	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ versión 3.8.26. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.8.26 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
13 de enero de 2022	Amazon MQ incorpora el código de estado RABBITMQ_MEMORY_ALARM para informar cuando el agente ha emitido una alarma de memoria elevada y se encuentra en mal estado. Amazon MQ proporciona información detallada y recomendaciones para ayudarlo a diagnosticar, resolver y evitar alarmas de memoria elevada. Para obtener más información, consulte lo siguiente. • the section called “ RABBITMQ_MEMORY_ALARM ”
6 de enero de 2022	Al configurar CloudWatch Logs for Amazon MQ para los agentes de ActiveMQ, Amazon MQ admite el uso de las claves de contexto de condición global aws:SourceAccount y las claves de contexto de condiciones globales en aws:SourceArn las políticas de IAM basadas en recursos para evitar el confuso problema de los diputados. Para obtener más información, consulte lo siguiente. • the section called “Prevención de la sustitución confusa entre servicios”

Date	Actualización de documentación
20 de diciembre de 2021	Amazon MQ para ActiveMQ incorpora un conjunto de nuevas métricas, que permiten monitorear el número máximo de conexiones que se pueden establecer con el agente mediante distintos protocolos de transporte compatibles, así como una nueva métrica adicional que permite monitorear el número de nodos conectados al agente en una red de agentes. Para obtener más información, consulte lo siguiente. • the section called “Métricas para ActiveMQ”
16 de noviembre de 2021	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ versión 3.8.23. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.8.23 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ.
12 de octubre de 2021	Amazon MQ ahora es compatible con ActiveMQ 5.16.3, una versión secundaria de motor principal. Para obtener más información, consulte los siguientes temas: • Página de la versión de ActiveMQ 5.16.3 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Actualización de una versión del motor del agente de Amazon MQ • Uso de archivos de configuración XML Spring

Date	Actualización de documentación
8 de septiembre de 2021	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ versión 3.8.22. Esta versión incluye una solución para un problema con las colas que utilizan el TTL (periodo de vida) por mensaje, que se identificó en la versión 3.8.17 de RabbitMQ, que se admitía antes. Le recomendamos que actualice sus agentes existentes a la versión 3.8.22. Para obtener más información acerca de las correcciones y características de esta versión, consulte los siguientes temas: • Notas de la versión 3.8.22 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ Para obtener más información acerca de las versiones admitidas de Amazon MQ para RabbitMQ y actualizaciones de agente de, consulte Administración de versiones del motor de Amazon MQ para RabbitMQ
25 de agosto de 2021	Amazon MQ para RabbitMQ ha desactivado temporalmente la versión 3.8.17 del motor RabbitMQ debido a un problema identificado con las colas que utilizan por mensaje (TTL). time-to-live Le recomendamos que utilice la versión 3.8.11.
29 de julio de 2021	Amazon MQ para RabbitMQ ahora es compatible con RabbitMQ versión 3.8.17. Para obtener más información acerca de las correcciones y características de esta actualización, consulte los siguientes temas: • Notas de la versión 3.8.17 de RabbitMQ GitHub en el repositorio del servidor RabbitMQ • Registro de cambios de RabbitMQ • Administración de versiones del motor de Amazon MQ para RabbitMQ

