aws

Guia do desenvolvedor

Amazon MQ

Amazon MQ: Guia do desenvolvedor

Table of Contents

O que é o Amazon MQ?

O Amazon MQ é um serviço gerenciado de agente de mensagens para o Apache ActiveMQ Classic e o RabbitMQ que gerencia a configuração, operação e manutenção de agentes de mensagens. Você pode criar um agente do Amazon MQ usando protocolos de mensagens padrão do setor ou migrar agentes de mensagens existentes para o Amazon MQ sem reescrever o código de mensagens.

Um agente é um ambiente de agente de mensagens em execução no Amazon MQ. É o bloco de criação básico do Amazon MQ. Um agente de mensagem permite que aplicações de software e componentes se comuniquem usando várias linguagens de programação, sistemas operacionais e protocolos de sistemas de mensagens formais. Você pode usar agentes do Amazon MQ para comunicação entre aplicações e componentes de grande escala nativos da nuvem.

Tópicos

Recursos do Amazon MQ

Manutenção gerenciada e atualizações de versão

O Amazon MQ realiza a manutenção e as atualizações de versão de um agente de mensagens durante a janela de manutenção programada.

Monitorar agentes com o CloudWatch

O Amazon MQ é integrado ao Amazon CloudWatch para que você possa visualizar e analisar métricas dos agentes e das filas. Você pode visualizar e analisar métricas usando o console do Amazon MQ, o console do CloudWatch, a linha de comandos e a API. As métricas são automaticamente coletadas e enviadas ao CloudWatch a cada minuto.

Segurança

O Amazon MQ oferece o recurso de criptografia de mensagens em repouso e em trânsito. As conexões com o agente usam SSL, e o acesso pode ser restrito a um endpoint privado dentro da

Amazon VPC. Além disso, você pode usar o AWS Identity and Access Management (IAM) para controlar quais ações os usuários e grupos do IAM podem realizar em agentes específicos do Amazon MQ.

Filas de quórum do RabbitMQ no Amazon MQ

As filas de quórum são um tipo de fila replicada composta de um nó líder (réplica primária) e de nós seguidores (outras réplicas). Cada nó está em uma zona de disponibilidade diferente; então, se um nó estiver temporariamente indisponível, a entrega de mensagens continuará com uma réplica líder recém-eleita em outra zona de disponibilidade. As filas de quórum são úteis para lidar com mensagens mal-intencionadas, que ocorrem quando uma mensagem falha e é enfileirada várias vezes.

Replicação de dados entre regiões para o ActiveMQ no Amazon MQ

A replicação de dados entre regiões (CRDR) permite a replicação assíncrona de mensagens do agente primário em uma região primária da AWS para o agente de réplica em uma região de réplica. Ao emitir uma solicitação de failover para a API do Amazon MQ, o agente de réplica atual é promovido à função de agente primário e o agente primário atual é rebaixado para a função de réplica.

Como posso começar a usar o Amazon MQ?

Para começar a usar o ActiveMQ no Amazon MQ, consulte a seguinte documentação:

- Conceitos básicos: criar e conectar a um agente do ActiveMQ
- the section called “Como implantar um agente”
- Tutoriais ActiveMQ
- the section called “Práticas recomendadas do Amazon MQ para ActiveMQ”

Para começar a usar o RabbitMQ no Amazon MQ, consulte a seguinte documentação:

- Conceitos básicos: criar e conectar a um agente do RabbitMQ
- the section called “Implantar um agente do RabbitMQ”
- the section called “Tutoriais do RabbitMQ”
- the section called “Práticas recomendadas do Amazon MQ para RabbitMQ”

Para saber mais sobre as APIs do Amazon MQ REST, consulte a Referência da API REST do Amazon MQ.

Para saber mais sobre os AWS CLI comandos do Amazon MQ, veja o Amazon MQ na AWS CLIReferência de Comandos.

Como posso fornecer feedback para o Amazon MQ?

Agradecemos e incentivamos seu feedback sobre a documentação. Você pode usar os ícones de polegar para cima e polegar para baixo no lado direito para enviar feedback ou pode usar o formulário “Fornecer feedback” no link abaixo.

Para entrar em contato com a equipe do Amazon MQ, use o Fórum de discussão do Amazon MQ.

Configuração do Amazon MQ

Antes de usar o Amazon MQ, é necessário executar as etapas a seguir.

Tópicos

Etapa 1: pré-requisitos

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS Centro de Identidade do AWS IAM, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login Console de gerenciamento da AWScomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

   Para obter instruções, consulte Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz (console) no Guia do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o Centro de Identidade do AWS IAM no Guia do usuário do Centro de Identidade do AWS IAM .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

   Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia Centro de Identidade do AWS IAM do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do Centro de Identidade do AWS IAM .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de logon único ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do Centro de Identidade do AWS IAM .

Etapa 2: criar um usuário e obter suas AWS credenciais

Os usuários precisam de acesso programático se quiserem interagir com pessoas AWS fora do Console de gerenciamento da AWS. A forma de conceder acesso programático depende do tipo de usuário que está acessando AWS.

Para conceder acesso programático aos usuários, selecione uma das seguintes opções:

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Recomendado) Use as credenciais do console como credenciais temporárias para assinar solicitações programáticas para o AWS CLI, AWS SDKs ou. AWS APIs	Siga as instruções da interface que deseja utilizar. • Para o AWS CLI, consulte Login para desenvolvimento AWS local no Guia AWS Command Line Interface do usuário. • Para AWS SDKs isso, consulte Login para desenvolvimento AWS local no Guia de referência de ferramentas AWS SDKs e ferramentas.

Qual usuário precisa de acesso programático?	Para	Por
Identidade da força de trabalho (Usuários gerenciados no Centro de Identidade do IAM)	Use credenciais temporárias para assinar solicitações programáticas para o AWS CLI AWS SDKs, ou. AWS APIs	Siga as instruções da interface que deseja utilizar. • Para o AWS CLI, consulte Configurando o AWS CLI para uso Centro de Identidade do AWS IAM no Guia do AWS Command Line Interface usuário. • Para AWS SDKs, ferramentas e AWS APIs, consulte a autenticação do IAM Identity Center no Guia de referência de ferramentas AWS SDKs e ferramentas.
IAM	Use credenciais temporárias para assinar solicitações programáticas para o AWS CLI AWS SDKs, ou. AWS APIs	Siga as instruções em Como usar credenciais temporárias com AWS recursos no Guia do usuário do IAM.

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitaç ões programáticas para o AWS CLI, AWS SDKs, ou. AWS APIs	Siga as instruções da interface que deseja utilizar. • Para isso AWS CLI, consulte Autenticação usando credenciais de usuário do IAM no Guia do AWS Command Line Interface usuário. • Para ferramentas AWS SDKs e ferramentas, consulte Autenticar usando credenciais de longo prazo no Guia de referência de ferramentas AWS SDKs e ferramentas. • Para isso AWS APIs, consulte Gerenciamento de chaves de acesso para usuários do IAM no Guia do usuário do IAM.

Etapa 3: Preparar-se para usar o código de exemplo

Os tutoriais a seguir mostram como você pode trabalhar com corretores Amazon MQ usando Console de gerenciamento da AWS o e como se conectar programaticamente aos corretores Amazon MQ para ActiveMQ e Amazon MQ para RabbitMQ. Se você quiser usar o código de exemplo ActiveMQ de Java, será necessário instalar o Java Standard Edition Development Kit e fazer algumas alterações de configuração no código de exemplo.

Você também pode criar e gerenciar corretores de forma programática usando a API REST do Amazon MQ e. AWS SDKs

Próximas etapas

Agora que você está preparado para trabalhar com o Amazon MQ, comece criando um agente. Dependendo do seu tipo de mecanismo de agente, você pode Conectar uma aplicação Java ao agente do Amazon MQ para ActiveMQ ou usar a biblioteca do cliente Java RabbitMQ para conectar uma aplicação baseada em JVM ao seu agente Amazon MQ para RabbitMQ.

Conceitos básicos: criar e conectar a um agente do ActiveMQ

Um agente é um ambiente de agente de mensagens em execução no Amazon MQ. É o bloco de criação básico do Amazon MQ. A descrição combinada da classe (`m5`) e do tamanho (`large`, `medium`) da instância do agente é um tipo de instância de agente (por exemplo,`mq.m5.large`). Para obter mais informações, consulte O que é um agente do Amazon MQ para ActiveMQ?.

Criar um operador do ActiveMQ

A tarefa inicial e mais comum do Amazon MQ é a criação de um agente. O exemplo a seguir mostra como é possível usar o Console de gerenciamento da AWS para criar um agente básico.

1. Faça login no console do Amazon MQ.
2. Na página Select broker engine (Selecionar mecanismo de agente), selecione Apache ActiveMQ (Apache ActiveMQ).
3. Na página Select deployment and storage (Selecionar implantação e armazenamento), na seção Deployment mode and storage type (Modo de implantação e tipo de armazenamento), faça o seguinte:

   a. Selecione o Deployment mode (Modo de implantação) (por exemplo:Agente ativo/em espera). Para obter mais informações, consulte Opções de implantação de agentes do Amazon MQ para ActiveMQ.

      - Um agente de instância única é composto por um agente em uma Zona de disponibilidade. O agente se comunica com sua aplicação e com um volume de armazenamento do Amazon EBS ou Amazon EFS. Para obter mais informações, consulte Opção 1: agentes de instância única do Amazon MQ.
      - Um agente ativo/em espera de alta disponibilidade é composto por dois agentes em duas zonas de disponibilidade diferentes, configuradas em um par redundante. Esses agentes se comunicam de forma síncrona com sua aplicação e com o Amazon EFS. Para obter mais informações, consulte Opção 2: active/standby corretores Amazon MQ para alta disponibilidade.

   b. Escolha o Tipo de armazenamento (por exemplo, EBS). Para obter mais informações, consulte Storage.

> **Note**
>
> O Amazon EBS replica dados em uma única zona de disponibilidade e não é compatível com o modo de implantação ativo/em espera do ActiveMQ.

c. Escolha Próximo.

4. Na página Definir configurações, faça o seguinte na seção Detalhes:

   a. Digite o Broker name (Nome do agente).

> **Important**
>
> Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de agente. Os nomes de agente são acessíveis a outros serviços de AWS, incluindo o CloudWatch Logs. Nomes de agente não devem ser usados para dados privados ou sigilosos.

> **Note**
>
> Na seção Configurações adicionais, você também pode configurar o seguinte:
>
> - Configurações
> - CloudWatch Logs
> - Acesso privado
> - Janela de manutenção de agente

   b. Selecione o Tipo de instância de agente (por exemplo, m5.large). Para obter mais informações, consulte Broker instance types.

5. Na seção ActiveMQ Web Console access (Acesso ao console da Web ActiveMQ), forneça um Username (Nome de usuário) e Password (Senha). As seguintes restrições se aplicam a nomes de usuário e senhas de agente:

   - Seu nome de usuário pode conter somente caracteres alfanuméricos, traços, pontos, sublinhados e tils (- . _ ~).

- Sua senha deve ter pelo menos 12 caracteres, deve conter pelo menos 4 caracteres exclusivos e não deve conter vírgulas, dois pontos ou sinais de igual (,:=).

> **Important**
>
> Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de usuário do agente. Nomes de usuário do agente são acessíveis a outros serviços de AWS, incluindo o CloudWatch Logs. Nomes de usuário do agente não devem ser usados para dados privados ou sigilosos.

6. Escolha Implantar.

   Enquanto o Amazon MQ cria seu agente, ele exibe o status Criação em andamento.

   A criação do agente leva cerca de 15 minutos.

   Quando o seu agente é criado com sucesso, o Amazon MQ exibe o status Running (Em execução).

7. Selecione ***MyBroker*** (MeuAgente).

   Na página ***MyBroker***, na seção Connect (Conectar), observe a URL do Console da Web do ActiveMQ do agente, por exemplo:

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

   Além disso, observe os Endpoints de protocolo de nível de conexão do seu agente. Veja a seguir um exemplo de endpoint OpenWire.

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

Conceitos básicos: criar e conectar a um agente do RabbitMQ

Um agente é um ambiente de agente de mensagens em execução no Amazon MQ. É o bloco de criação básico do Amazon MQ. A descrição combinada da classe (`m5`) e do tamanho (`large`, `medium`) da instância do agente é um tipo de instância de agente (por exemplo,`mq.m5.large`). Para obter mais informações, consulte . O que é um agente do Amazon MQ para RabbitMQ?

Criar um agente do RabbitMQ

A tarefa inicial e mais comum do Amazon MQ é a criação de um agente. O exemplo a seguir mostra como é possível usar o Console de gerenciamento da AWS para criar um agente básico.

Quando criar um agente do Amazon MQ para RabbitMQ, siga as práticas recomendadas de configuração do agente para RabbitMQ para maximizar o desempenho do agente e otimizar a eficiência do throughput de mensagens.

1. Faça login no console do Amazon MQ.
2. Na página Select broker engine (Selecionar mecanismo do agente), selecione RabbitMQ e, em seguida, selecione Next (Avançar).
3. Na página Select deployment mode (Selecionar modo de implementação), escolha o Deployment mode (Modo de implantação), por exemplo, Cluster deployment (Implantação de cluster) e, depois, escolha Next (Avançar).

   - Um Network Load Balancer (NLB) é composto por um agente em uma zona de disponibilidade atrás de um Network Load Balancer (NLB). O agente se comunica com sua aplicação e com um volume de armazenamento do Amazon EBS. Para obter mais informações, consulte Opção 1: agente de instância única do Amazon MQ para RabbitMQ.
   - A implantação de cluster RabbitMQ para alta disponibilidade é um agrupamento lógico de três nós do agente RabbitMQ atrás de um Network Load Balancer (NLB), cada um compartilhando usuários, filas e um estado distribuído em várias Zonas de Disponibilidade (AZ). Para obter mais informações, consulte Opção 2: implantação do cluster do Amazon MQ para RabbitMQ.
4. Na página Definir configurações, faça o seguinte na seção Detalhes:

   a. Digite o Broker name (Nome do agente).

> **Important**
>
> Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de agente. Os nomes de agente são acessíveis a outros serviços de AWS, incluindo o CloudWatch Logs. Nomes de agente não devem ser usados para dados privados ou sigilosos.

b. Selecione o Tipo de instância de agente (por exemplo, mq.m7g.large). Para obter mais informações, consulte Broker instance types.

5. Na página Configure settings (Definição de configurações), na seção RabbitMQ access (Acesso RabbitMQ), forneça um Username (Nome de usuário) e Password (Senha). As seguintes restrições se aplicam a credenciais de login do agente:

   - Seu nome de usuário pode conter somente caracteres alfanuméricos, traços, pontos e sublinhados (- . _). Este valor não deve conter quaisquer caracteres de til (~). O Amazon MQ proíbe o uso de `guest` como um nome de usuário.
   - Sua senha deve ter pelo menos 12 caracteres, deve conter pelo menos 4 caracteres exclusivos e não deve conter vírgulas, dois pontos ou sinais de igual (,:=).

   > **Important**
   >
   > Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de usuário do agente. Nomes de usuário do agente são acessíveis a outros serviços de AWS, incluindo o CloudWatch Logs. Nomes de usuário do agente não devem ser usados para dados privados ou sigilosos.

   > **Note**
   >
   > Na seção Configurações adicionais, você também pode configurar o seguinte:
   >
   > - Configurações
   > - CloudWatch Logs
   > - Acesso privado
   > - Janela de manutenção de agente

6. Escolha Próximo.
7. Na página Review and create (Revisar e criar), você pode revisar suas seleções e editá-las conforme necessário.
8. Escolha Criar agente.

   Enquanto o Amazon MQ cria seu agente, ele exibe o status Criação em andamento.

   A criação do agente leva cerca de 15 minutos.

   Quando o seu agente é criado com sucesso, o Amazon MQ exibe o status Running (Em execução).
9. Selecione ***MyBroker*** (MeuAgente).

   Na página ***MyBroker***, na seção Connect (Conectar), observe a URL do Console da Web do RabbitMQ do agente, por exemplo:

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

   Além disso, observe o Endpoint secure-AMQP. Veja a seguir um exemplo de endpoint `amqps` expondo a porta listener `5671`.

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

Gerenciando um agente do Amazon MQ

Depois de criar um agente, você pode gerenciar e manter os diferentes componentes do agente do Amazon MQ.

Tópicos

Conectando ao Amazon MQ

Você pode se conectar ao Amazon MQ de outros serviços do AWS usando endpoints de serviço e endpoints de agentes.

Service endpoints

Os seguintes métodos de conexão são usados pela API de serviço do Amazon MQ:

Domínios	Método de conexão
`mq.region.amazonaws.com`	IPv4
`mq.region.api.aws`	Pilha dupla (IPv4 e IPv6)
`mq-fips.region.amazonaws.com`	FIPS somente com IPv4

Domínios	Método de conexão
`mq-fips.region.api.aws`	FIPS com pilha dupla

Endpoints do agente

Os seguintes métodos de conexão são usados pelos corretores do Amazon MQ:

Domínios	Método de conexão
`brokerId.mq.region.amazonaws.com`	IPv4
`brokerId.mq.region.on.aws`	Pilha dupla (IPv4 e IPv6) **Note** Os agentes do Amazon MQ para ActiveMQ não são compatíveis com a pilha dupla.

Conectar-se ao Amazon MQ usando endpoints de pilha dupla (IPv4 e IPv6)

Endpoints de pilha dupla são compatíveis com tráfego IPv4 e IPv6. Quando você faz uma solicitação para um endpoint de pilha dupla, a URL do endpoint resolve para um endereço IPv4 ou IPv6. Para obter mais informações sobre endpoints de pilha dupla e FIPS, consulte o Guia de referência de SDK.

O Amazon MQ oferece suporte a endpoints de pilha dupla regionais, o que significa que você deve especificar a região do AWS como parte do nome do endpoint. Os nomes de endpoints de pilha dupla usam a seguinte convenção de nomenclatura: `mq.region.api.aws`. Por exemplo, o nome do endpoint de pilha dupla para a região `eu-west-1` é `mq.eu-west-1.api.aws`.

Para obter a lista completa dos endpoints do Amazon MQ, consulte a Referência geral do AWS.

Conectar-se ao Amazon MQ usando o AWS PrivateLink

Os endpoints AWS PrivateLink para a API do Amazon MQ com suporte para IPv4 e IPv6 fornecem conectividade privada entre nuvens privadas virtuais (VPCs) e a API do Amazon MQ sem expor seu tráfego à Internet pública.

> **Note**
>
> O suporte ao PrivateLink está disponível somente para o endpoint da API do Amazon MQ, não para o endpoint do agente. Para obter mais informações sobre como se conectar de forma privada a um endpoint de agente, consulte Configuring a private Amazon MQ broker.

Para acessar a API do Amazon MQ usando o PrivateLink, você deve primeiro criar endpoint da VPC de interface na VPC específica a partir da qual você deseja se conectar. Quando o endpoint da VPC, use o nome do serviço `com.amazonaws.region.mq` ou `com.amazonaws.region.mq-fips` para endpoints FIPS.

Quando chamar o Amazon MQ usando a CLI ou o SDK do AWS, você deve especificar a URL do endpoint para usar o nome de domínio de pilha dupla: `mq.region.api.aws` ou `mq-fips.region.api.aws`. O PrivateLink para Amazon MQ não oferece suporte ao nome de domínio padrão que termina em `amazonaws.com`. Para obter mais informações, consulte Endpoints FIPS e de pilha dupla no Guia de referência de SDK.

O exemplo de CLI a seguir mostra como chamar a `describe-broker-engine-type` na região Ásia-Pacífico (Sydney) por meio de um endpoint da VPC do Amazon MQ.

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

Para outras formas de configurar o endpoint na CLI, consulte Uso de endpoints na CLI da AWS

Você também pode determinar o acesso do usuário aos endpoints da VPC usando as políticas de endpoint da VPC. Para obter mais informações, consulte Controlar o acesso a endpoints de VPC usando políticas de endpoint.

Autenticação e autorização para corretores do Amazon MQ

O Amazon MQ oferece vários métodos de autenticação e autorização para proteger sua infraestrutura de mensagens de acordo com os requisitos da sua organização.

Autenticação e autorização do Amazon MQ for RabbitMQ

O Amazon MQ para RabbitMQ oferece suporte aos seguintes métodos de autenticação e autorização:

Autorização e autenticação simples

Nesse método, os usuários do agente são armazenados internamente no agente do RabbitMQ e gerenciados por meio do console Web ou da API de gerenciamento. As permissões para vhosts, trocas, filas e tópicos são configuradas diretamente no RabbitMQ. Esse é o método padrão. Para obter mais informações, consulte Autenticação e autorização simples.

OAuth Autenticação e autorização 2.0

Nesse método, os usuários do broker e suas permissões são gerenciados por um provedor de identidade (IdP) externo OAuth 2.0. A autenticação do usuário e as permissões de recursos para vhosts, trocas, filas e tópicos são centralizadas por meio do sistema de escopo do provedor OAuth 2.0. Isso simplifica o gerenciamento de usuários e permite a integração com os sistemas de identidade existentes. Para obter mais informações, consulte Autenticação e autorização OAuth 2.0.

Autenticação e autorização do IAM

Nesse método, os usuários do broker se autenticam usando credenciais AWS do IAM por meio da federação de saída do IAM. As credenciais do IAM são usadas para obter tokens JWT do AWS Security Token Service (STS), e esses tokens JWT servem como tokens OAuth 2.0 para autenticação. Esse método aproveita o suporte OAuth 2.0 existente no Amazon MQ para RabbitMQ, AWS onde atua como o provedor de identidade 2.0. OAuth A autenticação do usuário é gerenciada pelo AWS IAM, enquanto as permissões de recursos para vhosts, trocas, filas e tópicos são gerenciadas por meio de políticas do IAM e aliases de escopo configurados no RabbitMQ. Para obter mais informações, consulte Autenticação e autorização do IAM.

Autenticação e autorização LDAP

Nesse método, os usuários do broker e suas permissões são gerenciados por um serviço de diretório LDAP externo. A autenticação do usuário e as permissões de recursos são centralizadas por meio do

servidor LDAP, permitindo que os usuários acessem o RabbitMQ usando suas credenciais de serviço de diretório existentes. Para obter mais informações, consulte Autenticação e autorização LDAP.

Autenticação e autorização HTTP

Nesse método, os usuários do broker e suas permissões são gerenciados por um servidor HTTP externo. A autenticação do usuário e as permissões de recursos são centralizadas por meio do servidor HTTP, permitindo que os usuários acessem o RabbitMQ usando seu próprio provedor de autenticação e autorização. Para obter mais informações sobre esse método, consulte Autenticação e autorização HTTP.

Autenticação de certificado SSL

O Amazon MQ oferece suporte a TLS mútuo (mTLS) para corretores RabbitMQ. O plug-in de autenticação SSL usa certificados de cliente de conexões mTLS para autenticar usuários. Nesse método, os usuários do broker são autenticados usando certificados de cliente X.509 em vez de credenciais de nome de usuário e senha. O certificado do cliente é validado em relação a uma Autoridade Certificadora (CA) confiável e o nome de usuário é extraído de um campo no certificado, como Nome comum (CN) ou Nome alternativo do assunto (SAN). Esse método fornece autenticação forte sem transmitir credenciais pela rede. Para obter mais informações, consulte Autenticação de certificado SSL.

> **Note**
>
> O RabbitMQ suporta vários métodos de autenticação e autorização para serem usados simultaneamente. Por exemplo, você pode ativar a autenticação OAuth 2.0 e a autenticação simples (interna). Para obter mais informações, consulte a seção do tutorial OAuth 2.0 sobre como habilitar a autenticação OAuth 2.0 e simples (interna) e a documentação de controle de acesso do RabbitMQ.
>
> O Amazon MQ recomenda criar um usuário interno ao testar as configurações de autenticação. Isso permite que a configuração de acesso seja validada usando a API de gerenciamento do RabbitMQ. Para obter mais informações, consulte Validação de acesso.

Autenticação e autorização para Amazon MQ for ActiveMQ

O Amazon MQ para ActiveMQ suporta os seguintes métodos de autenticação e autorização:

Autorização e autenticação simples

Nesse método, os usuários do broker são criados e gerenciados por meio do console ou da API do Amazon MQ. Os usuários podem ser configurados com permissões específicas para acessar filas, tópicos e o ActiveMQ Web Console. Para obter mais informações sobre esse método, consulte Criando um usuário do ActiveMQ broker.

Autenticação e autorização LDAP

Nesse método, os usuários do broker se autenticam por meio de credenciais armazenadas em seu servidor LDAP. Você pode adicionar, excluir e modificar usuários e atribuir permissões a tópicos e filas por meio do servidor LDAP, fornecendo autenticação e autorização centralizadas. Para obter mais informações sobre esse método, consulte Integrando os corretores ActiveMQ com o LDAP.

Atualizando uma versão do mecanismo de agente do Amazon MQ

O Amazon MQ fornece regularmente novas versões do mecanismo do agente para todos os tipos de mecanismo de agente compatíveis. As novas versões do mecanismo incluem patches de segurança, correções de bugs e outras melhorias no mecanismo do agente.

O Amazon MQ organiza os números das versões de acordo com a especificação de versionamento semântico como `X.Y.Z`. Nas implementações do Amazon MQ, `X` denota a versão principal, `Y` representa a versão secundária e `Z` denota o número da versão de patch. O Amazon MQ oferece suporte a dois tipos de atualizações:

- Atualização da versão principal: ocorre quando os números de versão do mecanismo principal mudam. Por exemplo, a atualização do RabbitMQ versão 3.13 para a versão 4.2 é considerada uma atualização principal da versão.
- Atualização de versão secundária: ocorre quando apenas os números de versão secundários do mecanismo mudam. Por exemplo, atualizando a partir da versão 3. 11 para a versão 3. 12 é considerado um pequeno upgrade de versão.

Você pode atualizar seu agente manualmente a qualquer momento para a próxima versão principal ou secundária compatível. O Amazon MQ gerencia a atualização para a última versão de patch compatível com todos os corretores durante a janela de manutenção programada. As atualizações manuais e automáticas da versão ocorrem durante a janela de manutenção programada ou após a reinicialização do broker. O Amazon MQ atualiza seu agente para a próxima versão secundária quando a versão secundária atual chega ao fim do suporte.

Atualizar manualmente a versão do mecanismo

Você pode atualizar a versão do mecanismo de um corretor usando a API Console de gerenciamento da AWS AWS CLI, a ou a API do Amazon MQ.

Console de gerenciamento da AWS

Para atualizar a versão do mecanismo de um corretor usando o Console de gerenciamento da AWS

1. Na página de detalhes do agente, selecione Edit (Editar).
2. Em Especificações, para Versão do mecanismo de agente escolha o novo número de versão na lista suspensa.
3. Role até o final da página e selecione Programar modificações.
4. Em Programar modificações do agente, para Quando aplicar modificações, escolha uma das seguintes opções.

   - Selecione After the next reboot (Depois da próxima reinicialização) se você quiser que o Amazon MQ conclua a atualização da versão durante a próxima janela de manutenção programada.
   - Selecione Imediatamente se você quiser reiniciar o agente e atualizar a versão do mecanismo imediatamente.

   > **Important**
   >
   > Os agentes de instância única ficarão offline durante a reinicialização. Para agentes de cluster, somente um nó fica inativo por vez enquanto o agente é reinicializado.

5. Selecione Apply (Aplicar) para concluir a aplicação das alterações.

AWS CLI

Para atualizar a versão do mecanismo de um corretor usando o AWS CLI

1. Usar o comando CLI update-broker e especifique os seguintes parâmetros, conforme mostrado no exemplo.

   - `--broker-id` — O ID exclusivo que o Amazon MQ gera para o agente. Você pode analisar o ID do ARN do seu agente. Por exemplo, considerando o seguinte ARN, `arn:aws:mq:us-`

`east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, o ID do agente seria `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.

- `--engine-version` — O número da versão do mecanismo para a qual o a atualização do mecanismo de agente será feita.

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

2. (Opcional) Use o comando da CLI reboot-broker para reinicializar seu broker se quiser atualizar a versão do mecanismo imediatamente.

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

Se você não quiser reiniciar seu agente e aplicar as alterações imediatamente, o Amazon MQ atualizará o agente durante a próxima janela de manutenção agendada.

> **Important**
>
> Os agentes de instância única ficarão offline durante a reinicialização. Em agentes de cluster, somente um nó fica inativo por vez enquanto o agente é reinicializado.

API do Amazon MQ

Para atualizar a versão do mecanismo de um agente usando a API do Amazon MQ

1. Use a operação de API UpdateBroker. Especifique `broker-id` como um parâmetro de caminho. Os exemplos a seguir pressupõem um agente na região `us-west-2`. Para ter mais informações sobre os endpoints do Amazon MQ disponíveis, consulte Endpoints e cotas do Amazon MQ na Referência geral da 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
```

Use o `engineVersion` na carga útil da solicitação para especificar o número da versão para a qual o agente será atualizado.

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

2. (Opcional) Use a operação de RebootBrokerAPI para reinicializar seu broker se quiser atualizar a versão do mecanismo imediatamente. `broker-id`é especificado como um parâmetro de caminho.

```
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
```

Se você não quiser reiniciar seu agente e aplicar as alterações imediatamente, o Amazon MQ atualizará o agente durante a próxima janela de manutenção agendada.

> **Important**
>
> Os agentes de instância única ficarão offline durante a reinicialização. Para agentes de cluster, somente um nó fica inativo por vez enquanto o agente é reinicializado.

Atualização de um tipo de instância do agente do Amazon MQ

> **Important**
>
> As instância de `mq.m7g.x` só estão disponíveis no Amazon MQ para agentes do RabbitMQ. Os agentes do Amazon MQ para ActiveMQ só usam as instância de `mq.m5.x`.

A descrição combinada da classe (`m7g`) e do tamanho (`large`) da instância do agente é chamada de tipo de instância de agente (por exemplo, `mq.m7g.large`). Quando se escolhe um tipo de instância, é importante considerar os fatores que afetarão o desempenho do agente:

- o número de clientes e filas
- o volume de mensagens enviadas

- mensagens mantidas na memória
- mensagens redundantes

Tipos menores de instância do agente (`mq.m7g.medium`) são recomendados somente para testar o desempenho da aplicação. Recomendamos tipos maiores de instância do agente (`mq.m7g.large` e superiores) para níveis de produção de clientes e filas, alto throughput, mensagens na memória e mensagens redundantes.

Recomendamos a atualização para um tipo de instância maior (ou seja, de `micro` para`large`) se você estiver enfrentando problemas de desempenho ou se estiver migrando de um ambiente de teste para um ambiente de produção. Para atualizar o tipo de instância, você pode usar o Console de gerenciamento da AWS, a AWS CLI ou a API do Amazon MQ.

Console de gerenciamento da AWS

Para atualizar para um tipo de instância maior usando oConsole de gerenciamento da AWS, faça o seguinte:

1. Faça login no console do Amazon MQ.
2. No painel de navegação à esquerda, selecione Brookers (Agentes) e depois escolha o agente que você deseja atualizar na lista.
3. Na página de detalhes do agente, selecione Edit (Editar).
4. Em Specifications (Especificações), para Broker instance type (Tipo de instância do agente), escolha o novo tipo de instância na lista suspensa.
5. Role até o final da página e selecione Programar modificações.
6. Em Programar modificações do agente, para Quando aplicar modificações, escolha uma das seguintes opções.

    - Selecione After the next reboot (Depois da próxima reinicialização) se você quiser que o Amazon MQ conclua a atualização durante a próxima janela de manutenção programada.
    - Selecione Immediately (Imediatamente) se você quiser reiniciar o agente e atualizar o tipo de instância imediatamente.

> **Important**
>
> Os agentes de instância única ficarão offline durante a reinicialização. Para agentes de cluster, somente um nó fica inativo por vez enquanto o agente é reinicializado.

7. Selecione Apply (Aplicar) para concluir a aplicação das alterações.

AWS CLI

Para atualizar o tipo de instância de um agente usando o AWS CLI

1. Usar o comando da CLI modify-broker e especifique os parâmetros a seguir, conforme mostrado no exemplo.
   - `--broker-id` — O ID exclusivo que o Amazon MQ gera para o agente.
   - `--host-instance-type` — O número da versão do mecanismo para a qual o a atualização do mecanismo de agente será feita.

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

2. (Opcional) Use o comando CLI reboot-broker para reiniciar o agente, se você quiser atualizar o tipo de instância imediatamente.

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

   Se você não quiser reiniciar seu agente e aplicar as alterações imediatamente, o Amazon MQ atualizará o agente durante a próxima janela de manutenção agendada.

> **Important**
>
> Os agentes de instância única ficarão offline durante a reinicialização. Para agentes de cluster, somente um nó fica inativo por vez enquanto o agente é reinicializado.

API do Amazon MQ

Para atualizar o tipo de instância de um agente usando a API do Amazon MQ

1. Use a operação da API UpdateBroker. Especifique `broker-id` como um parâmetro de caminho. Os exemplos a seguir pressupõem um agente na região `us-west-2`. Para ter mais informações sobre os endpoints do Amazon MQ disponíveis, consulte Endpoints e cotas do Amazon MQ na Referência geral da 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
   ```

   Use o `host-instance-type` na carga útil da solicitação para especificar o tipo de instância para o qual o agente será atualizado.

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

2. (Opcional) Use a operação de API RebootBroker para reiniciar seu agente, se você quiser atualizar a versão do mecanismo imediatamente. `broker-id` é especificado como um parâmetro de caminho.

   ```
   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
   ```

   Se você não quiser reiniciar seu agente e aplicar as alterações imediatamente, o Amazon MQ atualizará o agente durante a próxima janela de manutenção agendada.

   > **Important**
   >
   > Os agentes de instância única ficarão offline durante a reinicialização. Para agentes de cluster, somente um nó fica inativo por vez enquanto o agente é reinicializado.

Tipos de armazenamento do Amazon MQ para o ActiveMQ

O Amazon MQ para ActiveMQ é compatível com o Amazon Elastic File System (EFS) e o Amazon Elastic Block Store (EBS). Por padrão, os agentes do ActiveMQ usam o Amazon EFS para armazenamento do agente. Para aproveitar a alta durabilidade e a replicação em várias zonas de disponibilidade, use o Amazon EFS. Para aproveitar a baixa latência e alta taxa de transferência, use o Amazon EBS.

Important

- Você pode usar o Amazon EBS somente com a família `mq.m5` de tipo de instância de agente.
- Embora você possa alterar o tipo de instância de agente, você não pode alterar o tipo de armazenamento do agente depois de criar o agente.
- O Amazon EBS replica dados em uma única zona de disponibilidade e não é compatível com o modo de implantação ativo/em espera do ActiveMQ.

Diferenças entre tipos de armazenamento

A tabela a seguir fornece uma breve visão geral das diferenças entre os tipos de armazenamento em memória, do Amazon EFS e do Amazon EBS.

Tipo de armazenamento	Persistência	Exemplo de caso de uso	Número máximo aproximado de mensagens enfileiradas por produtor, por segundo (mensagem de 1 KB)	Replicação
Na memória	Não persistente	• Cotações de ações	5.000	Nenhum

Tipo de armazenamento	Persistência	Exemplo de caso de uso	Número máximo aproximado de mensagens enfileiradas por produtor, por segundo (mensagem de 1 KB)	Replicação
		• Atualizações de dados de localização • Dados alterados com frequência		
Amazon EBS	Persistente	• Grandes volumes de texto • Processamento de pedidos	500	Várias cópias em uma única zona de disponibilidade (AZ)
Amazon EFS	Persistente	Transações financeiras	80	Várias cópias em várias AZs

O armazenamento de mensagens na memória fornece a latência mais baixa e a taxa de transferência mais alta. No entanto, as mensagens são perdidas durante a substituição da instância ou a reinicialização do agente.

O Amazon EFS foi projetado para ser altamente durável, replicado em vários componentes AZs para evitar a perda de dados resultante da falha de um único componente ou de um problema que afete a disponibilidade de uma AZ. O Amazon EBS é otimizado para taxa de transferência e é replicado em vários servidores em uma única zona de disponibilidade.

Configuração de um agente privado do Amazon MQ

Um agente privado não tem acessibilidade pública nem pode ser acessado de fora da VPC. Antes de configurar um agente privado, veja as seguintes informações sobre VPCs sub-redes e grupos de segurança:

- VPCs
  - As sub-redes e os grupos de segurança de um agente devem estar na mesma VPC.
  - Se você estiver usando um agente privado, poderá ver endereços IP que você não configurou com a VPC. Esses são endereços IP da infraestrutura do Amazon MQ e não exigem nenhuma ação.
- Sub-redes
  - Se as sub-redes estiverem em uma VPC compartilhada, a VPC deverá pertencer à mesma conta que criou o agente.
  - Se nenhuma sub-rede for fornecida, as sub-redes padrão na VPC padrão serão usadas.
  - Depois que o agente é criado, as subredes usadas não podem ser alteradas.
  - Para clusters e active/standby corretores, as sub-redes devem estar em zonas de disponibilidade diferentes.
  - Quanto a agentes de instância única, você pode especificar qual sub-rede usar e o agente será criado na mesma zona de disponibilidade.
- Grupos de segurança
  - Os grupos de segurança padrão na VPC padrão serão usados se nenhum grupo de segurança for fornecido.
  - Instância única, cluster e active/standby corretores exigem pelo menos um grupo de segurança (por exemplo, o grupo de segurança padrão).

    > **Note**
    >
    > Os agentes públicos do RabbitMQ não usam sub-redes nem grupos de segurança.

  - Depois que o agente é criado, o grupo de segurança usado não pode ser alterado. Os próprios grupos de segurança ainda podem ser modificados.

Configurando um corretor privado no Console de gerenciamento da AWS

Para configurar um agente privado, comece a criar um novo corretor no Console de gerenciamento da AWS. Em seguida, na seção Configurações de rede, para configurar a conectividade do seu agente, faça o seguinte:

1. Escolha o Acesso privado para o agente. Para se conectar a um corretor privado, você pode usar IPv4 IPv6, ou dual-stack (IPv4 e). IPv6 Para obter mais informações, consulte Connecting to Amazon MQ.
2. Em seguida, escolha Usar a VPC, a(s) sub-rede(s) e o(s) grupo(s) de segurança padrão ou Selecionar a VPC, a(s) sub-rede(s) e o(s) grupo(s) de segurança existentes. Se você não quiser usar a VPC, as sub-redes ou os grupos de segurança padrão ou existentes, crie um(a) para se conectar ao agente privado.

> **Note**
>
> Para acesso de agente privado, o método de conexão será o mesmo do tipo de IP selecionado da sub-rede. Depois que o agente for criado, o endpoint da VPC não poderá ser alterado e terá sempre o tipo IP das sub-redes selecionadas. Se você deseja usar um tipo de IP novo, precisará criar um agente.

> **Note**
>
> O Amazon MQ para ActiveMQ não usa endpoints da VPC. Quando você cria um agente do ActiveMQ pela primeira vez, o Amazon MQ provisiona uma interface de rede elástica (ENI) na VPC. Os grupos de segurança são colocados no ENI e podem ser usados por agentes públicos e privados.

Acessar console web do agente do Amazon MQ sem acessibilidade pública

Quando você desativa a acessibilidade pública do seu corretor, o ID da AWS conta que criou o corretor pode acessar o corretor privado. Se você desabilitar a acessibilidade pública do agente, deverá executar as etapas a seguir para acessar o console web do agente.

1. Crie uma instância do EC2 do Linux em `public-vpc` (com um IP público, se necessário).

2. Para verificar se a VPC está configurada corretamente, estabeleça uma conexão `ssh` com a instância do EC2 e use o comando `curl` com o URI do seu agente.
3. Na sua máquina, crie um `ssh` túnel para a instância do EC2 usando o caminho para o seu arquivo de chave privada e o endereço IP de sua instância pública do EC2. Por exemplo:

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

   Um servidor de proxy de encaminhamento é iniciado em sua máquina.
4. Instale um cliente proxy, como FoxyProxyem sua máquina.
5. Configure seu cliente proxy usando as seguintes configurações:

   - Para o tipo de proxy, especifique `SOCKS5`.
   - Para o endereço IP, nome do DNS e nome de servidor, especifique `localhost`.
   - Para a porta, especifique `8080`.
   - Remova qualquer padrão de URL existente.
   - Para o modelo de URL, especifique `*.mq.*.amazonaws.com*`.
   - Para o tipo de conexão, especifique `HTTP(S)`.

   Quando você habilita seu cliente proxy, é possível acessar o Console da Web em sua máquina.

> **Important**
>
> Se estiver usando um agente privado, poderá ver endereços IP que você não configurou com sua VPC. Esses são endereços IP do RabbitMQ na infraestrutura do Amazon MQ e não exigem nenhuma ação.

Agendar a janela de manutenção para um agente do Amazon MQ

Periodicamente, o Amazon MQ realiza a manutenção do hardware, do sistema operacional ou do software do mecanismo de um agente de mensagens durante a janela de manutenção. Por exemplo, se você alterou o tipo de instância do agente, o Amazon MQ aplicará as alterações durante a próxima janela de manutenção programada. A manutenção pode durar até duas horas, dependendo das operações agendadas para o agente de mensagens. Você pode minimizar o tempo

de inatividade durante uma janela de manutenção selecionando um modo de implantação do agente com alta disponibilidade em várias zonas de disponibilidade (AZs).

O Amazon MQ para o ActiveMQ fornece implantações em modo ativo/em espera para alta disponibilidade. No modo ativo/em espera, o Amazon MQ executa operações de manutenção uma instância de cada vez, e pelo menos uma instância permanece disponível. Além disso, você pode configurar uma rede de agentes com janelas de manutenção espalhadas por toda a semana. O Amazon MQ para o RabbitMQ fornece implantações de cluster para alta disponibilidade. Em implantações de cluster, o Amazon MQ executa operações de manutenção um nó de cada vez ao manter pelo menos dois nós em execução o tempo todo.

Ao criar seu agente pela primeira vez, você pode programar a janela de manutenção para ocorrer uma vez por semana em um horário especificado. Você só pode ajustar a janela de manutenção de um agente até quatro vezes antes da próxima janela de manutenção programada. Quando uma janela de manutenção do agente é concluída, o Amazon MQ redefine o limite, e você pode ajustar a programação antes da próxima janela de manutenção. A disponibilidade do agente não é afetada ao ajustar sua janela de manutenção.

Para ajustar a janela de manutenção do agente, você pode usar o Console de gerenciamento da AWS, o AWS CLI ou a API do Amazon MQ.

Programar a janela de manutenção do agente usando o Console de gerenciamento da AWS

Para ajustar a janela de manutenção do agente usando o Console de gerenciamento da AWS

1. Faça login no console do Amazon MQ.
2. No painel de navegação à esquerda, selecione Brookers (Agentes) e depois escolha o agente que você deseja atualizar na lista.
3. Na página de detalhes do agente, selecione Edit (Editar).
4. Em Manutenção, faça o seguinte.

    a. Para Start day (Dia de início), escolha um dia da semana, por exemplo, Sunday (domingo), da lista suspensa.

    b. Para Start time (Hora de início), escolha o horário (horas e minutos) do dia para o qual deseja agendar a próxima janela de manutenção do agente, por exemplo, 12:00.

> **Note**
>
> As opções de Hora de início são configuradas no fuso horário UTC+0.

5. Em seguida, selecione Programar modificações. Depois, escolha Após a próxima reinicialização ou Imediatamente. Escolher After the next reboot (Depois da próxima reinicialização) atualizará imediatamente a janela de manutenção sem reinicializar o agente. Ao escolher Imediatamente, o agente será reiniciado de imediato.
6. Na página de detalhes do agente, em Maintenance window (Janela de manutenção), verifique se sua nova programação preferencial é exibida.

Programar a janela de manutenção do agente usando o AWS CLI

Para ajustar a janela de manutenção do agente usando o AWS CLI

1. Usar o comando CLI update-broker e especifique os seguintes parâmetros, conforme mostrado no exemplo.

   - `--broker-id` — O ID exclusivo que o Amazon MQ gera para o agente. Você pode analisar o ID do ARN do seu agente. Por exemplo, considerando o seguinte ARN, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, o ID do agente seria `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - `--maintenance-window-start-time` — Os parâmetros que determinam a hora de início da janela de manutenção semanal fornecida na estrutura a seguir.
     - `DayOfWeek` – O dia da semana, na sintaxe a seguir: `MONDAY| TUESDAY | WEDNESDAY | THURSDAY | FRIDAY | SATURDAY | SUNDAY`
     - `TimeOfDay` — A hora, no formato de 24 horas.
     - `TimeZone` — (Opcional) O fuso horário, no formato País/Cidade ou no formato de deslocamento de UTC. Definido como UTC por padrão.

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

2. (Opcional) Use o comando CLI describe-broker para verificar se a janela de manutenção foi atualizada com sucesso.

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

Programar a janela de manutenção do agente usando a API do Amazon MQ

Para ajustar a janela de manutenção do agente usando a API do Amazon MQ

1. Use a operação da API UpdateBroker. Especifique `broker-id` como um parâmetro de caminho. Os exemplos a seguir pressupõem um agente na região `us-west-2`. Para ter mais informações sobre os endpoints do Amazon MQ disponíveis, consulte Endpoints e cotas do Amazon MQ na Referência geral da 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
```

   Use o parâmetro `maintenanceWindowStartTime` e o tipo de recurso `WeeklyStartTime` na carga útil da solicitação.

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

2. (Opcional) Use a operação DescribeBroker da API para verificar se a janela de manutenção foi atualizada com sucesso. `broker-id` é especificado como um parâmetro de caminho.