Date	Actualización de documentación
16 de julio de 2021	Ahora puede ajustar el período de mantenimiento de un agente de Amazon MQ mediante la Consola de administración de AWS, AWS CLI, o la API de Amazon MQ. Para obtener más información acerca de los períodos de mantenimiento del agente, consulte los siguientes temas. • Programación del período de mantenimiento de un agente de Amazon MQ
6 de julio de 2021	Amazon MQ para RabbitMQ introduce compatibilidad con el tipo de intercambio Hash consistente. Los intercambios Hash consistentes enrutan mensajes a colas en función de un valor hash que se calcula a partir de la clave de enrutamiento de un mensaje. Para obtener más información, consulte los siguientes temas: • Complemento de intercambio Hash consistente • Tipo de intercambio de hash coherente de RabbitMQ en el repositorio de RabbitMQ GitHub
7 de junio de 2021	Amazon MQ ahora es compatible con ActiveMQ 5.16.2, una nueva versión de motor principal. Para obtener más información, consulte los siguientes temas: • Página de la versión de ActiveMQ 5.16.2 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Actualización de una versión del motor del agente de Amazon MQ • Uso de archivos de configuración XML Spring
26 de mayo de 2021	Amazon MQ para RabbitMQ ya está disponible en las regiones China (Pekín) y China (Ningxia). Para obtener información acerca de las regiones disponibles, consulte Regiones y puntos de enlace de AWS.
18 de mayo de 2021	Amazon MQ para RabbitMQ implementa valores predeterminados del agente. Cuando crea un agente por primera vez, Amazon MQ crea un conjunto de políticas de agente y límites de anfitrión según el tipo de instancias y el modo de implementación que elija, con el fin de optimizar el rendimiento del agente. Para obtener más información, consulte lo siguiente: https://docs.aws.amazon.com/amazon-mq/latest/developer-guide/rabbitmq-defaults.html

Date	Actualización de documentación
5 de mayo de 2021	Ahora, Amazon MQ es compatible con ActiveMQ 5.15.15. Para obtener más información, consulte los siguientes temas: • Página de la versión de ActiveMQ 5.15.15 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Uso de archivos de configuración XML Spring
5 de mayo de 2021	Amazon MQ comenzó a realizar un seguimiento de los cambios en las políticas AWS gestionadas. Para obtener más información, consulte los siguientes temas: • the section called “AWS políticas gestionadas”
14 de abril de 2021	Amazon MQ ya está disponible en las regiones China (Pekín) y China (Ningxia) Para obtener información acerca de las regiones disponibles, consulte Regiones y puntos de enlace de AWS.
7 de abril de 2021	Amazon MQ ahora es compatible con RabbitMQ 3.8.11. Para obtener más información acerca de las correcciones y características de esta actualización, consulte los siguientes temas: • Notas de la versión 3.8.11 de RabbitMQ en el repositorio del servidor RabbitMQ GitHub • Registro de cambios de RabbitMQ • Administración de versiones del motor de Amazon MQ para RabbitMQ
1 de abril de 2021	Amazon MQ ya está disponible en la región Asia-Pacífico (Osaka). Para obtener información acerca de las regiones disponibles, consulte Regiones y puntos de enlace de Amazon MQ.

Date	Actualización de documentación
21 de diciembre de 2020	Ahora, Amazon MQ es compatible con ActiveMQ 5.15.14. Para obtener más información, consulte los siguientes temas: • Notas de la versión de ActiveMQ 5.15.14 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Uso de archivos de configuración XML Spring •  Important Debido a un problema conocido de Apache ActiveMQ en esta versión, el nuevo botón Pause queue (Pausar cola) de la consola web de ActiveMQ no se puede utilizar con los agentes de Amazon MQ para ActiveMQ. Para obtener información acerca de este problema, consulte AMQ-8104.

Date	Actualización de documentación
4 de noviembre de 2020	Amazon MQ ahora admite RabbitMQ, un popular agente de mensajes de código abierto. Esto le permite migrar sus agentes de mensajes RabbitMQ existentes a uno sin tener que volver a escribir el código. AWS Amazon MQ para RabbitMQ administra agentes de mensajes individuales y agrupados, y tareas como el aprovisionamiento de la infraestructura, la configuración del agente y la actualización del software. • Amazon MQ es compatible con RabbitMQ 3.8.6. Para obtener más información acerca de las versiones de motor admitidas, consulte the section called “Administración de versiones”. • El nivel gratuito de AWS incluye hasta 750 horas de un agente mq.t3.micro de una sola instancia y hasta 20 GB de almacenamiento al mes durante un año. Para obtener más información sobre los tipos de instancias admitidos, consulte Broker instance types. • Con Amazon MQ para RabbitMQ, puede acceder a sus agentes mediante AMQP 0-9-1 y en cualquiera de los idiomas que admiten las bibliotecas de cliente de RabbitMQ. Para obtener más información acerca de los conjuntos de cifrado compatibles, consulte the section called “Protocolos de Amazon MQ para RabbitMQ”. • Amazon MQ para RabbitMQ está disponible en todas las regiones en las que está disponible Amazon MQ actualmente. Para obtener más información acerca de todas las regiones disponibles, consulte la Tabla de regiones de AWS. Para empezar a utilizar Amazon MQ, crear un agente y conectar una aplicación basada en JVM con su agente de RabbitMQ, consulte Introducción: Cómo crear un agente de RabbitMQ y conectarse a él.

Date	Actualización de documentación
22 de octubre de 2020	Amazon MQ es compatible con ActiveMQ 5.15.13. Para obtener más información, consulte los siguientes temas: • Notas de la versión de ActiveMQ 5.15.13 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Uso de archivos de configuración XML Spring
30 de septiembre de 2020	Amazon MQ ahora está disponible en la región Europa (Milán). Para obtener información acerca de las regiones disponibles, consulte Regiones y puntos de enlace de Amazon MQ.
27 de julio de 2020	Puede autenticar usuarios de Amazon MQ utilizando las credenciales almacenadas en su Active Directory u otro servidor LDAP. También puede agregar, eliminar y modificar usuarios de Amazon MQ, y asignar permisos para temas y colas. Para obtener más información, consulte Integrar LDAP con ActiveMQ.
17 de julio de 2020	Amazon MQ ahora admite el tipo de instancias <code>mq.t3.micro</code>. Para obtener más información, consulte Broker instance types.
30 de junio de 2020	Amazon MQ es compatible con ActiveMQ 5.15.12. Para obtener más información, consulte los siguientes temas: • Notas de la versión de ActiveMQ 5.15.12 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Uso de archivos de configuración XML Spring

Date	Actualización de documentación
30 de abril de 2020	Amazon MQ admite un nuevo elemento secundario de colección, <code>systemUsage</code>, en el elemento <code>broker</code>. Para obtener más información, consulte systemUsage. Amazon MQ también admite tres nuevos atributos en el elemento secundario <code>kahaDB</code>. • <code>journalDiskSyncInterval</code> : intervalo (ms) para cuándo realizar una sincronización de disco si <code>journalDiskSyncStrategy=periodic</code>. • <code>journalDiskSyncStrategy</code> : configura la política de sincronización de disco. • <code>preallocationStrategy</code> : configura la forma en que el agente intentará preasignar los archivos de diario cuando se necesite un nuevo archivo de diario. Para obtener más información, consulte Atributos.
3 de marzo de 2020	Amazon MQ admite dos nuevas métricas CloudWatch • <code>TempPercentUsage</code> : porcentaje de almacenamiento temporal disponible utilizado por mensajes no persistentes. • <code>JobSchedulerStorePercentUsage</code> : porcentaje de espacio en disco utilizado por el almacén del programador de trabajos. Para obtener más información, consulte Monitoring and logging Amazon MQ brokers.
4 de febrero de 2020	Amazon MQ está disponible en las regiones Asia-Pacífico (Hong Kong) y Medio Oriente (Baréin). Para obtener información acerca de las regiones disponibles, consulte Regiones y puntos de enlace de AWS.

Date	Actualización de documentación
22 de enero de 2020	Amazon MQ es compatible con ActiveMQ 5.15.10. Para obtener más información, consulte los siguientes temas: • Notas de la versión de ActiveMQ 5.15.10 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Uso de archivos de configuración XML Spring
19 de diciembre de 2019	Amazon MQ está disponible en las regiones Europa (Estocolmo) y América del Sur (São Paulo). Para obtener información acerca de las regiones disponibles, consulte Regiones y puntos de enlace de AWS.