```
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
```

Reinicializar um agente do Amazon MQ

Para aplicar uma nova configuração a um agente, você pode reinicializá-lo.

> **Note**
>
> Se o agente do ActiveMQ não responder, você poderá reinicializá-lo para fazer a recuperação de um estado com defeito.

O exemplo a seguir mostra como reiniciar um agente do Amazon MQ utilizando o Console de gerenciamento da AWS.

Para reinicializar um agente do Amazon MQ

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, escolha o nome do seu corretor (por exemplo, MyBroker).
3. Na ***MyBroker***página, escolha Ações, Reinicialize o agente.

   > **Important**
   >
   > Os agentes de instância única ficarão offline durante a reinicialização. Os agentes de cluster estarão disponíveis, mas os nós serão reinicializados um por vez.

4. Na caixa de diálogo Reboot broker, escolha Reboot.

   A reinicialização de um operador leva cerca de 5 minutos. Se a reinicialização incluir alterações no tamanho da instância ou for executada em um agente com alta profundidade de fila, o processo de reinicialização poderá levar mais tempo.

Excluindo um agente do Amazon MQ

Se você não usa um agente do Amazon MQ (e não prevê usá-lo em um futuro próximo), é uma prática recomendada excluí-lo do Amazon MQ para reduzir seus custos. AWS

O exemplo a seguir mostra como excluir um agente utilizando o Console de gerenciamento da AWS.

Excluindo um agente do Amazon MQ

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, selecione seu corretor (por exemplo MyBroker) e escolha Excluir.
3. No Delete***MyBroker***? caixa de diálogo, digite `delete` e escolha Excluir.

   A exclusão de um agente leva cerca de 5 minutos.

Status do agente do Amazon MQ

A condição atual do agente é indicada por um status. A tabela a seguir lista os status de um agente Amazon MQ.

Console	solicitações de	Description
Falha na criação	`CREATION_FAILED`	Não foi possível criar o agente.
Criação em andamento	`CREATION_IN_PROGRESS`	O agente está sendo criado no momento.
Exclusão em andamento	`DELETION_IN_PROGRESS`	O agente está sendo excluído no momento.
Reinicialização em andamento	`REBOOT_IN_PROGRESS`	O agente está sendo reinicial izado no momento.
Executar	`RUNNING`	O agente está funcionando.
Ação crítica obrigatória	`CRITICAL_ACTION_RE QUIRED`	O agente está em execução, mas se encontra em um estado degradado e exige ação imediata. Você pode encontrar instruções para resolver o problema seleciona ndo o código de ação

Console	solicitações de	Description
		necessário na lista em Solução de problemas.

Adicionar tags aos recursos do Amazon MQ

Para organizar e identificar seus recursos do Amazon MQ para alocação de custo, você pode adicionar etiquetas de metadados que identificam um objetivo ou configuração de um agente. Isso é especialmente útil quando você tem vários agentes. Você pode usar tags de alocação de custos para organizar sua fatura da AWS para refletir sua própria estrutura de custos. Para isso, cadastre-se para obter a fatura da sua conta da AWS para incluir as chaves e valores das tags. Para obter mais informações, consulte Configuração de um relatório de alocação de custos mensal no Manual do usuário do AWS Billing.

Por exemplo, você pode adicionar etiquetas que representam o centro de custos e o objetivo dos seus recursos do Amazon MQ:

Recurso	Chave	Valor
`Broker1`	`Cost Center`	`34567`
	`Stack`	`Production`
`Broker2`	`Cost Center`	`34567`
	`Stack`	`Production`
`Broker3`	`Cost Center`	`12345`
	`Stack`	`Development`

Esse esquema de marcação permite que você agrupe dois agentes executando tarefas relacionadas no mesmo centro de custo e, ao mesmo tempo, etiquetar um agente não relacionado com outra etiqueta de alocação de custo.

Adicionar tags no console do Amazon MQ

Você pode adicionar rapidamente tags aos recursos que estiver criando no console do Amazon MQ seguindo estas etapas:

1. Na página Criar um agente, selecione Configurações adicionais.
2. Em Tags, selecione Adicionar tag.
3. Insira um par de chave e valor.
4. (Opcional) Selecione Adicionar tag para adicionar várias tags ao agente.
5. Selecione Criar agente.

Para adicionar tags ao criar uma configuração:

1. Na página Criar configuração, selecione Avançado.
2. Em Tags na página Criar configuração, selecione Adicionar tag.
3. Insira um par de chave e valor.
4. (Opcional) Selecione Adicionar tag para adicionar várias tags à sua configuração.
5. Selecione Criar configuração.

Depois de adicionar tags, você pode visualizar, editar e remover as tags dos seus recursos no console do Amazon MQ. Você também pode visualizar as tags dos seus recursos usando a API REST. Para obter mais informações, consulte Referência de API REST do Amazon MQ.

Usar o Amazon MQ para ActiveMQ

O Amazon MQ facilita a criação de um agente de mensagem com os recursos de processamento e armazenamento que atendem às suas necessidades. Você pode criar, gerenciar e excluir agentes usando o Console de gerenciamento da AWS, a API REST da Amazon MQ, ou a AWS Command Line Interface.

Os agentes do Amazon MQ para ActiveMQ podem ser implantados como agentes de instância única ou agentes em modo ativo/em espera. Para ambos os modos de implantação, o Amazon MQ oferece alta durabilidade armazenando seus dados de forma redundante.

> **Note**
>
> O Amazon MQ usa o Apache KahaDB como seu armazenamento de dados. Outros armazenamentos de dados, como JDBC e LevelDB, não são compatíveis.

Você pode acessar seus agentes usando qualquer linguagem de programação compatível com o ActiveMQ e habilitando o TLS explicitamente para os seguintes protocolos:

- AMQP
- MQTT
- MQTT pelo WebSocket
- OpenWire
- STOMP
- STOMP pelo WebSocket

Para saber mais sobre as APIs do Amazon MQ REST, consulte a Referência da API REST do Amazon MQ.

Agentes do Amazon MQ para ActiveMQ

O que é um agente do Amazon MQ para ActiveMQ?

Um agente é um ambiente de agente de mensagens em execução no Amazon MQ. É o bloco de criação básico do Amazon MQ. A descrição combinada da classe (`m5`) e do tamanho (`large`,

`medium`) da instância do agente é um tipo de instância de agente (por exemplo,`mq.m5.large`). Para obter mais informações, consulte Broker instance types.

- Um agente de instância única é composto por um agente em uma Zona de disponibilidade. O agente se comunica com sua aplicação e com um volume de armazenamento do Amazon EBS ou Amazon EFS.
- Uma agente ativo/em espera é composto por dois agentes em duas zonas de disponibilidade diferentes, configuradas em um Par redundante. Esses agentes se comunicam de forma síncrona com sua aplicação e com o Amazon EFS.

Para obter mais informações, consulte Opções de implantação de agentes do Amazon MQ para ActiveMQ.

É possível habilitar as atualizações secundárias de versão automáticas para novas versões secundárias do mecanismo de agente à medida que o Apache lança novas versões. Atualizações automáticas ocorrem durante a janela de manutenção definida pelo dia da semana, a hora do dia (no formato de 24 horas) e o fuso horário (UTC, por padrão).

Para obter informações sobre a criação e o gerenciamento de agentes, consulte o seguinte:

- Conceitos básicos: criar e conectar a um agente do ActiveMQ
- Operadores
- Broker statuses

Protocolos de nível de conexão compatíveis

Você pode acessar seus agentes usando qualquer linguagem de programação compatível com o ActiveMQ e habilitando o TLS explicitamente para os seguintes protocolos:

- AMQP
- MQTT
- MQTT pelo WebSocket
- OpenWire
- STOMP
- STOMP pelo WebSocket

Atributos

Um agente ActiveMQ tem vários atributos, por exemplo:

- Um nome (`MyBroker`)
- Um ID (`b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Um Nome do Recurso da Amazon (ARN) (`arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Uma URL do Console da Web ActiveMQ (`https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8162`)

  Para obter mais informações, consulte o console da Web na documentação do Apache ActiveMQ.

  > **Important**
  >
  > Se você especificar um mapa de autorização que não inclua o `activemq-webconsole`, você não poderá usar o Console da Web do ActiveMQ porque o grupo não estará autorizado a enviar mensagens ou receber mensagens do agente do Amazon MQ.

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

    > **Note**
    >
    > Este é um endpoint do OpenWire.

  - `stomp+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61614`
  - `wss://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61619`

Para obter mais informações, consulte Configuração de transportes na documentação do Apache ActiveMQ.

> **Note**
>
> Para um agente ativo/em espera, o Amazon MQ fornece duas URLs do Console da Web do ActiveMQ, mas apenas uma URL está ativo de cada vez. Da mesma forma, o Amazon MQ fornece dois endpoints para cada protocolo de nível de conexão, mas apenas um endpoint está ativo em cada par de cada vez. Os sufixos `-1` e `-2` denotam um par redundante.

Para obter uma lista completa de atributos do agente, consulte o seguinte na Referência de API Amazon MQ REST:

- ID da operação REST: Agente
- ID da operação REST: Agentes
- ID da operação REST: Reinicialização do agente

usuários do agente

Um usuário do ActiveMQ é uma pessoa ou uma aplicação que pode acessar as filas e tópicos de um agente ActiveMQ. Você pode configurar usuários para que tenham permissões específicas. Por exemplo, é possível permitir que alguns usuários acessem o Console da Web ActiveMQ.

Um grupo é um rótulo semântico. Você pode atribuir um grupo a um usuário e configurar permissões para grupos para enviar, receber e administrar filas e tópicos específicos.

> **Important**
>
> Fazer alterações em um usuário não aplica as alterações ao usuário imediatamente. Para aplicar as alterações, você deve aguardar a próxima janela de manutenção ou reiniciar o agente.

Para obter informações sobre usuários e grupos, consulte a documentação do Apache ActiveMQ a seguir:

- Autorização
- Exemplo de autorização

Para obter informações sobre a criação, edição e exclusão de usuários do ActiveMQ, consulte o seguinte:

- Criar um usuário do agente do ActiveMQ
- Usuários

Atributos de usuário

Para obter uma lista completa de atributos do usuário, consulte o seguinte na Referência de API Amazon MQ REST:

- ID da operação REST: Usuário
- ID da operação REST: Usuários

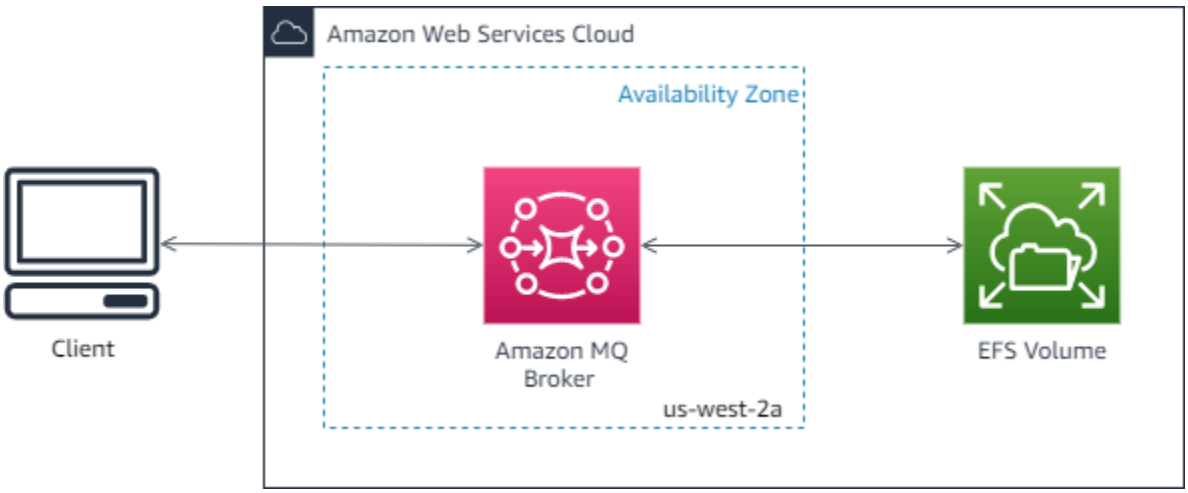
Opções de implantação de agentes do Amazon MQ para ActiveMQ

O Amazon MQ oferece opções de implantação de instância única e de cluster para os agentes.

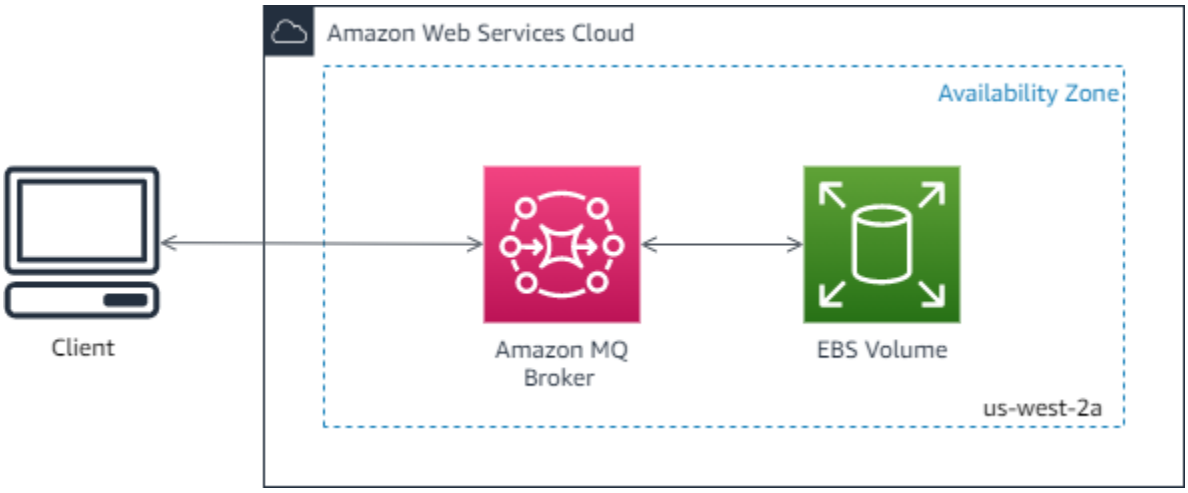
Opção 1: agentes de instância única do Amazon MQ

Um agente de instância única é composto por um agente em uma Zona de disponibilidade. O agente se comunica com sua aplicação e com um volume de armazenamento do Amazon EBS ou Amazon EFS. Os volumes de armazenamento do Amazon EFS foram projetados para fornecer o mais alto nível de durabilidade e disponibilidade, armazenando dados de forma redundante em várias zonas de disponibilidade ()AZs. O Amazon EBS fornece armazenamento em nível de bloco otimizado para baixa latência e alta taxa de transferência. Para obter mais informações sobre opções de armazenamento, consulte Storage.

O diagrama a seguir ilustra um agente de instância única com armazenamento Amazon EFS replicado em várias. AZs



O diagrama a seguir ilustra um agente de instância única com armazenamento do Amazon EBS replicado em vários servidores em uma única zona de disponibilidade.



Opção 2: active/standby corretores Amazon MQ para alta disponibilidade

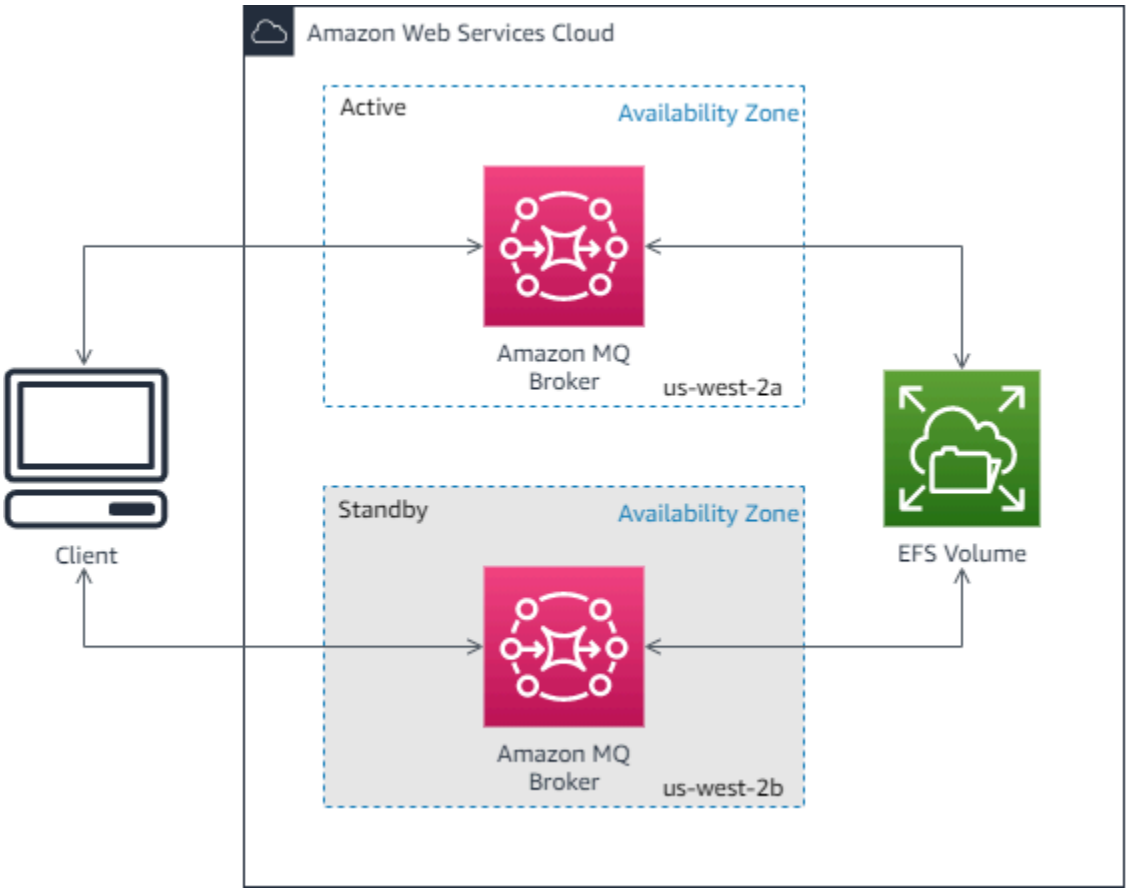
Uma agente ativo/em espera é composto por dois agentes em duas zonas de disponibilidade diferentes, configuradas em um Par redundante. Esses agentes se comunicam de forma síncrona com sua aplicação e com o Amazon EFS. Os volumes de armazenamento do Amazon EFS foram projetados para fornecer o mais alto nível de durabilidade e disponibilidade, armazenando dados de forma redundante em várias zonas de disponibilidade ()AZs. Para obter mais informações, consulte Storage.

Geralmente, apenas uma das instâncias do agente está sempre ativa, enquanto a outra está em espera. Se uma das instâncias do agente apresentar um defeito ou for submetida à manutenção, o Amazon MQ levará pouco tempo para tirar de serviço a instância inativa. Isso permite que a instância em espera saudável se torne ativa e comece a aceitar comunicações recebidas. As janelas de

manutenção e as reinicializações de agente iniciadas por você farão com que ocorra um failover. Quando você reinicia um operador, o failover leva apenas alguns segundos.

Para um active/standby corretor, o Amazon MQ fornece dois ActiveMQ Web Console URLs, mas somente um URL está ativo por vez. Da mesma forma, o Amazon MQ fornece dois endpoints para cada protocolo de nível de conexão, mas apenas um endpoint está ativo em cada par de cada vez. Os sufixos `-1` e `-2` denotam um par redundante. Para endpoints de protocolo no nível da conexão, você pode permitir que sua aplicação se conecte a qualquer endpoint usando o Transporte de failover.

O diagrama a seguir ilustra um active/standby agente com armazenamento Amazon EFS replicado em vários. AZs



Rede de agentes do Amazon MQ

O Amazon MQ é compatível com o recurso da rede de agentes do ActiveMQ.

Uma rede de corretores é composta por vários corretores ou corretores de instância única ativos simultaneamente. active/standby Criar uma rede de agentes pode aumentar a disponibilidade, a tolerância a falhas e o balanceamento de carga com várias instâncias de agentes.

Como funciona uma rede de agentes?

Uma rede de agentes é estabelecida por meio da conexão de um agente com outro usando-se os conectores de rede. Um conector de rede envia mensagens sob demanda de um agente para outro. Os conectores de rede são definidos na configuração do agente como conexões non-duplex ou duplex. Para conexões não duplex, as mensagens são encaminhadas apenas de um agente para o outro. Em conexões duplex, as mensagens são encaminhadas nos dois sentidos entre os dois agentes.

Se o conector de rede for configurado como duplex, as mensagens também serão encaminhadas do Broker2 para o Broker1.

Você pode usar conexões não duplex e duplex em uma rede de agentes. Talvez você queira introduzir uma conexão duplex com outro agente para melhorar o tráfego ou evitar um aumento de limite. As conexões duplex também são úteis para a migração parcial de agentes gerenciados on-premises para o Amazon MQ.

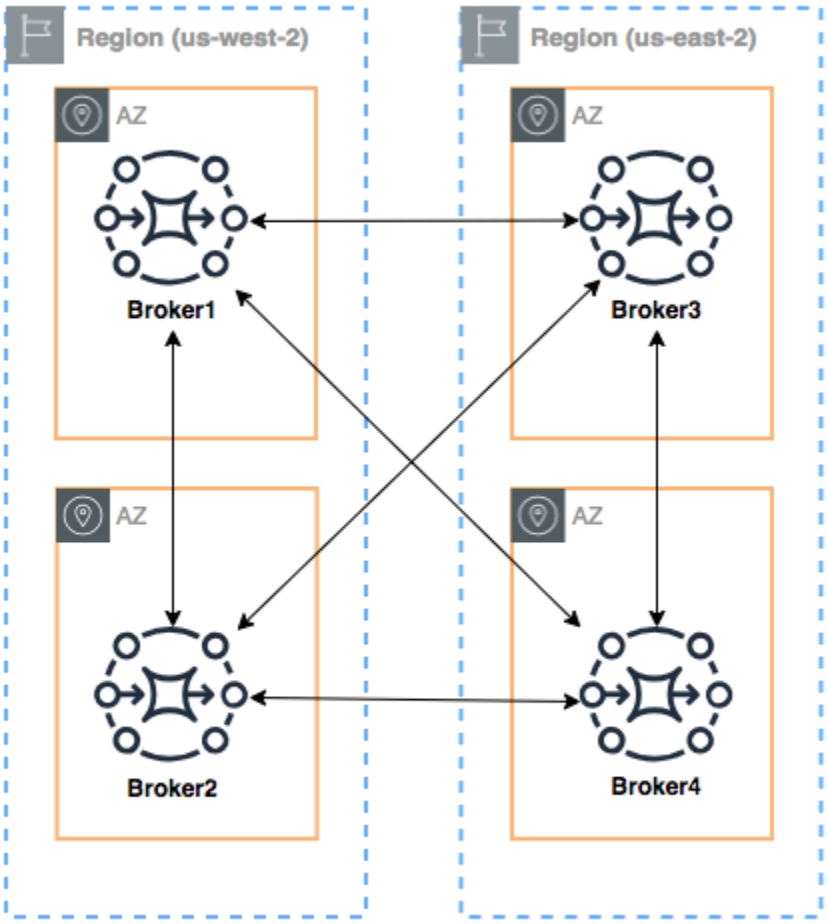
Como uma rede de agentes lida com as credenciais?

Para o agente A se conectar ao agente B em uma rede, o agente A deve usar credenciais válidas, como qualquer outro produtor ou consumidor. Em vez de fornecer uma senha em uma configuração do `<networkConnector>` do agente A, você deve primeiro criar um usuário no agente A com os mesmos valores como outro usuário no agente B (esses são usuários separados e exclusivos que compartilham os mesmos valores de nome de usuário e senha). Quando você especifica o atributo `userName` na configuração do `<networkConnector>`, o Amazon MQ adicionará a senha automaticamente no tempo de execução.

> **Important**
>
> Não especifique o atributo `password` para o `<networkConnector>`. Não recomendamos armazenar senhas em texto simples nos arquivos de configuração do agente, porque isso torna as senhas visíveis no console do Amazon MQ. Para obter mais informações, consulte Configure Network Connectors for Your Broker.

Dentro da região

Para configurar uma rede de corretores que abranja AWS regiões, implante corretores nessas regiões e configure conectores de rede para os endpoints desses corretores.



Para configurar uma rede de agentes, como nesse exemplo, você pode adicionar entradas do `networkConnectors` para as configurações do Broker1 e do Broker4 que fazem referência a endpoints de nível de conexão desses agentes.

Conectores de rede para o Broker1:

```
<networkConnectors>
    <networkConnector name="1_to_2" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)"/>
    <networkConnector name="1_to_3" userName="myCommonUser" duplex="true"
```

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

Conector de rede para o Broker2:

```
<networkConnectors>
    <networkConnector name="2_to_3" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Conectores de rede para o Broker4:

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

Failover dinâmico com conectores de transporte

Além de configurar elementos `networkConnector`, você pode configurar as opções `transportConnector` do agente para habilitar o failover dinâmico e para rebalancear as conexões quando os agentes são adicionados ou removidos da rede.

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

Nesse exemplo, tanto `updateClusterClients` como `rebalanceClusterClients` estão definidos como `true`. Nesse caso, os clientes receberão uma lista de agentes da rede e solicitarão que eles façam um rebalanceamento se um novo agente ingressar.

Opções disponíveis:

- `updateClusterClients`: transmite informações aos clientes sobre alterações na rede de topologia do agente.
- `rebalanceClusterClients`: faz com que os clientes realizem um rebalanceamento em todos os agentes quando um agente novo é adicionado a uma rede de agentes.
- `updateClusterClientsOnRemove`: atualiza os clientes com informações sobre topologia quando um agente sai de uma rede de agentes.

Quando `updateClusterClients` é definido como true (verdadeiro), os clientes podem ser configurados para se conectarem a um único agente em uma rede de agentes.

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

Quando um novo corretor se conectar, ele receberá uma lista URIs de todos os corretores da rede. Se a conexão com o agente falhar, ela poderá trocar de maneira dinâmica para um dos agentes fornecidos no momento da conexão.

Para obter mais informações sobre failover, consulte Opções do lado do agente para failover na documentação do ActiveMQ.

Tipos de instância do Amazon MQ para agentes do ActiveMQ

A descrição combinada da classe (`m5`) e do tamanho (`large`, `medium`) da instância do agente é um tipo de instância de agente (por exemplo,`mq.m5.large`). A tabela a seguir lista os tipos de instância do agente do Amazon MQ disponíveis para os agentes do ActiveMQ.

O Amazon MQ avisa com pelo menos 90 dias de antecedência quando um tipo de instância chegará ao fim do suporte. Recomendamos atualizar o agente para um novo tipo de instância antes da data do fim do suporte para evitar interrupções.

> **Important**
>
> Você não pode criar agentes em `t2.micro` ou `mq.m4.large` após 17 de março de 2025.

Tipo de instância	vCPU	Memória (GiB)	Uso recomendado	Armazenamento	Fim do suporte no Amazon MQ
`mq.t3.micro`	2	1	Avaliação	EFS	
`mq.m5.large`	2	8	Produção	EFS ou EBS	
`mq.m5.xlarge`	4	16	Produção	EFS ou EBS	
`mq.m5.2xlarge`	8	32	Produção	EFS ou EBS	
`mq.m5.4xlarge`	16	64	Produção	EFS ou EBS	

Para obter mais informações sobre considerações em relação à taxa de transferência, consulte Selecionar o tipo de instância de agente correto para obter a melhor taxa de transferência.

Configurações do agente do Amazon MQ para ActiveMQ

Uma configuração contém todas as definições do agente do ActiveMQ no formato XML (semelhante ao arquivo `activemq.xml` do ActiveMQ). Você pode criar uma configuração antes de criar qualquer agente. Em seguida, você pode aplicar a configuração a um ou mais agentes.

> **Important**
>
> Fazer alterações em uma configuração não aplica as alterações ao agente imediatamente. Para aplicar as alterações, você deve aguardar a próxima janela de manutenção ou reiniciar o agente.
>
> Você só pode excluir uma configuração usando a API do `DeleteConfiguration`. Para obter mais informações, consulte Configurações na Referência da API do Amazon MQ.

Atributos

A configuração de um agente tem vários atributos, por exemplo:

- Um nome (`MyConfiguration`)
- Um ID (`c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Um Nome do Recurso da Amazon (ARN) (`arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)

Para obter uma lista completa de atributos de configuração, consulte o seguinte na Referência de API Amazon MQ REST:

- ID da operação REST: Configuração
- ID da operação REST: Configurações

Para obter uma lista completa de atributos de revisão de configuração, consulte o seguinte:

- ID da operação REST: Revisão da configuração
- ID da operação REST: Revisões de configuração

Usar arquivos de configuração XML do Spring

Os agentes do ActiveMQ são configurados usando arquivos XML do Spring. É possível configurar vários aspectos do agente do ActiveMQ, como destinos pré-definidos, políticas de destino, políticas de autorização e plugins. O Amazon MQ controla alguns desses elementos de configuração, como transportes de rede e armazenamento. Outras opções de configuração, como a criação de redes de agentes, não são compatíveis atualmente.

O conjunto completo de opções de configuração compatíveis é especificado nos esquemas XML do Amazon MQ. Faça download de arquivos zip dos esquemas compatíveis usando os links a seguir.

- `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

Esses esquemas podem ser usados para validar e limpar seus arquivos de configuração. O Amazon MQ também permite que você forneça configurações enviando arquivos XML. Ao carregar um arquivo XML, o Amazon MQ limpa e remove automaticamente parâmetros de configuração inválidos e proibidos de acordo com o esquema.

> **Note**
>
> Para os atributos, é possível usar apenas valores estáticos. O Amazon MQ limpa elementos e atributos que contenham expressões, variáveis e referências de elementos do Spring da configuração que você fez.

Criar uma configuração do agente do Amazon MQ para ActiveMQ

Uma configuração contém todas as configurações do agente do ActiveMQ no formato XML (semelhante ao arquivo `activemq.xml` do ActiveMQ). Você pode criar uma configuração antes de criar qualquer agente. Em seguida, você pode aplicar a configuração a um ou mais agentes. As configurações podem ser aplicadas imediatamente ou durante uma janela de manutenção.

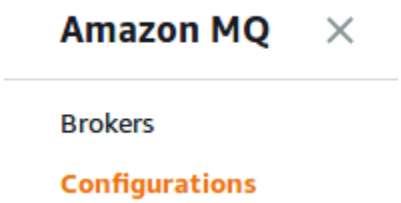
O exemplo a seguir mostra como criar e aplicar uma configuração de agente do Amazon MQ utilizando o Console de gerenciamento da AWS.

> **Important**
>
> Você só pode excluir uma configuração usando a API do `DeleteConfiguration`. Para obter mais informações, consulte Configurações na Referência da API do Amazon MQ.

Criar uma configuração

Para criar uma configuração do agente, primeiro crie a configuração.

1. Faça login no console do Amazon MQ.
2. Do lado esquerdo, expanda o painel de navegação e selecione Configurations (Configurações).



3. Na página Configurações, selecione Criar configuração.
4. Na página Criar configuração, na seção Detalhes, digite o Nome da configuração (por exemple, `MyConfiguration`) e selecione uma versão do Mecanismo do agente.

> **Note**
>
> Para saber mais sobre as versões do mecanismo ActiveMQ compatíveis com o Amazon MQ para ActiveMQ, consultethe section called “Gerenciamento de versão”.

5. Escolha Criar configuração.

Criar uma revisão de configuração

Depois de criar uma configuração do agente, você precisará editá-la usando uma revisão da configuração.

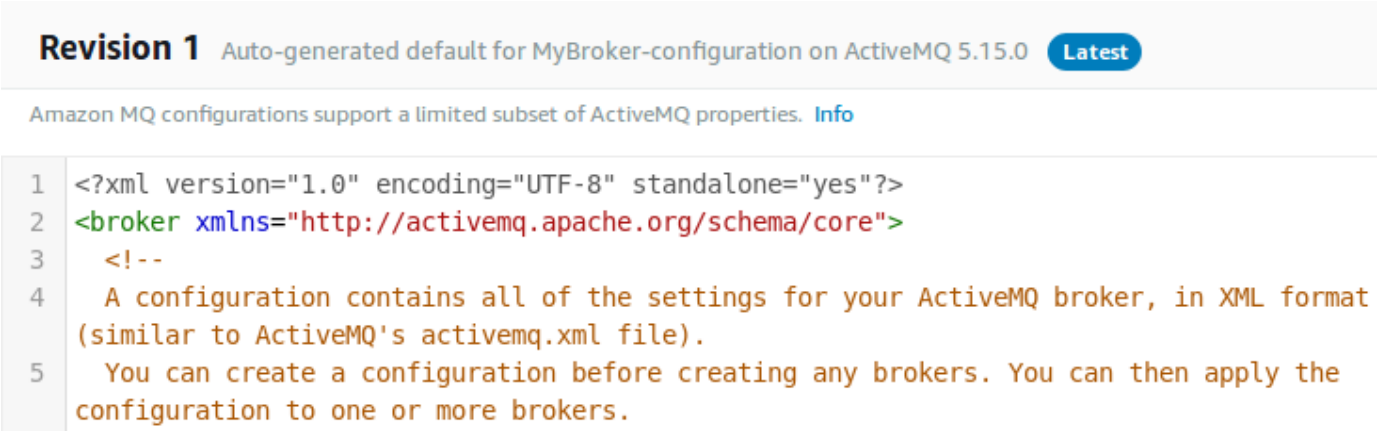
1. Na lista de configuração, escolha ***`MyConfiguration`***.

> **Note**
>
> A primeira revisão de configuração será sempre criada para você quando o Amazon MQ criar a configuração.

Na ***`MyConfiguration`***página, o tipo e a versão do broker engine que sua nova revisão de configuração usa (por exemplo, Apache ActiveMQ 5.15.16) são exibidos.

2. Na guia Configuration details (Detalhes da configuração), são exibidos o número de revisão da configuração, a descrição e a configuração do agente no formato XML.

> **Note**
>
> Editar a configuração atual irá criar uma nova revisão da configuração.



3. Selecione Edit configuration (Editar configuração) e faça as alterações na configuração XML.
4. Escolha Salvar.

   A caixa e diálogo Save revision (Salvar revisão) será exibida.
5. (Opcional) Tipo `A description of the changes in this revision`.
6. Escolha Salvar.

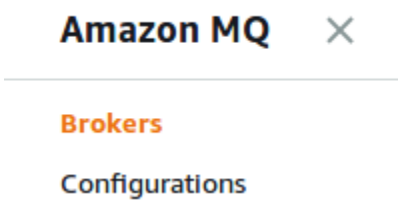
   A nova revisão da configuração é salva.

> **Important**
>
> O console do Amazon MQ limpa automaticamente parâmetros de configuração inválidos e proibidos de acordo com um esquema. Para obter mais informações e uma lista completa dos parâmetros XML permitidos, consulte Amazon MQ Broker Configuration Parameters.

Aplicar uma revisão de configuração ao operador

Depois de revisar a configuração, você pode aplicá-la ao agente.

1. Do lado esquerdo, expanda o painel de navegação e selecione Brokers (Agentes).



2. Na lista de corretores, selecione seu corretor (por exemplo MyBroker) e escolha Editar.
3. Na *MyBroker* página Editar, na seção Configuração, selecione uma Configuração e uma Revisão e, em seguida, escolha Programar Modificações.
4. Na seção Schedule broker modifications (Programar modificações no operador), escolha se deseja aplicar as modificações During the next scheduled maintenance window (Durante a próxima janela de manutenção programada) ou Immediately (Imediatamente).

> **Important**
>
> Os agentes de instância única ficarão offline durante a reinicialização. Para agentes de cluster, somente um nó fica inativo por vez enquanto o agente é reinicializado.

5. Escolha Aplicar.

   Sua revisão de configuração será aplicada ao agente no horário especificado.

Editar uma revisão de configuração do Amazon MQ para o ActiveMQ

Recomendamos que você edite uma revisão de configuração depois de aplicá-la ao agente. Use as instruções a seguir para editar uma revisão de configuração.

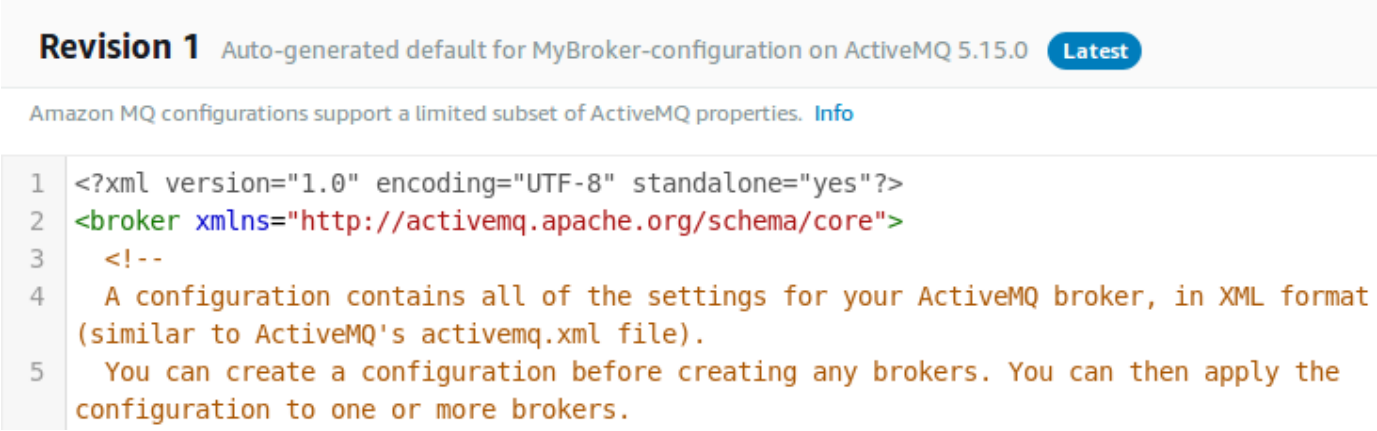
1. Faça login no console do Amazon MQ.
2. Na lista de corretores, selecione seu corretor (por exemplo MyBroker) e escolha Editar.
3. Na ***MyBroker***página, escolha Editar.
4. Na *MyBroker* página Editar, na seção Configuração, selecione uma Configuração e uma Revisão e escolha Editar.

> **Note**
>
> A menos que você selecione uma configuração ao criar um agente, a primeira revisão de configuração será sempre criada para você quando o Amazon MQ criar o agente.

Na ***MyBroker***página, o tipo e a versão do mecanismo do broker que a configuração usa (por exemplo, Apache ActiveMQ 5.15.8) são exibidos.

5. Na guia Configuration details (Detalhes da configuração), são exibidos o número de revisão da configuração, a descrição e a configuração do agente no formato XML.

> **Note**
>
> Editar a configuração atual irá criar uma nova revisão da configuração.



6. Selecione Edit configuration (Editar configuração) e faça as alterações na configuração XML.
7. Escolha Salvar.

   A caixa e diálogo Save revision (Salvar revisão) será exibida.
8. (Opcional) Tipo `A description of the changes in this revision`.
9. Escolha Salvar.

   A nova revisão da configuração é salva.

> **Important**
>
> O console do Amazon MQ limpa automaticamente parâmetros de configuração inválidos e proibidos de acordo com um esquema. Para obter mais informações e uma lista completa dos parâmetros XML permitidos, consulte Amazon MQ Broker Configuration Parameters.

Elementos permitidos nas configurações do Amazon MQ

Veja a seguir uma lista detalhada dos elementos permitidos nas configurações do Amazon MQ. Para obter mais informações, consulte Configuração de XML na documentação do Apache ActiveMQ.

Elemento
`abortSlowAckConsumerStrategy` (atributos)
`abortSlowConsumerStrategy` (atributos)
`authorizationEntry` (atributos)
`authorizationMap` (elementos de conjunto de filhos)
`authorizationPlugin` (elementos de conjunto de filhos)
`broker` (atributos | elementos de conjunto de filhos)
`cachedMessageGroupMapFactory` (atributos)
`compositeQueue` (atributos | elementos de conjunto de filhos)
`compositeTopic` (atributos | elementos de conjunto de filhos)
`constantPendingMessageLimitStrategy` (atributos)
`discarding` (atributos)
`discardingDLQBrokerPlugin` (atributos)
`fileCursor`
`fileDurableSubscriberCursor`
`fileQueueCursor`
`filteredDestination` (atributos)
`fixedCountSubscriptionRecoveryPolicy` (atributos)

Elemento
`fixedSizedSubscriptionRecoveryPolicy` (atributos)
`forcePersistencyModeBrokerPlugin` (atributos)
`individualDeadLetterStrategy` (atributos)
`lastImageSubscriptionRecoveryPolicy`
`messageGroupHashBucketFactory` (atributos)
`mirroredQueue` (atributos)
`noSubscriptionRecoveryPolicy`
`oldestMessageEvictionStrategy` (atributos)
`oldestMessageWithLowestPriorityEvictionStrategy` (atributos)
`policyEntry` (atributos | elementos de conjunto de filhos)
`policyMap` (elementos de conjunto de filhos)
`prefetchRatePendingMessageLimitStrategy` (atributos)
`priorityDispatchPolicy`
`priorityNetworkDispatchPolicy`
`queryBasedSubscriptionRecoveryPolicy` (atributos)
`queue` (atributos)
`redeliveryPlugin` (atributos | elementos de conjunto de filhos)
`redeliveryPolicy` (atributos)
`redeliveryPolicyMap` (elementos de conjunto de filhos)
`retainedMessageSubscriptionRecoveryPolicy` (elementos de conjunto de filhos)

Elemento
`roundRobinDispatchPolicy`
`sharedDeadLetterStrategy` (atributos | elementos de conjunto de filhos)
`simpleDispatchPolicy`
`simpleMessageGroupMapFactory`
`statisticsBrokerPlugin`
`storeCursor`
`storeDurableSubscriberCursor` (atributos)
`strictOrderDispatchPolicy`
`tempDestinationAuthorizationEntry` (atributos)
`tempQueue` (atributos)
`tempTopic` (atributos)
`timedSubscriptionRecoveryPolicy` (atributos)
`timeStampingBrokerPlugin` (atributos)
`topic` (atributos)
`transportConnector` (atributos)
`uniquePropertyMessageEvictionStrategy` (atributos)
`virtualDestinationInterceptor` (elementos de conjunto de filhos)
`virtualTopic` (atributos)
`vmCursor`
`vmDurableCursor`

Elemento
vmQueueCursor

Elementos e atributos permitidos nas configurações do Amazon MQ

A seguinte é uma lista detalhada dos elementos e de seus atributos permitidos nas configurações do Amazon MQ. Para obter mais informações, consulte Configuração de XML na documentação do Apache ActiveMQ.

Elemento	Atributo
abortSlowAckConsumerStrategy	abortConnection
	checkPeriod
	ignoreIdleConsumers
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	maxTimeSinceLastAck
	name
abortSlowConsumerStrategy	abortConnection
	checkPeriod
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	name

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

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

Elemento	Atributo
	`useVirtualTopics`
`cachedMessageGroupMapFactory`	`cacheSize`
`compositeQueue`	`concurrentSend`
	`copyMessage`
	`forwardOnly`
	`name`
	`sendWhenNotMatched`
`compositeTopic`	`concurrentSend`
	`copyMessage`
	`forwardOnly`
	`name`
	`sendWhenNotMatched`
conditionalNetworkBridgeFilterFactory	`rateDuration`
	`rateLimit`
	`replayDelay`
	`replayWhenNoConsumers`
	`selectorAware` Compatível com Apache ActiveMQ 5.16.x

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

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

Elemento	Atributo
oldestMessageWithLowestPriorityEvictionStrategy	evictExpiredMessagesHighWatermark
policyEntry	advisoryForConsumed
	advisoryForDelivery
	advisoryForDiscardingMessages
	advisoryForFastProducers
	advisoryForSlowConsumers
	advisoryWhenFull
	allConsumersExclusiveByDefault
	alwaysRetroactive
	blockedProducerWarningInterval
	consumersBeforeDispatchStarts
	cursorMemoryHighWaterMark
	doOptimzeMessageStorage
	durableTopicPrefetch
	enableAudit
	expireMessagesPeriod
	gcInactiveDestinations
	gcWithNetworkConsumers
	inactiveTimeoutBeforeGC
	inactiveTimoutBeforeGC

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

Elemento	Atributo
	`reduceMemoryFootprint`
	`sendAdvisoryIfNoConsumers`
	`sendFailIfNoSpace`
	`sendFailIfNoSpaceAfterTimeout` Compatível com Apache ActiveMQ 15.16.4 e posterior
	`sendDuplicateFromStoreToDLQ`
	`storeUsageHighWaterMark`
	`strictOrderDispatch`
	`tempQueue`
	`tempTopic`
	`timeBeforeDispatchStarts`
	`topic`
	`topicPrefetch`
	`useCache`
	`useConsumerPriority`
`usePrefetchExtension`	
`prefetchRatePendingMessageLimitStrategy`	`multiplier`

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

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

Elemento	Atributo
	processNetworkMessages
	ttlCeiling
topic	DLQ
	physicalName
transportConnector	name
	updateClusterClients
	rebalanceClusterClients
	updateClusterClientsOnRemove
uniquePropertyMessageEvictionStrategy	evictExpiredMessagesHighWatermark
	propertyName
virtualTopic	concurrentSend
	local
	dropOnResourceLimit
	name
	postfix
	prefix
	selectorAware
	setOriginalDestination
	transactedSend

Atributos de elementos pai do Amazon MQ

A seguinte é uma explicação detalhada dos atributos de elementos pai. Para obter mais informações, consulte Configuração de XML na documentação do Apache ActiveMQ.

Tópicos

- agente

agente

`broker` é um elemento de coleção pai.

Atributos

networkConnectionStartAssíncrono

Para reduzir a latência da rede e permitir que outras redes para iniciar em tempo hábil, use a tag `<networkConnectionStartAsync>`. A tag instrui o agente a usar um executor para iniciar conexões de rede em paralelo, assíncrona para um agente iniciar.

Padrão: `false`

Exemplo de configuração

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

Elementos, elementos de conjunto secundários e elementos secundários permitidos nas configurações do Amazon MQ

A seguinte é uma lista detalhada dos elementos, elementos do conjunto de filhos e de seus elementos filho permitidos nas configurações do Amazon MQ. Para obter mais informações, consulte Configuração de XML na documentação do Apache ActiveMQ.

Elemento	Elemento de coleção filho	Elemento filho
`authorizationMap`	`authorizationEntries`	`authorizationEntry`
		`tempDestinationAuthorizationEntry`

Elemento	Elemento de coleção filho	Elemento filho
	defaultEntry	authorizationEntry
		tempDestinationAuthorizationEntry
	tempDestinationAuthorizationEntry	tempDestinationAuthorizationEntry
authorizationPlugin	map	authorizationMap
broker	destinationInterceptors	mirroredQueue
		virtualDestinationInterceptor
	destinationPolicy	policyMap
	destinations	queue
		tempQueue
		tempTopic
		topic
	networkConnectors	networkConnector
	persistenceAdapter	kahaDB
	plugins	authorizationPlugin
		discardingDLQBrokerPlugin
		forcePersistencyModeBrokerPlugin
		redeliveryPlugin

Elemento	Elemento de coleção filho	Elemento filho
		statisticsBrokerPlugin
		timeStampingBrokerPlugin
	systemUsage	systemUsage
	transportConnector	name
		updateClusterClients
		rebalanceClusterClients
		updateClusterClientsOnRemove
compositeQueue	forwardTo	queue
		tempQueue
		tempTopic
		topic
		filteredDestination
compositeTopic	forwardTo	queue
		tempQueue
		tempTopic
		topic
		filteredDestination
policyEntry	deadLetterStrategy	discarding

Elemento	Elemento de coleção filho	Elemento filho
		individualDeadLetterStrategy
		sharedDeadLetterStrategy
	destination	queue
		tempQueue
		tempTopic
		topic
	dispatchPolicy	priorityDispatchPolicy
		priorityNetworkDispatchPolicy
		roundRobinDispatchPolicy
		simpleDispatchPolicy
		strictOrderDispatchPolicy
		clientIdFilterDispatchPolicy
	messageEvictionStrategy	oldestMessageEvictionStrategy
		oldestMessageWithLowestPriorityEvictionStrategy

Elemento	Elemento de coleção filho	Elemento filho
		uniquePropertyMessageEvictionStrategy
	messageGroupMapFactory	cachedMessageGroupMapFactory
		messageGroupHashBucketFactory
		simpleMessageGroupMapFactory
	pendingDurableSubscriberPolicy	fileDurableSubscriberCursor
		storeDurableSubscriberCursor
		vmDurableCursor
	pendingMessageLimitStrategy	constantPendingMessageLimitStrategy
		prefetchRatePendingMessageLimitStrategy
	pendingQueuePolicy	fileQueueCursor
		storeCursor
		vmQueueCursor
	pendingSubscriberPolicy	fileCursor
		vmCursor

Elemento	Elemento de coleção filho	Elemento filho
	slowConsumerStrategy	abortSlowAckConsumerStrategy
		abortSlowConsumerStrategy
	subscriptionRecoveryPolicy	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
timedSubscriptionRecoveryPolicy		
policyMap	defaultEntry	policyEntry
	policyEntries	policyEntry
redeliveryPlugin	redeliveryPolicyMap	redeliveryPolicyMap
redeliveryPolicyMap	defaultEntry	redeliveryPolicy
	redeliveryPolicyEntries	redeliveryPolicy

Elemento	Elemento de coleção filho	Elemento filho
retainedMessageSubscriptionRecoveryPolicy	wrapped	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
		timedSubscriptionRecoveryPolicy
sharedDeadLetterStrategy	deadLetterQueue	queue
		tempQueue
		tempTopic
		topic
virtualDestinationInterceptor	virtualDestinations	compositeQueue
		compositeTopic
		virtualTopic

Atributos de elementos filho do Amazon MQ

A seguinte é uma explicação detalhada dos atributos de elementos filho. Para obter mais informações, consulte Configuração de XML na documentação do Apache ActiveMQ.

Tópicos

authorizationEntry

`authorizationEntry` é um filho do elemento do conjunto de filhos `authorizationEntries`.

Atributos

admin|read|write

As permissões concedidas a um grupo de usuários. Para obter mais informações, consulte Sempre configurar um mapa de autorização.

Se você especificar um mapa de autorização que não inclua o `activemq-webconsole`, você não poderá usar o Console da Web do ActiveMQ porque o grupo não estará autorizado a enviar mensagens ou receber mensagens do agente do Amazon MQ.

Padrão: `null`

Exemplo de configuração

```
<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**
>
> O grupo `activemq-webconsole` do ActiveMQ no Amazon MQ tem permissões de administrador em todas as filas e tópicos. Todos os usuários desse grupo terão acesso de administrador.

networkConnector

`networkConnector` é um filho do elemento do conjunto de filhos `networkConnectors`.

Tópicos

- Atributos
- Exemplos de configuração

Atributos

conduitSubscriptions

Especifica se uma conexão de rede em uma rede de agentes trata vários consumidores que se inscreveram para o mesmo destino como um consumidor. Por exemplo, se `conduitSubscriptions` estiver definido como `true` e dois consumidores se conectarem ao agente B e consumirem a partir de um destino, o agente B combina as assinaturas em uma única assinatura lógica pela conexão de rede para o agente A, para que apenas uma única cópia de um agente de mensagem seja encaminhado do agente A para o B.

> **Note**
>
> Configurar `conduitSubscriptions` como `true` pode reduzir o tráfego de rede redundante. No entanto, usar esse atributo pode ter implicações para o balanceamento de carga de mensagens entre os consumidores e pode causar comportamento incorreto em determinados cenários (por exemplo, com seletores de mensagens JMS ou com tópicos duráveis).

Padrão: `true`

duplex

Especifica se a conexão na rede de agentes é usada para produzir e consumir mensagens. Por exemplo, se o agente A cria uma conexão para o agente B no modo não duplex, as mensagens podem ser encaminhadas apenas do agente A para o agente B. No entanto, se o agente A cria uma conexão duplex para o agente B, então, o agente B pode encaminhar mensagens para o agente A sem a necessidade de configurar um `<networkConnector>`.

Padrão: `false`

name

O nome da ponte na rede de agentes.

Padrão: `bridge`

uri

O endpoint do protocolo de nível de conexão para um dos dois agentes (ou para vários agentes) em uma rede de agentes.

Padrão: `null`

username

O nome de usuário comum aos agentes em uma rede de agentes.

Padrão: `null`

Exemplos de configuração

> **Note**
>
> Ao usar um `networkConnector` para definir uma rede de agentes, não inclua a senha de usuário comum para os agentes.

Uma rede de agentes com dois agentes

Nesta configuração, dois agentes são conectados em uma rede de agentes. O nome do conector de rede é `connector_1_to_2`, o nome de usuário comum aos corretores é `myCommonUser`, a

conexão é `duplex` e o URI do OpenWire endpoint é prefixado por`static:`, indicando uma one-to-one conexão entre os corretores.

```
<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 obter mais informações, consulte Configure Network Connectors for Your Broker.

Uma rede de agentes com vários agentes

Nesta configuração, vários agentes são conectados em uma rede de agentes. O nome do conector de rede é`connector_1_to_2`, o nome de usuário comum aos corretores é`myCommonUser`, a conexão é`duplex`, e a lista de OpenWire endpoints separados por vírgulas URIs é prefixada por`masterslave:`, indicando uma conexão de failover entre os corretores. O failover do agente para o agente não é aleatório e tentativas de reconexão continuam 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**
>
> Recomendamos usar o prefixo `masterslave:` para as redes de agentes. O prefixo é idêntico à sintaxe mais explícita `static:failover:()?randomize=false&maxReconnectAttempts=0`.

> **Note**
>
> Essa configuração de XML não permite espaços.

kahaDB

`kahaDB` é um filho do elemento do conjunto de filhos `persistenceAdapter`.