Date	Actualización de documentación
16 de diciembre de 2019	Amazon MQ admite la creación de agentes de rendimiento optimizado mediante el uso de Amazon Elastic Block Store (EBS), en lugar del Amazon Elastic File System (Amazon EFS) predeterminado, para el almacenamiento del agente. Para aprovechar el alto nivel de durabilidad y la replicación en varias zonas de disponibilidad, utilice Amazon EFS. Para aprovechar la baja latencia y el alto rendimiento, utilice Amazon EBS.  Important • Puede utilizar Amazon EBS únicamente con la familia de tipos de instancias de agente mq.m5. • Si bien puede cambiar el tipo de instancia de agente, no puede cambiar el tipo de almacenamiento de agente después de crear el agente. • Amazon EBS replica los datos dentro de una sola zona de disponibilidad y no admite el modo de implementación ActiveMQ activa/en espera. Para obtener más información, consulte los siguientes temas: • Storage • Elegir el tipo de almacenamiento de agente correcto para obtener el mejor rendimiento • La propiedad <code>storageType</code> del recurso broker-instance-options en la Referencia de la API REST de Amazon MQ • Las métricas <code>BurstBalance</code>, <code>VolumeReadOps</code> y <code>VolumeWriteOps</code> en la sección Monitoring and logging Amazon MQ brokers.
18 de octubre de 2019	Hay dos CloudWatch métricas de Amazon disponibles: <code>TotalEnqueueCount</code> y <code>TotalDequeueCount</code>. Para obtener más información, consulte Monitoring and logging Amazon MQ brokers

Date	Actualización de documentación
11 de octubre de 2019	Amazon MQ admite ahora puntos de enlace que cumplan los requisitos del estándar federal de procesamiento de información 140-2 (FIPS) en las regiones comerciales de EE. UU. Para obtener más información, consulte los siguientes temas: • Estándar de procesamiento de la información federal (FIPS, Federal Information Processing Standard) 140-2 • Regiones y puntos de conexión de Amazon MQ
30 de septiembre de 2019	Amazon MQ incluye ahora la posibilidad de escalar los agentes cambiando el tipo de instancias del anfitrión. Para obtener más información, consulte la propiedad <code>hostInstanceType</code> de UpdateBrokerInput y la propiedad <code>pendingHostInstanceType</code> de DescribeBrokerOutput.
30 de agosto de 2019	Ahora puede actualizar los grupos de seguridad asociados a un agente, tanto en la consola como con UpdateBrokerInput.
22 de julio de 2019	Amazon MQ se integra con AWS Key Management Service (KMS) para ofrecer cifrado en el servidor. Ahora puede seleccionar su propia CMK gestionada por el cliente o utilizar una clave de KMS AWS gestionada en su cuenta. AWS KMS Para obtener más información, consulte Cifrado en reposo. Amazon MQ admite el uso de AWS KMS claves de las siguientes maneras. • AWS clave KMS propia: la clave es propiedad de Amazon MQ y no está en tu cuenta. • AWS clave de KMS AWS gestionada: la clave de KMS gestionada (<code>aws/mq</code>) es una clave de KMS de su cuenta que Amazon MQ crea, gestiona y utiliza en su nombre. • Seleccione las CMK gestionadas por clientes existentes: las gestionadas por los clientes CMKs las crea y gestiona usted en (KMS). AWS Key Management Service