Atributos

concurrentStoreAndDispatchQueues

Especifica se é necessário usar armazenamento e despacho simultâneos para filas. Para obter mais informações, consulte Desativar o armazenamento e a expedição simultâneos para filas com consumidores lentos.

Padrão: `true`

cleanupOnStop

> Compatível com
>
> Apache ActiveMQ 15.16.x e versão superior

Quando desativada, a coleta de resíduos e a limpeza não ocorrem quando o agente é interrompido, o que agiliza o processo de desligamento. O aumento da velocidade é útil em casos com grandes bancos de dados ou bancos de dados do programador.

Padrão: `true`

journalDiskSyncIntervalo

Intervalo (ms) para quando executar uma sincronização de disco se `journalDiskSyncStrategy=periodic`. Para obter mais informações, consulte a documentação do Apache ActiveMQ kahaDB.

Padrão: `1000`

journalDiskSyncEstratégia

> Compatível com
>
> Apache ActiveMQ 15.14.x e versão superior

Configura a política de sincronização de disco. Para obter mais informações, consulte a documentação do Apache ActiveMQ kahaDB.

Padrão: `always`

> **Note**
>
> A documentação do ActiveMQ afirma que a perda de dados é limitada à duração de `journalDiskSyncInterval`, que tem um padrão de 1s. A perda de dados pode ser maior do que o intervalo, mas é difícil ser preciso. Tenha cuidado.

preallocationStrategy

Configura como o agente tentará pré-alocar os arquivos do diário quando um novo arquivo do diário for necessário. Para obter mais informações, consulte a documentação do Apache ActiveMQ kahaDB.

Padrão: `sparse_file`

Exemplo de configuração

Example

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

systemUsage

`systemUsage` é um filho do elemento do conjunto de filhos `systemUsage`. Ele controla a quantidade máxima de espaço que o agente usará antes de desacelerar os produtores. Para obter mais informações, consulte Controle do fluxo do produtor na documentação do Apache ActiveMQ.

Elemento filho

memoryUsage

`memoryUsage` é um filho do elemento filho `systemUsage`. Ele gerencia o uso de memória. Use `memoryUsage` para acompanhar quanto de um elemento está sendo usado, para que você possa controlar o uso do conjunto de trabalho de forma produtiva. Para obter mais informações, consulte o esquema na documentação do Apache ActiveMQ.

Elemento filho

`memoryUsage` é um filho do elemento filho `memoryUsage`.

Atributo

percentOfJvmPilha

Número inteiro entre 0 (inclusive) e 70 (inclusive).

Padrão: `70`

Atributos

sendFailIfNoSpace

Define se um método `send()` deverá falhar se não houver espaço livre. O valor padrão é false, o que bloqueia o método `send()` até haver espaço disponível. Para obter mais informações, consulte o esquema na documentação do Apache Active MQ.

Padrão: `false`

sendFailIfNoSpaceAfterTimeout

Padrão: `null`

Exemplo de configuração

Example

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

```
                                    <systemUsage sendFailIfNoSpace="true"
sendFailIfNoSpaceAfterTimeout="2000">
                                        <memoryUsage>
                                            <memoryUsage  percentOfJvmHeap="60" />
                                        </memoryUsage>>
                                    </systemUsage>
                                    </systemUsage>
                                </broker>
                                </persistenceAdapter>
```

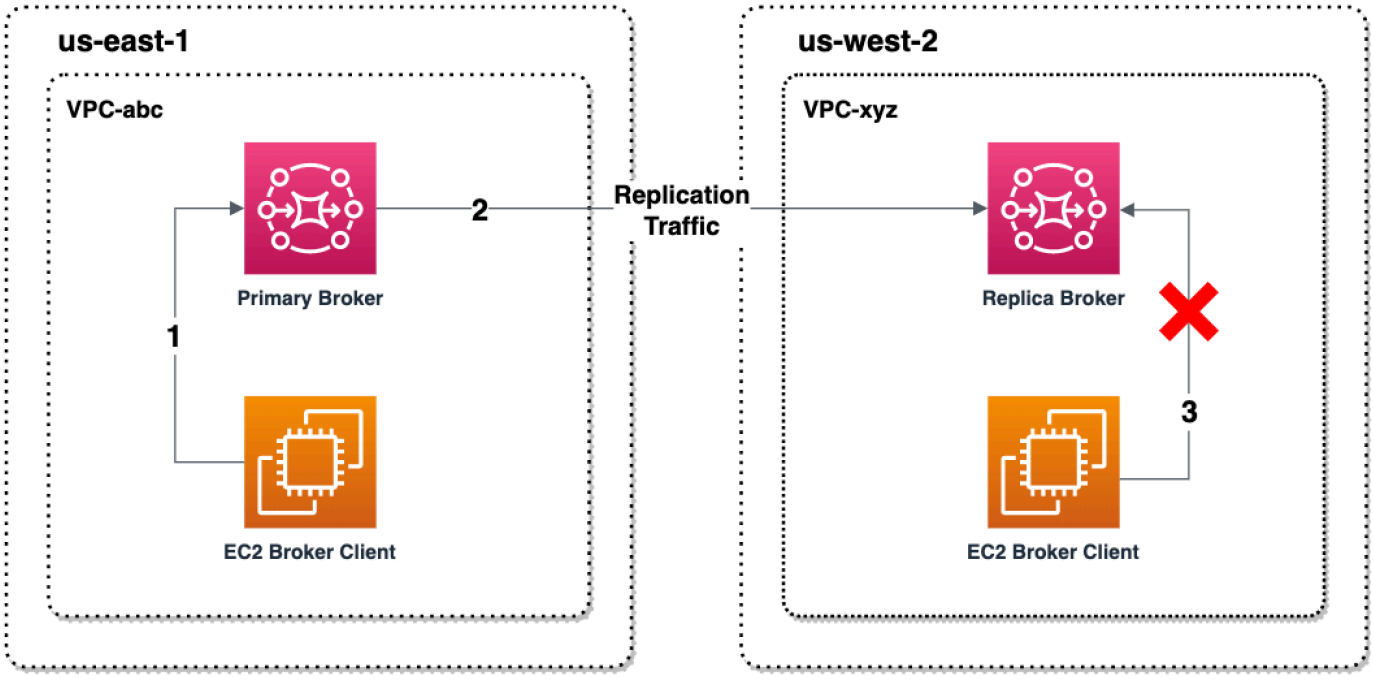
Replicação de dados entre regiões para o Amazon MQ for ActiveMQ

O Amazon MQ for ActiveMQ oferece um recurso de replicação de dados entre regiões (CRDR) que permite a replicação assíncrona de mensagens do agente primário em uma região primária da AWS para o agente de réplica em uma região de réplica. Ao emitir uma solicitação de failover para a API do Amazon MQ, o agente de réplica atual é promovido à função de agente primário e o agente primário atual é rebaixado para a função de réplica.

Agentes primários e de réplica para replicação de dados entre regiões

Você pode criar agentes primários e de réplica para replicação assíncrona de dados do agente primário em uma região AWS primária para o agente de réplica em uma região de réplica. A região primária consiste em um par redundante de agentes ativo/em espera, denominado agente primário. A região secundária consiste em um par redundante de agentes ativo/em espera, denominado agente de réplica.

O diagrama a seguir ilustra um agente de réplica em uma região secundária recebendo dados replicados assíncronos do agente primário na região primária.



Os agentes primários e de réplica atuam como uma solução de recuperação de dados entre regiões. Se o agente primário na região primária falhar, você poderá promover o agente de réplica na região secundária para primário iniciando uma transição ou um failover. O antigo agente primário então se torna o agente de réplica, e o antigo agente de réplica é promovido a agente primário. Para ter instruções sobre como criar um agente primário e um agente de réplica, consulte Criar um agente de replicação de dados do Amazon MQ entre regiões.

> **Note**
>
> Disponível apenas para agentes ativo/em espera.
> Não disponível para filas espelhadas.

Criar um agente de replicação de dados do Amazon MQ entre regiões

Com a replicação de dados entre regiões (CRDR), você pode alternar entre o Amazon MQ para agentes de mensagens ActiveMQ em duas regiões da AWS, conforme necessário. Você pode designar um agente existente como agente primário e criar uma réplica para esse agente, ou criar um agente primário e um agente de réplica juntos. Depois, é possível promover o agente de réplica à função de agente primário usando a operação `Promote` da API do Amazon MQ. Para ter mais

informações sobre agentes primários e de réplica, consulte Agentes primários e de réplica para replicação de dados entre regiões.

As instruções a seguir descrevem como criar e configurar um agente de réplica usando o Console de Gerenciamento do Amazon MQ.

Tópicos

Pré-requisitos

Para usar o recurso de replicação de dados entre regiões, você deve analisar e cumprir os seguintes pré-requisitos:

- Versão: o atributo de replicação de dados entre regiões só está disponível para agentes do Amazon MQ para ActiveMQ nas versões 5.17.6 e posterior.
- Região: a replicação de dados entre regiões é aceita nas seguintes regiões: Leste dos EUA (Ohio), Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon) e Oeste dos EUA (Norte da Califórnia).
- Tipo de instância: a replicação de dados entre regiões só está disponível para tamanhos de instância de agente `mq.m5.large` e posterior.
- Tipo de implantação: a replicação de dados entre regiões só está disponível para agentes ativos/em espera com implantação em várias zonas de disponibilidade.
- Status do agente: você só pode criar um agente de réplica para um agente primário com o status de agente `Running`.

Etapa 1 (opcional): Criar um agente primário

Criar um agente primário

1. Faça login no console do Amazon MQ.
2. Na página Agentes do console do Amazon MQ, escolha Criar agentes.
3. Na página Select broker engine (Selecionar mecanismo de agente), selecione Apache ActiveMQ (Apache ActiveMQ).

4. Na página Select deployment and storage (Selecionar implantação e armazenamento), na seção Deployment mode and storage type (Modo de implantação e tipo de armazenamento), faça o seguinte:

   - Em Modo de implantação, escolha Operador ativo/em espera de alta disponibilidade. Um agente ativo/em espera é composto por dois agentes em duas zonas de disponibilidade diferentes, configuradas em um par redundante. Esses agentes se comunicam de forma síncrona com sua aplicação e com o Amazon EFS. Para obter mais informações, consulte Opções de implantação de agentes do Amazon MQ para ActiveMQ.

5. Escolha Próximo.

6. Na página Definir configurações, faça o seguinte na seção Detalhes:

   a. Digite o Broker name (Nome do agente).

   > **Important**
   >
   > Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de agente. Os nomes de agente são acessíveis a outros serviços de AWS, incluindo o CloudWatch Logs. Nomes de agente não devem ser usados para dados privados ou sigilosos.

   b. Selecione o Tipo de instância de agente (por exemplo, m5.large). Para obter mais informações, consulte Broker instance types.

7. Na seção ActiveMQ Web Console access (Acesso ao console da Web ActiveMQ), forneça um Username (Nome de usuário) e Password (Senha). As seguintes restrições se aplicam a nomes de usuário e senhas de agente:

   - Seu nome de usuário pode conter somente caracteres alfanuméricos, traços, pontos, sublinhados e tils (- . _ ~).
   - Sua senha deve ter pelo menos 12 caracteres, deve conter pelo menos 4 caracteres exclusivos e não deve conter vírgulas, dois pontos ou sinais de igual (,:=).

   > **Important**
   >
   > Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de usuário do agente. Nomes de usuário do agente

são acessíveis a outros serviços de AWS, incluindo o CloudWatch Logs. Nomes de usuário do agente não devem ser usados para dados privados ou sigilosos.

A barra verde na parte superior da página confirma que o Amazon MQ está criando o agente de réplica na região de recuperação. Você também pode ver a função da CRDR e o status do RPO de seus agentes. Para desativar as colunas Função da CRDR e Status do RPO, escolha o ícone de engrenagem no canto superior direito da tabela Agentes. Depois, na página Preferências, desative a Função da CRDR ou o Status do RPO.

Etapa 2: Criar uma réplica de um agente existente

1. Na página Agentes do console do Amazon MQ, escolha Criar agente de réplica.
2. Na página Escolher agente primário, selecione um agente existente para usar como agente primário de CRDR. Em seguida, escolha Próximo.
3. Na página Configurar agente de réplica, use o menu suspenso para escolher a região da réplica.
4. Na seção Usuário do console ActiveMQ para agente de réplica, forneça um nome de usuário e uma senha para o usuário do console do agente de réplica. As seguintes restrições se aplicam a nomes de usuário e senhas de agente:

   - Seu nome de usuário pode conter somente caracteres alfanuméricos, traços, pontos, sublinhados e tils (- . _ ~).
   - Sua senha deve ter pelo menos 12 caracteres, deve conter pelo menos 4 caracteres exclusivos e não deve conter vírgulas, dois pontos ou sinais de igual (,:=).

   Important

   Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de usuário do agente. Nomes de usuário do agente são acessíveis a outros serviços de AWS, incluindo o CloudWatch Logs. Nomes de usuário do agente não devem ser usados para dados privados ou sigilosos.

5. Na seção Usuário de replicação de dados para conectar o acesso entre agentes, forneça um nome de usuário e uma senha para o usuário que acessará o agente primário e o agente de réplica. As seguintes restrições se aplicam a nomes de usuário e senhas de agente:

- Seu nome de usuário pode conter somente caracteres alfanuméricos, traços, pontos, sublinhados e tils (- . _ ~).
- Sua senha deve ter pelo menos 12 caracteres, deve conter pelo menos 4 caracteres exclusivos e não deve conter vírgulas, dois pontos ou sinais de igual (,:=).

> **Important**
>
> Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de usuário do agente. Nomes de usuário do agente são acessíveis a outros serviços de AWS, incluindo o CloudWatch Logs. Nomes de usuário do agente não devem ser usados para dados privados ou sigilosos.

Defina todas as configurações adicionais. Em seguida, escolha Próximo.

6. Na página Analisar e criar, revise os detalhes do agente de réplica. Depois, escolha Criar operador de réplica.
7. Depois, reinicialize o agente primário. Isso também reinicializará o agente de réplica. Para ter instruções sobre como reiniciar seu agente, consulte Rebooting a Broker.

Para ter mais informações sobre como definir configurações adicionais para seu agente do ActiveMQ, consulte Conceitos básicos: criar e conectar a um agente do ActiveMQ.

Excluir um agente de replicação de dados do Amazon MQ entre regiões

Para excluir um agente de replicação de dados entre regiões (CRDR) primário ou de réplica, você deve primeiro desemparelhar e depois reinicializar os agentes. As instruções a seguir mostram como cancelar o emparelhamento e reinicializar os agentes usando o Console de Gerenciamento da AWS.

1. Na página Agentes, selecione o agente de CRDR cujo emparelhamento você deseja cancelar e escolha Editar.
2. Na página Editar agente, na seção Replicação de dados, escolha Cancelar emparelhamento de agentes.
3. Digite “confirm (confirmar)” na janela pop-up para confirmar a escolha. Depois, escolha Cancelar emparelhamento de agentes.

4. Depois, reinicialize o agente primário não emparelhado. Isso também reinicializará o agente de réplica. Para ter instruções sobre como reiniciar seu agente, consulte Rebooting a Broker. Depois que o agente primário for reinicializado, o emparelhamento dos dois agentes será cancelado e eles poderão ser excluídos individualmente. Para excluir seu agente, consulte Deleting a broker.

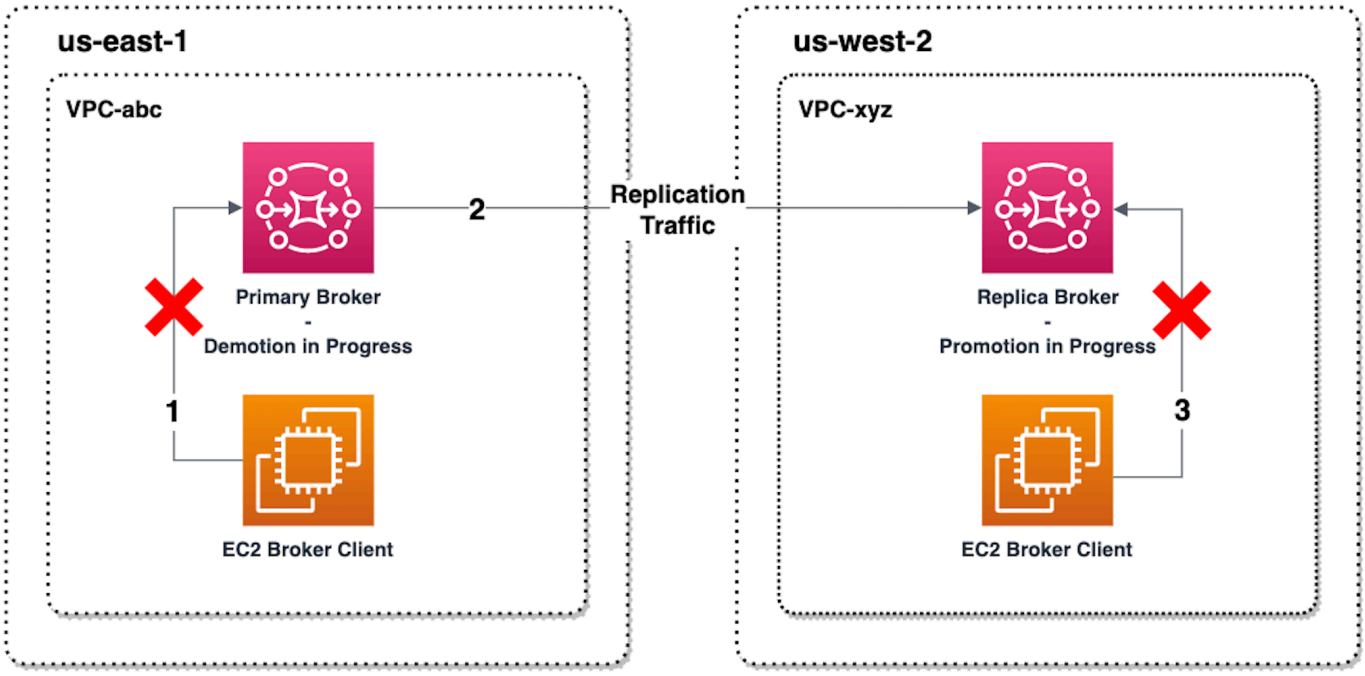
Iniciar a transição ou o failover para promover um agente de réplica do Amazon MQ à função de agente primário

Você pode iniciar uma transição ou um failover quando quiser promover o agente de réplica à função de agente primário. Quando você promove o agente de réplica, o agente primário é rebaixado para a função de agente de réplica.

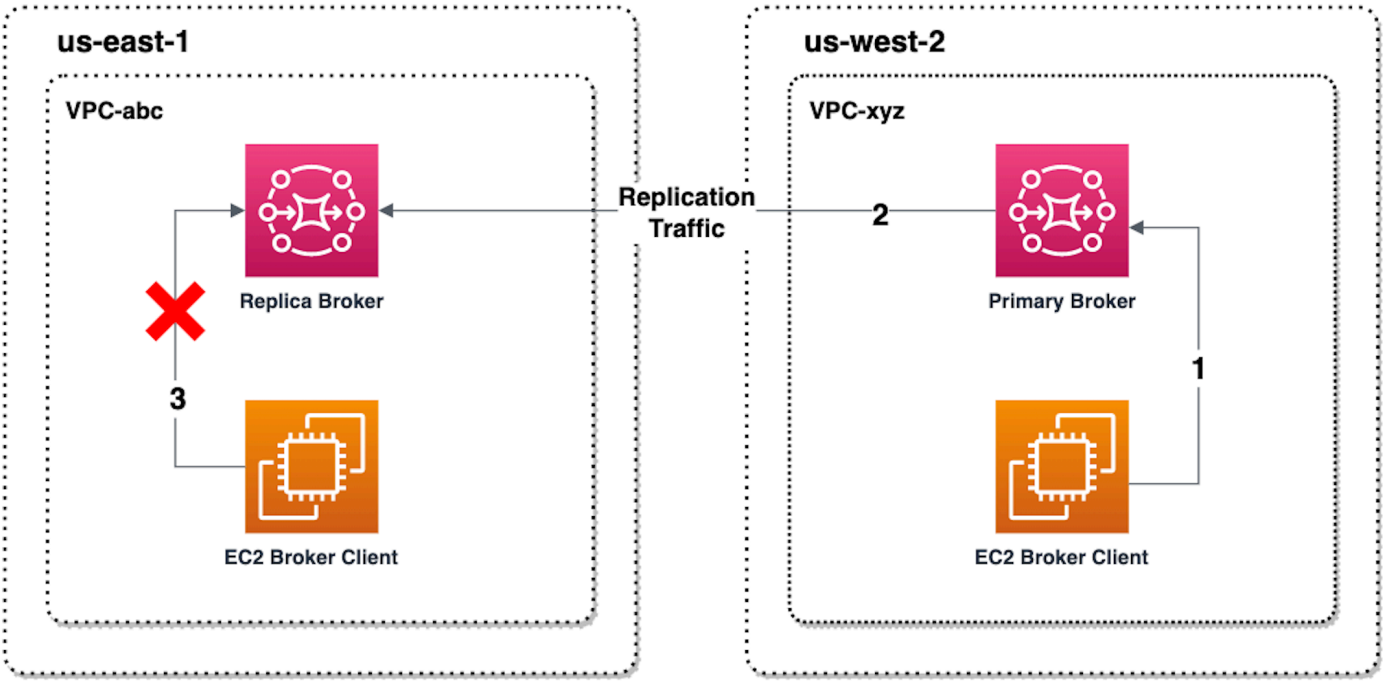
Uma transição prioriza a consistência em detrimento da disponibilidade. É garantido que os agentes tenham o mesmo estado quando essa operação de failover for concluída. Com uma transição, pode haver um período em que nenhum dos agentes esteja disponível para conexões com clientes e a consistência entre agentes seja estabelecida. Os dois agentes terão o mesmo estado no instante em que a réplica for promovida. O êxito da transição depende da integridade das duas regiões e da rede inter-regional.

Um failover prioriza a disponibilidade em detrimento da consistência. Não é garantido que os agentes tenham o mesmo estado quando essa operação for concluída. Com um failover, é garantido que o agente de réplica fique imediatamente disponível para atender ao tráfego do cliente, sem esperar que os dados de replicação sejam sincronizados ou que o primário receba o sinal de desligamento. O failover não depende da integridade da região primária original nem da rede inter-regional para ter êxito.

O diagrama a seguir ilustra uma transição na qual nenhum dos agentes aceita conexões de clientes enquanto a fila de replicação está sendo drenada e os estados do agente são sincronizados. Nesse processo, o cliente na VPC do agente primário não consegue produzir mais alterações de estado enquanto a operação está em andamento, e o agente primário está sendo rebaixado para uma réplica. Quando a fila de replicação é drenada e os dois agentes atingem o mesmo estado, o cliente na VPC do agente de réplica não consegue se conectar ao agente de réplica até que a operação de failover seja concluída e o agente de réplica seja promovido a primário.



O diagrama a seguir ilustra o status do agente após a conclusão do processo de transição. O agente de réplica original agora foi promovido à função de agente primário e está aceitando conexões de clientes. O cliente pode produzir e consumir dados do agente.



Promover o agente de réplica usando o console

Para promover o agente de réplica usando transição ou failover, siga estas etapas no console do Amazon MQ.

> **Note**
>
> Você não pode iniciar a transição nem o failover em um agente primário.

1. Mude para a região do seu agente de réplica. Na tabela Agentes, selecione o agente de réplica existente que você promoverá como primário.
2. Na página Detalhes do agente, faça o seguinte:
    1. Selecione Promover uma réplica.
    2. Na janela pop-up, escolha Transição ou Failover.
    3. Digite “confirmar” na caixa de texto para confirmar sua escolha.
    4. Escolha Confirmar.

Depois de iniciar o failover, o status do agente muda para Failover em andamento. A barra de progresso azul na parte superior da página Agentes fica verde quando o failover é concluído.

> **Note**
>
> A configuração só é replicada no momento em que o agente replicado é criado. Nenhuma atualização posterior é replicada.

Métricas de replicação de dados entre regiões no Amazon CloudWatch

O atributo de replicação de dados entre regiões do Amazon MQ for ActiveMQ oferece métricas para manter a confiabilidade, a disponibilidade e a performance de seus agentes primários e de réplica. Durante o processo de replicação, um agente de réplica em uma região secundária recebe dados replicados de forma assíncrona do agente primário na região primária. Se o agente primário na região primária falhar, você poderá promover o agente de réplica na região secundária para primário iniciando uma transição ou um failover. Para ter instruções sobre como visualizar métricas no Amazon CloudWatch, consulte Acessando CloudWatch métricas para o Amazon MQ.

Carimbos de data/hora da CRDR

Os carimbos de data/hora a seguir descrevem como as métricas encontradas no Amazon CloudWatch são calculadas. Há cinco carimbos de data/hora no processo de replicação de dados:

- Tempo de observação atual (TCO): o instante atual no tempo.
- Hora da criação (TC): o instante em que um evento foi criado na fila de replicação pelo agente primário. Disponível em agentes primários e de réplica.
- Hora da entrega (TD): o instante em que um evento foi entregue com êxito ao agente de réplicas. Disponível somente em agentes de réplica.
- Tempo de processamento (TP): o instante em que um evento foi processado com êxito pelo agente de réplica. Disponível somente em agentes de réplica.
- Tempo de confirmação (TA): o instante em que um evento foi reconhecido com êxito pelo agente principal. Disponível apenas em agentes primários.

Estime a performance de transição/failover com métricas do CRDR CloudWatch

O Amazon MQ habilita métricas para o seu agente por padrão. É possível visualizar as métricas do agente acessando o console do Amazon CloudWatch ou usando a API do CloudWatch. As métricas a seguir são úteis para entender a performance de replicação e de transição/failover de seus agentes de CRDR:

Métrica do Amazon MQ CloudWatch	Motivo do uso da CRDR	
`TotalReplicationLag`	O tempo estimado entre TA e TC do último evento não confirmado no agente primário.	
`ReplicationLag`	O tempo estimado entre TP e TC do último evento não confirmado no agente primário.	
`PrimaryWaitTime`	O tempo estimado entre TCO e TC do último evento	

Métrica do Amazon MQ CloudWatch	Motivo do uso da CRDR	
	processado no agente primário.	
`ReplicaWaitTime`	O tempo estimado entre TCO e TP do último evento processado no agente primário.	
`QueueSize`	O número total de eventos não confirmados na fila de replicação no agente primário.	

`TotalReplicationLag` e `ReplicationLag` descrevem o atraso na replicação entre os agentes primário e de réplica. As duas métricas também podem ser usadas para estimar o tempo até a conclusão da operação contínua de transição ou failover.

`PrimaryWaitTime` e `ReplicaWaitTime` podem ser usados para identificar quaisquer problemas contínuos com o processo de replicação. Se o valor da métrica estiver aumentando constantemente, isso poderá indicar que o processo de replicação está degradado ou pausado. A replicação lenta pode decorrer de problemas como particionamento de rede, inicialização de agentes e recuperação prolongada.

Tutoriais ActiveMQ

Os tutoriais a seguir mostram como você pode criar e se conectar aos agentes do ActiveMQ. Para usar o código de exemplo o ActiveMQ Java, será necessário instalar o Kit de Desenvolvimento da Edição Padrão do Java e fazer algumas alterações de configuração no código.

Tópicos

- Criar e configurar uma rede de agentes Amazon MQ.
- Conectar uma aplicação Java ao agente do Amazon MQ
- Integração de agentes ActiveMQ com LDAP
- Etapa 3: (opcional) conectar-se a uma AWS Lambda função
- Criar um usuário do agente do ActiveMQ

- Editar um usuário do agente do ActiveMQ
- Excluir um usuário do agente do ActiveMQ
- Exemplos funcionais de Como usar o Java Message Service (JMS) com o ActiveMQ

Criar e configurar uma rede de agentes Amazon MQ.

A rede de agentes é composta por vários agentes ativos simultaneamente, agentes de instância única ou agentes ativos/em espera. Neste tutorial, você aprende a criar uma rede de agentes de dois agentes com uma topologia de origem e de destino.

Para obter uma visão geral conceitual e informações detalhadas de configuração, consulte o seguinte:

- Rede de agentes do Amazon MQ
- Configurar sua rede de agentes corretamente
- `networkConnector`
- `networkConnectionStartAssíncrono`
- Redes de agentes na documentação do ActiveMQ

Você pode usar o console do Amazon MQ para criar uma rede de agentes do Amazon MQ. Como você pode iniciar a criação dos dois agentes em paralelo, esse processo leva cerca de 15 minutos.

Tópicos

- Pré-requisitos
- Etapa 1: Permitir tráfego entre agentes
- Etapa 2: Configurar os conectores de rede para o seu agente
- Próximas etapas

Pré-requisitos

Para criar uma rede de agentes, você deve ter o seguinte:

- Dois ou mais agentes simultaneamente ativos (chamado `MyBroker1` e `MyBroker2` neste tutorial). Para obter mais informações sobre como criar agentes, consulte Conceitos básicos: criar e conectar a um agente do ActiveMQ.

- Os dois corretores devem estar na mesma VPC ou em pares. VPCs Para obter mais informações VPCs, consulte O que é Amazon VPC? no Guia do usuário da Amazon VPC e o que é emparelhamento de VPC? no Guia de emparelhamento do Amazon VPC.

  > **Important**
  >
  > Se você não tem uma VPC padrão, uma sub-rede ou grupo de segurança, você deve criá-los primeiro. Para obter mais informações, consulte um dos tópicos a seguir no Manual do usuário da Amazon VPC.
  >
  > - Criar uma VPC padrão
  > - Criar uma sub-rede padrão
  > - Criar um grupo de segurança

- Dois usuários com credenciais de login idênticas para ambos os agentes. Para obter mais informações sobre como criar usuários, consulte Criar um usuário do agente do ActiveMQ.

  > **Note**
  >
  > Ao integrar a autenticação LDAP com uma rede de agentes, certifique-se de que o usuário existe tanto como um agente ActiveMQ como um usuário LDAP.

O exemplo a seguir usa dois agentes de instância única. No entanto, você pode criar redes de agentes usando os agentes ativos/em espera ou uma combinação de modos de implantação de agente.

Etapa 1: Permitir tráfego entre agentes

Depois de criar seus agentes, é necessário permitir o tráfego entre eles.

1. No console do Amazon MQ, na página MyBroker2, na seção Detalhes, em Segurança e rede, escolha o nome do seu grupo de segurança ou.

   A página Grupos de segurança do painel do EC2 é exibida.
2. Na lista de security group, escolha seu security group.
3. Na parte inferior da página, escolha Inbound (Entrada) e a seguir selecione Edit (Editar).

4. Na caixa de diálogo Editar regras de entrada, adicione uma regra para o OpenWire endpoint.

   a. Escolha Add Rule (Adicionar regra).

   b. Em Type (Tipo), selecione Custom TCP (TCP personalizado).

   c. Em Port Range, digite a OpenWire porta (61617).

   d. Execute um destes procedimentos:

   - Se você deseja restringir o acesso a determinado endereço IP, em Source (Origem), deixe a opção Custom (Personalizar) selecionada e insira o endereço IP de `MyBroker1`, seguido por `/32`. (Isso converte o endereço IP em um registro CIDR válido). Para obter mais informações, consulte Interfaces de rede elástica.

     > **Tip**
     >
     > Para recuperar o endereço IP do `MyBroker1`, no console do Amazon MQ, escolha o nome do agente e navegue até a seção Detalhes.

   - Se todos os agentes são privados e pertencem à mesma VPC, em Source (Origem), deixe a opção Custom (Personalizar) selecionada e insira o ID do grupo de segurança que você está editando.

     > **Note**
     >
     > Para agentes públicos, é necessário restringir o acesso usando endereços IP.

   e. Escolha Salvar.

   Agora seu agente pode aceitar conexões de entrada.

Etapa 2: Configurar os conectores de rede para o seu agente

Depois de permitir o tráfego entre os agentes, você deve configurar os conectores de rede para um deles.

1. Edite a revisão da configuração para o agente `MyBroker1`.

   a. Na página MyBroker1, escolha Editar.

   b. Na página Editar MyBroker 1, na seção Configuração, escolha Exibir.

O tipo e a versão do mecanismo de agente que a configuração usa (por exemplo, Apache ActiveMQ 5.15.0) são exibidos.

c. Na guia Detalhes da configuração, são exibidos o número de revisão da configuração, a descrição e a configuração do agente no formato XML.

d. Escolha Editar configuração.

e. Na parte inferior do arquivo de configuração, remova a seção `<networkConnectors>` e inclua as seguintes informações:

- O `name` para o conector de rede.
- O Console da Web ActiveMQ `username` que é comum para ambos os agentes.
- Ativa as conexões `duplex`.
- Execute um destes procedimentos:
  - Se você estiver conectando o broker a um broker de instância única, use o `static:` prefixo e o OpenWire endpoint `uri` para. `MyBroker2` Por exemplo:

```
<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>
```

  - Se você estiver conectando o broker a um broker ativo/em espera, use o `static +failover` transporte e o OpenWire endpoint `uri` para ambos os corretores com os seguintes parâmetros de consulta. `?randomize=false&maxReconnectAttempts=0` Por exemplo:

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

> **Note**
>
> Não inclua as credenciais de login para o usuário do ActiveMQ.

f. Escolha Salvar.

g. Na caixa de diálogo Salvar revisão, digite `Add network of brokers connector for MyBroker2`.

h. Escolha Salvar para salvar a nova revisão de configuração.

2. Edite `MyBroker1` para definir a última revisão de configuração para aplicar imediatamente.

a. Na página MyBroker1, escolha Editar.

b. Na página Editar MyBroker 1, na seção Configuração, escolha Programar modificações.

c. Na seção Programar modificações do agente, escolha para aplicar modificações imediatamente.

d. Escolha Aplicar.

`MyBroker1` é reinicializado e sua revisão de configuração será aplicada.

A rede de agentes é criada.

Próximas etapas

Depois de configurar a rede de agentes, você pode testá-la ao produzir e consumir mensagens.

> **Important**
>
> Certifique-se de habilitar conexões de entrada de sua máquina local para o broker `MyBroker1` na porta 8162 (para o ActiveMQ Web Console) e na porta 61617 (para o endpoint). OpenWire
> Também será necessário ajustar as configurações do grupo de segurança para permitir que o produtor e o consumidor se conectem à rede de agentes.

1. No console do Amazon MQ, navegue até a seção Connections (Conexões) e anote o endpoint do Console da Web ActiveMQ para o agente `MyBroker1`.

2. Navegue até o Console da Web ActiveMQ para o agente `MyBroker1`.
3. Para verificar se a ponte de rede está conectada, escolha Rede.

   Na seção Pontes de rede, o nome e o endereço de `MyBroker2` são listados nas colunas Agente remoto e Endereço do agente.
4. Em qualquer máquina que tem acesso ao agente do `MyBroker2`, crie um consumidor. Por exemplo:

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

   O consumidor se conecta ao OpenWire endpoint `MyBroker2` e começa a consumir mensagens da fila`MyQueue`.
5. Em qualquer máquina que tem acesso ao agente do `MyBroker1`, crie um produtor e envie algumas mensagens. Por exemplo:

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

   O produtor se conecta ao OpenWire endpoint `MyBroker1` e começa a produzir mensagens persistentes na fila`MyQueue`.

Conectar uma aplicação Java ao agente do Amazon MQ

Depois de criar um agente do Amazon MQ ActiveMQ, você pode conectar sua aplicação a ele. Os exemplos a seguir mostram como você pode usar o JMS (Java Message Service) para criar uma conexão com o agente, criar uma fila e enviar uma mensagem. Para obter um exemplo completo e funcional do Java, consulte Working Java Example.

Você pode se conectar a agentes do ActiveMQ usando vários clientes de ActiveMQ. Recomendamos usar o Cliente ActiveMQ.

Tópicos

Pré-requisitos

Habilitar atributos da VPC

Para garantir que seu agente esteja acessível dentro da sua VPC, você deve habilitar os atributos VPC `enableDnsHostnames` e `enableDnsSupport`. Para obter mais informações, consulte Compatibilidade com DNS para a sua VPC no Manual do usuário da Amazon VPC.

Habilitar conexões de entrada

Em seguida, habilite as conexões de entrada para sua aplicação.

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, escolha o nome do seu corretor (por exemplo, MyBroker).
3. Na ***`MyBroker`***página, na seção Conexões, observe os endereços e portas do URL do console web e dos protocolos de nível de fio do broker.
4. Na seção Details (Detalhes), em Security and network (Segurança e rede), escolha o nome do seu grupo de segurança ou

   .

   A página Grupos de segurança do painel do EC2 é exibida.
5. Na lista de security group, escolha seu security group.
6. Na parte inferior da página, escolha Inbound (Entrada) e a seguir selecione Edit (Editar).
7. Na caixa de diálogo Edit inbound rules (Editar regras de entrada), adicione uma regra para cada URL ou endpoint que você deseja que seja acessível publicamente (o exemplo a seguir mostra como fazer isso para um console da Web do agente).

   a. Escolha Add Rule (Adicionar regra).

   b. Em Type (Tipo), selecione Custom TCP (TCP personalizado).

c. Para o Intervalo de Portas, digite a porta do console da Web (`8162`).

d. Para Source (Origem), deixe Custom (Personalizado) selecionado e, depois, digite o endereço IP do sistema ao qual deseja ser capaz de acessar o console da Web (por exemplo, `192.0.2.1`).

e. Escolha Salvar.

Agora seu agente pode aceitar conexões de entrada.

Adicionar dependências de Java

Adicione os pacotes `activemq-client.jar` e `activemq-pool.jar` ao caminho da classe Java. O exemplo a seguir mostra essas dependências em um arquivo `pom.xml` do projeto 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 obter mais informações sobre `activemq-client.jar`, consulte Initial Configuration (Configuração inicial) na documentação do Apache ActiveMQ.

> **Important**
>
> No código de exemplo a seguir, os produtores e consumidores são executados em um único thread. Para sistemas de produção (ou para testar o failover de instância do agente), certifique-se de que seus produtores e consumidores sejam executados em hosts ou threads separados.

Para criar um produtor de mensagens e enviar uma mensagem

Use a instrução a seguir para criar um produtor de mensagens e receber uma mensagem.

1. Crie uma fábrica de conexão em grupo JMS para o produtor da mensagem usando o endpoint do seu agente e, em seguida, chame o método `createConnection` contra a fábrica.

> **Note**
>
> Para um active/standby corretor, o Amazon MQ fornece dois ActiveMQ Web Console URLs, mas somente um URL está ativo por vez. Da mesma forma, o Amazon MQ fornece dois endpoints para cada protocolo de nível de conexão, mas apenas um endpoint está ativo em cada par de cada vez. Os sufixos `-1` e `-2` denotam um par redundante. Para obter mais informações, consulte Opções de implantação de agentes do Amazon MQ para ActiveMQ).
> Para endpoints de protocolo no nível da conexão, você pode permitir que sua aplicação se conecte a qualquer endpoint usando o Transporte de failover.

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

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

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

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

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

> **Note**
>
> Os produtores de mensagens devem sempre usar a classe `PooledConnectionFactory`. Para obter mais informações, consulte Sempre usar pooling de conexão.

2. Crie uma sessão, uma fila chamada `MyQueue` e um produtor de mensagens.

```
// 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. Crie a string da mensagem `"Hello from Amazon MQ!"` e, em seguida, envie a mensagem.

```
// 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. Limpe o produtor.

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

Para criar um consumidor de mensagens e receber a mensagem

Use a instrução a seguir para criar um produtor de mensagens e receber uma mensagem.

1. Crie uma fábrica de conexão JMS para o produtor da mensagem usando o endpoint do seu agente e, em seguida, chame o método `createConnection` contra a 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**
>
> Os consumidores de mensagens nunca devem usar a classe `PooledConnectionFactory`. Para obter mais informações, consulte Sempre usar pooling de conexão.

2. Crie uma sessão, uma fila chamada `MyQueue` e um consumidor de mensagens.

```
// 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. Comece a aguardar mensagens e receba a mensagem quando ela chegar.

```
// 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**
>
> Diferentemente dos serviços de AWS mensagens (como o Amazon SQS), o consumidor está constantemente conectado ao agente.

4. Feche o consumidor, a sessão e a conexão.

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

Integração de agentes ActiveMQ com LDAP

> **Important**
>
> O Amazon MQ não aceita certificado de servidor emitido por uma CA privada.

Você pode acessar seus agentes do ActiveMQ usando os seguintes protocolos com TLS habilitado:

- AMQP
- MQTT
- Acabou o MQTT WebSocket
- OpenWire
- STOMP
- STOMP over WebSocket

O Amazon MQ oferece uma opção entre autenticação nativa do ActiveMQ e autenticação LDAP e autorização para gerenciar permissões de usuário. Para obter informações sobre restrições relacionadas a nomes de usuário e senhas do ActiveMQ, consulte Usuários.

Para autorizar os usuários e grupos do ActiveMQ para trabalhar com filas e tópicos, você deve editar a configuração do agente. O Amazon MQ usa o Plugin de autenticação simples do ActiveMQ para

restringir a leitura e a gravação em destinos. Para obter mais informações e exemplos, consulte Sempre configurar um mapa de autorização e `authorizationEntry`.

> **Note**
>
> Atualmente, o Amazon MQ não é compatível com a autenticação de certificado de cliente.

Tópicos

Integrar LDAP com ActiveMQ

Você pode autenticar usuários do Amazon MQ por meio das credenciais armazenadas em seu servidor Lightweight Directory Access Protocol (LDAP). Você também pode adicionar, excluir e modificar usuários do Amazon MQ e atribuir permissões a tópicos e filas por meio dele. As operações de gerenciamento, como criação, atualização e exclusão de agentes, ainda exigem credenciais do IAM e não estão integradas ao LDAP.

Os clientes que desejam simplificar e centralizar sua autenticação e autorização do Agente Amazon MQ usando um servidor LDAP podem usar esse recurso. Manter todas as credenciais do usuário no servidor LDAP economiza tempo e esforço fornecendo um local central para armazenar e gerenciar essas credenciais.

O Amazon MQ é compatível com a LDAP usando o plugin Apache ActiveMQ JAAS. Qualquer servidor LDAP, como Microsoft Active Directory ou OpenLDAP que for suportado pelo plugin, também é compatível com o Amazon MQ. Para obter mais informações sobre o plugin, consulte a seção Segurança da documentação do ActiveMQ.

Além dos usuários, você pode especificar o acesso a tópicos e filas para um grupo específico ou um usuário por meio do servidor LDAP. Para fazer isso, crie entradas que representam tópicos e filas no servidor LDAP e, em seguida, atribua permissões a um usuário ou grupo LDAP específico. Em seguida, você pode configurar o broker para recuperar dados de autorização do servidor LDAP.

> **Important**
>
> Quando se usa o LDAP, a autenticação não diferencia maiúsculas de minúsculas, mas a autorização diferencia maiúsculas de minúsculas para o nome de usuário.

Pré-requisitos

Antes de adicionar compatibilidade com a LDAP a um agente Amazon MQ novo ou existente, você deve configurar uma conta de serviço. Essa conta de serviço é necessária para iniciar uma conexão com um servidor LDAP e deve ter as permissões corretas para fazer essa conexão. Esta conta de serviço configurará a autenticação LDAP para o seu agente. Quaisquer conexões sucessivas de cliente serão autenticadas através da mesma conexão.

Uma conta de serviço é uma conta no servidor LDAP que tem acesso para iniciar uma conexão. Isto é um requisito LDAP padrão e você deve fornecer as credenciais da conta de serviço apenas uma vez. Após a configuração da conexão, todas as conexões futuras do cliente são autenticadas por meio do servidor LDAP. Suas credenciais de conta de serviço são armazenadas de forma segura em um formulário criptografado, acessível somente para o Amazon MQ.

Para integrar com o ActiveMQ, é necessária uma Árvore de Informações de Diretório (DIT) específica no servidor LDAP. Para um exemplo `ldif` que mostra claramente esta estrutura, veja Importe o seguinte arquivo LDIF para o servidor LDAP na Seção de Segurança da documentação do ActiveMQ.

Conceitos básicos do LDAP

Para começar, navegue até o console do Amazon MQ e escolha Autorização e autenticação LDAP quando você cria uma nova Amazon MQ ou edita uma instância de agente existente.

Forneça as seguintes informações sobre a conta de serviço:

- Nome de domínio totalmente qualificado O local do servidor LDAP para o qual as solicitações de autenticação e autorização devem ser emitidas.

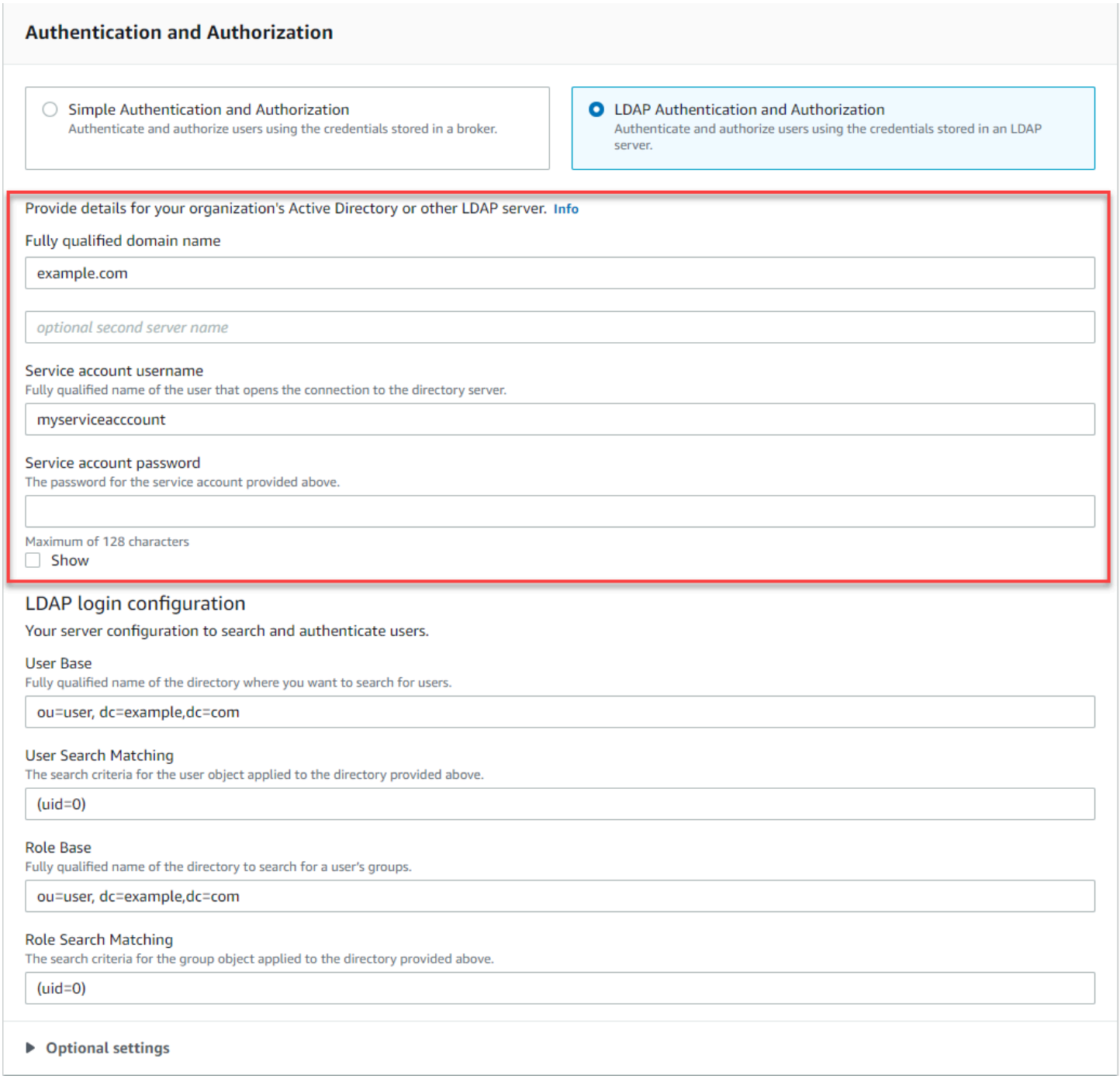
  > **Note**
  >
  > O nome de domínio totalmente qualificado do servidor LDAP fornecido não deve incluir o protocolo ou o número da porta. O Amazon MQ substituirá o nome de domínio totalmente qualificado com o protocolo `ldaps` e anexará o número da porta `636`.

Por exemplo, se você fornecer o seguinte domínio totalmente qualificado: `example.com`, o Amazon MQ acessará seu servidor LDAP usando o seguinte URL: `ldaps://example.com:636`.

Para que o host do agente possa se comunicar com êxito com o servidor LDAP, o nome de domínio totalmente qualificado deve ser resolvido publicamente. Para manter o servidor LDAP privado e seguro, restrinja o tráfego de entrada nas regras de entrada do servidor para permitir apenas o tráfego originado na VPC do agente.

- Nome de usuário da conta de serviço O nome distinto do usuário que será usado para executar a ligação inicial ao servidor LDAP.
- Senha da conta de serviço A senha do usuário que executa a vinculação inicial.

A imagem a seguir destaca onde fornecer esses detalhes.

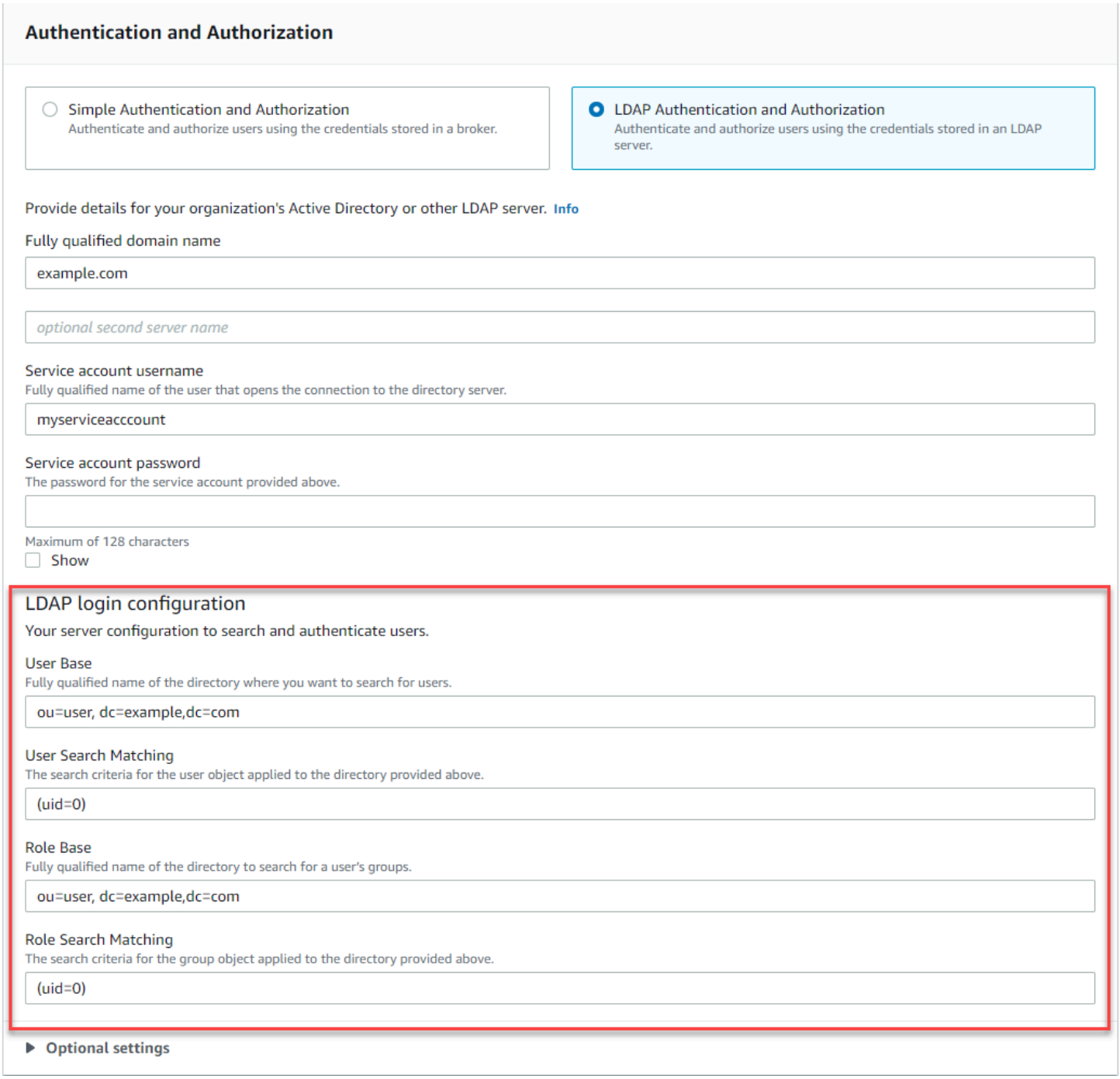


Na Configuração de login LDAP, forneça as seguintes informações obrigatórias:

- Base de usuários O nome distinto do nó na DIT (árvore de informações de diretório) que será pesquisado para os usuários.
- Correspondência de pesquisa de usuários O filtro de pesquisa LDAP que será usado para localizar usuários na `userBase`. O nome de usuário do cliente é substituído no espaço reservado `{0}` no filtro de pesquisa. Para obter mais informações, consulte Autenticação e Autorização.

- Base de Funções O nome distinto do nó na DIT que será pesquisado por funções. As funções podem ser configuradas como entradas de grupo LDAP explícitas em seu diretório. Uma entrada de função típica pode consistir em um atributo para o nome da função, como Nome comum (NC) e outro atributo, como `member`, com valores que representam os nomes distintos ou nomes de usuário dos usuários pertencentes ao grupo de funções. Por exemplo, dada a unidade organizacional,`group`, você pode fornecer o seguinte nome distinto: `ou=group,dc=example,dc=com`.
- Correspondência de pesquisa de usuários O filtro de pesquisa LDAP que será usado para localizar usuários na `roleBase`. O nome distinto do usuário resultante de comparado com `userSearchMatching` será substituído no espaço `{0}` reservado no filtro de pesquisa. O nome de usuário do cliente será substituído no lugar do `{1}` espaço reservado. Por exemplo, se as entradas de função em seu diretório incluírem um atributo chamado `member`, contendo os nomes de usuários para todos os usuários nessa função, você pode fornecer o seguinte filtro de pesquisa: `(member:=uid={1})`.

A imagem a seguir destaca onde especificar esses detalhes.



Nas Configurações opcionais, você pode fornecer as seguintes informações opcionais:

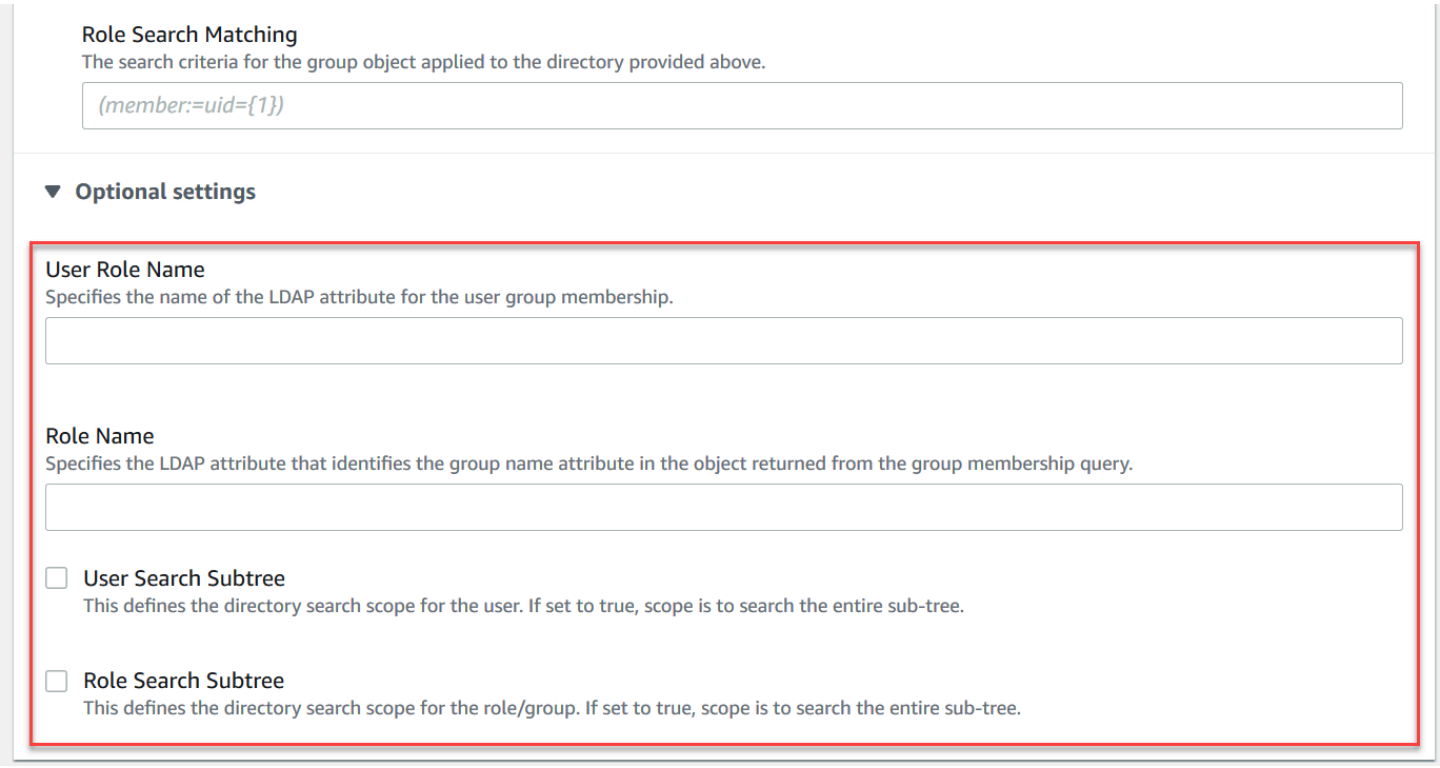
- Nome da Função do Usuário O nome do atributo LDAP na entrada do diretório do usuário para a associação a grupos do usuário. Em alguns casos, as funções de usuário podem ser identificadas pelo valor de um atributo na entrada do diretório do usuário. A opção `userRoleName` permite que você forneça o nome desse atributo. Por exemplo, vamos considerar a seguinte entrada de usuário:

```
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 fornecer o `userRoleName` correto para o exemplo acima, você deve especificar o `memberOf` atributo. Se a autenticação for bem-sucedida, o usuário receberá a função `role1`.

- Nome da Função O atributo do nome do grupo em uma entrada de função cujo valor é o nome dessa função. Por exemplo, você pode especificar `cn` para o nome comum de uma entrada de grupo. Se a autenticação for bem-sucedida, o usuário receberá o valor do `cn` atributo para cada entrada de função da qual ele é membro.
- Sub-árvore de pesquisa de usuários Define o escopo da consulta de pesquisa do usuário LDAP. Se verdadeiro, o escopo é definido para pesquisar toda a sub-árvore sob o nó definido por `userBase`.
- Sub-árvore de pesquisa de usuários Define o escopo da consulta de pesquisa do usuário LDAP. Se verdadeiro, o escopo é definido para pesquisar toda a sub-árvore sob o nó definido por `roleBase`.

A imagem a seguir destaca onde especificar essas configurações opcionais.



Como funciona a integração com LDAP

Podemos pensar na integração em duas categorias principais: a estrutura de autenticação e a estrutura de autorização.

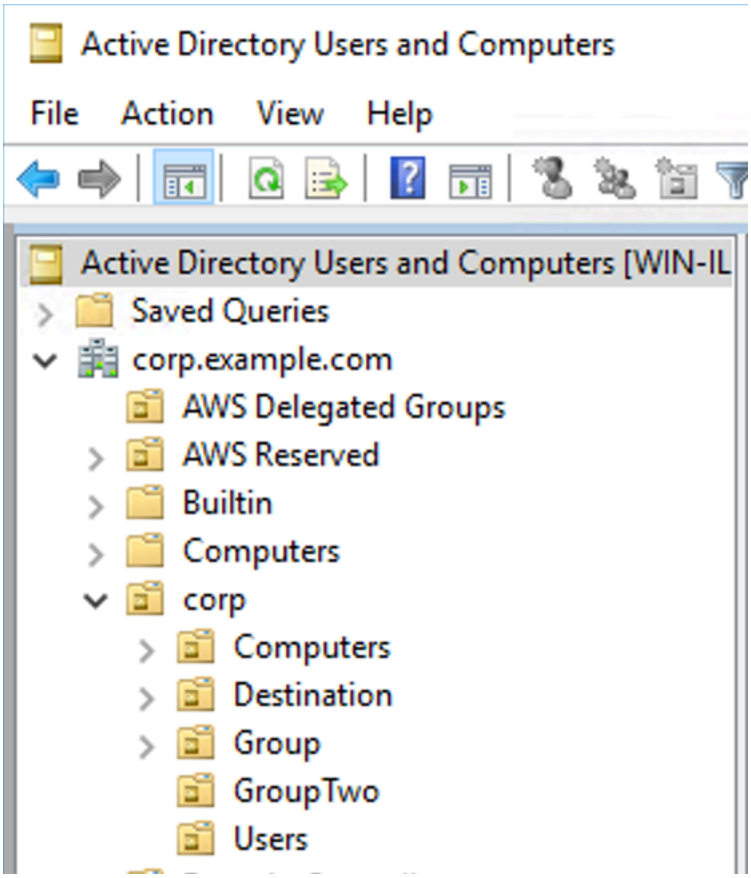
Autenticação

Para autenticação, as credenciais do cliente devem ser válidas. Essas credenciais são validadas em relação aos usuários na base de usuários no servidor LDAP.

A base de usuários fornecida ao agente ActiveMQ deve apontar para o nó na DIT onde os usuários são armazenados no servidor LDAP. Por exemplo, se você estiver usando AWS Managed Microsoft AD, e tiver os componentes de domínio`corp`, e `examplecom`, e dentro deles você tiver unidades organizacionais `corp` e`Users`, você usaria o seguinte como sua base de usuários:

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

O agente do ActiveMQ procuraria usuários nesse local na DIT para autenticar solicitações de conexão do cliente para o broker.



Como o código-fonte ActiveMQ codifica o nome do atributo para os usuários como `uid`, você deve se certificar de que cada usuário tem esse atributo definido. Para simplificar, você pode usar o nome de usuário da conexão do usuário. Para obter mais informações, consulte o ActiveMQ Código-fonte e Configurando mapeamentos de ID em Usuários e Computadores do Active Directory para Windows Server 2016 (e versões subsequentes).

Para habilitar o acesso ao console do ActiveMQ para usuários específicos, verifique se eles pertencem ao grupo `amazonmq-console-admins`.

Autorização

Para autorização, as bases de pesquisa de permissões são especificadas na configuração do agente. A autorização é feita por destino (ou caractere coringa, destino definido) por meio do `cachedLdapAuthorizationMap` elemento encontrado no `activemq.xml` Arquivo de configuração. Para obter mais informações, consulte o Módulo de autorização LDAP armazenado em cache.

> **Note**
>
> Para poder usar o `cachedLDAPAuthorizationMap` elemento no arquivo de `activemq.xml` configuração do seu agente, você deve escolher a opção Autenticação e Autorização LDAP ao criar uma configuração por meio do Console de gerenciamento da AWS, definir a criação de uma configuração por meio do Console de gerenciamento da AWS, ou definir a `authenticationStrategy`propriedade como LDAP ao criar uma nova configuração usando a API do Amazon MQ.

Você deve fornecer os três atributos a seguir como parte do Elemento `cachedLDAPAuthorizationMap`:

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

> **Important**
>
> Para evitar que informações confidenciais sejam colocadas diretamente no arquivo de configuração do agente, o Amazon MQ bloqueia que os seguintes atributos sejam usados no `cachedLdapAuthorizationMap`:
>
> - `connectionURL`
> - `connectionUsername`
> - `connectionPassword`
>
> Quando você cria um agente, o Amazon MQ substitui os valores fornecidos por meio de Console de gerenciamento da AWS, ou na `ldapServerMetadata`propriedade de sua solicitação de API, pelos atributos acima.

O seguinte exemplo demonstra o uso de deslocamentos `cachedLdapAuthorizationMap`.

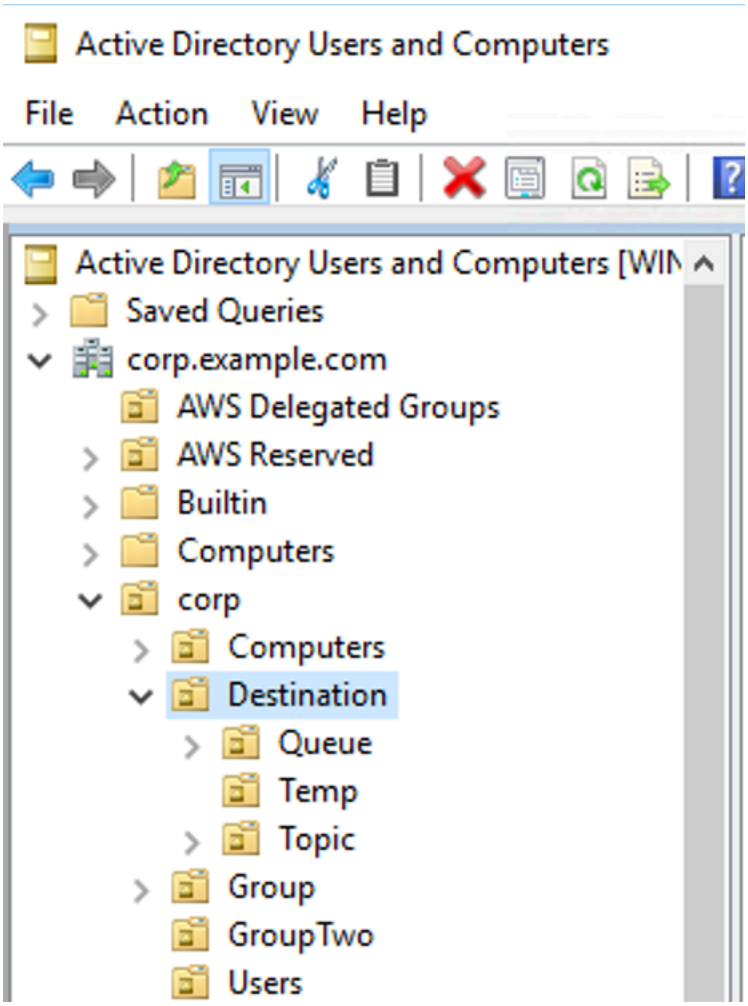
```
<authorizationPlugin>
    <map>
```

```
        <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>
```

Esses valores identificam os locais dentro da DIT onde as permissões para cada tipo de destino são especificadas. Portanto, para o exemplo acima AWS Managed Microsoft AD, usando os mesmos componentes de domínio `decorp`,, e `examplecom`, você especificaria uma unidade organizacional nomeada `destination` para conter todos os seus tipos de destino. Dentro dessa UO, você criaria um para `queues`, um para `topics`, e um para `temp`destinos.

Isso significaria que sua base de pesquisa de fila, que fornece informações de autorização para destinos da fila de tipo, teria o seguinte local em sua DIT:

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



Da mesma forma, as regras de permissões para tópicos e destinos temporários estariam localizadas no mesmo nível na 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 da UO para cada tipo de destino (fila, tópico, temporário), um coringa ou um nome de destino específico pode ser fornecido. Por exemplo, para fornecer uma regra de autorização para todas as filas que começam com o prefixo DEMO.EVENTS.$., você pode criar a seguinte OU:

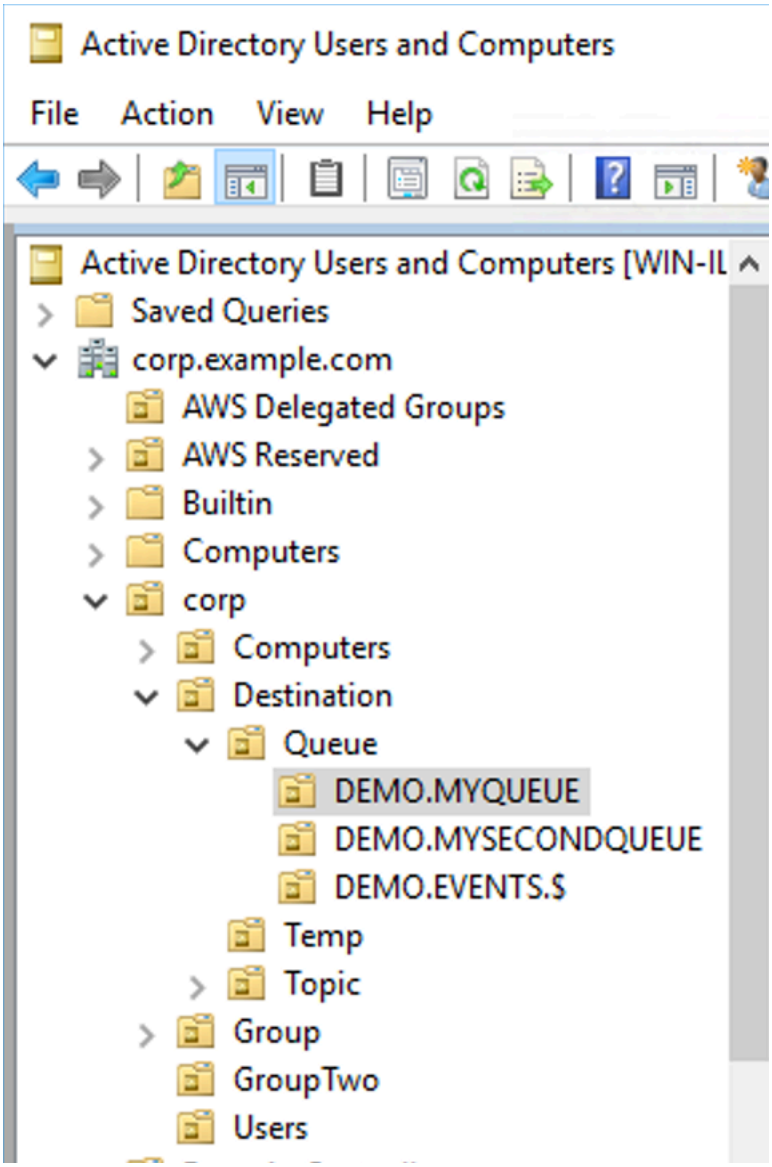
```
OU=DEMO.EVENTS.$,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```

> **Note**
>
> A `DEMO.EVENTS.$` UO está dentro da `Queue` UO.

Para obter mais informações sobre coringas no ActiveMQ, consulteWildcards (Coringas)

Para fornecer regras de autorização para filas específicas, como DEMO.MYQUEUE, especifique algo como o seguinte:

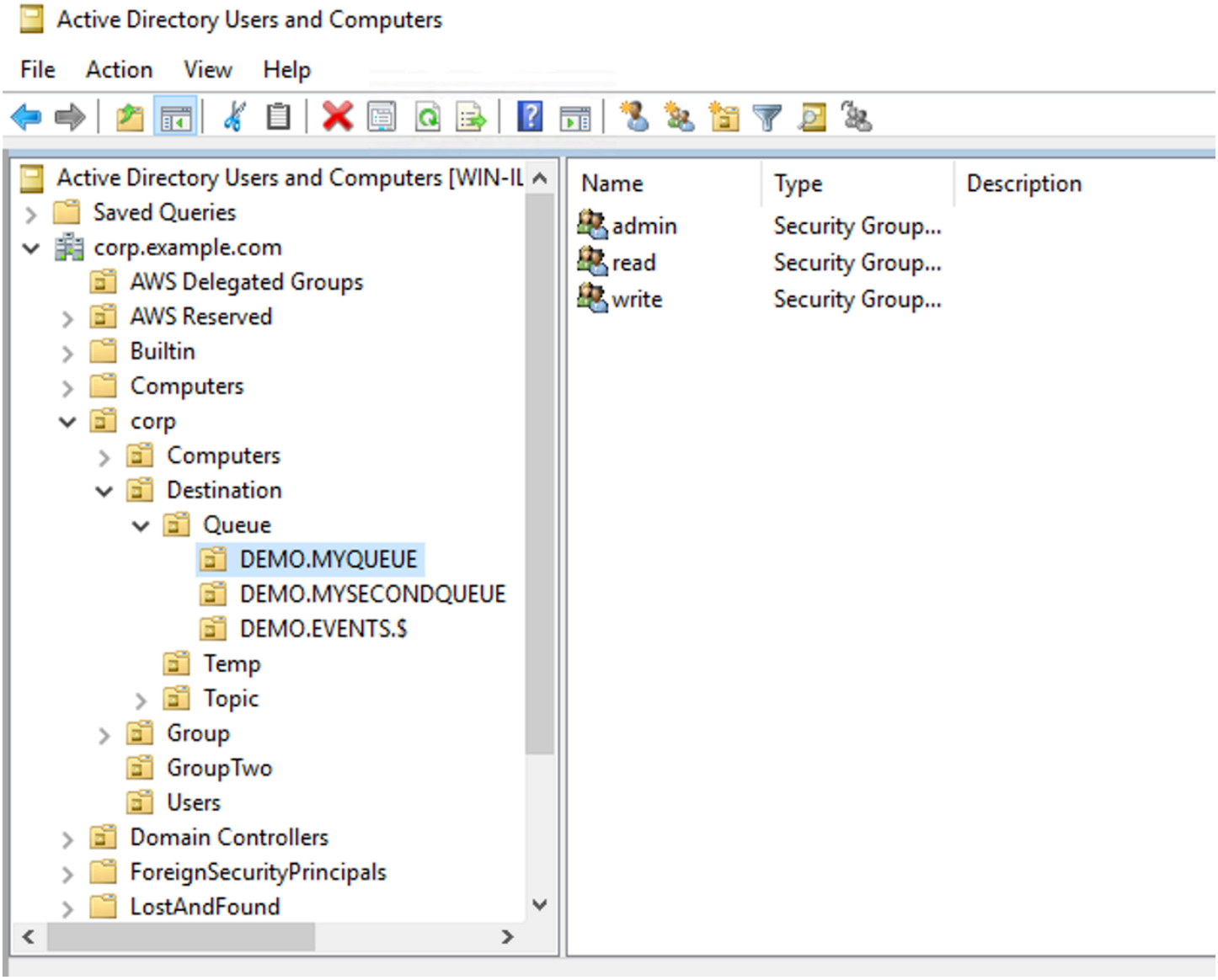
```
OU=DEMO.MYQUEUE,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



Grupos de segurança

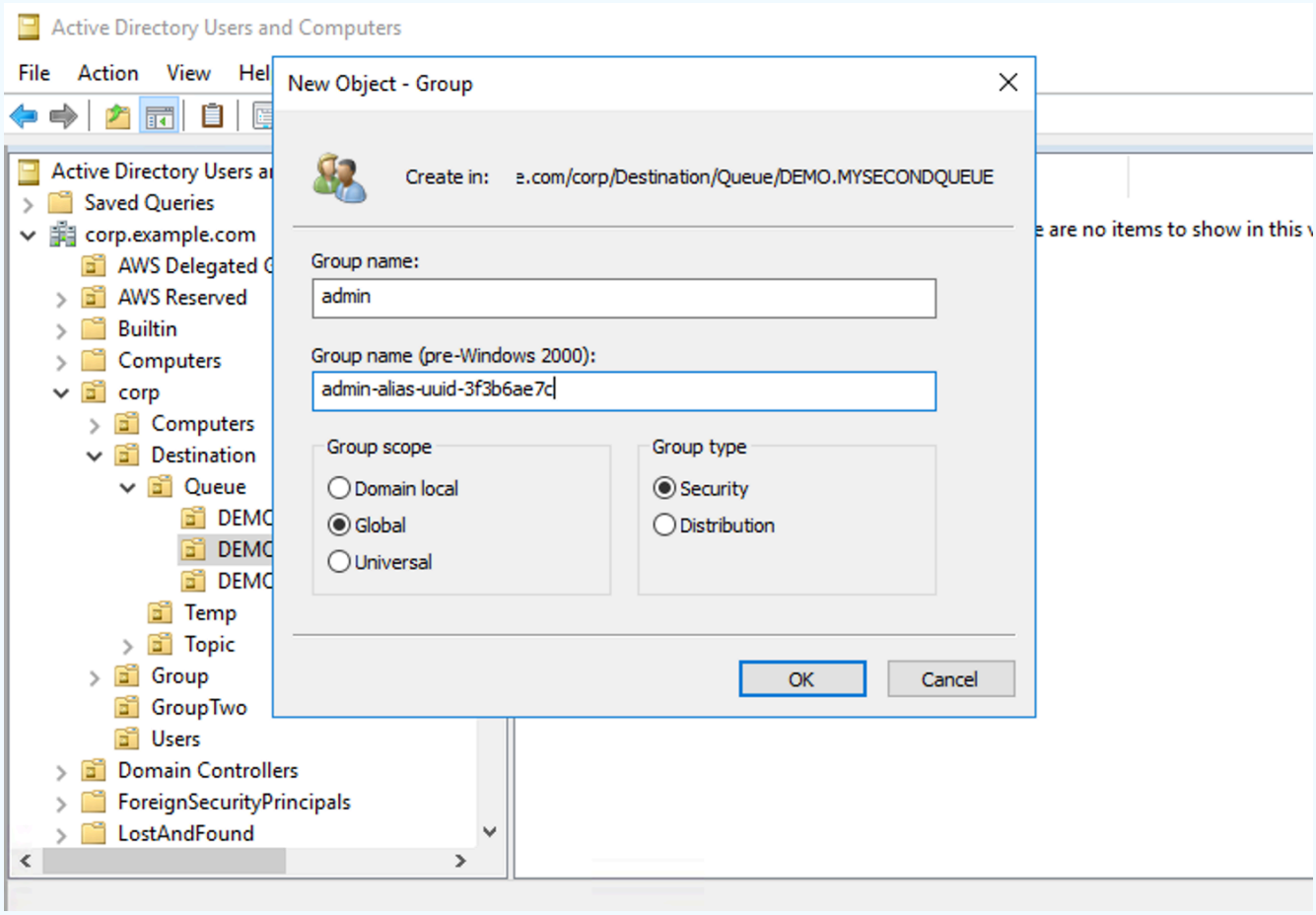
Dentro de cada UO que representa um destino ou um coringa, você deve criar três grupos de segurança. Como acontece com todas as permissões no ActiveMQ, essas são permissões. read/write/admin Para obter mais informações sobre o que cada uma dessas permissões permite que um usuário faça, consulte Segurança na documentação do ActiveMQ.

Você deve nomear esses grupos de segurança `read`, `write`, e `admin`. Dentro de cada um desses grupos de segurança, você pode adicionar usuários ou grupos que terão permissão para executar as ações associadas. Você precisará desses grupos de segurança para cada conjunto de destinos coringa ou destino individual.



Note

Quando você cria o grupo de administração, surgirá um conflito com o nome do grupo. Esse conflito ocorre porque as regras anteriores ao Windows 2000 herdadas não permitem que grupos compartilhem o mesmo nome, mesmo que os grupos estejam em locais diferentes da DIT. O valor na caixa de texto pré-Windows 2000 não tem impacto na configuração, mas deve ser globalmente única. Para evitar esse conflito, você pode acrescentar um `uuid` sufixo para cada `admin` grupo.



Adicionar um usuário ao `admin` grupo de segurança para um destino específico permitirá que o usuário crie e exclua esse tópico. Adicionando-os ao `read` grupo de segurança permitirá que eles leiam a partir do destino e adicionando-os ao grupo `write` permitirá que eles escrevam no destino.

Além de adicionar usuários individuais às permissões do grupo de segurança, você também pode adicionar grupos inteiros. No entanto, como o ActiveMQ novamente codifica nomes de atributo para

grupos, você deve garantir que o grupo que deseja adicionar tem a classe de objeto `groupOfNames`, conforme mostrado no código-fonte do ActiveMQ.

Para fazer isso, siga o mesmo processo que acontece com o `uid` para usuários. Consulte Configurando mapeamentos de ID em Usuários e Computadores do Active Directory para Windows Server 2016 (e versões subsequentes).

Etapa 3: (opcional) conectar-se a uma AWS Lambda função

AWS Lambda pode se conectar e consumir mensagens do seu agente Amazon MQ. Quando você conecta um agente ao Lambda, você cria um Mapeamento da origem do evento que lê mensagens de uma fila e invoca a função sincronicamente. O mapeamento da origem do evento que você cria lê mensagens de seu agente em lotes e as converte em uma carga útil do Lambda na forma de um objeto JSON.

Para conectar seu agente a uma função do Lambda

1. Adicione as permissões de Função do IAM a seguir à sua função de execução da função Lambda.

   - metros quadrados: DescribeBroker
   - ec2: CreateNetworkInterface
   - ec2: DeleteNetworkInterface
   - ec2: DescribeNetworkInterfaces
   - ec2: DescribeSecurityGroups
   - ec2: DescribeSubnets
   - ec2: DescribeVpcs
   - registros: CreateLogGroup
   - registros: CreateLogStream
   - registros: PutLogEvents
   - gerente de segredos: GetSecretValue

   > **Note**
   >
   > Sem as permissões necessárias do IAM, sua função não poderá ler registros com êxito dos recursos do Amazon MQ.

2. (Opcional) Se você criou um agente sem acessibilidade pública, você deve fazer um dos seguintes procedimentos para permitir que o Lambda se conecte ao seu agente:

   - Configure um gateway NAT por sub-rede pública. Para obter mais informações, consulte Acesso aos serviços e à Internet para funções conectadas à VPC no AWS Lambda Guia do desenvolvedor.
   - Crie uma conexão entre a Amazon Virtual Private Cloud (Amazon VPC) e o Lambda usando um endpoint da VPC. Sua Amazon VPC também deve se conectar aos endpoints AWS Security Token Service (AWS STS) e Secrets Manager. Para obter mais informações, consulte Configurar endpoints da VPC de interface para o Lambda no Guia do desenvolvedor AWS Lambda .

3. Configure seu agente como uma origem do evento para uma função do Lambda usando Console de gerenciamento da AWS. Você também pode usar o `create-event-source-mapping` AWS Command Line Interface comando.

4. Escreva algum código para sua função do Lambda para processar as mensagens consumidas pelo seu agente. A carga útil do Lambda recuperada pelo mapeamento da origem do evento depende do tipo de mecanismo do agente. Veja a seguir um exemplo de uma carga útil do Lambda para uma fila do Amazon MQ para ActiveMQ.

   > **Note**
   >
   > No exemplo, `testQueue` é o nome da fila.

```
{
  "eventSource": "aws:amq",
  "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
  "messages": {
    [
      {
        "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
        "messageType": "jms/text-message",
        "data": "QUJDOkFBQUE=",
        "connectionId": "myJMSCoID",
        "redelivered": false,
        "destination": {
          "physicalname": "testQueue"
```

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

Para obter mais informações sobre como conectar o Amazon MQ ao Lambda, as opções com as que o Lambda é compatível para uma origem de evento do Amazon MQ e erros de mapeamento da origem do evento, consulte Usar o Lambda com o Amazon MQ no Guia do desenvolvedor AWS Lambda .

Criar um usuário do agente do ActiveMQ

Um usuário do ActiveMQ é uma pessoa ou uma aplicação que pode acessar as filas e tópicos de um agente ActiveMQ. Você pode configurar usuários para que tenham permissões específicas. Por exemplo, é possível permitir que alguns usuários acessem o Console da Web ActiveMQ.

Um grupo é um rótulo semântico. Você pode atribuir um grupo a um usuário e configurar permissões para grupos para enviar, receber e administrar filas e tópicos específicos.

> **Note**
>
> Não é possível configurar grupos independentemente dos usuários. Um rótulo de grupo é criado quando você adiciona pelo menos um usuário a ele e é excluído quando você remove todos os usuários dele.

> **Note**
>
> O grupo `activemq-webconsole` do ActiveMQ no Amazon MQ tem permissões de administrador em todas as filas e tópicos. Todos os usuários desse grupo terão acesso de administrador.

Os exemplos a seguir mostram como criar, editar e excluir usuários de agente do Amazon MQ utilizando o Console de gerenciamento da AWS.

Criar um usuário do agente do ActiveMQ

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, escolha o nome do seu corretor (por exemplo, MyBroker) e escolha Exibir detalhes.

   Na ***MyBroker***página, na seção Usuários, todos os usuários dessa corretora estão listados.

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

3. Selecione Criar usuário.
4. Na caixa de diálogo Create user (Criar usuário), digite um Username (Nome de usuário) e uma Password (Senha).
5. (Opcional) Digite os nomes dos grupos aos quais o usuário pertence, separados por vírgulas (por exemplo: `Devs, Admins`).
6. (Opcional) Para permitir que o usuário acesse o Console da Web ActiveMQ, selecione Console da Web ActiveMQ.

7. Selecione Criar usuário.

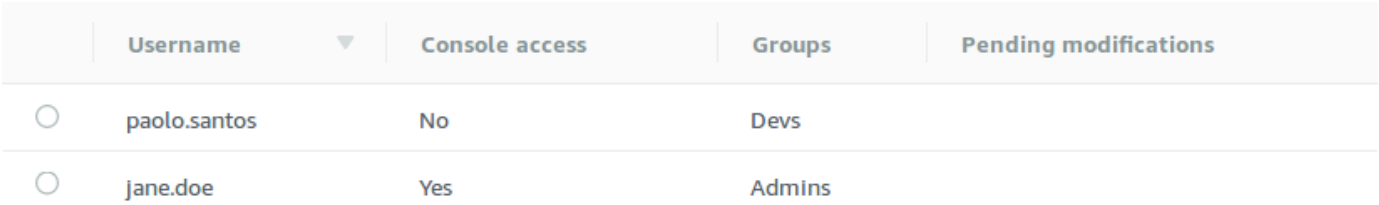
> **Important**
>
> Fazer alterações em um usuário não aplica as alterações ao usuário imediatamente. Para aplicar as alterações, você deve aguardar a próxima janela de manutenção ou reiniciar o agente.

Editar um usuário do agente do ActiveMQ

Para editar um usuário existente, faça o seguinte:

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, escolha o nome do seu corretor (por exemplo, MyBroker) e escolha Exibir detalhes.

   Na ***MyBroker***página, na seção Usuários, todos os usuários dessa corretora estão listados.



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

3. Escolha suas credenciais de login e selecione Editar.

   A caixa de diálogo Edit user (Editar usuário) será exibida.
4. (Opcional) Digite uma nova Password (Senha).
5. (Opcional) Adicione ou remova os nomes dos grupos aos quais o usuário pertence, separados por vírgulas (por exemplo: `Managers, Admins`).
6. (Opcional) Para permitir que o usuário acesse o Console da Web ActiveMQ, selecione Console da Web ActiveMQ.
7. Para salvar as alterações do usuário, selecione Done (Concluído).

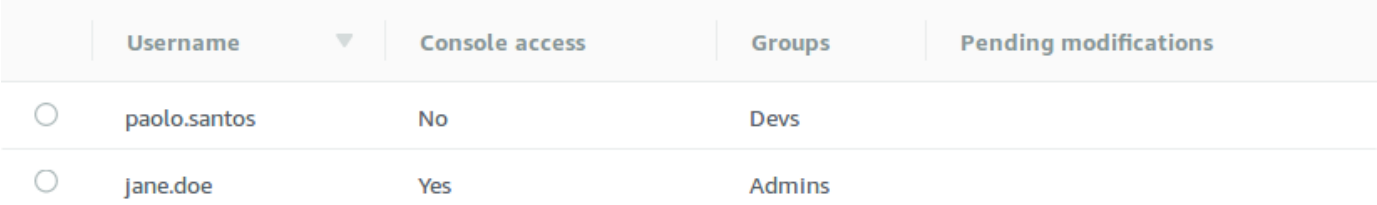
> **Important**
>
> Fazer alterações em um usuário não aplica as alterações ao usuário imediatamente. Para aplicar as alterações, você deve aguardar a próxima janela de manutenção ou reiniciar o agente.

Excluir um usuário do agente do ActiveMQ

Quando você não precisar mais de um usuário, exclua-o.

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, escolha o nome do seu corretor (por exemplo, MyBroker) e escolha Exibir detalhes.

   Na ***MyBroker***página, na seção Usuários, todos os usuários dessa corretora estão listados.



3. Selecione suas credenciais de login (por exemplo, ***MyUser***) e escolha Excluir.
4. Para confirmar a exclusão do usuário, na seção Excluir ***MyUser***? caixa de diálogo, escolha Excluir.

> **Important**
>
> Fazer alterações em um usuário não aplica as alterações ao usuário imediatamente. Para aplicar as alterações, você deve aguardar a próxima janela de manutenção ou reiniciar o agente.

Exemplos funcionais de Como usar o Java Message Service (JMS) com o ActiveMQ

Os exemplos a seguir mostram como você pode trabalhar com o ActiveMQ programaticamente:

- O código Java de OpenWire exemplo se conecta a um corretor, cria uma fila e envia e recebe uma mensagem. Para detalhamento e explicação, consulte Connecting a Java application to your broker.
- O código do exemplo Java de MQTT faz uma conexão com um agente e cria um tópico além de enviar e receber e uma mensagem.
- O código do exemplo Java de STOMP+WSS faz uma conexão com um agente e cria uma fila, além de enviar e receber e uma mensagem.

Pré-requisitos

Habilitar atributos da VPC

Para garantir que seu agente esteja acessível dentro da sua VPC, você deve habilitar os atributos VPC `enableDnsHostnames` e `enableDnsSupport`. Para obter mais informações, consulte Compatibilidade com DNS para a sua VPC no Manual do usuário da Amazon VPC.

Habilitar conexões de entrada

Para trabalhar com o Amazon MQ de forma programática, você deve usar conexões de entrada.

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, escolha o nome do seu corretor (por exemplo, MyBroker).
3. Na ***`MyBroker`***página, na seção Conexões, observe os endereços e portas do URL do console web e dos protocolos de nível de fio do broker.
4. Na seção Details (Detalhes), em Security and network (Segurança e rede), escolha o nome do seu grupo de segurança ou

   .

   A página Grupos de segurança do painel do EC2 é exibida.
5. Na lista de security group, escolha seu security group.
6. Na parte inferior da página, escolha Inbound (Entrada) e a seguir selecione Edit (Editar).
7. Na caixa de diálogo Edit inbound rules (Editar regras de entrada), adicione uma regra para cada URL ou endpoint que você deseja que seja acessível publicamente (o exemplo a seguir mostra como fazer isso para um console da Web do agente).

   a. Escolha Add Rule (Adicionar regra).

   b. Em Type (Tipo), selecione Custom TCP (TCP personalizado).

c. Para o Intervalo de Portas, digite a porta do console da Web (8162).

d. Para Source (Origem), deixe Custom (Personalizado) selecionado e, depois, digite o endereço IP do sistema ao qual deseja ser capaz de acessar o console da Web (por exemplo, `192.0.2.1`).

e. Escolha Salvar.

Agora seu agente pode aceitar conexões de entrada.

Adicionar dependências de Java

OpenWire

Adicione os pacotes `activemq-client.jar` e `activemq-pool.jar` ao caminho da classe Java. O exemplo a seguir mostra essas dependências em um arquivo `pom.xml` do projeto 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 obter mais informações sobre `activemq-client.jar`, consulte Initial Configuration (Configuração inicial) na documentação do Apache ActiveMQ.

MQTT

Adicione o pacote `org.eclipse.paho.client.mqttv3.jar` ao caminho da classe Java. O exemplo a seguir mostra essa dependência em um arquivo `pom.xml` do projeto Maven.

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

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

Para obter mais informações sobre `org.eclipse.paho.client.mqttv3.jar`, consulte Cliente Java Eclipse Paho.

STOMP+WSS

Adicione os seguintes pacotes ao caminho da classe Java:

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

O exemplo a seguir mostra essas dependências em um arquivo `pom.xml` do projeto 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 obter mais informações, consulte Suporte de STOMP na documentação do Spring Framework.

Amazon MQExample .java

> **Important**
>
> No código de exemplo a seguir, os produtores e consumidores são executados em um único thread. Para sistemas de produção (ou para testar o failover de instância do agente), certifique-se de que seus produtores e consumidores sejam executados em hosts ou threads separados.

OpenWire

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

```
 *
 */

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

                    import javax.jms.*;

                    public class AmazonMQExample {

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

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

                        sendMessage(pooledConnectionFactory);
                        receiveMessage(connectionFactory);

                        pooledConnectionFactory.stop();
                    }

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

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

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

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

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

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

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

                    private static void
                    receiveMessage(ActiveMQConnectionFactory connectionFactory)
throws JMSException {
                        // Establish a connection for the consumer.
                        // Note: Consumers should not use PooledConnectionFactory.
                        final Connection consumerConnection =
connectionFactory.createConnection();
                        consumerConnection.start();

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

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

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

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

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

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

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

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

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

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

MQTT

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

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

                    public class AmazonMQExampleMqtt implements MqttCallback {

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

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

                    private void run() throws MqttException, InterruptedException {

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

                        // Create the MQTT client and specify the connection
 options.
                        final String clientId = "abc123";
```

```
                        final MqttClient client = new
MqttClient(WIRE_LEVEL_ENDPOINT, clientId);
                        final MqttConnectOptions connOpts = new
MqttConnectOptions();

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

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

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

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

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

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

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

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

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

```
                    }
                    }
```

STOMP+WSS

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

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

                    import java.lang.reflect.Type;

                    public class AmazonMQExampleStompWss {

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

```
                    private final static String ACTIVE_MQ_PASSWORD =
"MyPassword456";

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

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

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

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

                        final WebSocketHttpHeaders headers = new
WebSocketHttpHeaders();

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

                        final StompSessionHandler sessionHandler = new
MySessionHandler();

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

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

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

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

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

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

Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ

O Apache ActiveMQ organiza números de versão de acordo com a especificação de versionamento semântico como `X.Y.Z`. Nas implementações do Amazon MQ para ActiveMQ, `X` denota a versão principal, `Y` representa a versão secundária e `Z` denota o número da versão de patch. O Amazon MQ considera que uma alteração de versão é principal se os números de versão principais mudarem. Por exemplo, a atualização da versão 5.17 para a 6.0 é considerada uma atualização de versão principal.

Uma alteração da versão é considerada secundária se apenas o número da versão secundária ou de patch for alterado. Por exemplo, atualizando a partir da versão 5. 18 a 5. 19 é considerado um pequeno upgrade de versão. Quando `autoMinorVersionUpgrade` está ativado, o Amazon MQ atualiza o agente para a versão de patch mais recente disponível.

O Amazon MQ para ActiveMQ recomenda que todos os agentes usem a versão secundária mais recente compatível. Para obter instruções de como atualizar a versão do mecanismo do agente, consulte Upgrading an Amazon MQ broker engine version.

Versões do mecanismo compatíveis no Amazon MQ para ActiveMQ

O calendário de suporte da versão do Amazon MQ indica quando uma versão do mecanismo do agente chegará ao fim do suporte. Quando uma versão chega ao fim do suporte, o Amazon MQ atualiza automaticamente todos os agentes dessa versão para a próxima versão compatível. Essa atualização ocorre durante as janelas de manutenção programada da sua corretora, dentro dos 45 dias após a end-of-support data.

O Amazon MQ avisa com pelo menos noventa dias de antecedência quando uma versão chegará ao fim do suporte. Recomendamos atualizar seu corretor antes da end-of-support data para evitar interrupções. Além disso, não é possível criar agentes em versões programadas para o fim do suporte dentro de trinta dias da data do fim do suporte.

Versão do Apache ActiveMQ	Fim do suporte no Amazon MQ
ActiveMQ 5.19 (recomendado)	
ActiveMQ 5.18	
ActiveMQ 5.17	16 de junho de 2025
ActiveMQ 5.16	15 de novembro de 2024
ActiveMQ 5.15	16 de setembro de 2024

Ao criar um novo agente do Amazon MQ para ActiveMQ, você pode especificar qualquer versão do mecanismo do ActiveMQ compatível. Se você não especificar o número da versão do mecanismo ao criar um agente, o Amazon MQ automaticamente definirá como padrão o número da versão mais recente do mecanismo.

Atualizações da versão do mecanismo

Você pode atualizar seu agente manualmente a qualquer momento para a próxima versão principal ou secundária compatível. Quando você ativa Atualizações automáticas de versões secundárias, o Amazon MQ atualiza seu agente para a versão de patch mais recente compatível durante a janela de manutenção.

Para obter mais informações sobre como atualizar seu agente manualmente, consulte the section called “Atualizar a versão do mecanismo”.

Listando as versões compatíveis do mecanismo

Você pode listar todas as versões de mecanismos secundários e principais compatíveis usando o `describe-broker-instance-options` AWS CLI comando.

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

Para filtrar os resultados por mecanismo e tipo de instância, use a opção `--engine-type` e `--host-instance-type`, conforme mostrado a seguir.

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

Por exemplo, para filtrar os resultados do ActiveMQ e do tipo de instância, *`engine-type`* `ACTIVEMQ` substitua por `mq.m5.large` e com. *`instance-type`* `mq.m5.large`

Práticas recomendadas do Amazon MQ para ActiveMQ

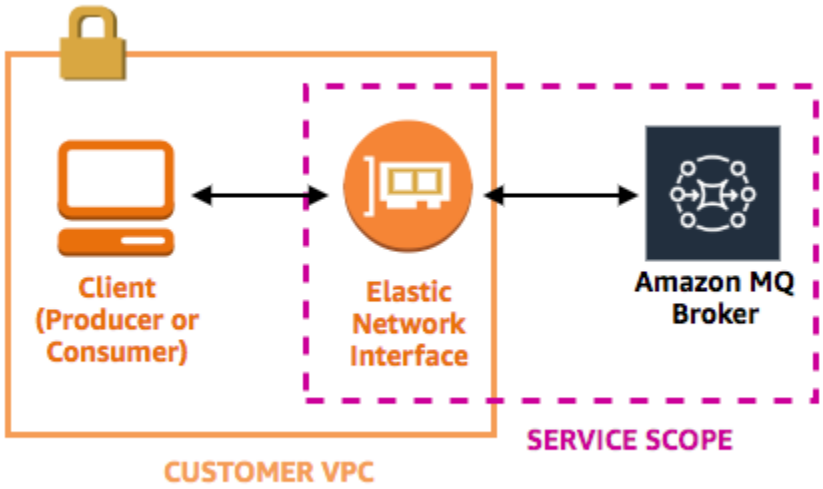
Use esta seção como referência para localizar rapidamente as recomendações para maximizar a performance e minimizar os custos de taxa de transferência para trabalhar com agentes do ActiveMQ no Amazon MQ.

Nunca modifique ou exclua a interface de rede elástica do Amazon MQ

Quando você cria um agente do Amazon MQ pela primeira vez, o Amazon MQ provisiona uma interface de rede elástica na Virtual Private Cloud (VPC) em sua conta e, por isso, requer uma série de permissões do EC2. A interface de rede permite que seu cliente (produtor ou consumidor) se

comunique com o agente do Amazon MQ. Considera-se que a interface de rede está dentro do escopo de serviço do Amazon MQ, apesar de fazer parte da VPC de sua conta.

> **Warning**
>
> Você não deve modificar ou excluir essa interface de rede. Modificar ou excluir a interface de rede pode causar uma perda permanente de conexão entre a VPC e o operador.



Sempre usar pooling de conexão

Em um cenário com um único produtor e um único consumidor (como o Conceitos básicos: criar e conectar a um agente do ActiveMQ tutorial), você pode usar uma única `ActiveMQConnectionFactory` classe para cada produtor e consumidor. Por exemplo:

```
// 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();
```

No entanto, em cenários mais realistas com vários produtores e consumidores, pode ser dispendioso e ineficiente criar um grande número de conexões para vários produtores. Nesses cenários, você deve agrupar solicitações de vários produtores usando a classe `PooledConnectionFactory`. Por exemplo:

> **Note**
>
> Os consumidores de mensagens nunca devem usar a classe `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();
```

Sempre usar o transporte de failover para conectar-se a vários endpoints de operador

Se você precisar que a aplicação se conecte a vários endpoints do agente — por exemplo, ao usar um modo de implantação ativo/em espera ou ao migrar de um agente de mensagens no local para o Amazon MQ — use o transporte de failover para permitir que os consumidores se conectem aleatoriamente a um deles. Por exemplo:

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

> **Important**
>
> Os agentes de várias zonas de disponibilidade podem experimentar failovers durante janelas de manutenção e reinicializações de agentes. Use o Transporte de failover para garantir a disponibilidade do seu corretor.