Date	Actualización de documentación
19 de junio de 2019	Amazon MQ está disponible en las regiones Europa (París) y Asia-Pacífico (Mumbai). Para obtener información acerca de las regiones disponibles, consulte Regiones y puntos de enlace de AWS .
12 de junio de 2019	Amazon MQ está disponible en la región Canadá (Central). Para obtener información acerca de las regiones disponibles, consulte Regiones y puntos de enlace de AWS .
3 de junio de 2019	Hay dos nuevas CloudWatch métricas de Amazon disponibles: <code>EstablishedConnectionsCount</code> y <code>InactiveDurableSubscribers</code>. Para obtener más información, consulte los siguientes temas: • Monitoring and logging Amazon MQ brokers • Monitoring and logging Amazon MQ brokers
10 de mayo de 2019	El almacenamiento de datos de los nuevos tipos de instancia <code>mq.t2.micro</code> está limitado a 20 GB. Para obtener más información, consulte los siguientes temas: • the section called “Almacenamiento de datos” • Broker instance types
29 de abril de 2019	A partir de ahora, puede utilizar políticas basadas en etiquetas y permisos de nivel de recursos. Para obtener más información, consulte los siguientes temas: • Cómo funciona Amazon MQ con IAM • Permisos de nivel de recurso para las acciones de la API de Amazon MQ
16 de abril de 2019	A partir de ahora, puede recuperar información sobre las opciones de las instancias y los motores del agente con la API REST. Para obtener más información, consulte los siguientes temas: • Opciones de instancias del agente • Tipos de motores del agente

Date	Actualización de documentación
8 de abril de 2019	Amazon MQ es compatible con ActiveMQ 5.15.9. Para obtener más información, consulte los siguientes temas: • Notas de la versión de ActiveMQ 5.15.9 • Administración de versiones del motor de Amazon MQ para ActiveMQ • Uso de archivos de configuración XML Spring
4 de marzo de 2019	Se ha mejorado la documentación para configurar la conmutación por error dinámica y el reequilibrio de clientes en una red de agentes. Habilite la conmutación por error dinámica configurando <code>transportConnectors</code> junto con las opciones de configuración de <code>networkConnectors</code> . Para obtener más información, consulte los siguientes temas: • Conmutación por error dinámica con conectores de transporte • Red de agentes de Amazon MQ • Amazon MQ Broker Configuration Parameters
27 de febrero de 2019	Amazon MQ está disponible en la región Europa (Londres), además de en las siguientes regiones: • Asia-Pacífico (Singapur) • Este de EE. UU. (Ohio) • Este de EE. UU. (Norte de Virginia) • Oeste de EE. UU. (Norte de California) • Oeste de EE. UU. (Oregón) • Asia-Pacífico (Tokio) • Asia-Pacífico (Seúl) • Asia-Pacífico (Sídney) • Europa (Fráncfort) • Europa (Irlanda)
24 de enero de 2019	La configuración predeterminada ahora incluye una política para purgar los destinos inactivos.

Date	Actualización de documentación
17 de enero de 2019	Los tipos de instancias <code>mq.t2.micro</code> de Amazon MQ ahora solo admiten 100 conexiones por protocolo de nivel de cable. Para obtener más información, consulte Quotas in Amazon MQ .
19 de diciembre de 2018	Puede configurar una serie de agentes de Amazon MQ en una red de agentes. Para obtener más información, consulte las siguientes secciones: • Red de agentes de Amazon MQ • Creating and Configuring a Network of Brokers • Configurar la red de agentes correctamente • networkConnector • networkConnectionStartAsincrónico
11 de diciembre de 2018	Amazon MQ es compatible con ActiveMQ 5.15.8, 5.15.6 y 5.15.0. • Se han resuelto los errores y se han realizado mejoras en ActiveMQ: • Notas de la versión de ActiveMQ 5.15.8 • Notas de la versión de ActiveMQ 5.15.7
5 de diciembre de 2018	AWS admite el etiquetado de recursos para ayudar a realizar un seguimiento de la asignación de costes. Puede etiquetar recursos al crearlos o visualizando los detalles de dicho recurso. Para obtener más información, consulte Etiquetado de los recursos .
19 de noviembre de 2018	AWS ha ampliado su programa de conformidad con el SOC para incluir Amazon MQ como servicio compatible con el SOC .
15 de octubre de 2018	• El número máximo de grupos por usuario es 20. Para obtener más información, consulte Usuarios. • El número máximo de conexiones por agente y protocolo de nivel de conexión es 1000. Para obtener más información, consulte Agentes.
2 de octubre de 2018	AWS ha ampliado su programa de conformidad con la HIPAA para incluir Amazon MQ como servicio que cumple con los requisitos de la HIPAA .