Evite usar seletores de mensagens

É possível usar seletores JMS para anexar filtros às assinaturas de tópico (para rotear mensagens a consumidores com base no conteúdo). No entanto, o uso de seletores JMS ocupa o buffer do filtro do agente do Amazon MQ, impedindo a filtragem de mensagens.

Em geral, evite permitir que os consumidores roteiem mensagens, pois, para um bom desacoplamento de consumidores e produtores, ambos devem ser temporários.

Preferir destinos virtuais a assinaturas duráveis

Uma assinatura durável pode ajudar a garantir que o consumidor receba todas as mensagens publicadas em um tópico, por exemplo, após a restauração de uma conexão perdida. No entanto, o uso de assinaturas duráveis também impede o uso de consumidores da concorrência e pode apresentar problemas de performance em escala. Considere o uso de destinos virtuais, em vez disso.

Se estiver usando o emparelhamento de Amazon VPC, evite clientes IPs na faixa de CIDR `10.0.0.0/16`

Se você estiver configurando o emparelhamento do Amazon VPC entre a infraestrutura local e seu agente do Amazon MQ, você não deve configurar conexões de clientes dentro do intervalo CIDR. IPs `10.0.0.0/16`

Desativar o armazenamento e a expedição simultâneos para filas com consumidores lentos

Por padrão, o Amazon MQ é otimizado para filas com consumidores rápidos:

- Os consumidores são considerados rápidos se conseguem acompanhar a taxa de mensagens geradas pelos produtores.

- Os consumidores são considerados lentos se uma fila cria um acúmulo de mensagens não confirmadas, o que pode reduzir a taxa de transferência do produtor.

Para instruir o Amazon MQ para ser otimizado para filas com consumidores lentos, defina o atributo `concurrentStoreAndDispatchQueues` como `false`. Para obter uma configuração de exemplo, consulte `concurrentStoreAndDispatchQueues`.

Selecionar o tipo de instância de agente correto para obter a melhor taxa de transferência

A taxa de transferência de mensagens de um tipo de instância de agentedepende do caso de uso da aplicação e dos seguintes fatores:

- Uso do ActiveMQ no modo persistente
- Tamanho da mensagem
- O número de produtores e consumidores
- O número de destinos

Noções básicas sobre o relacionamento entre o tamanho, a latência e a taxa de transferência de mensagens

Dependendo do caso de uso, um tipo de instância de agente maior pode não necessariamente melhorar a taxa de transferência do sistema. Quando o ActiveMQ grava mensagens em um armazenamento durável, o tamanho de suas mensagens determina o fator limitante do sistema:

- Se as mensagens forem menores que 100 KB, a latência do armazenamento persistente será o fator limitante.
- Se as mensagens forem maiores que 100 KB, a taxa de transferência do armazenamento persistente será o fator limitante.

Ao usar o ActiveMQ no modo persistente, a gravação no armazenamento ocorrerá normalmente quando houver alguns consumidores ou quando os consumidores forem lentos. No modo não persistente, a gravação no armazenamento também ocorrerá com consumidores lentos se a memória do heap da instância de agente estiver cheia.

Para determinar o melhor tipo de instância de agente para a sua aplicação, recomendamos testar diferentes tipos de instância de operador. Para obter mais informações, consulte Broker instance types e também Medição da taxa de transferência para o Amazon MQ usando o benchmark JMS.

Casos de uso de tipos de instância de agente maiores

Há três casos de uso comuns quando tipos de instância de agente maiores melhoram a taxa de transferência:

- Modo não persistente: quando sua aplicação é menos sensível à perda de mensagens durante o failover de instância de agente (por exemplo, ao transmitir placares de esportes), muitas vezes você pode usar o modo não persistente do ActiveMQ. Nesse modo, o ActiveMQ grava mensagens no armazenamento persistente somente se a memória do heap da instância de agente está cheia. Os sistemas que usam o modo não persistente podem se beneficiar da quantidade maior de memória, CPU mais rápida e redes mais rápidas e disponíveis em tipos de instância de agente maiores.
- Consumidores rápidos: quando os consumidores ativos estão disponíveis e o sinalizador `concurrentStoreAndDispatchQueues` está habilitado, o ActiveMQ permite o fluxo das mensagens diretamente do produtor para o consumidor sem enviar mensagens ao armazenamento (mesmo em modo persistente). Se a sua aplicação pode consumir mensagens rapidamente (ou se você pode projetar seus consumidores para fazer isso), a aplicação pode se beneficiar de um tipo de instância de agente maior. Para permitir que seu aplicativo consuma mensagens com mais rapidez, adicione threads de consumidor às instâncias do aplicativo ou expanda as instâncias do aplicativo verticalmente ou horizontalmente.
- Transações em lote: quando você usa o modo persistente e envia várias mensagens por transação, você pode obter uma taxa de transferência de mensagens em geral mais alta usando tipos de instância de agente maiores. Para obter mais informações, consulte Devo usar transações? na documentação do ActiveMQ.

Escolha o tipo de armazenamento de agente correto para obter a melhor taxa de transferência

Para aproveitar a alta durabilidade e a replicação em várias zonas de disponibilidade, use o Amazon EFS. Para aproveitar a baixa latência e alta taxa de transferência, use o Amazon EBS. Para obter mais informações, consulte Storage.

Configurar sua rede de agentes corretamente

Quando você cria uma rede de agentes, configure-a corretamente para seu aplicativo:

- Ativar modo persistente: Como (em relação a seus pares) cada instância de agente atua como um produtor ou um consumidor, redes de agentes não fornecem a replicação distribuída de mensagens. O primeiro agente que atua como um consumidor recebe uma mensagem e a mantém para armazenamento. Esse agente envia uma confirmação para o produtor e encaminha a mensagem para o próximo agente. Quando o segundo agente reconhece a persistência da mensagem, o primeiro agente exclui a mensagem.

  Se o modo persistente é desativado, o primeiro agente reconhece o produtor sem manter a mensagem para armazenamento. Para obter mais informações, consulte Armazenamento de mensagem replicada e Qual é a diferença entre entrega persistente e não persistente? na documentação do Apache ActiveMQ.
- Não desative mensagens de aviso para instâncias de agente: Para obter mais informações, consulte Mensagens de aviso na documentação do Apache ActiveMQ.
- Não use a descoberta do agente multicast: O Amazon MQ não é compatível com a descoberta do agente usando multicast. Para obter mais informações, consulte Qual é a diferença entre a descoberta, multicast e zeroconf? na documentação do Apache ActiveMQ.

Evite reinicializações lentas recuperando transações XA preparadas

O ActiveMQ oferece suporte a transações distribuídas (XA). Saber como o ActiveMQ processa transações XA pode ajudar a evitar tempos de recuperação mais lentos para reinicializações do agente e failovers no Amazon MQ.

Transações XA preparadas não resolvidas são reproduzidas novamente em cada reinicialização. Se elas permanecerem não resolvidas, o número de transações ficará cada vez maior com o tempo, aumentando significativamente o tempo necessário para inicializar o operador. Isso afeta o tempo de reinicialização e de failover. Você deve resolver essas transações com um `commit()` ou um, de `rollback()` para que não haja degradação de performance ao longo do tempo.

Para monitorar suas transações XA preparadas não resolvidas, você pode usar a `JournalFilesForFastRecovery` métrica no Amazon CloudWatch Logs. Se esse número estiver aumentando ou for consistentemente maior que 1, você deve recuperar suas transações não resolvidas com um código semelhante ao exemplo a seguir. Para obter mais informações, consulte Cotas no Amazon MQ.

O código de exemplo a seguir aborda transações XA preparadas e as encerra com um `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) {
        }
    }
}
```

Em um cenário do mundo real, você pode verificar suas transações XA preparadas em relação ao Gerenciador de transações XA. Em seguida, você pode decidir se deseja tratar de cada transação preparada com um `rollback()` ou um `commit()`.

Usar o Amazon MQ para RabbitMQ

O Amazon MQ facilita a criação de um agente de mensagem com os recursos de processamento e armazenamento que atendem às suas necessidades. Você pode criar, gerenciar e excluir corretores usando a Console de gerenciamento da AWS API REST do Amazon MQ ou a. AWS Command Line Interface

Esta seção descreve os elementos básicos de um agente de mensagens para os tipos de mecanismo ActiveMQ e RabbitMQ, lista os tipos de instâncias de agente de Amazon MQ disponíveis e seus status e fornece uma visão geral da arquitetura de agente e das opções de configuração.

Para saber mais sobre o Amazon MQ REST APIs, consulte a Referência da API REST do Amazon MQ.

O que é um agente do Amazon MQ para RabbitMQ?

Um agente é um ambiente de agente de mensagens em execução no Amazon MQ. É o bloco de criação básico do Amazon MQ. A descrição combinada da classe (`m7g`) e do tamanho (`large`, `medium`) da instância do agente é um tipo de instância de agente (por exemplo, `mq.m7g.large`).

- Um agente de instância única consiste em um agente em uma zona de disponibilidade atrás de um Network Load Balancer (NLB). O agente se comunica com sua aplicação e com um volume de armazenamento do Amazon EBS.
- A implantação de cluster é um agrupamento lógico de três nós do agente RabbitMQ por trás de um Balanceador de Carga da Rede, cada um compartilhando usuários, filas e um estado distribuído em várias Zonas de Disponibilidade (AZ).

Para obter mais informações, consulte Implantando um corretor RabbitMQ.

Portas listener

Os corretores RabbitMQ gerenciados pelo Amazon MQ oferecem suporte às seguintes portas de ouvinte para conectividade em nível de aplicativo via. `amqps` Você também pode usar essas portas para conexões de clientes usando o console web RabbitMQ e a API de gerenciamento. Todas as conexões usam criptografia TLS para segurança.

- Porta do ouvinte `5671` - usada para conexões seguras do AMQP feitas por meio do URL seguro do AMQP. Essa porta suporta os protocolos AMQP 0-9-1 e AMQP 1.0 no

RabbitMQ 4. Por exemplo, considerando um agente com ID de agente `b-c8352341-ec91-4a78-ad9c-a43f23d325bb`, implantado na região `us-west-2`, o seguinte é a URL `amqps` completo do agente: `b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com:5671`.

- Portas de ouvinte `443` e `15671` - Você pode usar as duas portas de ouvinte de forma intercambiável para acessar um corretor por meio do console web do RabbitMQ ou da API de gerenciamento. A porta 443 fornece acesso HTTPS padrão, enquanto a porta 15671 é a porta de gerenciamento tradicional do RabbitMQ com criptografia TLS.

Atributos

Um agente RabbitMQ tem vários atributos:

- Um nome. Por exemplo, .`MyBroker`
- Um ID. Por exemplo, .`b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`
- Um Nome do Recurso da Amazon (ARN). Por exemplo, .`arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`
- Uma URL do console da Web RabbitMQ. Por exemplo, .`https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`

  Para obter mais informações, consulte o Console da Web RabbitMQ na documentação do RabbitMQ.
- Um endpoint AMQP seguro. Por exemplo, .`amqps://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`

Para obter uma lista completa de atributos do agente, consulte o seguinte na Referência de API do Amazon MQ REST:

- ID da operação REST: Operador
- ID da operação REST: Operadores
- ID da operação REST: Reinicialização do operador

Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ

O RabbitMQ organiza números de versão de acordo com a especificação de versionamento semântico como `X.Y.Z`. No Amazon MQ para implementações do RabbitMQ, `X` denota a versão principal, `Y` representa a versão secundária e `Z` denota o número de versão de patch. O Amazon MQ considera que uma alteração de versão é principal se os números de versão principais mudarem. Por exemplo, a atualização da versão 3.13 para a 4.0 é considerada uma atualização importante da versão. Uma alteração da versão é considerada secundária se apenas o número da versão secundária ou de patch for alterado. Por exemplo, atualizando a partir da versão 3. 11,28 a 3. 12.13 é considerado um pequeno upgrade de versão.

O Amazon MQ para RabbitMQ recomenda que todos os corretores usem a versão mais recente compatível, o RabbitMQ 4.2. Para obter instruções de como atualizar a versão do mecanismo do agente, consulte Upgrading an Amazon MQ broker engine version.

Ao criar um novo agente Amazon MQ para RabbitMQ, você só precisa especificar os números da versão principal e secundária. Por exemplo, RabbitMQ 4.2. Se você não especificar a versão do mecanismo ao criar um agente, o Amazon MQ automaticamente usará como padrão a versão mais recente do mecanismo.

> **Important**
>
> O Amazon MQ não é compatível com fluxos. Criar um fluxo resultará em perda de dados. O Amazon MQ não oferece suporte ao uso de registros estruturados em JSON.

O Amazon MQ oferece suporte a duas versões principais do RabbitMQ:

- RabbitMQ 4

  O Amazon MQ oferece suporte ao RabbitMQ 4.2 na série de lançamento do RabbitMQ 4 somente no tipo de instância mq.m7g em todos os tamanhos de instância compatíveis.

- RabbitMQ 3

  O Amazon MQ oferece suporte ao RabbitMQ 3.13 na série de lançamento do RabbitMQ 3 nos tipos de instância mq.t3, mq.m5 e mq.m7g em todos os tamanhos de instância compatíveis.

Listando as versões compatíveis do mecanismo

Você pode listar todas as versões de mecanismos secundários e principais compatíveis usando o `describe-broker-instance-options` AWS CLI comando.

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

Para filtrar os resultados por mecanismo e tipo de instância, use a opção `--engine-type` e `--host-instance-type`, conforme mostrado a seguir.

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

Por exemplo, para filtrar os resultados do RabbitMQ e do tipo de `mq.m7g.large` instância, *`engine-type`* `RABBITMQ` substitua por e por. *`instance-type`* `mq.m7g.large`

RabbitMQ 4

O Amazon MQ oferece suporte ao RabbitMQ 4.2 na série de lançamento do RabbitMQ 4 somente no tipo de instância mq.m7g em todos os tamanhos de instância compatíveis.

> **Important**
>
> Você só pode criar novos corretores no RabbitMQ 4.2. Atualmente, as atualizações em vigor do RabbitMQ 3.13 não são suportadas.

> **Important**
>
> O tipo de fila padrão no Amazon MQ para corretores do RabbitMQ 4.2 será “quorum”. Se nenhum argumento do tipo de fila for especificado durante a criação da fila, uma fila de quórum será criada.
> É altamente recomendável usar filas de quórum no RabbitMQ 4 para necessidades de durabilidade, pois não é garantido que as filas clássicas sejam duráveis em todos os casos.

As seguintes alterações foram introduzidas no RabbitMQ 4 no Amazon MQ

- AMQP 1.0 como protocolo principal: para obter mais informações, consulte Protocolos.

- Pás locais: as escavadeiras agora oferecem suporte a um novo protocolo chamado “local”, além do AMQP 0-9-1 e do AMQP 1.0. Os shovels locais são baseados internamente no AMQP 1.0, mas em vez de usar conexões TCP separadas, eles usam conexões intra-cluster entre os nós do cluster e internas para publicar e consumir mensagens. APIs Isso só pode ser usado para consumo e publicação no mesmo cluster e pode oferecer maior taxa de transferência usando menos recursos do que o AMQP 0-9-1 e o AMQP 1.0.
- As filas de quórum oferecem suporte às prioridades de mensagens: as prioridades de mensagens da fila de quórum estão sempre ativas e não exigem uma política para funcionar. Assim que uma fila de quórum receber uma mensagem com uma prioridade definida, ela habilitará a priorização. As filas de quórum internamente suportam apenas duas prioridades: alta e normal. As mensagens sem um conjunto de prioridades serão mapeadas para o normal, assim como as prioridades de 0 a 4. Mensagens com prioridade maior que 4 serão mapeadas para alta. As mensagens de alta prioridade serão favorecidas em relação às mensagens de prioridade normal na proporção de 2:1, ou seja, para cada 2 mensagens de alta prioridade, a fila entregará 1 mensagem de prioridade normal (se disponível). Portanto, as filas de quórum implementam um tipo de processamento prioritário não estrito e de “compartilhamento justo”. Isso garante que o progresso seja sempre feito nas mensagens de prioridade normal, mas as prioridades altas são favorecidas na proporção de 2:1.
- Khepri: Khepri é usado como o armazenamento de metadados padrão para corretores RabbitMQ 4
- TLS mútuo (mTLS): o Amazon MQ oferece suporte a TLS mútuo (mTLS) para corretores RabbitMQ, permitindo que os clientes se autentiquem usando certificados. Para obter mais informações, consulte Configuração do mTLS.
- Plug-in de autenticação de certificado SSL: O plug-in de autenticação SSL usa certificados de cliente de conexões mTLS para autenticar usuários, permitindo a autenticação usando certificados de cliente X.509 em vez de credenciais de nome de usuário e senha. Para obter mais informações, consulte Autenticação de certificado SSL.
- Plugin de autenticação HTTP: O plug-in de back-end de autenticação HTTP permite delegar autenticação e autorização a um serviço HTTP externo. Para obter mais informações, consulte Autenticação e autorização HTTP.
- Suporte JMS: o broker agora suporta cargas de trabalho JMS com o plug-in de troca de tópicos JMS ativado, permitindo que aplicativos JMS se conectem usando o cliente RabbitMQ JMS.

Os seguintes recursos foram descontinuados do RabbitMQ 4 no Amazon MQ

- Espelhamento de filas clássicas: as filas clássicas continuam sendo suportadas sem nenhuma alteração significativa nas bibliotecas e aplicativos do cliente, mas agora são um tipo de fila não replicada. Os clientes poderão se conectar a qualquer nó para publicar e consumir a partir de qualquer fila clássica não replicada. As filas de quórum são recomendadas para replicação e segurança de dados.
- Remoção da QoS global: recomenda-se que os clientes definam a QoS por consumidor (não global) em vez da QoS global, em que uma única pré-busca compartilhada é usada para um canal inteiro.
- Support para filas transitórias e não exclusivas: filas transitórias são filas cuja vida útil está vinculada ao tempo de atividade do nó em que foram declaradas. Em um único agente de instância, eles são removidos quando o nó é reiniciado. Em uma implantação de cluster, eles são removidos quando o nó em que estão hospedados é reiniciado. Recomendamos usar o TTL de fila para excluir automaticamente filas inativas e não utilizadas após algum tempo de inatividade. As filas exclusivas continuam sendo suportadas e são excluídas quando todas as conexões com a fila forem removidas.

As seguintes alterações importantes podem afetar seus aplicativos ao atualizar para o RabbitMQ 4.2 no Amazon MQ

- Tipo de fila padrão: o tipo de fila padrão em um broker RabbitMQ 4 é definido como quorum. Se nenhum argumento do tipo de fila for especificado durante a criação da fila, uma fila de quórum será criada.
- O limite padrão de reentrega nas filas de quórum é definido como 20: as mensagens que forem reenviadas 20 vezes ou mais serão excluídas (removidas). Se 20 entregas por mensagem for um cenário comum para uma fila, um destino de letras mortas ou um limite maior deverá ser configurado para essas filas para evitar perda de dados. A forma recomendada de fazer isso é por meio de uma política.
- amqplib: as versões amqplib do cliente Node JS anteriores à 0.10.7 ou qualquer biblioteca cliente AMQP usando frame_max < 8192 não conseguirão se conectar ao RabbitMQ
- Limites de recursos padrão: o Amazon MQ para RabbitMQ introduziu limites padrão de uso de recursos para conexões, canais, consumidores por canal, filas, vhosts, escavadeiras, trocas e tamanho máximo de mensagens. Eles servem como barreiras para proteger a disponibilidade do

corretor e podem ser personalizados usando configurações para atender às suas necessidades específicas.

Os seguintes recursos não são compatíveis com o RabbitMQ 4 no Amazon MQ

- Trocas aleatórias locais: trocas aleatórias locais não são suportadas no Amazon MQ, pois os nós do Amazon MQ estão atrás de um balanceador de carga de rede.
- Interceptor de mensagens: os interceptores de mensagens do RabbitMQ não são compatíveis com o Amazon MQ.
- Métricas por fila: o Amazon MQ não fornecerá métricas de fila do RabbitMQ para os corretores do RabbitMQ 4. AWS CloudWatch O Amazon MQ ainda fornecerá métricas em nível de corretor por meio de. AWS CloudWatch Você pode consultar métricas de fila usando a API de gerenciamento do RabbitMQ. Recomendamos consultar métricas para filas específicas com uma frequência de um minuto ou mais.

Suporte à versão RabbitMQ

O calendário de suporte da versão Amazon MQ abaixo indica quando uma versão do broker Engine chegará ao fim do suporte. Quando uma versão chega ao fim do suporte, o Amazon MQ atualiza automaticamente todos os agentes dessa versão para a próxima versão compatível. Essa atualização ocorre durante as janelas de manutenção programada da sua corretora, dentro de 45 dias após a end-of-support data.

O Amazon MQ avisa com pelo menos noventa dias de antecedência quando uma versão chegará ao fim do suporte. Recomendamos atualizar seu corretor antes da end-of-support data para evitar interrupções. Além disso, não é possível criar agentes em versões programadas para o fim do suporte dentro de trinta dias da data do fim do suporte.

Versão do RabbitMQ	Fim do suporte no Amazon MQ
4.2 (Recomendado)	
3.13	
3.12	17 de março de 2025

Atualizações de versão

Você pode atualizar seu agente manualmente a qualquer momento para a próxima versão principal ou secundária compatível. Para obter mais informações sobre como atualizar manualmente sua corretora, consulte Atualização de uma versão do mecanismo de corretora Amazon MQ.

O Amazon MQ gerencia atualizações para a última versão de patch compatível com todos os corretores RabbitMQ usando a versão 3.13 e superior. As atualizações de versões manuais e automáticas ocorrem durante a janela de manutenção agendada ou depois de reiniciar seu agente.

> **Important**
>
> O RabbitMQ só permite atualizações incrementais de versão (por exemplo, 3.9.x para 3.10.x). Não é possível pular versões secundárias ao atualizar (por exemplo, 3.8.x para 3.11.x).

Os agentes de instância única ficarão offline durante a reinicialização. Para agentes de cluster, as filas espelhadas devem ser sincronizadas durante a reinicialização. Com filas mais longas, o processo de sincronização das filas pode demorar mais. Durante o processo de sincronização das filas, a fila não fica disponível para consumidores e produtores. Quando o processo de sincronização das filas for concluído, o agente ficará disponível novamente. Para minimizar o impacto, recomendamos a atualização durante um período de baixo tráfego. Para obter mais informações sobre as práticas recomendadas de atualização de versão, consulte Práticas recomendadas do Amazon MQ para RabbitMQ.

Opções de implantação de agentes do Amazon MQ para RabbitMQ

Agentes RabbitMQ podem ser criados como agentes de instância única ou em uma implantação de cluster. Para ambos os modos de implantação, o Amazon MQ oferece alta durabilidade armazenando seus dados de forma redundante.

Você pode acessar seus agentes do RabbitMQ usando qualquer linguagem de programação compatível com o RabbitMQ e habilitando o TLS para os seguintes protocolos:

- AMQP (0-9-1)

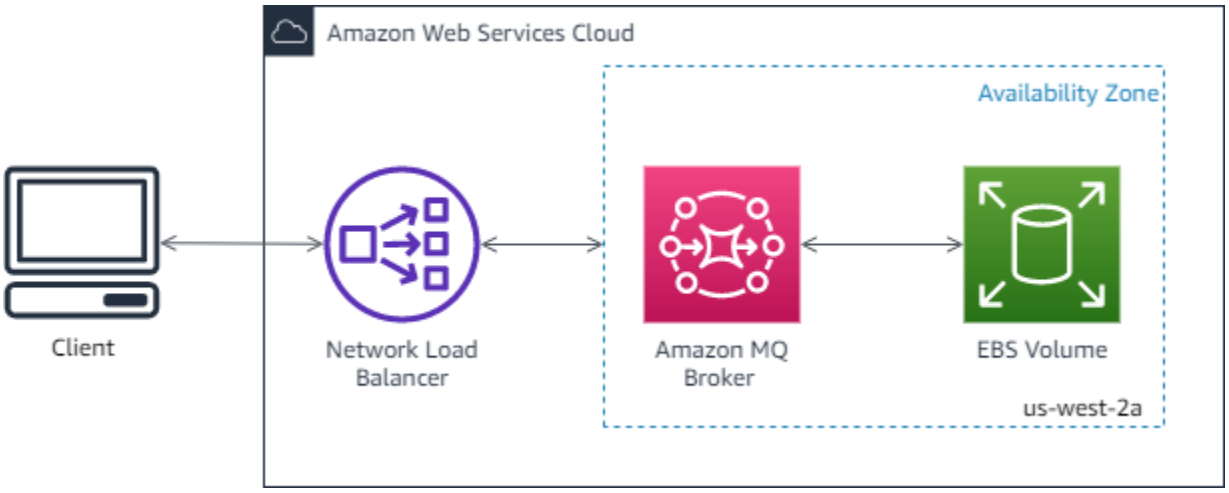
Tópicos

Opção 1: agente de instância única do Amazon MQ para RabbitMQ

Um agente de instância única é composto por um agente em uma zona de disponibilidade atrás de um Balanceador de carga da rede (NLB). O agente se comunica com sua aplicação e com um volume de armazenamento do Amazon EBS. O Amazon EBS fornece armazenamento em nível de bloco otimizado para baixa latência e alta taxa de transferência.

O uso de um Balanceador de carga da rede garante que seu endpoint do agente do Amazon MQ para RabbitMQ permaneça inalterado se a instância do agente for substituída durante uma janela de manutenção ou devido a falhas de hardware subjacentes do Amazon EC2. Um Balanceador de carga da rede permite que suas aplicações e usuários continuem a usar o mesmo endpoint para se conectar ao agente.

O diagrama a seguir ilustra um agente de instância única do Amazon MQ para RabbitMQ.



Opção 2: implantação do cluster do Amazon MQ para RabbitMQ

A implantação de cluster é um agrupamento lógico de três nós do agente RabbitMQ por trás de um Balanceador de Carga da Rede, cada um compartilhando usuários, filas e um estado distribuído em várias Zonas de Disponibilidade (AZ).

Em uma implantação de cluster, o Amazon MQ gerencia automaticamente as políticas de agente para habilitar o espelhamento clássico em todos os nós, garantindo alta disponibilidade (HA). Cada fila espelhada consiste em um nó principal e um ou mais espelhos. Cada fila tem seu próprio nó principal. Todas as operações para uma determinada fila são aplicadas primeiro no nó principal

da fila e depois propagadas para espelhos. O Amazon MQ cria uma política de sistema padrão que define o `ha-mode` para `all` e `ha-sync-mode` para `automatic`. Isso garante que os dados sejam replicados para todos os nós do cluster em diferentes zonas de disponibilidade para maior durabilidade.

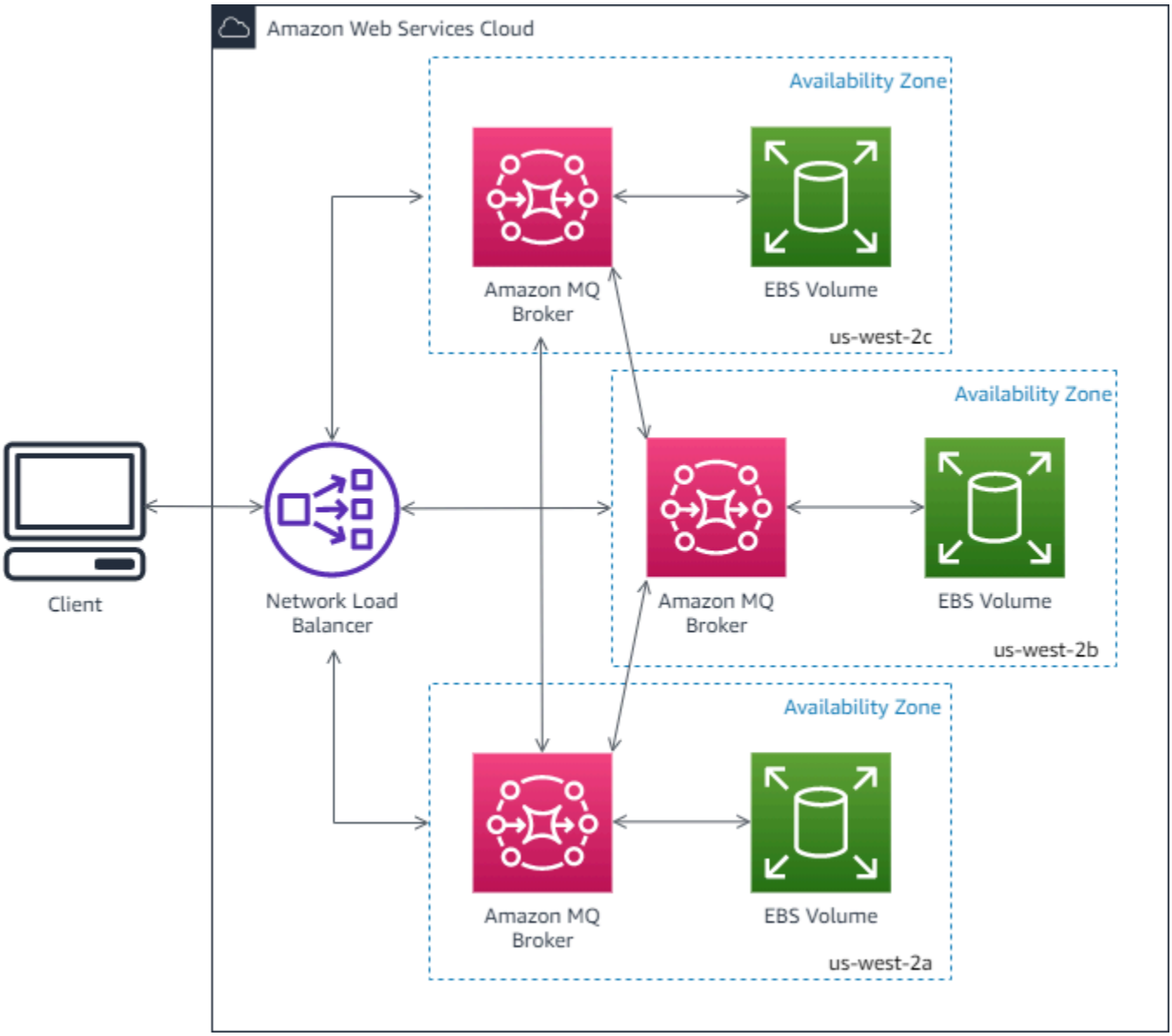
> **Note**
>
> Em uma implantação de cluster, se ocorrer uma interrupção na zona de disponibilidade, o Amazon MQ tentará automaticamente realocar os nós afetados do RabbitMQ para uma zona de disponibilidade diferente para manter o tamanho do cluster. Quando a interrupção for resolvida, o cluster será automaticamente rebalanceado entre as zonas de disponibilidade.

> **Note**
>
> Durante uma janela de manutenção, toda a manutenção de um cluster é realizada em um nó de cada vez, mantendo pelo menos dois nós em execução o tempo todo. Cada vez que um nó é derrubado, as conexões de cliente para esse nó são cortadas e precisam ser restabelecidas. Você deve garantir que seu código de cliente foi projetado para se reconectar automaticamente ao cluster. Para obter mais informações sobre a recuperação de conexões, consulte the section called “Etapa 1: Recuperação automática de falhas de rede”.
> Como o Amazon MQ define `ha-sync-mode: automatic`, durante uma janela de manutenção, as filas serão sincronizadas quando cada nó voltar a ingressar no cluster. A sincronização de filas bloqueia todas as outras operações de fila. Você pode atenuar o impacto da sincronização de filas durante as janelas de manutenção mantendo as filas curtas.

A política padrão não deve ser excluída. Se você excluí-la, o Amazon MQ vai recriá-la automaticamente. O Amazon MQ também garantirá que as propriedades de HA sejam aplicadas a todas as outras políticas criadas em um agente em cluster. Se você adicionar uma política sem as propriedades de HA, o Amazon MQ as adicionará para você. Se você adicionar uma política com diferentes propriedades de alta disponibilidade, o Amazon MQ as substituirá. Para obter mais informações sobre o espelhamento clássico, consulte filas espelhadas.

O diagrama a seguir ilustra uma implantação do agente de cluster RabbitMQ com três nós em três zonas de disponibilidade (AZ), cada um com seu próprio volume do Amazon EBS e um estado

compartilhado. O Amazon EBS fornece armazenamento em nível de bloco otimizado para baixa latência e alta taxa de transferência.



Tipos de instância do agente do Amazon MQ para RabbitMQ

A descrição combinada da classe (m7g) e do tamanho (grande, médio) da instância do broker é chamada de tipo de instância do broker (por exemplo, mq.m7g.large).

Recomendamos usar os tipos de instância mq.m7g para implantações de cluster e de instância única.

O Amazon MQ avisa com pelo menos 90 dias de antecedência quando um tipo de instância chegará ao fim do suporte. Recomendamos atualizar seu corretor para um novo tipo de instância antes da end-of-support data para evitar interrupções.

> **Important**
>
> Você não pode fazer o downgrade de um agente de um tipo de `mq.m5` instância `mq.m7g` ou para um tipo de `mq.t3.micro` instância.
> O tipo de `mq.t3.micro` instância não é compatível com a implantação de clusters.

Tipos de instância para implantação de clusters m7g

Recomendamos o uso de tipos de instância do `mq.m7g.x` com a implantação de clusters. A tabela a seguir mostra os tipos de instância disponíveis do `mq.m7g.x` para implantação de cluster.

Tipo de instância	vCPU	Memória (GiB)	Linha de base de rede/Largura de banda máxima (Gbps)	Uso recomendado	Armazenamento	Tamanho do volume de disco por nó (GB)
mq.m7g.medium	1	4	0,52 / 12,5	Avaliação	EBS	5
mq.m7g.large	2	8	0.937/12.5	Produção	EBS	15
mq.m7g.xlarge	4	16	1,876/12,5	Produção	EBS	25
mq.m7g.2xlarge	8	32	3.75/15.0	Produção	EBS	45
mq.m7g.4xlarge	16	64	7.5/15.0	Produção	EBS	90

Tipo de instância	vCPU	Memória (GiB)	Linha de base de rede/Largura de banda máxima (Gbps)	Uso recomendado	Armazenamento	Tamanho do volume de disco por nó (GB)
mq.m7g.8xlarge	32	128	15 gigabits	Produção	EBS	175
mq.m7g.12xlarge	48	192	22,5 gigabits	Produção	EBS	260
mq.m7g.16xlarge	64	256	30 gigabits	Produção	EBS	345

Tipos de instância para implantação de instância única m7g

A tabela a seguir mostra os tipos de instância disponíveis do `mq.m7g.x` para implantação de cluster.

Tipo de instância	vCPU	Memória (GiB)	Linha de base de rede/Largura de banda máxima (Gbps)	Uso recomendado	Armazenamento	Tamanho do volume de disco por nó (GB)
mq.m7g.medium	1	4	0,52 / 12,5	Avaliação	EBS	200
mq.m7g.large	2	8	0.937/12.5	Produção	EBS	200
mq.m7g.xlarge	4	16	1,876/12,5	Produção	EBS	200

Tipo de instância	vCPU	Memória (GiB)	Linha de base de rede/Largura de banda máxima (Gbps)	Uso recomendado	Armazenamento	Tamanho do volume de disco por nó (GB)
mq.m7g.2xlarge	8	32	3.75/15.0	Produção	EBS	200
mq.m7g.4xlarge	16	64	7.5/15.0	Produção	EBS	200
mq.m7g.8xlarge	32	128	15 gigabits	Produção	EBS	200
mq.m7g.12xlarge	48	192	22,5 gigabits	Produção	EBS	200
mq.m7g.16xlarge	64	256	39 Gigabit	Produção	EBS	200

Tipos de instância para implantação de instância única **mq.m5**

A tabela a seguir mostra os tipos de instância disponíveis do `mq.m5.x` para implantação de cluster.

Tipo de instância	vCPU	Memória (GiB)	Linha de base de rede/Largura de banda máxima (Gbps)	Uso recomendado	Armazenamento	Tamanho do volume de disco por nó (GB)
mq.t3.micro	2	1	0,064/5,0	Avaliação	EBS	20

Tipo de instância	vCPU	Memória (GiB)	Linha de base de rede/Largura de banda máxima (Gbps)	Uso recomendado	Armazenamento	Tamanho do volume de disco por nó (GB)
mq.m5.large	2	8	0,75/10,0	Produção	EBS	200
mq.m5.xlarge	4	16	1.25/10.0	Produção	EBS	200
mq.m5.2xlarge	8	32	2.5/10.0	Produção	EBS	200
mq.m5.4xlarge	16	64	5.0/10.0	Produção	EBS	200

Tipos de instância para implantação de clusters **`mq.m5`**

A tabela a seguir mostra os tipos de instância disponíveis do `mq.m5.x` para implantação de cluster.

Tipo de instância	vCPU	Memória (GiB)	Linha de base de rede/Largura de banda máxima (Gbps)	Uso recomendado	Armazenamento	Tamanho do volume de disco por nó (GB)
mq.m5.large	2	8	0,75/10,0	Produção	EBS	200
mq.m5.xlarge	4	16	1.25/10.0	Produção	EBS	200

Tipo de instância	vCPU	Memória (GiB)	Linha de base de rede/Largura de banda máxima (Gbps)	Uso recomendado	Armazenamento	Tamanho do volume de disco por nó (GB)
mq.m5.2xlarge	8	32	2.5/10.0	Produção	EBS	200
mq.m5.4xlarge	16	64	5.0/10.0	Produção	EBS	200

Diretrizes de dimensionamento do Amazon MQ para RabbitMQ

Você pode escolher o tipo de instância do agente que melhor forneça suporte à aplicação. Ao escolher um tipo de instância, considere os fatores que afetarão o desempenho do corretor:

- o número de clientes e filas
- o volume de mensagens enviadas
- mensagens mantidas na memória
- mensagens redundantes

Tipos menores de instâncias de broker `m7g.medium` são recomendados somente para testar o desempenho do aplicativo. Recomendamos tipos de instância de broker maiores `m7g.large` e níveis superiores ou superiores ou de produção de clientes e filas, alta taxa de transferência, mensagens na memória e mensagens redundantes.

> **Important**
>
> Você não pode fazer o downgrade de um agente de um tipo de `mq.m7g` instância `mq.m5` ou para um tipo de `mq.t3.micro` instância.

É importante testar os agentes para determinar o tipo e o tamanho da instância adequados para seus requisitos de mensagens de workloads.

Sempre use os limites de recursos padrão no agente RabbitMQ 4 para determinar o tamanho apropriado da instância para seu aplicativo de acordo com as melhores práticas do Amazon MQ. Esses limites de recursos padrão são baseados em tipos, tipo de `m7g` instância e filas de quórum.

- Limites de recursos padrão para implantação de instância única m7g
- Limites de recursos padrão para implantação do cluster m7g

Você pode aumentar o valor de qualquer limite até os valores máximos, conforme definido pelo tipo de instância e modo de implantação. No entanto, é altamente recomendável que você teste o desempenho do corretor com os valores aumentados antes de usá-lo na produção.

- Limites máximos de recursos para implantação de instância única m7g
- Limites máximos de recursos para implantação do cluster m7g
- Limites máximos de recursos para implantação de instância única m5
- Limites máximos de recursos para implantação do cluster m5
- Mensagens de erro

> **Note**
>
> Os corretores RabbitMQ 3.13 não vêm com limites de recursos padrão, mas recomendamos que você use os padrões sugeridos.

Limites de recursos padrão

O Amazon MQ para RabbitMQ suporta a configuração dos limites de recursos do broker a partir do RabbitMQ 4. Quando você cria um agente, o Amazon MQ aplica automaticamente valores padrão a esses limites de recursos. Esses padrões atuam como barreiras para proteger a disponibilidade de seu corretor e, ao mesmo tempo, acomodar os padrões comuns de uso do cliente. Você pode personalizar o comportamento do seu agente alterando os valores de configuração limite para melhor atender aos seus requisitos específicos de carga de trabalho.

Antes de fazer alterações, observe:

> **Important**
>
> 1. Alterações na configuração podem afetar o desempenho e a disponibilidade da corretora
> 2. Entenda o impacto antes de alterar qualquer opção de configuração padrão
> 3. Teste primeiro as alterações de configuração em ambientes que não sejam de produção
> 4. Monitore a saúde do corretor após aplicar as alterações

> **Important**
>
> Os valores padrão e os intervalos suportados para essas configurações variam de acordo com a versão do RabbitMQ, o tipo de instância e o modo de implantação do broker.

> **Important**
>
> Observação: associar ou criar um agente com valores de configuração fora do intervalo suportado resultará em uma resposta de erro.

Para saber como personalizar esses limites de recursos padrão para seu corretor, consultethe section called “Configurando o limite de recursos”.

Os limites de recursos padrão aplicados aos corretores RabbitMQ 4.2 são

- Limites de recursos padrão para implantação de instância única m7g
- Limites de recursos padrão para implantação do cluster m7g

Limites de recursos padrão

> **Important**
>
> Amazon MQ para corretores RabbitMQ 3, o padrão é configurado com o limite máximo de recursos e o Amazon MQ não oferece a capacidade de substituir a configuração do limite de recursos.

Valores padrão para corretores de instância única

Tipo de instância	Conexões por nó	Canais por Node	Consumidores por canal	Queues (Filas)	fantasmas	Shovels	Trocas	Tamanho da mensagem em bytes
mq.m7g.medium	100	500	10	500	10	30	500	52428
mq.m7g.large	1.500	4.500	10	1.000	50	50	1.000	52428
mq.m7g.xlarge	3.000	9.000	10	2.000	100	100	2.000	52428
mq.m7g.2large	6.000	18.000	10	4.000	150	200	4.000	52428
mq.m7g.4large	12.000	36.000	10	8.000	200	400	8.000	52428
mq.m7g.8large	24.000	72.000	10	16.000	250	800	16.000	52428
mq.m7g.1xlarge	36.000	108.000	10	24.000	300	1.200	24.000	52428
mq.m7g.1xlarge	48.000	144.000	10	32.000	350	1.600	32.000	52428

Valores padrão para agentes de cluster

Tipo de instância	Conexões por nó	Canais por Node	Consumidores por canal	Queues (Filas)	fantasmas	Shovels	Trocas	Tamanho da mensagem em bytes
mq.m7g.medium	100	300	10	100	10	10	100	52428
mq.m7g.large	500	1500	10	1.000	50	30	1.000	52428
mq.m7g.xarge	1000	3000	10	2.000	100	60	2.000	52428
mq.m7g.2large	2000	6000	10	4.000	150	120	4.000	52428
mq.m7g.4large	4000	12.000	10	8.000	200	240	8.000	52428
mq.m7g.8large	8000	24.000	10	16.000	250	480	16.000	52428
mq.m7g.1xlarge	12000	36.000	10	24.000	300	720	24000	52428
mq.m7g.1xlarge	16.000	48.000	10	32.000	350	960	32.000	52428

Limite máximo de recursos do Amazon MQ para RabbitMQ

Você pode configurar limites de recursos até os valores máximos mostrados nas tabelas a seguir. Para saber como atualizar os limites de recursos do seu corretor, consultethe section called “Configurando o limite de recursos”.

Diretrizes de dimensionamento para m7g com filas de quórum para implantação de instância única

A tabela a seguir mostra os valores-limite máximos de cada tipo de instância para agentes de instância única.

Tipo de instância	Conexões	Canais	Consumidores por canal	Queues (Filas)	Vhosts	Shovels	Trocas	Tamanho da mensagem em bytes
mq.m7g.medium	300	900	1.000	2.500	10	150	12500	134217728
mq.m7g.large	5.000	15.000	1.000	20.000	1500	250	100.000	134217728
mq.m7g.xlarge	10.000	30.000	1.000	30.000	1.500	500	150.000	134217728
mq.m7g.2large	20.000	60.000	1.000	40.000	1.500	1.000	200.000	134217728
mq.m7g.4large	40.000	120.000	1.000	60.000	1.500	2000	300,000	134217728
mq.m7g.8large	80.000	240.000	1.000	80.000	1.500	4000	400.000	134217728
mq.m7g.1xlarge	120.000	360.000	1.000	100.000	1.500	6.000	500.000	134217728
mq.m7g.1xlarge	160.000	480.000	1.000	120.000	1.500	8.000	600.000	134217728

Diretrizes de dimensionamento para m7g com filas de quórum para implantação de clusters

A tabela a seguir mostra os valores-limite máximos de cada tipo de instância para agentes de cluster.

Tipo de instância	Conexões por nó	Canais por Node	Consumidores por canal	Queues (Filas)	Vhosts	Shovels	Trocas	Tamanho da mensagem em bytes
mq.m7g.medium	300	900	1.000	500	10	50	500	134217728
mq.m7g.large	5.000	15.000	1.000	10.000	1.500	150	50.000	134217728
mq.m7g.xlarge	10.000	30.000	1.000	15.000	1.500	300	75,000	134217728
mq.m7g.2large	20.000	60.000	1.000	20.000	1.500	600	100.000	134217728
mq.m7g.4large	40.000	120.000	1.000	30.000	1.500	1200	150.000	134217728
mq.m7g.8large	80.000	240.000	1.000	40.000	1.500	2.400	200.000	134217728
mq.m7g.1xlarge	120.000	360.000	1.000	50.000	1.500	3.600	250.000	134217728
mq.m7g.1xlarge	160.000	480.000	1.000	60.000	1.500	4.800	300,000	134217728

Limites máximos de recursos para implantação de instância única M5

A tabela a seguir mostra os valores-limite máximos de cada tipo de instância para agentes de instância única.

Tipo de instância	Conexões	Canais	Consumidores por canal	Queues (Filas)	Vhosts	Shovels
m5.large	5.000	15.000	1.000	30.000	1500	250
m5.xlarge	10.000	30.000	1.000	60.000	1500	500
m5.2xlarge	20.000	60.000	1.000	120.000	1500	1.000
m5.4xlarge	40.000	120.000	1000	240.000	1.000	2.000

Limites máximos de recursos para implantação do cluster m5

A tabela a seguir mostra os valores-limite máximos de cada tipo de instância para agentes de cluster.

Tipo de instância	Queues (Filas)	Consumidores por canal	Shovels
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

Os limites de conexão e canal a seguir são aplicados por nó.

Tipo de instância	Conexões	Canais
m5.large	5000	15.000
m5.xlarge	10.000	30.000
m5.2xlarge	20.000	60.000
m5.4xlarge	40.000	120.000

Os valores-limite exatos para um agente de cluster podem ser menores do que o valor indicado, dependendo do número de nós disponíveis e de como o RabbitMQ distribui os recursos entre os nós disponíveis. Se você exceder os valores-limite, poderá criar uma conexão com um nó diferente e tentar outra vez, ou poderá atualizar o tamanho da instância para aumentar os limites máximos.

Mensagens de erro

As mensagens de erro a seguir são exibidas quando os limites são excedidos. Todos os valores são baseados nos limites de instância única **`m7.large`**.

> **Note**
>
> Os códigos de erro das mensagens a seguir podem mudar com base na biblioteca de cliente que você estiver usando.

Conexão

```
ConnectionClosedByBroker 500 "NOT_ALLOWED - connection refused: node
connection limit (5000) is reached"
```

Channel (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')
```

Tamanho máximo da mensagem

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

Troca

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

> **Note**
>
> As mensagens de erro a seguir usam o formato da API de gerenciamento em HTTP.

Queue (Fila)

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

Shovel

```
{"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"}
```

Padrões de agentes do Amazon MQ for RabbitMQ

Quando você cria um Amazon MQ para agente RabbitMQ, o Amazon MQ aplica um conjunto padrão de políticas de agente e limites de vhost para otimizar a performance do seu agente. O Amazon MQ aplica limites de vhost somente ao vhost padrão (`/`). O Amazon MQ não aplicará políticas padrão a vhosts recém-criados. Recomendamos manter esses padrões para todos os agentes novos e existentes. No entanto, você pode modificar, substituir ou excluir esses padrões a qualquer momento.

O Amazon MQ cria diferentes políticas de corretor e limites de vhost para o Amazon MQ for RabbitMQ 3 e RabbitMQ 4. As diferenças serão discutidas em detalhes nas subseções a seguir.

O Amazon MQ cria políticas e limites com base no tipo de instância e no modo de implantação do agente que você escolhe ao criar seu agente. As políticas padrão são nomeadas de acordo com o modo de implantação, da seguinte maneira:

Amazon MQ para RabbitMQ 3:

- Instância única — `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Implantação de clusters — `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ` && `AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Amazon MQ para RabbitMQ 4:

- Instância única — `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Implantação de clusters — `AWS-DEFAULT-POLICY-CLUSTER && AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Para os agentes de instância única, o Amazon MQ define o valor de prioridade da política como `0`. Para substituir o valor de prioridade padrão, você pode criar suas próprias políticas personalizadas com valores de prioridade mais altos. Para implantações de cluster, o Amazon MQ define o valor de prioridade como 1 para padrões do agente. Para criar sua própria política personalizada para clusters, atribua um valor de prioridade maior que 1.

> **Note**
>
> Em implantações de cluster, as políticas de agente `ha-mode` e `ha-sync-mode` são necessárias para espelhamento clássico e alta disponibilidade (HA). Essas configurações são aplicáveis somente para o Amazon MQ para RabbitMQ 3 e não para o RabbitMQ 4. Se você exclui a política padrão `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ`, o Amazon MQ usa a política `ha-all-AWS-OWNED-DO-NOT-DELETE` com um valor de prioridade `0`. Isso garante que as políticas `ha-mode` e `ha-sync-mode` necessárias ainda estejam em vigor. Se você criar sua própria política personalizada, o Amazon MQ anexará automaticamente o `ha-mode` e `ha-sync-mode` nas suas definições de política.

Tópicos

Descrições de políticas e limites

A lista a seguir descreve as políticas e limites padrão que o Amazon MQ aplica a um agente recém-criado. Os valores para `max-length`, `max-queues` e `max-connections` variam de acordo com o tipo de instância e o modo de implantação do seu agente. Esses valores estão listados na seção Valores padrão recomendados.

Configurações nos corretores RabbitMQ 3 e RabbitMQ 4

- **`queue-mode: lazy`** (política) — Habilita filas lentas. Por padrão, as filas mantêm um cache na memória de mensagens, permitindo que o agente entregue mensagens aos consumidores o mais rápido possível. Isso pode fazer o agente ficar sem memória e acionar um alarme de alta memória. As filas lentas tentam mover as mensagens para o disco o mais cedo possível. Isso significa que menos mensagens são mantidas na memória em condições normais de operação. Usando filas lentas, o Amazon MQ para RabbitMQ pode ser compatível com sistemas de mensagens muito maiores e filas mais longas. Observe que, para determinados casos de uso, os agentes com filas lentas podem ter uma performance ligeiramente mais lenta. Isso ocorre porque as mensagens são movidas do disco para o agente, em vez de entregar mensagens de um cache na memória.

  > Modos de implantação
  >
  > Instância única, cluster

- **`max-length: number-of-messages`** (política) — Define um limite para o número de mensagens em uma fila. Em implantações de cluster, o limite impede a sincronização de fila pausada em casos como reinicializações de agente ou após uma janela de manutenção.

  > Modos de implantação
  >
  > Cluster

- **`overflow: reject-publish`** (política) — Impõe filas com uma política `max-length` para rejeitar novas mensagens depois do número de mensagens na fila atingir o valor `max-length`. Para garantir que as mensagens não sejam perdidas se uma fila estiver em um estado sobrecarregado, as aplicações dos clientes que publicam mensagens no agente devem implementar a confirmação do editor. Para obter informações sobre como implementar a confirmação do editor, consulte Confirmações do editor no site do RabbitMQ.

  > Modos de implantação
  >
  > Cluster

Configurações específicas do RabbitMQ 3

- **`max-queues: number-of-queues-per-vhost`** (limite de vhost) — Define o limite para o número de filas em um agente. Similar à definição de política `max-length`, limitar o número de

filas em implantações de cluster impede a sincronização de filas pausada após reinicializações de agente ou janelas de manutenção. Limitar filas também impede quantidades excessivas de uso da CPU para manter filas.

> **Modos de implantação**
>
> Instância única, cluster

- **`max-connections: number-of-connections-per-vhost`** (limite de vhost) — Define o limite para o número de conexões de cliente com o agente. Limitar o número de conexões de acordo com os valores recomendados impede o uso excessivo de memória pelo agente o que poderia resultar na sinalização de um alarme de alto uso de memória e na interrupção das operações.

> **Modos de implantação**
>
> Instância única, cluster

Valores padrão recomendados

> **Important**
>
> `max-queues`e `max-connections` são aplicados somente ao Amazon MQ para RabbitMQ 3.

> **Note**
>
> Os limites padrão `max-length` e `max-queue` são testados e avaliados com base em um tamanho médio de mensagem de 5 kB. Se as suas mensagens forem significativamente maiores do que 5 kB, você precisará ajustar e reduzir os limites `max-length` e `max-queue`.

A tabela a seguir lista os valores de limite padrão para um agente recém-criado. O Amazon MQ aplica esses valores de acordo com o tipo de instância e o modo de implantação do agente.

Tipo de instância	Modo de implantação	**max-length**	**max-queues**	**max-connections**
mq.m7g.medium	Instância única	N/D	2.500	100
	Cluster	500.000	100	100
mq.m7g.large	Instância única	N/D	20.000	5.000
	Cluster	8.000.000	10.000	5.000
mq.m7g.xlarge	Instância única	N/D	30.000	10.000
	Cluster	9.000.000	15.000	10.000
mq.m7g.2xlarge	Instância única	N/D	40.000	20.000
	Cluster	10,000,000	40.000	20.000
mq.m7g.4xlarge	Instância única	N/D	60.000	40.000
	Cluster	12.000.000	30.000	40.000
mq.m7g.8xlarge	Instância única	N/D	80.000	80.000
	Cluster	20.000.000	40.000	80.000
mq.m7g.12xlarge	Instância única	N/D	100.000	120.000
	Cluster	30.000.000	20.000	120.000
mq.m7g.16xlarge	Instância única	N/D	120.000	160.000
	Cluster	40.000.000	50.000	160.000

Tipo de instância	Modo de implantação	**max-length**	**max-queues**	**max-connections**
t3.micro	Instância única	N/D	500	500

Tipo de instância	Modo de implantação	`max-length`	`max-queues`	`max-connections`
m5.large	Instância única	N/D	20.000	4.000
m5.large	Cluster	8.000.000	10.000	15.000
m5.xlarge	Instância única	N/D	30.000	8.000
m5.xlarge	Cluster	9.000.000	10.000	20.000
m5.2xlarge	Instância única	N/D	60.000	15.000
m5.2xlarge	Cluster	10,000,000	10.000	40.000
m5.4xlarge	Instância única	N/D	150.000	30.000
m5.4xlarge	Cluster	12.000.000	10.000	100.000

Configurando um corretor RabbitMQ

Uma configuração contém todas as configurações do seu corretor RabbitMQ no formato Cuttlefish. Você pode criar uma configuração antes de criar qualquer agente. Em seguida, você pode aplicar a configuração a um ou mais agentes.

Atributos

A configuração de um agente tem vários atributos, por exemplo:

- Um nome (MyConfiguration)
- Uma identificação (c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9)
- Um nome de recurso da Amazon (ARN) (arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b678cd-901e-2fgh-3i45j6k178l9)

Para obter uma lista completa de atributos de configuração, consulte o seguinte na Referência de API Amazon MQ REST:

- ID da operação REST: Configuração

- ID da operação REST: Configurações

Para obter uma lista completa de atributos de revisão de configuração, consulte o seguinte:

- ID da operação REST: Revisão da configuração
- ID da operação REST: Revisões de configuração

Tópicos

- Criando e aplicando configurações do corretor RabbitMQ
- Editar uma revisão de configuração do Amazon MQ para RabbitMQ
- Valores configuráveis para RabbitMQ no Amazon MQ
- Suporte de ARN na configuração do RabbitMQ

Criação e aplicação de configurações do agente do RabbitMQ

Uma configuração contém todas as definições do agente do RabbitMQ, no formato Cuttlefish. Você pode criar uma configuração antes de criar qualquer agente. Depois, você pode aplicar a configuração a um ou mais agentes

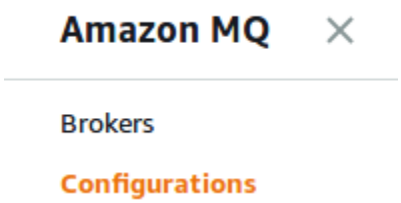
Os exemplos a seguir mostram como criar e aplicar uma configuração do agente do RabbitMQ utilizando o Console de gerenciamento da AWS.

> **Important**
>
> Você só pode excluir uma configuração usando a API do `DeleteConfiguration`. Para obter mais informações, consulte Configurações na Referência da API do Amazon MQ.

Criar uma configuração

Para aplicar uma configuração ao agente, primeiro você deve criar a configuração.

1. Faça login no console do Amazon MQ.
2. Do lado esquerdo, expanda o painel de navegação e selecione Configurations (Configurações).



3. Na página Configurações, selecione Criar configuração.
4. Na página Criar configuração, na seção Detalhes, digite o Nome da configuração (por exemple, `MyConfiguration`) e selecione uma versão do Mecanismo do agente.

   Para saber mais sobre as versões do mecanismo do RabbitMQ compatíveis com o Amazon MQ para RabbitMQ, consulte the section called “Gerenciamento de versão”.
5. Escolha Criar configuração.

Criar uma revisão de configuração

Depois de criar uma configuração, você deverá editá-la usando uma revisão de configuração.

1. Na lista de configuração, escolha ***`MyConfiguration`***.

   > **Note**
   >
   > A primeira revisão de configuração será sempre criada para você quando o Amazon MQ criar a configuração.

   Na ***`MyConfiguration`***página, o tipo e a versão do mecanismo do broker que sua nova revisão de configuração usa (por exemplo, RabbitMQ 3.xx.xx) são exibidos.
2. Na guia Detalhes da configuração, são exibidos o número de revisão da configuração, a descrição e a configuração do agente no formato Cuttlefish.

   > **Note**
   >
   > Editar a configuração atual irá criar uma nova revisão da configuração.
3. Selecione Editar configuração e faça as alterações na configuração do Cuttlefish.
4. Escolha Salvar.

A caixa e diálogo Save revision (Salvar revisão) será exibida.

5. (Opcional) Tipo `A description of the changes in this revision.`
6. Escolha Salvar.

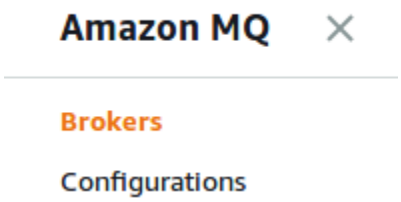
   A nova revisão da configuração é salva.

> **Important**
>
> Fazer alterações em uma configuração não aplica as alterações ao agente imediatamente. Para aplicar as alterações, você deve aguardar a próxima janela de manutenção ou reiniciar o agente.
> No momento, não é possível excluir uma configuração.

Aplicar uma revisão de configuração ao operador

Depois de criar a revisão da configuração, você pode aplicá-la ao agente.

1. Do lado esquerdo, expanda o painel de navegação e selecione Brokers (Agentes).



2. Na lista de corretores, selecione seu corretor (por exemplo MyBroker) e escolha Editar.
3. Na *`MyBroker`* página Editar, na seção Configuração, selecione uma Configuração e uma Revisão e, em seguida, escolha Programar Modificações.
4. Na seção Schedule broker modifications (Programar modificações no operador), escolha se deseja aplicar as modificações During the next scheduled maintenance window (Durante a próxima janela de manutenção programada) ou Immediately (Imediatamente).

> **Important**
>
> Os agentes de instância única ficarão offline durante a reinicialização. Para agentes de cluster, somente um nó fica inativo por vez enquanto o agente é reinicializado.

5. Escolha Aplicar.

   Sua revisão de configuração será aplicada ao agente no horário especificado.

Editar uma revisão de configuração do Amazon MQ para RabbitMQ

As instruções a seguir descrevem como editar uma revisão de configuração para o agente.

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, selecione seu corretor (por exemplo MyBroker) e escolha Editar.
3. Na ***MyBroker***página, escolha Editar.
4. Na *MyBroker* página Editar, na seção Configuração, selecione uma Configuração e uma Revisão e escolha Editar.

   > **Note**
   >
   > A menos que você selecione uma configuração ao criar um agente, a primeira revisão de configuração será sempre criada para você quando o Amazon MQ criar o agente.

   Na ***MyBroker***página, o tipo e a versão do mecanismo do broker que a configuração usa (por exemplo, RabbitMQ 3.xx.xx) são exibidos.
5. Na guia Detalhes da configuração, são exibidos o número de revisão da configuração, a descrição e a configuração do agente no formato Cuttlefish.

   > **Note**
   >
   > Editar a configuração atual irá criar uma nova revisão da configuração.
6. Selecione Editar configuração e faça as alterações na configuração do Cuttlefish.
7. Escolha Salvar.

   A caixa e diálogo Save revision (Salvar revisão) será exibida.
8. (Opcional) Tipo `A description of the changes in this revision`.
9. Escolha Salvar.

   A nova revisão da configuração é salva.

> **Important**
>
> Fazer alterações em uma configuração não aplica as alterações ao agente imediatamente. Para aplicar as alterações, você deve aguardar a próxima janela de manutenção ou reiniciar o agente.
> No momento, não é possível excluir uma configuração.

Valores configuráveis

Você pode definir o valor das opções de configuração do agente a seguir modificando o arquivo de configuração do agente no Console de gerenciamento da AWS.

Além dos valores descritos na tabela a seguir, o Amazon MQ oferece suporte a opções adicionais de configuração de agentes relacionadas à autenticação e autorização, bem como aos limites de recursos. Para obter mais informações sobre essas opções de configuração, consulte

- OAuth Configuração 2.0
- Configuração LDAP
- Configuração HTTP
- Configuração do SSL
- Configuração mTLS
- Suporte para ARN
- Limites de recurso
- Configuração SSL do cliente AMQP

Configuração	Valor padrão	Valores recomendados	Valores	Versões aplicáveis	Description
consumer_timeout	1.800.000 ms (30 minutos)	1.800.000 ms (30 minutos)	0 a 2.147.483.647 ms. O Amazon MQ também suporta o	Todas as versões	Um tempo limite na confirmação da entrega do consumidor

Configuração	Valor padrão	Valores recomendados	Valores	Versões aplicáveis	Description
			valor 0, que significa “infinito”.		para detectar quando os consumidores não confirmam as entregas.
heartbeat	60 segundos	60 segundos	De 60 a 3.600 segundos	Todas as versões	Define o tempo antes de uma conexão ser considerada indisponível pelo RabbitMQ.
management.restrictions.operator_policy_changes.disabled	true	true	true, false	Todas as versões	Desabilita a realização de alterações nas políticas do operador. Se você fizer essa alteração, é altamente recomendável incluir as propriedades de HA em suas próprias políticas de operador.

Configuração	Valor padrão	Valores recomendados	Valores	Versões aplicáveis	Description
quorum_queue.property_equivalence.relaxed_checks_on_redeclaration	true	true	true, false	Todas as versões	Quando definido como TRUE, a aplicação evita uma exceção do canal ao redeclarar uma fila de quórum.
secure.management.http.headers.enabled	true	true	true, false	Todas as versões	Habilita cabeçalhos de segurança HTTP não modificáveis.

Configurar uma confirmação de entrega do consumidor

Você pode configurar consumer_timeout para detectar quando os consumidores não embalam as entregas. Se o consumidor não enviar uma confirmação dentro do tempo limite, o canal será fechado. Por exemplo, se você estiver usando o valor padrão de 1.800.000 milissegundos, se o consumidor não enviar uma confirmação de entrega dentro de 1.800.000 milissegundos, o canal será fechado. O Amazon MQ também suporta o valor 0, que significa “infinito”.

Configurar pulsação

Você pode configurar um tempo limite de pulsação para descobrir quando as conexões foram interrompidas ou falharam. O valor da pulsação define o limite de tempo antes de uma conexão ser considerada inativa.

Configurar políticas do operador

A política de operador padrão em cada host virtual tem as seguintes propriedades de HA recomendadas:

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

As alterações nas políticas do operador por meio da API de gerenciamento Console de gerenciamento da AWS ou não estão disponíveis por padrão. Você pode ativar as alterações adicionando a seguinte linha à configuração do agente:

```
management.restrictions.operator_policy_changes.disabled=false
```

Se você fizer essa alteração, é altamente recomendável incluir as propriedades de HA em suas próprias políticas de operador.

Configurar verificações flexíveis na declaração de filas

Se você migrou suas filas clássicas para filas de quórum, mas não atualizou seu código de cliente, você pode evitar uma exceção de canal ao redeclarar uma fila de quórum configurando quorum_queue.property_equivalence.relaxed_checks_on_redeclaration definido como true.

Configurar cabeçalhos de segurança HTTP

A configuração secure.management.http.headers.enabled ativa os seguintes cabeçalhos de segurança HTTP:

- X-Content-Type-Options: nosniff: impede que os navegadores realizem a detecção de conteúdo, algoritmos usados para deduzir o formato dos arquivos dos sites.
- X-Frame-Options: DENY: impede que alguém incorpore o plug-in de gerenciamento em um quadro em seu próprio site para enganar outras pessoas.

- Strict-Transport-Security: max-age=47304000; includeSubDomains: obriga os navegadores a usarem HTTPS ao fazer conexões subsequentes ao site e seus subdomínios por um longo período de tempo (1,5 anos).

Os corretores do Amazon MQ para RabbitMQ criados nas versões 3.10 e superiores terão secure.management.http.headers.enabled definido como true por padrão. Você pode ativar esses cabeçalhos de segurança HTTP definindo secure.management.http.headers.enabled como true. Se você quiser desativar esses cabeçalhos de segurança HTTP, defina secure.management.http.headers.enabled como false.

Configurando a autenticação e autorização OAuth 2.0

Para obter informações sobre as opções de configuração OAuth OAuth 2.0 e a configuração da autenticação 2.0 para seus corretores, consulte Configurações OAuth 2.0 suportadas e Usando autenticação e OAuth autorização 2.0.

Configurando a autenticação e autorização LDAP

Para obter informações sobre as opções de configuração do LDAP e a configuração da autenticação LDAP para seus corretores, consulte Configurações de LDAP suportadas e. Usando autenticação e autorização LDAP

Configurando a autenticação e autorização HTTP

Para obter informações sobre os valores de configuração da autenticação HTTP e a configuração da autenticação HTTP para seus corretores, consulte Autenticação e Usando autenticação e autorização HTTP autorização HTTP e.

> **Note**
>
> O plug-in de autenticação HTTP está disponível somente para o Amazon MQ for RabbitMQ versão 4 e superior.

Configurando a autenticação do certificado SSL

Para obter informações sobre os valores de configuração da autenticação do certificado SSL e a configuração da autenticação do certificado SSL para seus corretores, consulte Autenticação do certificado SSL e. Usando a autenticação de certificado SSL

> **Note**
>
> O plug-in de autenticação de certificado SSL está disponível somente para o Amazon MQ for RabbitMQ versão 4 e superior.

Configurando o mTLS

O Amazon MQ para RabbitMQ oferece suporte a TLS mútuo (mTLS) para conexões seguras com vários endpoints e serviços externos. O mTLS fornece segurança aprimorada ao exigir que o cliente e o servidor se autentiquem usando certificados.

> **Note**
>
> O uso de autoridades de certificação privadas para mTLS está disponível somente para Amazon MQ para RabbitMQ versão 4 e superior.

> **Important**
>
> O Amazon MQ para RabbitMQ impõe o uso de arquivos de AWS ARNs certificado e chave privada. Consulte Suporte de ARN na configuração do RabbitMQ para obter mais detalhes.

Nesta página

Endpoint AMQP

Configure o mTLS para conexões de clientes com o endpoint AMQP. Isso é usado com a autenticação do certificado SSL. Para configurações compatíveis, consulteAutenticação de certificado SSL.

Plugin de gerenciamento RabbitMQ

Configure o mTLS para conexões com a interface de gerenciamento do RabbitMQ.

> **Note**
>
> O mTLS estrito não é compatível com a API de gerenciamento.

Configurações compatíveis

`aws.arns.management.ssl.cacertfile`

Arquivo de autoridade de certificação para validar certificados de clientes conectados à interface de gerenciamento.

`management.ssl.verify`

Modo de verificação por pares. Valores suportados:`verify_none`, `verify_peer`

`management.ssl.depth`

Profundidade máxima da cadeia de certificados para verificação.

`management.ssl.hostname_verification`

Modo de verificação do nome do host. Valores suportados:`wildcard`, `none`

Opções de SSL não suportadas

Os seguintes valores de configuração de SSL não são suportados:

Veja a 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`

Plug-in RabbitMQ 2.0 OAuth

Configure mTLS para conexões do Amazon MQ com OAuth o provedor de identidade 2.0. Para configurações compatíveis, consulteOAuth Autenticação e autorização 2.0.

Plugin de autenticação HTTP RabbitMQ

Configure o mTLS para conexões do Amazon MQ com o servidor de autenticação HTTP. Para configurações compatíveis, consulteAutenticação e autorização HTTP.

Plug-in LDAP do RabbitMQ

Configure mTLS para conexões do Amazon MQ com o servidor LDAP. Para configurações compatíveis, consulteAutenticação e autorização LDAP.

Conexões de cliente AMQP

Configure a verificação por pares TLS para conexões de clientes AMQP usadas pela federação e pelo shovel. Para obter mais informações, consulte Configuração SSL do cliente AMQP.

> **Important**
>
> No momento, o Amazon MQ não oferece suporte à configuração de certificados de cliente para conexões de clientes AMQP. Como resultado, a federação e o shovel não podem se

conectar a corretores habilitados para mTLS que exigem autenticação de certificado de cliente.

Configuração do limite de recursos

O Amazon MQ para RabbitMQ suporta a configuração dos limites de recursos do agente a partir do RabbitMQ 4. Quando você cria um agente, o Amazon MQ aplica automaticamente valores padrão a esses limites de recursos. Esses padrões atuam como barreiras para proteger a disponibilidade de seu corretor e, ao mesmo tempo, acomodar os padrões comuns de uso do cliente. Você pode personalizar o comportamento do seu agente alterando os valores de configuração limite para melhor atender aos seus requisitos específicos de carga de trabalho. Para obter mais detalhes sobre os valores padrão e máximos permitidos, consultethe section called “Diretrizes de dimensionamento”.

Nomes de recursos e chaves de configuração

Nome do recurso	Chave de configuração
Conexão	`connection_max`
Canal	`channel_max_per_node`
Fila	`cluster_queue_limit`
Vhost	`vhost_max`
Shovel	`runtime_parameters.limits.shovel`
Exchange	`cluster_exchange_limit`
Consumidor por canal	`consumer_max_per_channel`
Tamanho máximo de mensagem	`max_message_size`

Como substituir os limites de recursos

Você pode substituir os limites de recursos usando a API do Amazon MQ e o console do Amazon MQ.

O exemplo a seguir mostra como substituir o limite padrão de contagem de filas usando: AWS CLI

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

Uma invocação bem-sucedida cria uma revisão de configuração. Você deve associar a configuração ao seu broker RabbitMQ e reinicializar o broker para aplicar a substituição. Para obter mais detalhes, consulte RabbitMQ Broker Configurations

Erros de substituição do limite de recursos

Associar ou criar um agente com valores de configuração fora do intervalo suportado resulta em uma resposta de erro semelhante à seguinte:

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

Suporte de ARN na configuração do RabbitMQ

O Amazon MQ para RabbitMQ suporta AWS ARNs os valores de algumas definições de configuração do RabbitMQ. Isso é habilitado pelo plug-in da comunidade RabbitMQ rabbitmq-aws. Esse plug-in é desenvolvido e mantido pelo Amazon MQ e também pode ser usado em corretores RabbitMQ auto-hospedados e não gerenciados pelo Amazon MQ.

⚠ Considerações importantes

- Os valores de ARN resolvidos recuperados pelo plug-in aws são passados diretamente para o processo RabbitMQ em tempo de execução. Eles não são armazenados em outro lugar no nó RabbitMQ.
- O Amazon MQ para RabbitMQ exige uma função do IAM que possa ser assumida pelo Amazon MQ para acessar o configurado. ARNs Isso é configurado pela configuraçãoaws.arns.assume_role_arn.
- Os usuários que UpdateBroker APIs ligam CreateBroker ou têm uma configuração de agente que inclui uma função do IAM devem ter a iam:PassRole permissão para essa função.

- A função do IAM deve existir na mesma AWS conta do broker RabbitMQ. Tudo ARNs na configuração deve estar presente na mesma AWS região do broker RabbitMQ.
- O Amazon MQ adiciona chaves condicionais globais do IAM `aws:SourceAccount` e `aws:SourceArn` ao assumir a função do IAM. Esses valores devem ser usados na política do IAM anexada à função de proteção delegada confusa.

Nesta página

Chaves compatíveis

Função obrigatória do IAM

`aws.arns.assume_role_arn`

ARN da função do IAM que o Amazon MQ assume para acessar outros recursos. AWS Obrigatório quando qualquer outra configuração de ARN é usada.

Endpoint AMQP

Chave de configuração	Description
`aws.arns.ssl_options.cacertfile`	Arquivo de autoridade de certificação para conexões de SSL/TLS clientes. O Amazon MQ exige o uso do Amazon S3 ou o armazenamento do certificado.

Plugin de gerenciamento RabbitMQ

Chave de configuração	Description
`aws.arns.management.ssl.cacertfile`	Arquivo de autoridade de certificação para SSL/TLS conexões de interface de gerenciamento. O Amazon MQ exige o uso do Amazon S3 ou o armazenamento do certificado.

Plug-in RabbitMQ 2.0 OAuth

Chave de configuração	Description
`aws.arns.auth_oauth2.https.cacertfile`	Arquivo de autoridade de certificação para conexões HTTPS OAuth 2.0. O Amazon MQ exige o uso do Amazon S3 ou o armazenamento do certificado.

Plugin de autenticação HTTP RabbitMQ

Chave de configuração	Description
`aws.arns.auth_http.ssl_options.cacertfile`	Arquivo de autoridade de certificação para SSL/TLS conexões de autenticação HTTP. O Amazon MQ exige o uso do Amazon S3 ou o armazenamento do certificado.
`aws.arns.auth_http.ssl_options.certfile`	Arquivo de certificado para conexões TLS mútuas entre o Amazon MQ e o servidor de autenticação HTTP. O Amazon MQ exige o uso do Amazon S3 ou o armazenamento do certificado.
`aws.arns.auth_http.ssl_options.keyfile`	Arquivo de chave privada para conexões TLS mútuas entre o Amazon MQ e o servidor de autenticação HTTP. O Amazon MQ exige o uso AWS Secrets Manager para armazenar a chave privada.

Plug-in LDAP do RabbitMQ

Chave de configuração	Description
`aws.arns.auth_ldap.ssl_options.cacertfile`	Arquivo de autoridade de certificação para conexões LDAP. SSL/TLS O Amazon MQ exige o uso do Amazon S3 ou o armazenamento do certificado.
`aws.arns.auth_ldap.ssl_options.certfile`	Arquivo de certificado para conexões TLS mútuas entre o Amazon MQ e o servidor LDAP. O Amazon MQ exige o uso do Amazon S3 ou o armazenamento do certificado.
`aws.arns.auth_ldap.ssl_options.keyfile`	Arquivo de chave privada para conexões TLS mútuas entre o Amazon MQ e o servidor LDAP. O Amazon MQ exige o uso AWS Secrets Manager para armazenar a chave privada.
`aws.arns.auth_ldap.dn_lookup_bind.password`	Senha para associação de pesquisa LDAP DN. O Amazon MQ exige o uso AWS Secrets Manager para armazenar a senha como um valor de texto simples.
`aws.arns.auth_ldap.other_bind.password`	Senha para outro vínculo LDAP. O Amazon MQ exige o uso AWS Secrets Manager para armazenar a senha como um valor de texto simples.

Exemplos de políticas do IAM

Para exemplos de políticas do IAM, incluindo documentos de política de assumir funções e documentos de política de funções, consulte o exemplo de implementação do CDK.

Consulte Usando autenticação e autorização LDAP as etapas sobre como configurar AWS Secrets Manager os recursos do Amazon S3.

Validação de acesso

Para solucionar cenários em que os valores do ARN não podem ser buscados, o plug-in aws oferece suporte a um endpoint da API de gerenciamento do RabbitMQ que pode ser chamado para verificar se o Amazon MQ é capaz de assumir a função e resolver com êxito. AWS ARNs Isso evita a necessidade de atualizar a configuração do agente, atualizar o agente com a nova revisão de configuração e reinicializar o agente para testar as alterações na configuração.

> **Note**
>
> O uso dessa API requer um usuário administrador existente do RabbitMQ. O Amazon MQ recomenda a criação de agentes de teste com um usuário interno, além de outros métodos de acesso. Consulte habilitar a autenticação OAuth 2.0 e a autenticação simples (interna). Esse usuário pode então ser usado para acessar a API de validação.

> **Note**
>
> Embora o plug-in aws ofereça suporte à transmissão de uma nova função como entrada para a API de validação, esse parâmetro não é suportado pelo Amazon MQ. A função do IAM usada para validação deve corresponder ao valor `aws.arns.assume_role_arn` da configuração do agente.

Estados de quarentena de corretores relacionados

Para obter informações sobre os estados de quarentena do corretor relacionados a problemas de suporte do ARN, consulte:

- RABBITMQ_INVALID_ASSUME ROLE
- RABBITMQ_INVALID_ARN_LDAP
- RABBITMQ_INVALID_ARN

Exemplo de cenário

- `b-f0fc695e-2f9c-486b-845a-988023a3e55b`O corretor foi configurado para usar a função IAM `<role>` para acessar o AWS Secrets Manager segredo `<arn>`
- Se a função fornecida ao Amazon MQ não tiver permissão de leitura no AWS Secrets Manager segredo, o seguinte erro será mostrado nos registros do RabbitMQ:

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

  Além disso, o corretor entrará no estado de `INVALID_ASSUMEROLE` quarentena. Para obter mais informações, consulte INVALID_ASSUMEROLE.

- As tentativas de autenticação LDAP falharão com o seguinte erro:

```
[error] <0.254.0> LDAP bind failed: invalid_credentials
```

- Corrija a função do IAM com as permissões adequadas
- Chame o endpoint de validação para verificar se o RabbitMQ agora consegue acessar o segredo:

```
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 '.'
```

Configuração SSL do cliente AMQP

Federation e shovel usam o AMQP para comunicação entre corretores upstream e downstream. Por padrão, a verificação por pares TLS está habilitada para clientes AMQP no Amazon MQ para RabbitMQ 4. Com essa configuração, os clientes AMQP da federação e do shovel que executam nos corretores Amazon MQ realizarão a verificação por pares ao estabelecer conexões com o corretor upstream.

Os clientes AMQP executados em corretores Amazon MQ oferecem suporte às mesmas autoridades de certificação da Mozilla. Se você não usa o ACM, use um certificado emitido por uma CA na Lista de certificados de CA incluídos da Mozilla. Se seu agente local usar certificados de outras autoridades de certificação, a verificação de SSL falhará. Você pode desativar a verificação por pares do TLS para esses casos de uso.

> **Important**
>
> No momento, o Amazon MQ não oferece suporte à configuração de certificados de cliente para conexões de clientes AMQP. Como resultado, a federação e o shovel não podem se conectar a corretores habilitados para mTLS que exigem autenticação de certificado de cliente.

> **Important**
>
> No Amazon MQ para RabbitMQ 3, as propriedades SSL dos clientes AMQP são configuradas com os padrões do RabbitMQ (verify_none). O Amazon MQ para RabbitMQ 3 não suporta a substituição desses padrões.

> **Note**
>
> Com a `verify_peer` configuração padrão, você pode estabelecer conexões de federação e escavação entre quaisquer 2 agentes do Amazon MQ, mas isso não dá suporte ao estabelecimento da conexão entre o agente do Amazon MQ e os corretores privados ou agentes locais que estejam executando com certificados CA que não sejam do Amazon MQ. Para se conectar com corretores particulares ou locais, você precisa desativar a verificação por pares no agente downstream do Amazon MQ.

Chave de configuração SSL do cliente AMQP

Configuração	Chave de configuração	Valores suportados
Verificação por pares SSL do cliente AMQP	`amqp_client.ssl_options.verify`	`verify_none` , `verify_peer`

Como substituir a verificação por pares SSL do cliente AMQP

Você pode substituir a verificação por pares SSL do cliente AMQP usando a API Amazon MQ e o console Amazon MQ nos corretores RabbitMQ 4.

O exemplo a seguir mostra como substituir a verificação por pares SSL do cliente AMQP usando o: AWS CLI

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

Uma invocação bem-sucedida cria uma revisão de configuração. Você deve associar a configuração ao seu broker RabbitMQ e reinicializar o broker para aplicar a substituição. Para obter mais detalhes, consulte Creating and applying broker configurations

> **Important**
>
> Durante o uso`verify_none`, a criptografia SSL ainda está ativa, mas a identidade do par não é verificada. Use essa configuração somente quando necessário e certifique-se de confiar no caminho da rede até o agente de destino.

Autenticação e autorização do Amazon MQ para RabbitMQ

O Amazon MQ para RabbitMQ oferece suporte aos seguintes métodos de autenticação e autorização:

Autorização e autenticação simples

Nesse método, os usuários do agente são armazenados internamente no agente do RabbitMQ e gerenciados por meio do console Web ou da API de gerenciamento. As permissões para vhosts, trocas, filas e tópicos são configuradas diretamente no RabbitMQ. Esse é o método padrão. Para obter mais informações, consulte Autenticação e autorização simples.

OAuth Autenticação e autorização 2.0

Nesse método, os usuários do broker e suas permissões são gerenciados por um provedor de identidade (IdP) externo OAuth 2.0. A autenticação do usuário e as permissões de recursos para vhosts, trocas, filas e tópicos são centralizadas por meio do sistema de escopo do provedor OAuth 2.0. Isso simplifica o gerenciamento de usuários e permite a integração com os sistemas de identidade existentes. Para obter mais informações, consulte Autenticação e autorização OAuth 2.0.

Autenticação e autorização do IAM

Nesse método, os usuários do broker se autenticam usando credenciais AWS do IAM por meio da federação de saída do IAM. As credenciais do IAM são usadas para obter tokens JWT do AWS Security Token Service (STS), e esses tokens JWT servem como tokens OAuth 2.0 para autenticação. Esse método aproveita o suporte OAuth 2.0 existente no Amazon MQ para RabbitMQ,

AWS onde atua como o provedor de identidade 2.0. OAuth A autenticação do usuário é gerenciada pelo AWS IAM, enquanto as permissões de recursos para vhosts, trocas, filas e tópicos são gerenciadas por meio de políticas do IAM e aliases de escopo configurados no RabbitMQ. Para obter mais informações, consulte Autenticação e autorização do IAM.

Autenticação e autorização LDAP

Nesse método, os usuários do broker e suas permissões são gerenciados por um serviço de diretório LDAP externo. A autenticação do usuário e as permissões de recursos são centralizadas por meio do servidor LDAP, permitindo que os usuários acessem o RabbitMQ usando suas credenciais de serviço de diretório existentes. Para obter mais informações, consulte Autenticação e autorização LDAP.

Autenticação e autorização HTTP

Nesse método, os usuários do broker e suas permissões são gerenciados por um servidor HTTP externo. A autenticação do usuário e as permissões de recursos são centralizadas por meio do servidor HTTP, permitindo que os usuários acessem o RabbitMQ usando seu próprio provedor de autenticação e autorização. Para obter mais informações sobre esse método, consulte Autenticação e autorização HTTP.

Autenticação de certificado SSL

O Amazon MQ oferece suporte a TLS mútuo (mTLS) para corretores RabbitMQ. O plug-in de autenticação SSL usa certificados de cliente de conexões mTLS para autenticar usuários. Nesse método, os usuários do broker são autenticados usando certificados de cliente X.509 em vez de credenciais de nome de usuário e senha. O certificado do cliente é validado em relação a uma Autoridade Certificadora (CA) confiável e o nome de usuário é extraído de um campo no certificado, como Nome comum (CN) ou Nome alternativo do assunto (SAN). Esse método fornece autenticação forte sem transmitir credenciais pela rede. Para obter mais informações, consulte Autenticação de certificado SSL.

> **Note**
>
> O RabbitMQ suporta vários métodos de autenticação e autorização para serem usados simultaneamente. Por exemplo, você pode ativar a autenticação OAuth 2.0 e a autenticação simples (interna). Para obter mais informações, consulte a seção do tutorial OAuth 2.0 sobre como habilitar a autenticação OAuth 2.0 e simples (interna) e a documentação de controle de acesso do RabbitMQ.

O Amazon MQ recomenda criar um usuário interno ao testar as configurações de autenticação. Isso permite que a configuração de acesso seja validada usando a API de gerenciamento do RabbitMQ. Para obter mais informações, consulte Validação de acesso.

Autorização e autenticação simples

Usuários do agente do Amazon MQ para RabbitMQ

> **Note**
>
> Este tópico descreve o gerenciamento de usuários do broker com o mecanismo interno padrão de autenticação e autorização do RabbitMQ. Para obter informações sobre todos os métodos de autenticação e autorização compatíveis, consulte Amazon MQ for RabbitMQ Authentication and Authorization.

Cada conexão de cliente AMQP 0-9-1 tem um usuário associado. Esse usuário deve ser autenticado. Cada conexão de cliente também tem como alvo um host virtual (vhost). O usuário deve ter um conjunto de permissões para esse vhost. Um usuário pode ter permissão para configure (configurar), write (gravar) em, e read (ler) de filas e trocas em um vhost. Você especifica as credenciais do usuário e o vhost de destino quando a conexão é estabelecida.

Quando você cria um agente do Amazon MQ para RabbitMQ pela primeira vez, o Amazon MQ usa as credenciais de login que você fornece para criar um usuário do RabbitMQ com a tag `administrator`. Em seguida, você pode adicionar e gerenciar usuários através do management API (API de gerenciamento) ou o console da Web RabbitMQ. Você também pode usar o console da Web RabbitMQ ou a API de gerenciamento para definir ou modificar permissões e etiquetas de usuário.

> **Note**
>
> Os usuários do RabbitMQ não serão armazenados ou exibidos por meio da API de Users (Usuários) do Amazon MQ.

> **Important**
>
> O Amazon MQ para RabbitMQ não permite o nome de usuário “convidado” e excluirá a conta de convidado padrão quando você criar um agente. O Amazon MQ também excluirá periodicamente qualquer conta de “convidado” criada pelo cliente.

Para criar um novo usuário com a API de gerenciamento RabbitMQ, use o seguinte endpoint da API e o corpo da solicitação. Substitua *username* e *password* por suas novas credenciais de login.

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

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

> **Important**
>
> - Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de usuário do agente. Os nomes de usuário dos corretores podem ser acessados por outros AWS serviços, incluindo CloudWatch registros. Nomes de usuário do agente não devem ser usados para dados privados ou sigilosos.
> - Se você perder o acesso a todas as contas de administrador, consulte Recuperando o acesso do agente para usar a autenticação do IAM para recuperação.

A chave `tags` é obrigatória e é uma lista de etiquetas separadas por vírgulas para o usuário. O Amazon MQ é compatível com as tags de usuário `administrator`, `management`, `monitoring` e `policymaker`.

Você pode definir permissões para um usuário individual usando o seguinte endpoint da API e o corpo da solicitação. Substitua *vhost* e *username* por suas informações. Para o vhost padrão `/`, use `%2F`.

```
PUT /api/permissions/vhost/username HTTP/1.1

    {"configure":".*","write":".*","read":".*"}
```

> **Note**
>
> As chaves `configure`, `read` e `write` são obrigatórias.

Usando o valor de caractere curinga `.*`, esta operação concederá permissões de leitura, gravação e configuração para o usuário, em todas as filas no vhost especificado. Para obter mais informações sobre como gerenciar usuários por meio da API de gerenciamento do RabbitMQ, consulte RabbitMQ Management HTTP API (API HTTP de gerenciamento do RabbitMQ).

OAuth Autenticação e autorização 2.0 para Amazon MQ para RabbitMQ

O Amazon MQ para RabbitMQ oferece suporte a vários métodos de autenticação e autorização. Para obter informações sobre todos os métodos compatíveis, consulte Autenticação e autorização do Amazon MQ para corretores do RabbitMQ.

Na autenticação e autorização OAuth 2.0, os usuários do broker e suas permissões são gerenciados por um provedor de identidade (IdP) externo OAuth 2.0. A autenticação do usuário e as permissões de recursos para vhosts, trocas, filas e tópicos são centralizadas por meio do sistema de escopo do provedor OAuth 2.0. Isso simplifica o gerenciamento de usuários e permite a integração com os sistemas de identidade existentes.

> **Considerações importantes**
>
> - OAuth A integração 2.0 não é suportada no Amazon MQ para corretores ActiveMQ.
> - O Amazon MQ para RabbitMQ não oferece suporte a certificado de servidor emitido por uma CA privada.
> - O plug-in RabbitMQ OAuth 2.0 não oferece suporte a endpoints de introspecção de tokens e tokens de acesso opacos. Também não realiza verificações de revogação de token.
> - Você deve incluir a permissão do IAM,`mq:UpdateBrokerAccessConfiguration`, para habilitar OAuth 2.0 em corretores existentes.
> - O Amazon MQ cria automaticamente um usuário do sistema chamado `monitoring-AWS-OWNED-DO-NOT-DELETE` com permissões somente de monitoramento. Esse usuário usa o sistema de autenticação interna do RabbitMQ mesmo em corretores OAuth habilitados para 2.0 e está restrito apenas ao acesso à interface de loopback.

Para obter informações sobre como configurar OAuth 2.0 para seus corretores Amazon MQ para RabbitMQ, consulte. Uso da autorização e da autenticação OAuth 2.0

Nesta página

Configurações OAuth 2.0 suportadas

O Amazon MQ para RabbitMQ oferece suporte a todas as variáveis configuráveis no plug-in OAuth RabbitMQ 2.0, com as seguintes exceções:

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

  Como o Amazon MQ não oferece suporte a essa chave, não oferecemos suporte a UAA como IdP.
- `management.oauth_resource_servers.{id/index}.oauth_client_secret`
- `auth_oauth2.signing_keys.{id/index}`

Validações adicionais para autenticação OAuth 2.0

O Amazon MQ também impõe as seguintes validações adicionais para a autenticação 2.0: OAuth

- Tudo URLs precisa começar`https://`.
- Algoritmos de assinatura compatíveis: `Ed25519`, `Ed25519ph`, `Ed448`, `Ed448ph`, `EdDSA`, `ES256K`, `ES256`, `ES384`, `ES512`, `HS256`, `HS384`, `HS512`, `PS256`, `PS384`, `PS512`, `RS256`, `RS384` e `RS512`.

Autenticação e autorização do IAM para Amazon MQ para RabbitMQ

O Amazon MQ para RabbitMQ oferece suporte a vários métodos de autenticação e autorização. Para obter informações sobre todos os métodos compatíveis, consulte Autenticação e autorização do Amazon MQ para corretores do RabbitMQ.

A autenticação e autorização do IAM permitem que os usuários do broker se autentiquem usando credenciais AWS do IAM por meio da federação de saída do IAM. Nesse método, as credenciais do IAM são usadas para obter tokens JWT do AWS Security Token Service (STS). Esses tokens JWT

servem como tokens OAuth 2.0 para autenticação, aproveitando o suporte OAuth 2.0 existente no Amazon MQ para RabbitMQ, AWS onde atua como o provedor de identidade 2.0. OAuth AWS O IAM lida com a autenticação do usuário, enquanto as permissões de recursos para hosts virtuais, trocas, filas e tópicos são gerenciadas por meio de políticas do IAM e aliases de escopo configurados no RabbitMQ.

⚠ Considerações importantes

- A autenticação do IAM é compatível com as versões 3.13, 4.2 e superiores do RabbitMQ. Ele não é compatível com o Amazon MQ para corretores ActiveMQ.
- A autenticação do IAM exige que a federação de saída do IAM esteja configurada e disponível em sua AWS conta.
- Esse método se baseia na infraestrutura OAuth 2.0 existente no Amazon MQ para RabbitMQ, AWS servindo como provedor de identidade 2.0. OAuth
- O Amazon MQ cria automaticamente um usuário do sistema chamado `monitoring-AWS-OWNED-DO-NOT-DELETE` com permissões somente de monitoramento. Esse usuário usa o sistema de autenticação interna do RabbitMQ mesmo em corretores habilitados para IAM e está restrito apenas ao acesso à interface de loopback.

Nesta página

Como funciona a autenticação do IAM

A autenticação do IAM para o Amazon MQ for RabbitMQ usa a federação de saída do IAM para permitir que as credenciais do IAM sejam autenticadas com os corretores AWS do RabbitMQ. As credenciais do IAM são usadas para obter tokens JWT do AWS Security Token Service (STS), e esses tokens JWT servem como tokens OAuth 2.0 para autenticação com o corretor RabbitMQ.

Limitações

A autenticação do IAM para Amazon MQ para RabbitMQ tem a seguinte limitação:

- Configuração de declaração de escopo — Você não pode usar uma declaração de escopo diretamente porque o token JWT do STS está aninhado. A chave é `sts.amazonaws.com` que

requer o uso de aliases de escopo na configuração do RabbitMQ para mapear as funções do IAM para as permissões do RabbitMQ. Essa limitação também impede o uso total das políticas do IAM para autorização, exigindo a configuração do RabbitMQ para autorização.

Para obter informações sobre como configurar a autenticação e autorização do IAM para seus corretores Amazon MQ para RabbitMQ, consulte. Usando autenticação e autorização do IAM

Autenticação e autorização HTTP para Amazon MQ para RabbitMQ

O Amazon MQ para RabbitMQ oferece suporte à autenticação e autorização de usuários do broker usando um servidor HTTP externo. Para outros métodos compatíveis, consulte Autenticação e autorização do Amazon MQ para corretores do RabbitMQ.

> **Note**
>
> O plug-in de autenticação HTTP está disponível somente para o Amazon MQ for RabbitMQ versão 4 e superior.

> **Considerações importantes**
>
> - O servidor HTTP precisa estar acessível pela Internet pública. O Amazon MQ para RabbitMQ pode ser configurado para autenticação no servidor HTTP usando TLS mútuo.
> - O Amazon MQ para RabbitMQ impõe o uso de AWS ARNs para configurações que exigem acesso ao sistema de arquivos local. Consulte Suporte de ARN na configuração do RabbitMQ para obter mais detalhes.
> - Você deve incluir a permissão do IAM,`mq:UpdateBrokerAccessConfiguration`, para habilitar a autenticação HTTP em corretores existentes.
> - O Amazon MQ cria automaticamente um usuário do sistema chamado `monitoring-AWS-OWNED-DO-NOT-DELETE` com permissões somente de monitoramento. Esse usuário usa o sistema de autenticação interna do RabbitMQ mesmo em corretores habilitados para HTTP e está restrito apenas ao acesso à interface de loopback. O Amazon MQ impede a exclusão desse usuário adicionando a tag de usuário protegida.

Para obter informações sobre como configurar a autenticação HTTP para seus corretores Amazon MQ para RabbitMQ, consulte. Usando autenticação e autorização HTTP

Nesta página

Configurações HTTP suportadas

O Amazon MQ para RabbitMQ oferece suporte a todas as variáveis configuráveis no plug-in de autenticação HTTP do RabbitMQ, com as seguintes exceções que exigem. AWS ARNs Para obter detalhes sobre o suporte ao ARN, consulte Suporte ao ARN na configuração do RabbitMQ.

Configurações que exigem ARNs

`auth_http.ssl_options.cacertfile`

Use `aws.arns.auth_http.ssl_options.cacertfile` em vez disso

`auth_http.ssl_options.certfile`

Use `aws.arns.auth_http.ssl_options.certfile` em vez disso

`auth_http.ssl_options.keyfile`

Use `aws.arns.auth_http.ssl_options.keyfile` em vez disso

Opções de SSL não suportadas

As seguintes opções de configuração de SSL também não são suportadas:

Veja a 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`

Validações adicionais para configurações HTTP no Amazon MQ

O Amazon MQ também impõe as seguintes validações adicionais para autenticação e autorização HTTP:

- `auth_http.http_method`deve ser um `get` ou `post`
- As configurações de caminho a seguir devem usar HTTPS: URLs
  - `auth_http.user_path`
  - `auth_http.vhost_path`
  - `auth_http.resource_path`
  - `auth_http.topic_path`
- Se alguma configuração exigir o uso de um AWS ARN, `aws.arns.assume_role_arn` deverá ser fornecida.

Autenticação de certificado SSL para Amazon MQ para RabbitMQ

O Amazon MQ para RabbitMQ oferece suporte à autenticação de usuários do broker usando certificados de cliente X.509. Para outros métodos compatíveis, consulte Autenticação e autorização do Amazon MQ para corretores do RabbitMQ.

> **Note**
>
> O plug-in de autenticação de certificado SSL está disponível somente para o Amazon MQ for RabbitMQ versão 4 e superior.

> **Considerações importantes**
>
> - Os certificados do cliente devem ser assinados por uma Autoridade Certificadora (CA) confiável. O Amazon MQ para RabbitMQ valida a cadeia de certificados durante a autenticação.
> - O Amazon MQ para RabbitMQ impõe o uso de configurações relacionadas a certificados, como certificados CA, e AWS ARNs para configurações que exigem acesso ao sistema de arquivos local. Consulte Suporte de ARN na configuração do RabbitMQ para obter mais detalhes.
> - O Amazon MQ cria automaticamente um usuário do sistema chamado `monitoring-AWS-OWNED-DO-NOT-DELETE` com permissões somente de monitoramento. Esse usuário usa o sistema de autenticação interna do RabbitMQ mesmo em corretores habilitados para certificados SSL e está restrito apenas ao acesso à interface de loopback. O Amazon MQ impede a exclusão desse usuário adicionando a tag de usuário protegida.

Para obter informações sobre como configurar a autenticação de certificado SSL para seus corretores Amazon MQ para RabbitMQ, consulte. Usando a autenticação de certificado SSL

Nesta página

Configurações SSL suportadas

O Amazon MQ para RabbitMQ oferece suporte SSL/TLS à configuração de conexões de clientes. Para obter detalhes sobre o suporte ao ARN, consulte Suporte ao ARN na configuração do RabbitMQ.