Date	Actualización de documentación
27 de septiembre de 2018	Amazon MQ es compatible con ActiveMQ 5.15.6, además de la versión 5.15.0. Para obtener más información, consulte los siguientes temas: • Introducción: Creación y conexión con un agente de ActiveMQ • Se han resuelto los errores y se han realizado mejoras en la documentación de ActiveMQ: • Notas de la versión de ActiveMQ 5.15.6 • Notas de la versión de ActiveMQ 5.15.5 • Notas de la versión de ActiveMQ 5.15.4 • Notas de la versión de ActiveMQ 5.15.3 • Notas de la versión de ActiveMQ 5.15.2 • Notas de la versión de ActiveMQ 5.15.1 • Cliente de ActiveMQ 5.15.6
31 de agosto de 2018	• Están disponibles las siguientes métricas: • <code>CurrentConnectionsCount</code> • <code>TotalConsumerCount</code> • <code>TotalProducerCount</code> Para obtener más información, consulte la sección Monitoring and logging Amazon MQ brokers. • La dirección IP del agente se muestra en la página Details (Detalles).  Note Para los agentes con el acceso público deshabilitado, se muestra la dirección IP interna.

Date	Actualización de documentación
30 de agosto de 2018	Amazon MQ está disponible en la región Asia-Pacífico (Singapur), además de en las siguientes regiones: • Este de EE. UU. (Ohio) • Este de EE. UU. (Norte de Virginia) • Oeste de EE. UU. (Norte de California) • Oeste de EE. UU. (Oregón) • Asia-Pacífico (Tokio) • Asia-Pacífico (Seúl) • Asia-Pacífico (Sídney) • Europa (Fráncfort) • Europa (Irlanda)
30 de julio de 2018	Puede configurar Amazon MQ para publicar registros generales y de auditoría en Amazon CloudWatch Logs. Para obtener más información, consulte Monitoring and logging Amazon MQ brokers.
25 de julio de 2018	Amazon MQ está disponible en las regiones Asia-Pacífico (Tokio) y Asia-Pacífico (Seúl), además de en las siguientes regiones: • Este de EE. UU. (Ohio) • Este de EE. UU. (Norte de Virginia) • Oeste de EE. UU. (Norte de California) • Oeste de EE. UU. (Oregón) • Asia-Pacífico (Sídney) • Europa (Fráncfort) • Europa (Irlanda)
19 de julio de 2018	Se puede utilizar AWS CloudTrail para registrar las llamadas a la API de Amazon MQ. Para obtener más información, consulte Logging Amazon MQ API calls using CloudTrail.

Date	Actualización de documentación
29 de junio de 2018	Además de <code>mq.t2.micro</code> y <code>mq.m4.large</code>, están disponibles los siguientes tipos de instancia de agente para cargas de trabajo de desarrollo, pruebas y producción que requieren un alto desempeño: • <code>mq.m5.large</code> • <code>mq.m5.xlarge</code> • <code>mq.m5.2xlarge</code> • <code>mq.m5.4xlarge</code> Para obtener más información, consulte Broker instance types.
27 de junio de 2018	Amazon MQ está disponible en la región EE. UU. Oeste (Norte de California), además de en las siguientes regiones: • Este de EE. UU. (Ohio) • Este de EE. UU. (Norte de Virginia) • Oeste de EE. UU. (Oregón) • Asia-Pacífico (Sídney) • Europa (Fráncfort) • Europa (Irlanda)

Date	Actualización de documentación
14 de junio de 2018	- Puede utilizar el AWS::Amazon MQ::Broker AWS CloudFormation recurso para realizar las siguientes acciones: - Crear un agente. - Añadir cambios de configuración o modificar los usuarios del agente especificado. - Devolver información sobre el agente especificado. - Eliminar el agente especificado.  Note Al cambiar cualquier propiedad del tipo de propiedad Amazon MQ Broker ConfigurationId o Amazon MQ Broker User, el corredor se reinicia inmediatamente. - Puede utilizar el AWS::Amazon MQ::Configuration AWS CloudFormation recurso para realizar las siguientes acciones: - Crear una configuración. - Actualizar la configuración especificada. - Devolver información sobre la configuración especificada.  Note Se puede utilizar CloudFormation para modificar, pero no para eliminar, una configuración de Amazon MQ.
7 de junio de 2018	La consola de Amazon MQ admite alemán, portugués de Brasil, español, italiano y chino tradicional.
17 de mayo de 2018	El límite de número de usuarios por agente es 250. Para obtener más información, consulte Usuarios .
13 de marzo de 2018	El proceso de creación de un agente tarda unos 15 minutos. Para obtener más información, consulte Terminar de crear el agente .

Date	Actualización de documentación
1 de marzo de 2018	- Puede configurar el almacenamiento y el envío simultáneos para Apache KahaDB mediante el atributo concurrentStoreAndDispatchQueueues. - La CpuCreditBalance CloudWatch métrica > está disponible para el tipo de instancia de bróker. mq.t2.micro
10 de enero de 2018	Los siguientes cambios afectan a la consola de Amazon MQ: - En la lista de agentes, la columna Creation está oculta de forma predeterminada. Para personalizar el tamaño de la página y las columnas, seleccione . - En la MyBroker página, en la sección Conexiones, elija el nombre del grupo de seguridad o  abra la consola EC2 (en lugar de la consola de VPC). La consola de EC2 le permite configurar de forma intuitiva las reglas de entrada y salida. Para obtener más información, consulte la información actualizada de la sección Connecting a Java application to your broker.
9 de enero de 2018	- El permiso para el ID de operación de REST UpdateBroker aparece correctamente como mq:UpdateBroker en la consola de IAM. - Se ha eliminado el permiso mq:DescribeEngine erróneo de la consola de IAM;.

Date	Actualización de documentación
28 de noviembre de 2017	Esta es la versión inicial Amazon MQ y de la Guía para desarrolladores de Amazon MQ. - Amazon MQ está disponible en las siguientes regiones: - Este de EE. UU. (Ohio) - Este de EE. UU. (Norte de Virginia) - Oeste de EE. UU. (Oregón) - Asia-Pacífico (Sídney) - Europa (Fráncfort) - Europa (Irlanda) El uso del tipo de instancia de <code>mq.t2.micro</code> está sujeto a créditos de CPU y rendimiento de referencia, con la capacidad de ampliarse por encima del nivel de referencia (para obtener más información, consulte la métrica CpuCreditBalance). Si su aplicación requiere rendimiento fijo, considere el uso de un tipo de instancias <code>mq.m5.large</code>. - Puede crear agentes <code>mq.m4.large</code> y <code>mq.t2.micro</code>. El uso del tipo de instancia de <code>mq.t2.micro</code> está sujeto a créditos de CPU y rendimiento de referencia, con la capacidad de ampliarse por encima del nivel de referencia (para obtener más información, consulte la métrica CpuCreditBalance). Si su aplicación requiere rendimiento fijo, considere el uso de un tipo de instancias <code>mq.m5.large</code>. - Puede usar el motor del agente ActiveMQ 5.15.0. - También puede crear y administrar corredores mediante programación mediante la API REST de Amazon MQ y AWS SDKs - Para obtener acceso a sus agentes, puede usar cualquier lenguaje de programación que admita ActiveMQ y habilitar TLS explícitamente para los siguientes protocolos: AMQP MQTT - MQTT sobre WebSocket OpenWire

Date	Actualización de documentación
	• STOMP • STOMP WebSocket • Puede conectarse a los agentes de ActiveMQ mediante distintos clientes de ActiveMQ. Le recomendamos que utilice el cliente de ActiveMQ. Para obtener más información, consulte Connecting a Java application to your broker. • El agente puede enviar y recibir mensajes de cualquier tamaño.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.