Configurações que exigem ARNs

`ssl_options.cacertfile`

Use `aws.arns.ssl_options.cacertfile` em vez disso

Configurações de login do certificado SSL

As configurações a seguir controlam como os nomes de usuário são extraídos dos certificados do cliente:

`ssl_cert_login_from`

Especifica qual campo de certificado usar para extração do nome de usuário. Valores com suporte:

- `distinguished_name`- Use o nome distinto completo
- `common_name`- Use o campo Nome comum (CN)
- `subject_alternative_name`ou `subject_alt_name` - Use o nome alternativo do assunto

`ssl_cert_login_san_type`

Ao usar o Subject Alternative Name, especifica o tipo de SAN. Valores suportados:`dns`,`ip`,`email`,`uri`, `other_name`

`ssl_cert_login_san_index`

Ao usar o Subject Alternative Name, especifica o índice da entrada SAN a ser usada (com base em zero). Deve ser um número inteiro não negativo.

Opções de SSL para conexões de clientes

As seguintes opções de SSL se aplicam às conexões do cliente:

`ssl_options.verify`

Modo de verificação por pares. Valores suportados:`verify_none`, `verify_peer`

`ssl_options.fail_if_no_peer_cert`

Se as conexões devem ser rejeitadas se o cliente não fornecer um certificado. Valor booleano.

`ssl_options.depth`

Profundidade máxima da cadeia de certificados para verificação.

`ssl_options.hostname_verification`

Modo de verificação do nome do host. Valores suportados:`wildcard`, `none`

Opções de SSL não suportadas

As seguintes opções de configuração de SSL não são suportadas:

Veja a 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`

Validações adicionais para configurações de SSL no Amazon MQ

O Amazon MQ também impõe as seguintes validações adicionais para autenticação de certificados SSL:

- Se alguma configuração exigir o uso de um AWS ARN, `aws.arns.assume_role_arn` deverá ser fornecida.

Autenticação e autorização LDAP para Amazon MQ para RabbitMQ

O Amazon MQ para RabbitMQ oferece suporte à autenticação e autorização de usuários do broker usando um servidor LDAP externo. Para outros métodos compatíveis, consulte Autenticação e autorização do Amazon MQ para corretores do RabbitMQ.

⚠ Considerações importantes

- O servidor LDAP precisa estar acessível pela Internet pública. O Amazon MQ para RabbitMQ pode ser configurado para autenticação no servidor LDAP usando TLS mútuo.
- O Amazon MQ para RabbitMQ impõe o uso de AWS ARNs configurações LDAP confidenciais, como senhas, e configurações que exigem acesso ao sistema de arquivos local. Consulte Suporte de ARN na configuração do RabbitMQ para obter mais detalhes.
- Você deve incluir a permissão do IAM,`mq:UpdateBrokerAccessConfiguration`, para habilitar o LDAP em corretores existentes.
- O Amazon MQ cria automaticamente um usuário do sistema chamado `monitoring-AWS-OWNED-DO-NOT-DELETE` com permissões somente de monitoramento. Esse usuário usa o sistema de autenticação interna do RabbitMQ mesmo em corretores habilitados para LDAP e está restrito apenas ao acesso à interface de loopback. O Amazon MQ impede a exclusão desse usuário adicionando a tag de usuário protegida.

Para obter informações sobre como configurar o LDAP para seus corretores Amazon MQ para RabbitMQ, consulte. Usando autenticação e autorização LDAP

Nesta página

Configurações LDAP suportadas

O Amazon MQ para RabbitMQ oferece suporte a todas as variáveis configuráveis no plug-in LDAP do RabbitMQ, com as seguintes exceções que exigem. AWS ARNs Para obter detalhes sobre o suporte ao ARN, consulte Suporte ao ARN na configuração do RabbitMQ.

Configurações que exigem ARNs

`auth_ldap.dn_lookup_bind.password`

Use `aws.arns.auth_ldap.dn_lookup_bind.password` em vez disso

`auth_ldap.other_bind.password`

Use `aws.arns.auth_ldap.other_bind.password` em vez disso

`auth_ldap.ssl_options.cacertfile`

Use `aws.arns.auth_ldap.ssl_options.cacertfile` em vez disso

`auth_ldap.ssl_options.certfile`

Use `aws.arns.auth_ldap.ssl_options.certfile` em vez disso

`auth_ldap.ssl_options.keyfile`

Use `aws.arns.auth_ldap.ssl_options.keyfile` em vez disso

Opções de SSL não suportadas

As seguintes opções de configuração de SSL também não são suportadas:

Veja a 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`

Validações adicionais para configurações LDAP no Amazon MQ

O Amazon MQ também impõe as seguintes validações adicionais para autenticação e autorização LDAP:

- `auth_ldap.log`não pode ser definido como `network_unsafe`
- O servidor LDAP deve usar LDAPS. `auth_ldap.use_ssl`Ou `auth_ldap.use_starttls` deve ser habilitado explicitamente
- Se alguma configuração exigir o uso de um AWS ARN, `aws.arns.assume_role_arn` deverá ser fornecida.
- `auth_ldap.servers`deve ser um endereço IP válido ou um FQDN válido
- As chaves a seguir devem ser um nome distinto 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

O Amazon MQ para RabbitMQ também oferece suporte aos seguintes plug-ins.

- Plugin de gerenciamento RabbitMQ

- Plugin Shovel
- plug-in de federação
- Plugin consistente de troca de hash
- OAuth 2 plug-ins
- Plug-in LDAP
- Plug-in HTTP
- Plug-in de certificado SSL
- plug-in aws
- Plug-in JMS Topic Exchange

Plugin de gerenciamento RabbitMQ

O Amazon MQ para RabbitMQ oferece suporte ao plug-in de gerenciamento RabbitMQ, que fornece uma API de gerenciamento baseada em HTTP junto com uma interface de usuário baseada em navegador para o console web do RabbitMQ. Você pode usar o console da Web e a API de gerenciamento para criar e gerenciar usuários e políticas do agente.

Plugin shovel

O Amazon MQ para RabbitMQ oferece suporte ao plug-in shovel RabbitMQ, que permite mover mensagens de filas e trocas de uma corretora para outra. Você pode usar o shovel para conectar agentes de baixo acoplamento e distribuir mensagens longe dos nós com cargas de mensagens mais pesadas.

> **Important**
>
> Você não pode configurar shovels entre filas ou trocas se o destino do shovel for um agente privado.
> O Amazon MQ não é compatível com o uso de shovels estáticas.

Somente escavadeiras dinâmicas são suportadas. As escavadeiras dinâmicas são configuradas usando parâmetros de tempo de execução e podem ser iniciadas e interrompidas a qualquer momento de forma programática por uma conexão de cliente. Por exemplo, usando a API de

gerenciamento do RabbitMQ, você pode criar uma solicitação PUT para o seguinte endpoint da API para configurar uma escavadeira dinâmica. No exemplo, {vhost} pode ser substituído pelo nome do vhost do corretor e {name} substituído pelo nome da nova escavadeira dinâmica.

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

No corpo da solicitação, você deve especificar uma fila ou uma troca, mas não ambos. O exemplo abaixo configura uma escavadeira dinâmica entre uma fila local especificada em src-queue e uma fila remota definida em dest-queue. Da mesma forma, você pode usar os parâmetros src-exchange e dest-exchange para configurar uma escavadeira entre duas trocas.

```
{
"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"
}
}
```

Plugin de federação

O Amazon MQ oferece suporte a trocas e filas federadas usando o plug-in de federação RabbitMQ. Com a federação, você pode replicar o fluxo de mensagens entre filas, trocas e consumidores em agentes separados. Filas e trocas federadas usam point-to-point links para se conectar a colegas em outras corretoras. Enquanto as trocas federadas, por padrão, roteiam mensagens uma vez, as filas federadas podem mover mensagens várias vezes conforme necessário pelos consumidores.

Você pode usar federação para permitir que um agente downstream consuma uma mensagem de uma troca ou de uma fila em um upstream. Você pode habilitar a federação em agentes downstream usando o console da Web do RabbitMQ ou a API de gerenciamento.

> **Important**
>
> Não será possível configurar a federação se a fila ou troca de upstream estiver em um agente privado. Só será possível configurar a federação entre filas ou trocas em agentes

públicos ou entre uma fila ou troca de upstream em um agente público e uma fila ou troca de downstream em um agente privado.

Por exemplo, usando a API de gerenciamento, você pode configurar a federação fazendo o seguinte:

- Configure um ou mais upstreams que definem conexões de federação com outros nós. Você pode definir conexões de federação usando o console da Web do RabbitMQ ou a API de gerenciamento. Usando a API de gerenciamento, você pode criar uma solicitação POST para/api/parameters/federation-upstream/%2f/myupstreamcom o corpo da solicitação a seguir.

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

- Configure uma política para permitir que suas filas ou trocas se tornem federadas. Você pode configurar políticas usando o console da Web do RabbitMQ ou a API de gerenciamento. Usando a API de gerenciamento, você pode criar uma solicitação POST para/api/policies/%2f/federate-me com o corpo da solicitação a seguir.

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

> **Note**
>
> O corpo da solicitação pressupõe que as trocas no servidor sejam nomeadas começando com amq. Usar a expressão regular ^amq\. garantirá que a federação seja habilitada para todas as trocas cujos nomes comecem com “amq”. As trocas no seu servidor RabbitMQ podem ser nomeadas de forma diferente.

Plugin de troca de hash consistente

O Amazon MQ para RabbitMQ oferece suporte ao plug-in RabbitMQ Consistent Hash Exchange Type. As trocas de hash consistentes fazem o roteamento de mensagens para filas com base em um valor de hash calculado a partir da routing key (chave de roteamento) de uma mensagem. Considerando uma chave de roteamento razoavelmente uniforme, as trocas de Hash consistentes podem distribuir mensagens entre filas de maneira razoavelmente uniforme.

Para filas vinculadas a uma troca de hash consistente, a chave de vinculação determina number-as-a-string o peso de vinculação de cada fila. As filas com um peso de vinculação maior receberão uma distribuição proporcionalmente maior de mensagens da troca de hash consistente à qual estão vinculadas. Em uma topologia de troca de hash consistente, os editores podem simplesmente publicar mensagens no Exchange, mas os consumidores devem ser explicitamente configurados para consumir mensagens de filas específicas.

OAuth Plug-in 2.0

O Amazon MQ para RabbitMQ oferece suporte ao plug-in de back-end de autenticação 2OAuth . Esse plug-in é ativado condicionalmente com base na configuração do seu corretor. Quando ativado, esse plug-in fornece autenticação e autorização OAuth 2.0 com integração com provedores de identidade OAuth 2.0 externos para gerenciamento centralizado de usuários e controle de acesso. Para obter mais informações sobre a autenticação OAuth 2.0, consulteOAuth Autenticação e autorização 2.0.

Plug-in LDAP

O Amazon MQ para RabbitMQ oferece suporte ao plug-in de back-end de autenticação LDAP. Esse plug-in é ativado condicionalmente com base na configuração do seu corretor. Quando ativado, esse plug-in fornece autenticação e autorização LDAP com integração com serviços de diretório LDAP externos para autenticação e autorização centralizadas do usuário. Para obter mais informações sobre a autenticação LDAP, consulte. Autenticação e autorização LDAP

Plug-in HTTP

O Amazon MQ para RabbitMQ oferece suporte ao plug-in de back-end de autenticação HTTP. Esse plug-in é ativado condicionalmente com base na configuração do seu corretor. Quando ativado, esse plug-in fornece autenticação e autorização HTTP com integração com servidores HTTP externos para autenticação e autorização centralizadas do usuário. Para obter mais informações sobre a autenticação HTTP, consulteAutenticação e autorização HTTP.

> **Note**
>
> O plug-in de autenticação HTTP está disponível somente para o Amazon MQ for RabbitMQ versão 4 e superior.

Plug-in de certificado SSL

O Amazon MQ oferece suporte a TLS mútuo (mTLS) para corretores RabbitMQ. O plug-in de autenticação SSL usa certificados de cliente de conexões mTLS para autenticar usuários. Esse plug-in é ativado condicionalmente com base na configuração do seu corretor. Quando ativado, ele fornece autenticação baseada em certificado usando certificados de cliente X.509 para autenticação forte sem transmitir credenciais pela rede. Para obter mais informações sobre a autenticação do certificado SSL, consulteAutenticação de certificado SSL.

> **Note**
>
> O plug-in de autenticação de certificado SSL está disponível somente para o Amazon MQ for RabbitMQ versão 4 e superior.

plug-in aws

O plug-in aws é habilitado condicionalmente pelo Amazon MQ para RabbitMQ com base na configuração do seu agente. Esse plug-in comunitário, desenvolvido e mantido pela Amazon MQ, fornece recuperação segura de credenciais e certificados de AWS serviços usados AWS ARNs nas configurações do RabbitMQ. Para obter mais informações sobre o suporte ao ARN, consulte. ARN support in RabbitMQ configuration

Plug-in JMS Topic Exchange

O plug-in JMS Topic Exchange é sempre ativado pelo Amazon MQ para o RabbitMQ. Ele funciona com o cliente RabbitMQ JMS para permitir que aplicativos JMS novos e existentes se conectem ao Amazon MQ para RabbitMQ.

> **Note**
>
> O plug-in JMS Topic Exchange está disponível somente para Amazon MQ para RabbitMQ versão 4 e superior. Ele é ativado por padrão, mas só é ativado quando o cliente RabbitMQ JMS é usado para executar cargas de trabalho JMS.

Protocolos compatíveis

Você pode acessar seus corretores RabbitMQ usando qualquer linguagem de programação compatível com o RabbitMQ e habilitando o TLS para qualquer uma das seguintes especificações de protocolo:

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

Amazon MQ para suporte ao RabbitMQ JMS

Agora você pode executar cargas de trabalho do JMS 1.1, 2.0 e 3.1 no Amazon MQ para RabbitMQ 4 com o cliente RabbitMQ JMS.

Cliente RabbitMQ JMS

O cliente RabbitMQ JMS é uma biblioteca cliente JMS de código aberto que você precisa para conectar seu aplicativo JMS aos corretores Amazon MQ RabbitMQ. Para obter mais informações, visite o GitHub repositório oficial.

Compatível com JMS 1.1, 2.0 e 3.1 APIs

Do Amazon MQ para RabbitMQ 4 em diante, o plug-in está sempre ativado. `jms-topic-exchange` Portanto, você pode usar o Amazon MQ para RabbitMQ 4 e o cliente RabbitMQ JMS para sua carga de trabalho do JMS. Todos os JMS APIs definidos no JMS 1.1 são suportados, exceto:

- As sessões do servidor não APIs são suportadas.
- APIs As transações XA não são suportadas.
- O seletor JMS para o destino da fila JMS não é suportado.
- O atributo de `NoLocal` assinatura do JMS não é suportado.

Todos os recém-adicionados APIs no JMS 2.0 e no JMS 3.1 são suportados, exceto:

- `JMSProducer.setDeliveryDelay`A API não é suportada.

Para saber mais sobre como conectar seu aplicativo JMS ao Amazon MQ for RabbitMQ broker, consulte o tutorial sobre Conectando seu aplicativo JMS ao Amazon MQ for RabbitMQ broker

Autenticação e autorização

Todos os mecanismos de autenticação e autorização listados nesta seção são suportados. As credenciais usadas para se conectar ao broker usando o cliente JMS são as mesmas que se você estivesse se conectando ao broker RabbitMQ usando um cliente Java AMQP.

Interoperabilidade com filas AMQP no RabbitMQ

Você pode usar o cliente JMS RabbitMQ para enviar mensagens JMS para uma troca AMQP e consumir mensagens de uma fila AMQP (esse recurso não oferece suporte a tópicos de JMS). Isso permite que você interopere ou migre determinadas cargas de trabalho do JMS para cargas de trabalho do AMQP. Para obter mais informações, visite a documentação oficial do cliente.

Aplicar políticas ao Amazon MQ para RabbitMQ

Você pode aplicar políticas e limites personalizados com valores padrão recomendados pelo Amazon MQ. Se você excluiu as políticas e limites padrão recomendados e deseja recriá-los, ou se tiver criado vhosts adicionais e quiser aplicar as políticas e limites padrão aos novos vhosts, você pode usar as etapas a seguir.

Important

Nas versões 3.13 e inferiores do mecanismo Amazon MQ para RabbitMQ, a política atual padrão do operador é:

```
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
```

Nas versões 4.0 e superiores, a política padrão do operador foi alterada para:

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

Essa alteração é necessária porque o espelhamento de filas clássico e as configurações de política de HA não são compatíveis com o RabbitMQ 4.

Não é possível criar uma política que se aplique tanto a filas clássicas espelhadas quanto a filas de quórum. Se você quiser que sua política se aplique somente a filas de quórum, defina `--apply-to` como `quorum_queues`. Se você estiver usando filas clássicas espelhadas e filas de quórum, deverá criar uma política separada com `--apply-to:classic_queues` bem como uma política de filas de quórum.

Important

Para executar as etapas a seguir, é necessário ter um usuário do agente do Amazon MQ para RabbitMQ com permissões de administrador. Você pode usar o usuário administrador criado quando criou o agente pela primeira vez ou outro usuário que você possa ter criado posteriormente. A tabela a seguir fornece a etiqueta de usuário administrador necessária e as permissões como padrões de expressão regular (regexp).

Etiquetas	Ler regexp	Configurar regexp	Escrever regexp
`administrator`	`.*`	`.*`	`.*`

Para obter mais informações sobre criar usuários RabbitMQ e gerenciar etiquetas e permissões de usuário, consulte Usuários do agente do Amazon MQ para RabbitMQ.

Para aplicar políticas padrão e limites de host virtual usando o console da Web RabbitMQ

1. Faça login no console do Amazon MQ.
2. No painel de navegação à esquerda, escolha Agentes.
3. Na lista de agentes, escolha o nome do agente ao qual você deseja aplicar a nova política.
4. Na página de detalhes do agente, na seção Conexões, selecione a URL do Console Web do RabbitMQ. O console da Web do RabbitMQ é aberto em uma nova guia ou janela do navegador.
5. Faça login no console da Web do RabbitMQ com o nome de usuário e a senha do administrador do agente.
6. No console da Web do RabbitMQ, na parte superior da página, escolha Admin.
7. Na página Admin, no painel de navegação da direita, selecione Políticas.

8. Na página Políticas, você pode ver uma lista das Políticas de usuário atuais do agente. Abaixo das Políticas de usuário, expanda Adicionar/atualizar uma política.
9. Para criar uma política de agente, em Adicionar/atualizar uma política, faça o seguinte:

   a. Para o Host virtual, escolha o nome do vhost ao qual você deseja anexar as políticas da lista suspensa. Para escolher o vhost padrão, escolha /.

   > **Note**
   >
   > Se você não tiver criado vhosts adicionais, a opção Virtual host (Host virtual) não aparecerá no console RabbitMQ, e as políticas serão aplicadas somente ao vhost padrão.

   b. Em Name (Nome), insira um nome para a sua política, por exemplo **`policy-defaults`**.

   c. Para Pattern (Padrão), insira o padrão de expressão regular **`.*`** para que a política corresponda a todas as filas no agente.

   d. Para Apply to (Aplicar em), escolha Exchanges and queues (Trocas e filas) na lista suspensa.

   e. Para Priority (Prioridade), insira um número inteiro maior que todas as outras políticas aplicadas ao vhost. Você pode aplicar exatamente um conjunto de definições de política a filas e trocas RabbitMQ a qualquer momento. O RabbitMQ escolhe a política correspondente com o valor de prioridade mais alto. Para obter mais informações sobre prioridades de política e como combinar políticas, consulte Policies (Políticas) na Documentação do Servidor RabbitMQ.

   f. Para Definition (Definição), adicione os seguintes pares de chave-valor:

   - **`queue-mode=lazy`**. Selecione String (String) na lista suspensa.
   - **`overflow=reject-publish`**. Selecione String (String) na lista suspensa.

     > **Note**
     >
     > Não se aplica aos agentes de instância única.

   - **`max-length=`*`number-of-messages`***. *`number-of-messages`*Substitua pelo valor recomendado do Amazon MQ de acordo com o tamanho da instância e o modo de implantação do broker, por exemplo, **`8000000`** para um `mq.m7g.large` cluster. Selecione Number (Número) na lista suspensa.

> **Note**
>
> Não se aplica aos agentes de instância única.

g. Escolha Add/update policy (Adicionar/atualizar política).

10. Confirme se a nova política aparece na lista de User policies (Políticas de usuário).

> **Note**
>
> Para agentes de cluster, o Amazon MQ aplica automaticamente as definições de política `ha-mode: all` e `ha-sync-mode: automatic`.

11. No painel de navegação da direita, escolha Limits (Limites).
12. Na página Limites você poderá ver uma lista dos Virtual host limits (Limites de host virtual) atuais do agente. Abaixo dos Limites de host virtual, expanda Set/update a virtual host limit (Definir/atualizar um limite de host virtual).
13. Para criar um novo limite vhost, em Set/update a virtual host limit (Definir/atualizar um limite de host virtual), faça o seguinte:

    a. Para o Virtual host (Host virtual), escolha o nome do vhost ao qual você deseja anexar as políticas da lista suspensa. Para escolher o vhost padrão, escolha /.

    b. Para Limit (Limite), escolha máximo de conexões nas opções suspensas.

    c. Para Value (Valor), insira o Amazon MQ recommended value (Valor recomendado pelo Amazon MQ) de acordo com o tamanho da instância do agente e o modo de implantação, por exemplo, **`15000`** para um cluster `mq.m5.large`.

    d. Selecione Set/update limit (Definir/atualizar limite).

    e. Repita as etapas acima e, para Limit (Limite), escolha máximo de filas nas opções suspensas.

14. Confirme se os novos limites aparecem na lista de Virtual host limits (Limites de host virtual).

Para aplicar políticas padrão e limites de host virtual usando a API de gerenciamento RabbitMQ

1. Faça login no console do Amazon MQ.
2. No painel de navegação à esquerda, escolha Agentes.
3. Na lista de agentes, escolha o nome do agente ao qual você deseja aplicar a nova política.

4. Na página do agente, na seção Connections (Conexões), anote a URL do RabbitMQ web console (Console da Web RabbitMQ). Este é o endpoint do agente que você usa em uma solicitação HTTP.
5. Abra uma nova janela de terminal ou linha de comando de sua escolha.
6. Para criar uma nova política de agente, insira o comando `curl` a seguir. Este comando assume uma fila no vhost `/` padrão, que é codificada como `%2F`. Para aplicar a política a outro vhost, substitua `%2F` pelo nome do vhost.

   > **Note**
   >
   > Substitua *`username`* e *`password`* por suas credenciais de login de administrador. *`number-of-messages`*Substitua pelo valor recomendado do Amazon MQ de acordo com o tamanho da instância e o modo de implantação do broker. *`policy-name`*Substitua por um nome para sua política. *`broker-endpoint`*Substitua pelo URL que você anotou 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 se a nova política foi adicionada às políticas de usuário do seu agente, insira o seguinte comando `curl` para listar todas as políticas de agente.

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

8. Para criar um novo limites `max-connections` de host virtual, insira o seguinte comando `curl`. Este comando assume uma fila no vhost `/` padrão, que é codificada como `%2F`. Para aplicar a política a outro vhost, substitua `%2F` pelo nome do vhost.

   > **Note**
   >
   > Substitua *`username`* e *`password`* por suas credenciais de login de administrador. *`max-connections`*Substitua pelo valor recomendado do Amazon MQ de acordo com

o tamanho da instância e o modo de implantação do broker. Substitua o endpoint do agente com a URL que você anotou 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 criar um novo limite de host virtual `max-queues`, repita a etapa anterior, mas modifique o comando curl conforme mostrado a seguir.

```
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 se os novos limites foram adicionados aos limites de host virtual do seu agente, insira o comando a seguir `curl` para listar todos os limites de host virtual do agente.

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

Filas de quórum do RabbitMQ no Amazon MQ

As filas de quórum são um tipo de fila replicada composta de um líder (réplica primária) e de seguidores (outras réplicas). Se o líder ficar indisponível, as filas de quórum usarão o algoritmo de consenso Raft para eleger um novo nó líder pela maioria dos votos, e o líder anterior será rebaixado a um nó seguidor no mesmo cluster. Os seguidores restantes continuam se replicando como antes. Como cada nó está em uma zona de disponibilidade diferente, se um nó estiver temporariamente indisponível, a entrega de mensagens continuará com a réplica líder recém-eleita em outra zona de disponibilidade.

As filas de quórum são úteis para lidar com mensagens mal-intencionadas, que ocorrem quando uma mensagem falha e é enfileirada várias vezes.

Filas de quórum não devem ser usadas se você:

- usa filas transitórias;

- tem longas filas de pendências;
- prioriza a baixa latência.

Para declarar uma fila de quórum, defina o cabeçalho `x-queue-type` como `quorum`.

Tópicos

Migrar de filas clássicas para filas de quórum no Amazon MQ para RabbitMQ

Você pode migrar filas clássicas espelhadas para filas de quórum nos agentes do Amazon MQ na versão 3.13 ou posterior criando um host virtual no mesmo cluster ou migrando no local.

Opção 1: migrar de filas clássicas espelhadas para filas de quórum com um novo host virtual

Você pode migrar filas clássicas espelhadas para filas de quórum nos agentes do Amazon MQ na versão 3.13 ou posterior criando um host virtual no mesmo cluster.

1. No cluster existente, crie um host virtual (vhost) com o tipo de fila padrão como quórum.
2. Use filas clássicas espelhadas para criar o Plugin de federação com base no novo vhost, com o URI apontando para o antigo vhost.
3. Usando o `rabbitmqadmin`, exporte as definições do vhost antigo para um novo arquivo. Você deve fazer alterações no arquivo do esquema para que ele seja compatível com as filas de quórum. Para ver a lista completa das alterações que você precisa fazer no arquivo, consulte Moving definitions na documentação de filas de quórum do RabbitMQ. Depois de aplicar as alterações necessárias no arquivo, reimporte as definições para o novo vhost.
4. Crie uma política no novo vhost. Para obter recomendações de configurações de política do Amazon MQ para filas de quórum, consulte Configurações de política de filas de quórum do Amazon MQ para RabbitMQ. Em seguida, inicie a federação que você criou anteriormente do vhost antigo para o novo vhost.
5. Direcione consumidores e produtores para o novo vhost.

6. Configure o plug-in Shovel para mover quaisquer mensagens restantes. Quando a fila estiver vazia, exclua o Shovel.

Migrar de filas clássicas espelhadas para filas de quórum no local

Você pode migrar filas clássicas espelhadas para filas de quórum nos agentes do Amazon MQ na versão 3.13 ou posterior migrando no local.

1. Interrompa os consumidores e produtores.
2. Crie uma fila de quórum temporária.
3. Configure o plug-in Shovel para mover qualquer mensagem da antiga fila clássica espelhada para a nova fila de quórum temporária. Depois que todas as mensagens forem movidas para a fila de quórum temporária, exclua o Shovel.
4. Exclua a fila clássica espelhada original. Em seguida, recrie uma fila de quórum com o mesmo nome e vínculos da fila clássica espelhada original.
5. Crie um Shovel para mover as mensagens da fila de quórum temporária para a nova fila de quórum.

Configurações de política de filas de quórum do Amazon MQ para RabbitMQ

Você pode adicionar configurações de política específicas às filas de quórum para seu agente do RabbitMQ no Amazon MQ.

Ao criar uma política para filas de quórum, você deve fazer o seguinte:

- Remova todos os atributos da política que começam com ha, como `ha-mode`, `ha-params`, `ha-sync-mode`, `ha-sync-batch-size`, `ha-promote-on-shutdown` e `ha-promote-on-failure`.
- Remova `queue-mode`.
- Altere o estouro quando ele estiver definido como `reject-publish-dlx`.

> **Important**
>
> O Amazon MQ para RabbitMQ aplica todos ou nenhum dos atributos em uma política. Não é possível criar uma política que se aplique tanto a filas clássicas espelhadas quanto a filas

de quórum. Se você quiser que sua política se aplique somente a filas de quórum, defina `--apply-to` como `quorum_queues`. Se você estiver usando filas clássicas espelhadas e filas de quórum, deverá criar uma política separada com `--apply-to:classic_queues`, bem como uma política de filas de quórum.

Você não precisa modificar as políticas `AWS-DEFAULT` porque elas adotam automaticamente o novo tipo de fila no parâmetro “aplica-se a”. Para obter mais informações sobre políticas padrão do Amazon MQ para RabbitMQ, consulte Configurar políticas do operador.

Práticas recomendadas para filas de quórum do Amazon MQ para RabbitMQ

Recomendamos usar as práticas recomendadas a seguir para melhorar o desempenho ao trabalhar com filas de quórum.

Definir um limite de entrega para lidar com mensagens mal-intencionadas

Mensagens mal-intencionadas ocorrem quando uma mensagem falha e é reenviada várias vezes. Você pode definir um limite de entrega de mensagens usando o argumento `delivery-limit` da política para descartar mensagens que são reenviadas várias vezes. Se uma mensagem for reenviada mais vezes do que o limite de entrega permitido, ela será descartada e excluída pelo RabbitMQ. Quando você define um limite de entrega, a mensagem é enfileirada novamente perto do início da fila.

Prioridade de mensagens para filas de quórum

As filas de quórum não têm prioridade de mensagens. Se você precisar de prioridade de mensagens, deverá criar várias filas de quórum. Para obter mais informações sobre como priorizar mensagens com várias filas de quórum, consulte Message priority na documentação do RabbitMQ.

Usar o fator de replicação padrão

O Amazon MQ para RabbitMQ usa como padrão um fator de replicação de três nós para agentes de cluster que usam filas de quórum. Se você fizer alterações em `x-quorum-initial-group-size`, o Amazon MQ voltará a usar como padrão o fator de replicação de três.

Práticas recomendadas do Amazon MQ para RabbitMQ

Siga essas diretrizes de preparação para produção para maximizar o desempenho do agente e otimizar a eficiência de throughput de mensagens quando trabalhar com agentes do Amazon MQ para RabbitMQ.

> **Important**
>
> Atualmente, o Amazon MQ não é compatível com transmissões, nem com o uso do registro estruturado em JSON, apresentado no RabbitMQ 3.9.x.

Tópicos

Práticas recomendadas para configuração de agentes e gerenciamento de conexões no Amazon MQ para RabbitMQ

O gerenciamento de conexões e a configuração do agente são a primeira etapa para evitar problemas com a taxa de throughput de mensagens do agente, a utilização de recursos e a capacidade de lidar com workloads de produção. Quando criar e configurar um agente do Amazon MQ para RabbitMQ, siga as práticas recomendadas a seguir para selecionar os tipos de instância apropriados, gerenciar as conexões de forma eficiente e configurar a pré-busca de mensagens para maximizar o desempenho do agente.

> **Important**
>
> O Amazon MQ para RabbitMQ não permite o nome de usuário “convidado” e excluirá a conta de convidado padrão quando você criar um agente. O Amazon MQ também excluirá periodicamente qualquer conta de “convidado” criada pelo cliente.

Etapa 1: usar implantações de cluster

Em workloads de produção, recomendamos o uso de implantações de cluster em vez de agentes de instância única para garantir alta disponibilidade e resiliência de mensagens. As implantações de cluster removem pontos únicos de falha e oferecem melhor tolerância a falhas.

As implantações de cluster consistem em três nós de agente do RabbitMQ distribuídos em três zonas de disponibilidade, fornecendo failover automático e garantindo que as operações continuem mesmo se uma zona de disponibilidade inteira ficar indisponível. O Amazon MQ replica automaticamente as mensagens em todos os nós para garantir a disponibilidade durante falhas ou manutenção dos nós.

As implantações de clusters são essenciais para ambientes de produção e são suportadas pelo Acordo de Nível de Serviço do Amazon MQ.

Para saber mais, consulte Implantação de clusters no Amazon MQ para RabbitMQ.

Etapa 2: Escolher o tipo correto de instância do agente

O throughput de mensagens de um tipo de instância do agente depende do caso de uso da aplicação. O `M7g.medium` deve ser usado somente para testar o desempenho da aplicação. Usar essa instância menor antes de usar instâncias maiores na produção pode melhorar o desempenho da aplicação. Em tipos de instância `m7g.large` e superiores, você pode usar implantações de cluster para obter alta disponibilidade e durabilidade de mensagens. Tipos maiores de instância do agente conseguem lidar com níveis de produção de clientes e filas, alto throughput, mensagens na memória e mensagens redundantes.

Para obter mais informações sobre como escolher o tipo de instância correto, consulte Diretrizes de dimensionamento no Amazon MQ para RabbitMQ.

Etapa 3: Usar filas de quórum

As filas de quórum, com implantação de cluster, devem ser a escolha padrão para tipos de fila replicados em ambientes de produção para os agente do RabbitMQ na versão 3.13 e posterior. As filas de quórum são um tipo de fila replicada moderna que fornece alta confiabilidade, alto throughput e latência estável.

As filas de quórum usam o algoritmo de consenso Raft para fornecer melhor tolerância a falhas. Quando o nó líder fica indisponível, as filas de quórum elegem automaticamente um novo líder por

maioria de votos, garantindo que a entrega de mensagens continue com o mínimo de interrupção. Como cada nó está em uma zona de disponibilidade diferente, o sistema de mensagens permanece disponível mesmo que uma zona de disponibilidade inteira fique temporariamente indisponível.

Para declarar uma fila de quórum, defina o cabeçalho `x-queue-type` como `quorum` quando criar as filas.

Para saber mais sobre filas de quórum, incluindo estratégias de migração e práticas recomendadas, consulte Filas de quórum no Amazon MQ para RabbitMQ.

Etapa 4: Usar vários canais

Para evitar a perda de conexão, use vários canais em uma única conexão. As aplicações devem evitar uma relação de conexão de 1:1 com o canal. Recomendamos usar uma conexão por processo e um canal por fio. Evite o uso excessivo de um canal para impedir vazamentos no canal.

Práticas recomendadas para durabilidade e confiabilidade de mensagens no Amazon MQ para RabbitMQ

Antes de mover o aplicativo para a produção, siga as práticas recomendadas para evitar a perda de mensagens e a sobreutilização de recursos.

Etapa 1: Usar mensagens persistentes e filas duráveis

Mensagens persistentes podem ajudar a proteger a durabilidade em situações em que um agente falha ou reinicia. Mensagens persistentes são gravadas no disco assim que chegam. Ao contrário das filas lazy, no entanto, as mensagens persistentes são armazenadas em cache tanto na memória quanto no disco, a menos que o agente necessite de mais memória. Nos casos em que mais memória é necessária, as mensagens são removidas da memória pelo mecanismo do agente RabbitMQ que gerencia o armazenamento de mensagens no disco, comumente chamado de camada de persistência.

Para habilitar a persistência de mensagens, você pode declarar suas filas como `durable` e definir o modo de entrega de mensagens como `persistent`. O exemplo a seguir demonstra declarar uma fila durável usando a biblioteca do cliente Java RabbitMQ. Ao trabalhar com o AMQP 0-9-1, você pode marcar mensagens como persistentes definindo o modo de entrega como “2”.

```
boolean durable = true;
```

```
channel.queueDeclare("my_queue", durable, false, false, null);
```

Depois de configurar sua fila como durável, você pode enviar uma mensagem persistente para a fila definindo `MessageProperties` como `PERSISTENT_TEXT_PLAIN`, da forma mostrada no exemplo a seguir.

```
import com.rabbitmq.client.MessageProperties;

channel.basicPublish("", "my_queue",
            MessageProperties.PERSISTENT_TEXT_PLAIN,
            message.getBytes());
```

Etapa 2: Configurar a confirmação do publicador e o reconhecimento de entrega do consumidor

O processo de confirmar que uma mensagem foi enviada ao agente é conhecido como confirmação do publicador. As confirmações do publicador avisam a aplicação quando as mensagens foram armazenadas de forma confiável. As confirmações do publicador também podem ajudar a controlar a taxa de mensagens armazenadas no agente. Sem as confirmações do publicador, não há confirmação de que uma mensagem foi processada com sucesso, e o agente talvez envie mensagens que não consiga processar.

De modo similar, quando uma aplicação cliente envia uma confirmação de entrega e consumo de mensagens de volta ao agente, isso é conhecido como confirmação de entrega do consumidor. Os dois tipos de confirmação são essenciais para garantir a segurança dos dados ao trabalhar com agentes do RabbitMQ.

A confirmação de entrega do consumidor geralmente é configurada na aplicação do cliente. Ao trabalhar com o AMQP 0-9-1, a confirmação pode ser habilitada configurando o método `basic.consume`. Os clientes AMQP 0-9-1 também podem configurar as confirmações do publicador enviando o método `confirm.select`.

Normalmente, a confirmação de entrega está habilitada em um canal. Por exemplo, ao trabalhar com a biblioteca do cliente Java RabbitMQ, você pode usar o `Channel#basicAck` para configurar um reconhecimento positivo `basic.ack`, conforme mostrado no exemplo a seguir.

```
// 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**
>
> Mensagens não reconhecidas devem ser armazenadas em cache na memória. Você pode limitar o número de mensagens que um consumidor busca antecipadamente configurando Pré-busca para uma aplicação do cliente.

Você pode configurar o `consumer_timeout` para detectar quando os consumidores não confirmarem as entregas. Se o consumidor não enviar uma confirmação dentro do tempo limite, o canal será fechado e você receberá `PRECONDITION_FAILED`. Para diagnosticar o erro, use a UpdateConfigurationAPI para aumentar o `consumer_timeout` valor.

Etapa 3: Manter as filas curtas

Em implantações de cluster, filas com um grande número de mensagens podem levar à utilização excessiva de recursos. Quando um agente é utilizado em excesso, a reinicialização de um agente do Amazon MQ para RabbitMQ pode causar maior degradação da performance. Se reinicializados, os agentes usados em excesso podem deixar de responder no estado `REBOOT_IN_PROGRESS`.

Durante as janelas de manutenção, o Amazon MQ executa todos os trabalhos de manutenção um nó de cada vez para garantir que o agente permaneça operacional. Como resultado, as filas podem precisar sincronizar à medida que cada nó retoma a operação. Durante a sincronização, as mensagens que precisam ser replicadas em espelhos são carregadas na memória do volume correspondente do Amazon Elastic Block Store (Amazon EBS) para serem processadas em lotes. O processamento de mensagens em lotes permite que as filas sejam sincronizadas mais rapidamente.

Se as filas forem mantidas curtas e as mensagens forem pequenas, as filas serão sincronizadas com êxito e retomarão a operação conforme esperado. No entanto, se a quantidade de dados em um lote se aproximar do limite de memória do nó, o nó gera um alarme de memória alta, pausando a sincronização de fila. Você pode confirmar o uso da memória comparando as métricas do nó `RabbitMemUsed` e do `RabbitMqMemLimit` broker em CloudWatch. A sincronização não pode ser concluída até que as mensagens sejam consumidas ou excluídas ou o número de mensagens no lote seja reduzido.

Se a sincronização de filas estiver pausada para uma implantação de cluster, recomendamos consumir ou excluir mensagens para diminuir o número de mensagens em filas. Quando a profundidade da fila for reduzida e a sincronização da fila for concluída, o status do agente mudará para `RUNNING`. Para resolver uma sincronização de fila pausada, você também pode aplicar uma política para reduzir o tamanho do lote de sincronização de filas.

Você também pode definir políticas de exclusão automática e TTL para reduzir proativamente o uso de recursos, bem como reduzir ao mínimo o NACKs alcance dos consumidores. O enfileiramento de mensagens na corretora consome muita CPU, portanto, um grande número de mensagens pode afetar o desempenho da corretora. NACKs

Práticas recomendadas para otimização e eficiência de desempenho no Amazon MQ para RabbitMQ

Você pode otimizar o desempenho do agente do Amazon MQ para RabbitMQ maximizando o throughput, minimizando a latência e garantindo a utilização eficiente dos recursos. Conclua as práticas recomendadas a seguir para otimizar o desempenho da aplicação.

Etapa 1: Manter o tamanho das mensagens abaixo de 1 MB

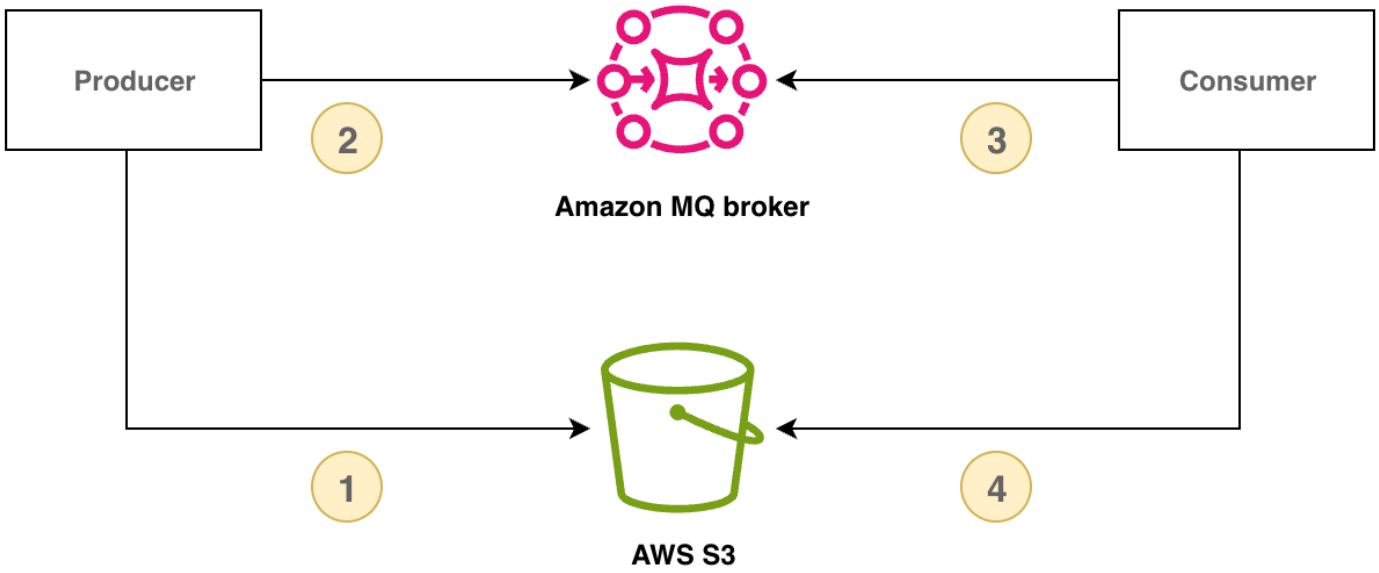
Recomendamos manter as mensagens com menos de 1 megabyte (MB) para desempenho e confiabilidade ideais.

O RabbitMQ 3.13 é compatível com tamanhos de mensagem de até 128 MB por padrão. Porém, mensagens grandes podem acionar alarmes de memória imprevisíveis que bloqueiam a publicação e potencialmente criam pressão de alta memória quando se replica mensagens entre os nós. As mensagens superdimensionadas também podem afetar os processos de reinicialização e recuperação do agente, o que aumenta os riscos à continuidade do serviço e pode causar degradação do desempenho.

Armazenar e recuperar cargas úteis grandes usando o padrão de verificação de reivindicações

Para gerenciar mensagens grandes, você pode implementar o padrão de verificação de reinvindicação armazenando a carga útil da mensagem no armazenamento externo e enviando somente o identificador de referência da carga útil por meio do RabbitMQ. O consumidor usa o identificador de referência da carga útil para recuperar e processar a mensagem grande.

O diagrama a seguir demonstra como usar o Amazon MQ para RabbitMQ e o Amazon S3 para implementar o padrão de verificação de reivindicações.



O exemplo a seguir demonstra esse padrão usando-se o Amazon MQ, o AWS SDK para Java 2.x e o Amazon S3:

1. Primeiro, defina uma classe de mensagem que conterá o identificador de referência do Amazon S3.

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

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

2. Crie um método de publicador que armazene a carga útil no Amazon S3 e envie uma mensagem de referência por meio do 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 um método de consumidor que recupere a carga útil do Amazon S3, processe a carga útil e exclua o objeto do 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());
}
```

Etapa 2: Usar **`basic.consume`** e consumidores duradouros

Usar `basic.consume` com um consumidor de longa data é mais eficiente do que pesquisar o uso de mensagens individuais do `basic.get`. Para saber mais, consulte Pesquisa de mensagens individuais.

Etapa 3: Configurar pré-busca

Você pode usar o valor de pré-busca RabbitMQ para otimizar como seus consumidores consomem mensagens. O RabbitMQ implementa o mecanismo de pré-busca do canal fornecido pelo AMQP 0-9-1 aplicando a contagem de pré-busca aos consumidores em oposição aos canais. O valor de pré-busca é usado para especificar quantas mensagens estão sendo enviadas ao consumidor em um determinado momento. Por padrão, o RabbitMQ define um tamanho ilimitado de buffer para aplicações do cliente.

Há muitos fatores a serem considerados ao definir uma contagem de pré-busca para seus consumidores RabbitMQ. Primeiro, considere o ambiente e a configuração dos seus consumidores. Como os consumidores precisam manter todas as mensagens na memória enquanto estão sendo processadas, um alto valor de pré-busca pode ter um impacto negativo na performance de seus consumidores e, em alguns casos, pode resultar em um consumidor potencialmente travando tudo. Da mesma forma, o próprio agente RabbitMQ mantém todas as mensagens que envia armazenadas em cache na memória até receber reconhecimento do consumidor. Um valor de pré-busca alto pode fazer com que o servidor RabbitMQ fique sem memória rapidamente se a confirmação automática não estiver configurada para os consumidores e se os consumidores demorarem um tempo relativamente longo para processar mensagens.

Com as considerações acima em mente, recomendamos sempre definir um valor de pré-busca para evitar situações em que um agente RabbitMQ ou seus consumidores ficam sem memória devido a um grande número de mensagens não processadas ou não confirmadas. Se você precisar otimizar seus agentes para processar grandes volumes de mensagens, você pode testar seus agentes e consumidores usando uma gama de contagens de pré-busca para determinar o valor em que ponto a sobrecarga de rede se torna em grande parte insignificante em comparação com o tempo que um consumidor leva para processar mensagens.

> **Note**
>
> - Se as aplicações do seu cliente tiverem configurado para confirmar automaticamente a entrega de mensagens aos consumidores, a definição de um valor de pré-busca não terá efeito.
> - Todas as mensagens pré-buscadas são removidas da fila.

O exemplo a seguir demonstra a configuração de um valor de pré-busca de `10` para um único consumidor usando a biblioteca do cliente Java 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**
>
> Na biblioteca do cliente Java RabbitMQ, o valor padrão para `global` está definido como `false`, de modo que o exemplo acima pode ser escrito simplesmente como `channel.basicQos(10)`.

Etapa 4: Usar o Celery 5.5 ou posterior com filas de quórum

O Python Celery, um sistema distribuído de filas de tarefas, pode gerar muitas mensagens não críticas ao enfrentar uma alta carga de tarefas. Essa atividade adicional do agente pode desencadear the section called “ RABBITMQ_MEMORY_ALARM ” e levar à indisponibilidade do agente. Para reduzir a chance de acionar o alarme de memória, faça o seguinte:

Para todas as versões do Celery

1. Desative `task_create_missing_queues` para mitigar a rotatividade da fila.
2. Em seguida, desative `worker_enable_remote_control` para interromper a criação dinâmica de filas do `celery@...pidbox`. Isso reduzirá a rotatividade de filas no agente.

   ```
   worker_enable_remote_control = false
   ```

3. Para reduzir ainda mais a atividade de mensagens não críticas, desative o Celery worker-send-task-eventssem incluir `-E` ou `--task-events` sinalizar ao iniciar seu aplicativo Celery.
4. Inicie a aplicação Celery usando os seguintes parâmetros:

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

Para as versões 5.5 e posteriores do Celery

1. Atualize para a versão 5.5 do Celery, a versão mínima que oferece suporte a filas de quórum, ou uma versão posterior. Para verificar qual versão do Celery você está usando, use `celery --version`. Para obter mais informações sobre quóruns, consulte the section called “Filas de quórum”.
2. Depois de atualizar para o Celery 5.5 ou posterior, configure `task_default_queue_type` como “quórum”.
3. Em seguida, ative também a opção Publicar confirmações nas Opções de transporte de agente:

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

Práticas recomendadas para resiliência e monitoramento de rede no Amazon MQ para RabbitMQ

A resiliência da rede e as métricas do agente de monitoramento são essenciais para manter as aplicações de mensagens confiáveis. Conclua as práticas recomendadas a seguir para implementar mecanismos de recuperação automática e estratégias de monitoramento de recursos.

Etapa 1: Recuperação automática de falhas de rede

Recomendamos sempre habilitar a recuperação automática de rede para evitar tempo de inatividade significativo nos casos em que as conexões do cliente com os nós RabbitMQ falham. A biblioteca do cliente Java RabbitMQ é compatível com a recuperação automática de rede por padrão, começando com a versão `4.0.0`.

A recuperação automática da conexão é acionada se uma exceção não tratada for lançada no I/O loop da conexão, se o tempo limite da operação de leitura do soquete for detectado ou se o servidor perder uma pulsação.

Nos casos em que a conexão inicial entre um cliente e um nó RabbitMQ falha, a recuperação automática não será acionada. Recomendamos escrever o código da aplicação para levar em conta as falhas de conexão iniciais tentando a conexão novamente. O exemplo a seguir demonstra a repetição de falhas iniciais de rede usando a biblioteca de cliente Java 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**
>
> Se uma aplicação fecha uma conexão usando o método `Connection.Close`, a recuperação automática de rede não será ativada ou acionada.

Etapa 2: Monitorar métricas e alarmes do agente

Recomendamos monitorar regularmente CloudWatch métricas e alarmes de seu agente Amazon MQ for RabbitMQ para identificar e resolver possíveis problemas antes que eles afetem seu aplicativo de mensagens. O monitoramento proativo é essencial para manter uma aplicação de mensagens resiliente e garantir o desempenho ideal.

O Amazon MQ for RabbitMQ publica métricas CloudWatch que fornecem informações sobre o desempenho do agente, a utilização de recursos e o fluxo de mensagens. As principais métricas a serem monitoradas incluem o uso da memória e do disco. Você pode configurar CloudWatch alarmes para quando seu corretor se aproximar dos limites de recursos ou sofrer uma degradação do desempenho.

Monitore as seguintes métricas essenciais:

`RabbitMQMemUsed` e **`RabbitMQMemLimit`**

Monitore o uso da memória para evitar alarmes de memória que possam bloquear a publicação de mensagens.

`RabbitMQDiskFree` e **`RabbitMQDiskFreeLimit`**

Monitore o uso do disco para evitar problemas de espaço em disco que podem causar falhas no agente.

Em implantações de cluster, monitore também as métricas específicas do nó para identificar problemas específicos do nó.

> **Note**
>
> Para saber mais sobre como evitar o alarme de alta memória, consulte Abordar e evitar o alarme de alta memória.

Tutoriais do RabbitMQ

Os tutoriais a seguir mostram como é possível configurar e usar o RabbitMQ no Amazon MQ. Para saber mais sobre como trabalhar com bibliotecas de clientes compatíveis com várias linguagens de programação como Node.js, Python, .NET e muito mais, consulte Tutoriais do RabbitMQ no Guia de conceitos básicos do RabbitMQ.

Tópicos

- Editar as preferências de agente
- Como usar Python Pika com o Amazon MQ para RabbitMQ
- Resolvendo a sincronização de fila pausada do RabbitMQ
- Reduzir o número de conexões e canais
- Etapa 2: conectar uma aplicação baseada em JVM ao seu agente
- Etapa 3: (opcional) conectar-se a uma AWS Lambda função
- Uso da autorização e da autenticação OAuth 2.0 para Amazon MQ para RabbitMQ
- Usando autenticação e autorização do IAM para Amazon MQ para RabbitMQ
- Usando autenticação e autorização LDAP para Amazon MQ para RabbitMQ
- Usando autenticação e autorização HTTP para Amazon MQ para RabbitMQ
- Usando a autenticação de certificado SSL para Amazon MQ para RabbitMQ
- Usando mTLS para AMQP e endpoints de gerenciamento
- Conectando seu aplicativo JMS

Editar as preferências de agente

Você pode editar suas preferências de corretor, como ativar ou desativar CloudWatch registros usando o. Console de gerenciamento da AWS

Editar opções do agente RabbitMQ

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, selecione seu corretor (por exemplo MyBroker) e escolha Editar.
3. Na *MyBroker* página Editar, na seção Especificações, selecione uma versão do mecanismo Broker ou um tipo de Instância do Broker.
4. Na seção CloudWatch Registros, clique no botão de alternância para ativar ou desativar os registros gerais. Nenhuma outra etapa é necessária.

   > **Note**
   >
   > - Para os corretores do RabbitMQ, o Amazon MQ usa automaticamente uma função vinculada ao serviço (SLR) para publicar registros gerais. CloudWatch Para obter mais informações, consulte the section called “Uso de perfis vinculados ao serviço”.
   > - O Amazon MQ não é compatível com registros de auditoria para agentes RabbitMQ.

5. Na seção Maintenance (Manutenção), configure a programação de manutenção do agente:

   Para atualizar o broker para novas versões à medida que as AWS libera, escolha Habilitar atualizações automáticas de versões secundárias. Atualizações automáticas ocorrem durante a janela de manutenção definida pelo dia da semana, a hora do dia (no formato de 24 horas) e o fuso horário (UTC, por padrão).
6. Selecione Schedule modifications (Programar modificações).

   > **Note**
   >
   > Se você selecionar somente Enable automatic minor version upgrades (Habilitar atualizações automáticas de versão secundária), o botão será alterado para Save (Salvar), pois não será necessária nenhuma reinicialização do agente.

   Suas preferências serão aplicadas ao agente no horário especificado.

Como usar Python Pika com o Amazon MQ para RabbitMQ

O tutorial a seguir mostra como você pode configurar um cliente Python Pika com o TLS configurado para estabelecer conexão com o agente Amazon MQ para RabbitMQ. O Pika é uma implementação Python do protocolo AMQP 0-9-1 para RabbitMQ. Este tutorial orienta você na instalação do Pika, na declaração de uma fila, na configuração de um editor para enviar mensagens para a bolsa padrão da corretora e na configuração de um consumidor para receber mensagens da fila.

Tópicos

Pré-requisitos

Para concluir as etapas neste tutorial, você precisa dos seguintes pré-requisitos:

- Um agente Amazon MQ para RabbitMQ. Para mais informações, consulte Criar um agente Amazon MQ para RabbitMQ.
- O Python 3 instalado para o seu sistema operacional.
- O Pika instalado usando o Python `pip`. Para instalar o Pika, abra uma nova janela de terminal e execute o seguinte.

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

Permissões

Para este tutorial, você precisa de pelo menos um usuário do agente Amazon MQ para RabbitMQ com permissão para gravação em e leitura de um vhost. A tabela a seguir descreve as permissões mínimas necessárias como padrões de expressão regular (regexp).

Tags	Configurar regexp	Escrever regexp	Ler regexp
none		`.*`	`.*`

As permissões de usuário listadas fornecem apenas permissões de leitura e gravação para o usuário, sem conceder acesso ao plugin de gerenciamento para executar operações administrativas no agente. Você pode restringir ainda mais as permissões fornecendo padrões regexp que limitem o acesso do usuário às filas especificadas. Por exemplo, se você alterar o padrão regexp de leitura para `^[hello world].*`, o usuário só terá permissão de leitura para as filas que começam com `hello world`.

Para obter mais informações sobre criar usuários RabbitMQ e gerenciar etiquetas e permissões de usuário, consulte Usuários do agente do Amazon MQ para RabbitMQ.

Etapa um: criar um cliente Python Pika básico

Para criar uma classe base de cliente Python Pika que define um construtor e fornece o contexto SSL necessário para a configuração TLS durante a interação com um agente Amazon MQ para RabbitMQ, faça o seguinte.

1. Abra uma nova janela de terminal, crie um novo diretório para seu projeto e acesse o diretório.

```
$ mkdir pika-tutorial
$ cd pika-tutorial
```

2. Crie um novo arquivo chamado `basicClient.py` contendo o seguinte 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()
```

Agora você pode definir classes adicionais para seu publicador e consumidor que herdam de `BasicPikaClient`.

Etapa dois: criar um publicador e enviar uma mensagem

Para criar um publicador que declara uma fila e envia uma única mensagem, faça o seguinte.

1. Copie o conteúdo da amostra de código a seguir e salve localmente como `publisher.py` no mesmo diretório que você criou na etapa 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()
```

A classe `BasicMessageSender` herda de `BasicPikaClient` e implementa métodos adicionais para declarar uma fila, enviar uma mensagem para a fila e fechar conexões. A amostra de código encaminha uma mensagem para a troca padrão, com uma chave de roteamento igual ao nome da fila.

2. Em `if __name__ == "__main__":`, substitua os parâmetros transmitidos para a declaração do construtor `BasicMessageSender` com as seguintes informações.

   - **<broker-id>** — O ID exclusivo que o Amazon MQ gera para o agente. Você pode analisar o ID do ARN do seu agente. Por exemplo, considerando o seguinte ARN, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, o ID do agente seria `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - **<username>**: o nome de usuário de um usuário agente com permissões suficientes para gravação de mensagens no agente.
   - **<password>**: a senha de um usuário agente com permissões suficientes para gravação de mensagens no agente.
   - **<region>**— A AWS região na qual você criou seu corretor Amazon MQ para RabbitMQ. Por exemplo, `.us-west-2`

3. Execute o seguinte comando no mesmo diretório que você criou `publisher.py`.

```
$ python3 publisher.py
```

Se o código for executado com êxito, você verá o resultado a seguir na janela do seu terminal.

```
Trying to declare queue(hello world queue)...
Sent message. Exchange: , Routing Key: hello world queue, Body: b'Hello World!'
```

Etapa três: criar um consumidor e receber uma mensagem

Para criar um consumidor que receba uma única mensagem da fila, faça o seguinte.

1. Copie o conteúdo da amostra de código a seguir e salve localmente como `consumer.py` no mesmo diretório.

```
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()
```

Semelhante ao editor que você criou na etapa anterior, `BasicMessageReceiver` herda `BasicPikaClient` e implementa métodos adicionais para receber uma única mensagem e fechar conexões.

2. Na declaração `if __name__ == "__main__":`, substitua os parâmetros transmitidos ao construtor `BasicMessageReceiver` com suas informações.
3. Execute o seguinte comando no diretório do projeto.

```
$ python3 consumer.py
```

Se o código for executado com êxito, você verá o corpo da mensagem e os cabeçalhos, incluindo a chave de roteamento, exibidos na janela do seu terminal.

```
<Basic.GetOk(['delivery_tag=1', 'exchange=', 'message_count=0',
 'redelivered=False', 'routing_key=hello world queue'])> <BasicProperties> b'Hello
 World!'
```

Etapa quatro: (opcional) configurar um loop de eventos e consumir mensagens

Para consumir várias mensagens de uma fila, use o método `basic_consume` do Pika e uma função de retorno de chamada conforme mostrado a seguir

1. Em `consumer.py`, adicione a definição de método a seguir à classe `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. Em `consumer.py`, sob `if __name__ == "__main__":`, invoque o método `consume_messages` definido por você na etapa 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. Execute `consumer.py` novamente e, se bem-sucedidas, as mensagens na fila serão exibidas na janela do seu terminal.

```
[*] Waiting for messages. To exit press CTRL+C
[x] Received b'Hello World!'
[x] Received b'Hello World!'
...
```

Próximas etapas

- Para mais informações sobre outras bibliotecas suportadas de cliente do RabbitMQ, consulte Documentação do cliente RabbitMQ no site do RabbitMQ.

Resolvendo a sincronização de fila pausada do RabbitMQ

Em uma implantação de cluster do Amazon MQ para RabbitMQ, as mensagens publicadas em cada fila são replicadas em três nós de agente. Esta replicação, chamada de espelhamento, fornece alta disponibilidade (HA) para agentes RabbitMQ. As filas em uma implantação de cluster consistem em uma réplica principal em um nó e um ou mais espelhos. Cada operação aplicada a uma fila espelhada, incluindo o enfileiramento de mensagens, é aplicada primeiro à fila principal e, em seguida, replicada em seus espelhos.

Por exemplo, considere uma fila espelhada replicada em três nós: o nó principal (`main`) e dois espelhos (`mirror-1` e `mirror-2`). Se todas as mensagens nessa fila espelhada forem propagadas com êxito para todos os espelhos, a fila será sincronizada. Se um nó (`mirror-1`) se tornar indisponível por um intervalo de tempo, a fila ainda estará operacional e poderá continuar a enfileirar mensagens. No entanto, para sincronizar a fila, as mensagens publicadas no `main` enquanto `mirror-1` estiver indisponível devem ser replicadas para `mirror-1`.

Para obter mais informações sobre o espelhamento, consulte Filas Espelhadas Clássicas no site RabbitMQ.

Manutenção e sincronização de filas

Durante as janelas de manutenção, o Amazon MQ executa todos os trabalhos de manutenção um nó de cada vez para garantir que o agente permaneça operacional. Como resultado, as filas podem precisar sincronizar à medida que cada nó retoma a operação. Durante a sincronização, as mensagens que precisam ser replicadas em espelhos são carregadas na memória do volume correspondente do Amazon Elastic Block Store (Amazon EBS) para serem processadas em lotes. O processamento de mensagens em lotes permite que as filas sejam sincronizadas mais rapidamente.

Se as filas forem mantidas curtas e as mensagens forem pequenas, as filas serão sincronizadas com êxito e retomarão a operação conforme esperado. No entanto, se a quantidade de dados em um lote se aproximar do limite de memória do nó, o nó gera um alarme de memória alta, pausando a sincronização de fila. Você pode confirmar o uso da memória comparando as métricas do nó `RabbitMemUsed` e do `RabbitMqMemLimit` broker em CloudWatch. A sincronização não pode ser concluída até que as mensagens sejam consumidas ou excluídas ou o número de mensagens no lote seja reduzido.

> **Note**
>
> Reduzir o tamanho do lote de sincronização de fila pode resultar em um número maior de transações de replicação.

Para resolver uma sincronização de fila pausada, siga as etapas deste tutorial, que demonstra a aplicação de uma política `ha-sync-batch-size` e reiniciar a sincronização de filas.

Tópicos

Pré-requisitos

Para este tutorial, você deve ter um usuário do agente Amazon MQ para RabbitMQ com permissões de administrador. Você pode usar o usuário administrador criado quando criou o agente pela primeira vez ou outro usuário que você possa ter criado posteriormente. A tabela a seguir fornece a etiqueta de usuário administrador necessária e as permissões como padrões de expressão regular (regexp).

Etiquetas	Ler regexp	Configurar regexp	Escrever regexp
`administrator`	`.*`	`.*`	`.*`

Para obter mais informações sobre criar usuários RabbitMQ e gerenciar etiquetas e permissões de usuário, consulte Usuários do agente do Amazon MQ para RabbitMQ.

Etapa 1: Aplicar uma política **`ha-sync-batch-size`**

Os procedimentos a seguir demonstram a adição de uma política que se aplica a todas as filas criadas no agente. Você pode usar o console da Web do RabbitMQ ou a API de gerenciamento do RabbitMQ. Para obter mais informações, consulte Plugin Gerenciamento no site do RabbitMQ.

Para aplicar uma política **`ha-sync-batch-size`** usando o console da Web RabbitMQ

1. Faça login no console do Amazon MQ.
2. No painel de navegação à esquerda, escolha Agentes.
3. Na lista de agentes, escolha o nome do agente ao qual você deseja aplicar a nova política.
4. Na página do agente, na seção Conexões, escolha a URL RabbitMQ web console (Console da Web RabbitMQ). O console da Web do RabbitMQ é aberto em uma nova guia ou janela do navegador.
5. Faça login no console da Web do RabbitMQ com as credenciais de login de administrador do agente.
6. No console da Web do RabbitMQ, na parte superior da página, escolha Admin.
7. Na página Admin, no painel de navegação da direita, selecione Políticas.
8. Na página Políticas, você pode ver uma lista das Políticas de usuário atuais do agente. Abaixo das User policies (Políticas de usuário), expanda Add/update a policy (Adicionar/atualizar uma política).

   > **Note**
   >
   > Por padrão, os clusters Amazon MQ para RabbitMQ são criados com uma política de agente inicial chamada `ha-all-AWS-OWNED-DO-NOT-DELETE`. O Amazon MQ gerencia essa política para garantir que cada fila no agente seja replicada para todos os três nós e que as filas sejam sincronizadas automaticamente.

9. Para criar uma política de agente, em Add/update a policy (Adicionar/atualizar uma política), faça o seguinte:

   a. Em Nome, insira um nome para a sua política, por exemplo **`batch-size-policy`**.

   b. Para Pattern (Padrão), insira o padrão de expressão regular **`.*`** para que a política corresponda a todas as filas no agente.

   c. Para Apply to (Aplicar em), escolha Exchanges and queues (Trocas e filas) na lista suspensa.

   d. Para Priority (Prioridade), insira um número inteiro maior que todas as outras políticas aplicadas ao vhost. Você pode aplicar exatamente um conjunto de definições de política a filas e trocas RabbitMQ a qualquer momento. O RabbitMQ escolhe a política correspondente com o valor de prioridade mais alto. Para obter mais informações

sobre prioridades de política e como combinar políticas, consulte Policies (Políticas) na Documentação do Servidor RabbitMQ.

e. Para Definition (Definição), adicione os seguintes pares de chave-valor:

- **`ha-sync-batch-size`**=*`100`*. Escolha Número na lista suspensa.

> **Note**
>
> Talvez seja necessário ajustar e calibrar o valor de `ha-sync-batch-size` com base no número e tamanho das mensagens não sincronizadas nas filas.

- **`ha-mode=all`**. Selecione String (String) na lista suspensa.

> **Important**
>
> A definição `ha-mode` é necessária para todas as políticas relacionadas a HA. Omitir isso resulta em uma falha de validação.

- **`ha-sync-mode=automatic`**. Selecione String (String) na lista suspensa.

> **Note**
>
> A definição `ha-sync-mode` é necessária para todas as políticas personalizadas. Se isso for omitido, o Amazon MQ anexará automaticamente a definição.

f. Escolha Add/update policy (Adicionar/atualizar política).

10. Confirme se a nova política aparece na lista de User policies (Políticas de usuário).

Para aplicar uma política **`ha-sync-batch-size`** usando a API de gerenciamento RabbitMQ

1. Faça login no console do Amazon MQ.
2. No painel de navegação à esquerda, escolha Agentes.
3. Na lista de agentes, escolha o nome do agente ao qual você deseja aplicar a nova política.
4. Na página do agente, na seção Connections (Conexões), anote a URL do RabbitMQ web console (Console da Web RabbitMQ). Este é o endpoint do agente que você usa em uma solicitação HTTP.
5. Abra uma nova janela de terminal ou linha de comando de sua escolha.

6. Para criar uma nova política de agente, insira o comando `curl` a seguir. Este comando assume uma fila no vhost `/` padrão, que é codificada como `%2F`.

> **Note**
>
> Substitua *`username`* e *`password`* por suas credenciais de login de administrador do corretor. Talvez seja necessário ajustar e calibrar o valor de `ha-sync-batch-size` (*`100`*) com base no número e no tamanho das mensagens não sincronizadas em suas filas. Substitua o endpoint do agente com a URL que você anotou 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/%2F/batch-size-policy
```

7. Para confirmar se a nova política foi adicionada às políticas de usuário do seu agente, insira o seguinte comando `curl` para listar todas as políticas de agente.

```
curl -i -u username:password https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-
west-2.amazonaws.com/api/policies
```

Etapa 2: Reiniciar a sincronização de filas

Depois de aplicar uma nova política `ha-sync-batch-size` para seu agente, reinicie a sincronização de fila.

Para reiniciar a sincronização de filas usando o console da Web RabbitMQ

> **Note**
>
> Para abrir o console da Web RabbitMQ, consulte as instruções anteriores na Etapa 1 deste tutorial.

1. No console da Web do RabbitMQ, na parte superior da página, escolha Queues (Filas).

2. Na página Filas, em Todas as filas, localize sua fila pausada. Na coluna Política, sua fila deve listar o nome da nova política que você criou (por exemplo, `batch-size-policy`).
3. Para reiniciar o processo de sincronização com um tamanho de lote reduzido, primeiro cancele a sincronização da fila. Em seguida, reinicie a sincronização da fila.

> **Note**
>
> Se a sincronização for interrompida e não for concluída com êxito, tente reduzir o valor `ha-sync-batch-size` e reiniciar a sincronização de fila novamente.

Próximas etapas

- Depois que sua fila for sincronizada com sucesso, você poderá monitorar a quantidade de memória que seus nós do RabbitMQ usam visualizando a métrica da Amazon. CloudWatch `RabbitMQMemUsed` Você também pode visualizar a métrica `RabbitMQMemLimit` para monitorar o limite de memória de um nó. Para obter mais informações, consulte Acessando CloudWatch métricas para o Amazon MQ e CloudWatch Métricas disponíveis para Amazon MQ para corretores RabbitMQ.
- Para evitar pausar a sincronização de filas, recomendamos manter as filas curtas e processar as mensagens. Para workloads com tamanhos de mensagem maiores, também recomendamos atualizar o tipo de instância do agente para um tamanho de instância maior com mais memória. Para obter mais informações sobre os tipos de instância do agente e como editar as preferências do agente consulte Editar as preferências de agente.
- Quando você cria um novo Amazon MQ para o agente RabbitMQ, o Amazon MQ aplica um conjunto de políticas padrão e limites de host virtual para otimizar a performance do agente. Se o seu agente não tiver as políticas e limites padrão recomendados, recomendamos criá-las você mesmo. Para obter mais informações sobre como criar políticas padrão e limites vhost, consulte https://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html.

Recursos relacionados

- UpdateBrokerInput— Use essa propriedade de agente para atualizar um tipo de instância de agente usando a API do Amazon MQ.

- Parâmetros e Políticas (Documentação do Servidor RabbitMQ): Saiba mais sobre os parâmetros e políticas do RabbitMQ no site do RabbitMQ.
- API HTTP de gerenciamento do RabbitMQ — Saiba mais sobre a API de gerenciamento do RabbitMQ.

Reduzir o número de conexões e canais

As conexões com o agente do RabbitMQ no Amazon MQ podem ser encerradas por suas aplicações cliente ou manualmente com o uso do console da Web do RabbitMQ. Para encerrar uma conexão usando o console da Web do RabbitMQ, faça o seguinte.

1. Faça login Console de gerenciamento da AWS e abra o console web RabbitMQ do seu corretor.
2. No console do RabbitMQ, escolha a guia Connections (Conexões).
3. Na página Connections (Conexões), em All connections (Todas as conexões), escolha na lista o nome da conexão que você deseja encerrar.
4. Na página de detalhes da conexão, escolha Close this connection (Encerrar esta conexão) para expandir a seção e depois escolha Force Close (Forçar encerramento). Como opção, você pode substituir o texto padrão do campo Reason (Motivo) pela sua própria descrição. O RabbitMQ no Amazon MQ retornará o motivo especificado para o cliente quando você encerrar a conexão.
5. Escolha OK na caixa de diálogo para confirmar e encerrar a conexão.

Quando você encerrar uma conexão, todos os canais associados à conexão encerrada também serão encerrados.

> **Note**
>
> Suas aplicações cliente podem ser configuradas para restabelecer automaticamente as conexões com o agentes depois que estas são encerradas. Nesse caso, encerrar conexões pelo console da Web do agente não será suficiente para reduzir a contagem de conexões ou canais.

Para agentes sem acesso público, você pode bloquear as conexões temporariamente, negando o tráfego de entrada na porta de protocolo de mensagem apropriada, por exemplo, a porta `5671` para conexões AMQP. É possível bloquear a porta no grupo de segurança que você forneceu ao Amazon MQ ao criar o agente. Para obter mais informações sobre como modificar seu grupo de segurança,

consulte o tópico sobre como Adicionar regras a um grupo de segurança, no Guia do usuário da Amazon VPC.

Etapa 2: conectar uma aplicação baseada em JVM ao seu agente

Depois de criar um agente do RabbitMQ, você pode conectar sua aplicação a ele. Os exemplos a seguir mostram como usar a Biblioteca de cliente Java para criar uma conexão com seu agente, criar uma fila e enviar uma mensagem. Você pode se conectar a agentes RabbitMQ usando bibliotecas de cliente RabbitMQ compatíveis para vários idiomas. Para obter mais informações sobre bibliotecas de cliente do RabbitMQ com suporte, consulte Bibliotecas de cliente e ferramentas de desenvolvedor do RabbitMQ.

Pré-requisitos

> **Note**
>
> As etapas de pré-requisito a seguir são aplicáveis somente a agentes RabbitMQ criados sem acessibilidade pública. Se você estiver criando um agente com acessibilidade pública, pode ignorar essas etapas.

Habilitar atributos da VPC

Para garantir que seu agente esteja acessível dentro da sua VPC, você deve habilitar os atributos VPC `enableDnsHostnames` e `enableDnsSupport`. Para obter mais informações, consulte Compatibilidade com DNS para a sua VPC no Manual do usuário da Amazon VPC.

Habilitar conexões de entrada

1. Faça login no console do Amazon MQ.
2. Na lista de corretores, escolha o nome do seu corretor (por exemplo, MyBroker).
3. Na ***`MyBroker`***página, na seção Conexões, observe os endereços e portas do URL do console web e dos protocolos de nível de fio do broker.
4. Na seção Details (Detalhes), em Security and network (Segurança e rede), escolha o nome do seu grupo de segurança ou

   .

   A página Grupos de segurança do painel do EC2 é exibida.

5. Na lista de security group, escolha seu security group.
6. Na parte inferior da página, escolha Inbound (Entrada) e a seguir selecione Edit (Editar).
7. Na caixa de diálogo Edit inbound rules (Editar regras de entrada), adicione uma regra para cada URL ou endpoint que você deseja que seja acessível publicamente (o exemplo a seguir mostra como fazer isso para um console da Web do agente).

   a. Escolha Add Rule (Adicionar regra).
   b. Em Type (Tipo), selecione Custom TCP (TCP personalizado).
   c. Para Source (Origem), deixe Custom (Personalizado) selecionado e, depois, digite o endereço IP do sistema ao qual deseja ser capaz de acessar o console da Web (por exemplo, `192.0.2.1`).
   d. Escolha Salvar.

   Agora seu agente pode aceitar conexões de entrada.

Adicionar dependências de Java

Se você estiver usando o Apache Maven para automatizar compilações, adicione a seguinte dependência a seu arquivo `pom.xml`. Para obter mais informações sobre arquivos do Project Object Model no Apache Maven, consulte Introdução ao POM.

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

Se você estiver usando o Gradle para automatizar compilações, declare a seguinte dependência.

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

Importar **`Connection`** e classes **`Channel`**

O cliente Java do RabbitMQ usa `com.rabbitmq.client` como seu pacote de nível superior, com as classes da API `Connection` e `Channel` representando uma conexão AMQP 0-9-1 e um canal,

respectivamente. Importe as classes `Connection` e `Channel` antes de usá-las, conforme mostrado no exemplo a seguir.

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

Crie um **`ConnectionFactory`** e conecte ao seu agente

Use o exemplo a seguir para criar uma instância da classe `ConnectionFactory` com os parâmetros fornecidos. Use o método `setHost` para configurar o endpoint do agente que você anotou anteriormente. Para conexões AMQPS de nível de conexão, use a porta `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 uma mensagem em uma troca

Você pode usar o `Channel.basicPublish` para publicar mensagens em uma troca. O exemplo a seguir usa a classe AMQP `Builder` para construir um objeto de propriedades de mensagem com tipo de conteúdo `plain/text`.

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

```
                messageBodyBytes);
```

> **Note**
>
> Observe que `BasicProperties` é uma classe interna da classe titular gerada automaticamente, `AMQP`.

Inscrever-se em uma fila e receber uma mensagem

Você pode receber uma mensagem inscrevendo-se em uma fila usando a Interface `Consumer`. Depois de inscrito, as mensagens serão entregues automaticamente à medida que chegarem.

A maneira mais fácil de implementar um `Consumer` é usar a subclasse `DefaultConsumer`. Um objeto `DefaultConsumer` pode ser transmitido como parte de uma chamada `basicConsume` para configurar a assinatura, conforme mostrado no exemplo a seguir.

```
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**
>
> Como nós especificamos `autoAck = false`, é necessário reconhecer as mensagens entregues ao `Consumer`, o que é feito de maneira mais conveniente no método `handleDelivery`, conforme mostrado no exemplo.

Fechar sua conexão e desconectar do agente

Para se desconectar do seu agente RabbitMQ, feche o canal e a conexão, conforme mostrado a seguir.

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

> **Note**
>
> Para obter mais informações sobre como trabalhar com a biblioteca de cliente Java do RabbitMQ, consulte o Guia da API do cliente Java do RabbitMQ.

Etapa 3: (opcional) conectar-se a uma AWS Lambda função

AWS Lambda pode se conectar e consumir mensagens do seu agente Amazon MQ. Quando você conecta um agente ao Lambda, você cria um Mapeamento da origem do evento que lê mensagens de uma fila e invoca a função sincronicamente. O mapeamento da origem do evento que você cria lê mensagens de seu agente em lotes e as converte em uma carga útil do Lambda na forma de um objeto JSON.

Para conectar seu agente a uma função do Lambda

1. Adicione as permissões de Função do IAM a seguir à sua função de execução da função Lambda.

   - metros quadrados: DescribeBroker
   - ec2: CreateNetworkInterface
   - ec2: DeleteNetworkInterface
   - ec2: DescribeNetworkInterfaces
   - ec2: DescribeSecurityGroups
   - ec2: DescribeSubnets
   - ec2: DescribeVpcs
   - troncos: CreateLogGroup
   - troncos: CreateLogStream
   - troncos: PutLogEvents

- gerente de segredos: GetSecretValue

> **Note**
>
> Sem as permissões necessárias do IAM, sua função não poderá ler registros com êxito dos recursos do Amazon MQ.

2. (Opcional) Se você criou um agente sem acessibilidade pública, você deve fazer um dos seguintes procedimentos para permitir que o Lambda se conecte ao seu agente:

   - Configure um gateway NAT por sub-rede pública. Para obter mais informações, consulte Acesso aos serviços e à Internet para funções conectadas à VPC no AWS Lambda Guia do desenvolvedor.
   - Crie uma conexão entre a Amazon Virtual Private Cloud (Amazon VPC) e o Lambda usando um endpoint da VPC. Sua Amazon VPC também deve se conectar aos endpoints AWS Security Token Service (AWS STS) e Secrets Manager. Para obter mais informações, consulte Configurar endpoints da VPC de interface para o Lambda no Guia do desenvolvedor AWS Lambda .

3. Configure seu agente como uma origem do evento para uma função do Lambda usando Console de gerenciamento da AWS. Você também pode usar o `create-event-source-mapping` AWS Command Line Interface comando.

4. Escreva algum código para sua função do Lambda para processar as suas mensagens consumidas pelo seu agente. A carga útil do Lambda recuperada pelo mapeamento da origem do evento depende do tipo de mecanismo do agente. Veja a seguir um exemplo de uma carga útil do Lambda para uma fila do Amazon MQ para RabbitMQ.

> **Note**
>
> No exemplo, `test` é o nome da fila e `/` é o nome do host virtual padrão. Ao receber mensagens, a origem do evento lista as mensagens em `test::/`.

```
{
  "eventSource": "aws:rmq",
  "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
```

```
  "rmqMessagesByQueue": {
    "test::/": [
      {
        "basicProperties": {
          "contentType": "text/plain",
          "contentEncoding": null,
          "headers": {
            "header1": {
              "bytes": [
                118,
                97,
                108,
                117,
                101,
                49
              ]
            },
            "header2": {
              "bytes": [
                118,
                97,
                108,
                117,
                101,
                50
              ]
            },
            "numberInHeader": 10
          }
          "deliveryMode": 1,
          "priority": 34,
          "correlationId": null,
          "replyTo": null,
          "expiration": "60000",
          "messageId": null,
          "timestamp": "Jan 1, 1970, 12:33:41 AM",
          "type": null,
          "userId": "AIDACKCEVSQ6C2EXAMPLE",
          "appId": null,
          "clusterId": null,
          "bodySize": 80
        },
        "redelivered": false,
        "data": "eyJ0aW1lb3V0IjowLCJkYXRhIjoiQ1pybWYwR3c4T3Y0YnFMUXhENEUifQ=="
```

```
        }
      ]
    }
}
```

Para obter mais informações sobre como conectar o Amazon MQ ao Lambda, as opções para as quais o Lambda oferece suporte para uma origem de evento do Amazon MQ e erros de mapeamento da origem do evento, consulte Usar o Lambda com o Amazon MQ no Guia do desenvolvedor AWS Lambda .

Uso da autorização e da autenticação OAuth 2.0 para Amazon MQ para RabbitMQ

Esse tutorial descrever como configura a autenticação OAuth 2.0 para agentes do Amazon MQ para RabbitMQ usando o Amazon Cognito como provedor do OAuth 2.0.

> **Note**
>
> O Amazon Cognito está disponível nas regiões China (Pequim) e China (Ningxia).

> **Important**
>
> Esse tutorial é específico para o Amazon Cognito, mas você pode usar outros provedores de identidade (IDPs). Para obter mais informações, consulte Exemplos de autenticação OAuth 2.0.

Nesta página

Pré-requisitos para configurar a autenticação OAuth 2.0

Você pode definir os recursos do Amazon Cognito necessários neste tutorial implantando a pilha AWS CDK, pilha do Amazon Cognito para plug-in OAuth 2 do RabbitMQ. Se você estiver

configurando o Amazon Cognito manualmente, certifique-se de cumprir os seguintes pré-requisitos antes de configurar o OAuth 2.0 nos agentes do Amazon MQ para RabbitMQ:

Pré-requisitos para configurar o Amazon Cognito

- Configure um endpoint do Amazon Cognito criando um grupo de usuários. Para fazer isso, consulte o blog intitulado Como usar o OAuth 2.0 no Amazon Cognito: Saiba mais sobre as diferentes concessões do OAuth 2.0.
- Crie um servidor de recursos chamado `rabbitmq` no grupo de usuários com os seguintes escopos definidos: `read:allwrite:all`,`configure:all` e `tag:administrator`. Esses escopos serão associados às permissões do RabbitMQ.

  Para obter informações sobre a criação de um servidor de recursos, consulte Definição de um servidor de recursos para o grupo de usuários (Console de gerenciamento da AWS) no Guia do desenvolvedor do Amazon Cognito.
- Crie as aplicações a seguir.
  - Cliente de aplicativo para o grupo de usuários do tipo `Machine-to-Machine application`. Esse é um cliente confidencial com um segredo de cliente que será usado pelos clientes do RabbitMQ AMQP. Para obter mais informações sobre clientes de aplicações e como criar um, consulte Tipos de clientes de aplicação e Criação de um cliente de aplicação.
  - Cliente de aplicativo para o grupo de usuários do tipo `Single-page application`. Esse é um cliente público que será usado para fazer login de usuários no console de gerenciamento do RabbitMQ. Você deve atualizar esse cliente do aplicativo para incluir o endpoint do agente do Amazon MQ para RabbitMQ que você criará no procedimento a seguir como uma URL de retorno de chamada permitida. Para obter mais informações, consulte Configuração do login gerenciado com o console do Amazon Cognito.

Pré-requisito para configurar o Amazon MQ

- Uma instalação funcional do Docker para executar um script bash que verifica se a configuração do OAuth 2.0 foi bem-sucedida ou não.
- Versão da AWS CLI >= `2.28.23` para tornar opcional a adição de um nome de usuário e senha durante a criação do agente.

Configuração da autenticação OAuth 2.0 com o Amazon Cognito usando-se a AWS CLI

O procedimento a seguir mostra como configurar a autenticação OAuth 2.0 para os agentes do Amazon MQ para RabbitMQ usando-se o Amazon Cognito como IdP. Esse procedimento usa a AWS CLI para criar e configurar os recursos necessários.

No procedimento a seguir, certifique-se de substituir os valores do espaço reservado, como configurationID e Revision, *<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>* e *<2>*, por seus valores reais.

1. Crie uma nova configuração usando o comando da AWS CLI create-configuration conforme mostrado no exemplo a seguir.

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

   Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Crie um arquivo de configuração chamado **`rabbitmq.conf`** para usar o OAuth 2.0 como método de autenticação e autorização, conforme mostrado no exemplo a seguir.

```
auth_backends.1 = oauth2
```

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

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

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

Essa configuração usa aliases de escopo para mapear os escopos definidos no Amazon Cognito para escopos compatíveis do RabbitMQ.

3. Atualize a configuração usando o comando da AWS CLI update-configuration conforme mostrado no exemplo a seguir. Nesse comando, adicione o ID de configuração que você recebeu na resposta da Etapa 1 do procedimento. Por exemplo, **`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)"
```

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Crie um agente com a configuração do OAuth 2.0 criada na Etapa 2 deste procedimento. Para fazer isso, use o comando create-broker da AWS CLI, conforme mostrado no exemplo a seguir. Nesse comando, forneça o ID de configuração e o número da revisão obtidos nas respostas das etapas 1 e 2, respectivamente. Por exemplo, **`c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`** e **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>}' \
```

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Verifique se o status do agente muda de `CREATION_IN_PROGRESS` para `RUNNING`, usando o comando da AWS CLI describe-broker, conforme mostrado no exemplo a seguir. Nesse

comando, forneça o ID do corretor que você obteve no resultado da etapa anterior. Por exemplo,**`b-2a1b5133-a10c-49d2-879b-8c176c34cf73`**.

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

Esse comando retorna uma resposta semelhante ao exemplo a seguir. A resposta a seguir é uma versão abreviada da saída completa que o comando `describe-broker` retorna. Essa resposta mostra o status do agente e a estratégia de autenticação usada para proteger o agente. Nesse caso, a estratégia de autenticação `config_managed` indica que o agente usa o método de autenticação OAuth 2.

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

Para fazer login no console de gerenciamento do RabbitMQ usando o OAuth2, o endpoint do agente precisa ser adicionado como uma URL de retorno de chamada válida no cliente da aplicação Amazon Cognito correspondente. Para obter mais informações, consulte a Etapa 5 na configuração do nosso exemplo de pilha de CDK do Amazon Cognito.

6. Verifique a autenticação e a autorização do OAuth 2.0 com o script a seguir `perf-test.sh`.

   Use esse script bash para testar a conectividade com o agente do Amazon MQ para RabbitMQ. Esse script obtém um token do Amazon Cognito e verifica se a conexão foi configurada corretamente. Se for configurado com sucesso, você verá seu agente publicar e consumir mensagens.

   Se você receber um erro `ACCESS_REFUSED`, poderá solucionar os problemas de configuração usando os CloudWatch Logs para o agente. Você pode encontrar o link para o grupo de logs do CloudWatch para o agente no console do Amazon MQ.

   Nesse script, é necessário fornecer os seguintes valores:

   - `CLIENT_ID` e `CLIENT_SECRET`: Você pode encontrar esses valores na página de clientes de aplicativos do console do Amazon Cognito.

- Domínio Cognito: você pode encontrar isso no console do Amazon Cognito. Em Branding (Marca), escolha Domain (Domínio). Na página Domain (Domínio), você pode descobrir esse valor na seção Resource servers (Servidores de recursos).
- Endpoint do agente Amazon MQ: você pode encontrar esse valor em Conexões na página de detalhes do agente no console do 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=${RabbitMqOAuth2TestStack.AmqpAppClientId}
CLIENT_SECRET=${RabbitMqOAuth2TestStack.AmqpAppClientSecret}

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


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

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

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

docker run -it --rm --ulimit nofile=40960:40960 pivotalrabbitmq/perf-test:latest \
```

```
    --queue-pattern 'test-queue-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --flag persistent --rate $PRODUCER_RATE
```

Configuração do OAuth 2.0 e autenticação simples com o Amazon Cognito

Quando criar um agente com autenticação OAuth 2.0, você pode especificar um dos seguintes métodos de autenticação:

- Somente o OAuth 2.0: para usar esse método, forneça um nome de usuário e uma senha quando criar o agente. O procedimento anterior mostra como usar somente o método de autenticação OAuth 2.0.
- Autenticação simples e a OAuth 2.0: para usar esse método, forneça um nome de usuário e uma senha quando criar o agente. Além disso, adicione `auth_backends.2 = internal` à configuração do agente, conforme mostrado no procedimento a seguir.

No procedimento a seguir, certifique-se de substituir os valores do espaço reservado, como<ConfigurationId> *e* <Revision>, por seus valores reais.

1. Para usar os dois métodos de autenticação, crie a configuração do agente, conforme mostrado no exemplo a seguir.

```
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:*/*
```

Essa configuração usa aliases de escopo para mapear os escopos definidos no Amazon Cognito para escopos compatíveis do RabbitMQ.

2. Crie um agente que use os dois métodos de autenticação, conforme mostrado no exemplo a seguir.

```
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. Verifique se o status do agente e a configuração para definir o método de autenticação foram bem-sucedidos, conforme descrito nas etapas 5 e 6 do procedimento Configuração da autenticação OAuth 2.0 com o Amazon Cognito.

Usando autenticação e autorização do IAM para Amazon MQ para RabbitMQ

O procedimento a seguir demonstra como habilitar a autenticação e autorização AWS do IAM para um agente Amazon MQ for RabbitMQ. Depois de habilitar o IAM, os usuários podem se autenticar usando as credenciais AWS do IAM para acessar a API de gerenciamento do RabbitMQ e se conectar via AMQP. Para obter detalhes sobre como a autenticação do IAM funciona com o Amazon MQ para RabbitMQ, consulte. the section called “Autenticação e autorização do IAM”

Pré-requisitos

- AWS credenciais de administrador para a AWS conta proprietária do agente Amazon MQ for RabbitMQ
- Um ambiente de shell configurado com essas credenciais de administrador (usando perfis AWS CLI ou variáveis de ambiente)

- AWS CLI instalada e configurada
- `jq`processador JSON de linha de comando instalado
- `curl`ferramenta de linha de comando instalada

Configurando a autenticação e autorização do IAM usando AWS CLI

1. Definir variáveis de ambiente

   Defina as variáveis de ambiente necessárias para seu corretor:

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

2. Ativar tokens JWT de saída

   Ative a federação externa de identidade da web para sua AWS conta:

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

   A saída exibe um URL de identificador de emissor exclusivo para sua conta no formato`https://<id>.tokens.sts.global.api.aws`.

3. Crie o documento de política do IAM

   Crie um documento de política que conceda permissões para obter tokens de identidade da web:

   ```
   cat > policy.json << 'EOF'
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "VisualEditor0",
               "Effect": "Allow",
   ```

```
            "Action": [
                "sts:GetWebIdentityToken",
                "sts:TagGetWebIdentityToken"
            ],
            "Resource": "*"
        }
    ]
}
EOF
```

4. Crie a política de confiança

   Recupere sua identidade de chamador e crie um documento de política de confiança:

```
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. Crie a função do IAM

   Crie a função do IAM e anexe a 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. Definir as configurações do OAuth2 RabbitMQ

   Crie um arquivo de configuração do RabbitMQ com configurações de OAuth2 autenticação e autorização:

```
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. Atualizar a configuração do broker

   Aplique a nova configuração ao seu corretor:

```
# 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
```

Aguarde até que o status do corretor retorne `RUNNING` antes de prosseguir para a próxima etapa.

8. Obtenha um token JWT

   Assuma a função do IAM e obtenha um token de identidade da 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. Acesse a API de gerenciamento do RabbitMQ

   Use o token JWT para acessar a API de gerenciamento do RabbitMQ:

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

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

Uma resposta bem-sucedida confirma que a autenticação do IAM está funcionando corretamente. A resposta contém informações de visão geral do corretor no formato JSON.

10. Conecte-se via AMQP usando o token JWT

    Teste a conectividade AMQP usando o token JWT com a ferramenta perf-test:

```
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
```

    Se você receber um `ACCESS_REFUSED` erro, poderá solucionar seus problemas de configuração usando os CloudWatch registros do seu corretor. Você pode encontrar o link para o grupo de CloudWatch registros de registros do seu agente no console do Amazon MQ.

Usando autenticação e autorização LDAP para Amazon MQ para RabbitMQ

Este tutorial descreve como configurar a autenticação e autorização LDAP para seu Amazon MQ para corretores RabbitMQ usando. AWS Managed Microsoft AD

Nesta página

Pré-requisitos para configurar a autenticação e autorização LDAP

Você pode configurar os AWS recursos necessários neste tutorial implantando a pilha AWS CDK para a integração do Amazon MQ for RabbitMQ LDAP com. AWS Managed Microsoft AD

Essa pilha de CDK cria automaticamente todos os AWS recursos necessários AWS Managed Microsoft AD, incluindo usuários e grupos LDAP, Network Load Balancer, certificados e funções do IAM. Consulte o pacote README para obter uma lista completa dos recursos criados pela pilha.

Se você estiver configurando os recursos manualmente em vez de usar a pilha de CDK, certifique-se de ter a infraestrutura equivalente antes de configurar o LDAP em seus corretores Amazon MQ para RabbitMQ.

Pré-requisito para configurar o Amazon MQ

AWS Versão CLI >= 2.28.23 para tornar opcional a adição de um nome de usuário e senha durante a criação do broker.

Configurando o LDAP no RabbitMQ usando CLI AWS

Esse procedimento usa a AWS CLI para criar e configurar os recursos necessários. No procedimento a seguir, certifique-se de substituir os valores do espaço reservado, como configurationId e Revision, `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` e`<2>`, por seus valores reais.

1. Crie uma nova configuração usando o comando `create-configuration` AWS CLI, conforme mostrado no exemplo a seguir.

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

   Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
"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. Crie um arquivo de configuração chamado `rabbitmq.conf` para usar o LDAP como método de autenticação e autorização, conforme mostrado no exemplo a seguir. Substitua todos os valores de espaço reservado no modelo (marcados com`${RabbitMqLdapTestStack.*}`) pelos valores reais das saídas de pilha de AWS CDK pré-requisitos implantadas ou da infraestrutura 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. Atualize a configuração usando o comando `update-configuration` AWS CLI, conforme mostrado no exemplo a seguir. Nesse comando, adicione o ID de configuração recebido na

resposta da Etapa 1 do procedimento. Por exemplo, .`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)"
```

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Crie um broker com a configuração LDAP que você criou na Etapa 2 deste procedimento. Para fazer isso, use o comando `create-broker` AWS CLI conforme mostrado no exemplo a seguir. Nesse comando, informe o ID de configuração e o número da revisão obtidos nas respostas das etapas 1 e 2, respectivamente. Por exemplo, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` e 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>}'
```

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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"
}
```

> **Restrição de nomenclatura do corretor**
>
> A função do IAM criada pela pilha CDK de pré-requisito restringe os nomes dos corretores para começar. `rabbitmq-ldap-test` Certifique-se de que o nome do seu corretor siga esse padrão ou a função do IAM não terá permissão para assumir a função na resolução do ARN.

5. Verifique se o status do broker muda de `CREATION_IN_PROGRESS` para`RUNNING`, usando o comando `describe-broker` AWS CLI, conforme mostrado no exemplo a seguir. Nesse comando, informe o ID do agente obtido no resultado da etapa anterior. Por exemplo, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

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

Esse comando retorna uma resposta semelhante ao exemplo a seguir. A resposta a seguir é uma versão abreviada da saída completa que o comando `describe-broker` retorna. Essa resposta mostra o status do agente e a estratégia de autenticação usada para proteger o agente. Nesse caso, a estratégia de `config_managed` autenticação indica que o agente usa o método de autenticação LDAP.

```
{
```

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

6. Valide o acesso ao RabbitMQ usando um dos usuários de teste criados pela pilha CDK de pré-requisito

```
# 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"}'
```

Usando autenticação e autorização HTTP para Amazon MQ para RabbitMQ

Este tutorial descreve como configurar a autenticação e autorização HTTP para seus corretores Amazon MQ para RabbitMQ usando um servidor HTTP externo.

> **Note**
>
> O plug-in de autenticação HTTP está disponível somente para o Amazon MQ for RabbitMQ versão 4 e superior.

Nesta página

Pré-requisitos para configurar a autenticação e autorização HTTP

Você pode configurar os AWS recursos necessários neste tutorial implantando a pilha AWS CDK para Amazon MQ para integração de autenticação HTTP RabbitMQ.

Essa pilha de CDK cria automaticamente todos os AWS recursos necessários, incluindo o servidor de autenticação HTTP, certificados e funções do IAM. Consulte o pacote README para obter uma lista completa dos recursos criados pela pilha.

Se você estiver configurando os recursos manualmente em vez de usar a pilha CDK, certifique-se de ter a infraestrutura equivalente antes de configurar a autenticação HTTP em seus corretores Amazon MQ para RabbitMQ.

Pré-requisito para configurar o Amazon MQ

AWS Versão CLI >= 2.28.23 para tornar opcional a adição de um nome de usuário e senha durante a criação do broker.

Configurando a autenticação HTTP no RabbitMQ usando CLI AWS

Esse procedimento usa a AWS CLI para criar e configurar os recursos necessários. No procedimento a seguir, certifique-se de substituir os valores do espaço reservado por seus valores reais.

1. Crie uma nova configuração usando o comando `create-configuration` AWS CLI, conforme mostrado no exemplo a seguir.

   ```
   aws mq create-configuration \
     --name "rabbitmq-http-config" \
     --engine-type "RABBITMQ" \
     --engine-version "4.2"
   ```

   Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Crie um arquivo de configuração chamado `rabbitmq.conf` para usar HTTP como método de autenticação e autorização, conforme mostrado no exemplo a seguir. Substitua todos os valores de espaço reservado no modelo (marcados com`${...}`) pelos valores reais das saídas de pilha de AWS CDK pré-requisitos implantadas ou da infraestrutura 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. Atualize a configuração usando o `update-configuration` AWS comando CLI. Use o ID de configuração da Etapa 3.

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

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Crie um broker com a configuração HTTP. Use o ID de configuração e o número da revisão das etapas 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>}'
```

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Verifique se o status do broker muda de `CREATION_IN_PROGRESS` para`RUNNING`, usando o comando `describe-broker` AWS CLI.

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

Esse comando retorna uma resposta semelhante ao exemplo a seguir. A estratégia de `config_managed` autenticação indica que o agente usa o método de autenticação HTTP.

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

6. Valide o acesso ao RabbitMQ usando um dos usuários de teste criados pela pilha CDK de pré-requisito

```
# 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 '{}'
```

Usando a autenticação de certificado SSL para Amazon MQ para RabbitMQ

Este tutorial descreve como configurar a autenticação de certificado SSL para seus corretores Amazon MQ para RabbitMQ usando uma autoridade de certificação privada.

> **Note**
>
> O plug-in de autenticação de certificado SSL está disponível somente para o Amazon MQ for RabbitMQ versão 4 e superior.

Nesta página

Pré-requisitos para configurar a autenticação do certificado SSL

A autenticação de certificado SSL usa TLS mútuo (mTLS) para autenticar clientes usando certificados X.509. Você pode configurar os AWS recursos necessários neste tutorial implantando a pilha de AWS CDK para a integração entre Amazon MQ e RabbitMQ mTLS.

Essa pilha de CDK cria automaticamente todos os AWS recursos necessários, incluindo autoridade de certificação, certificados de cliente e funções do IAM. Consulte o pacote README para obter uma lista completa dos recursos criados pela pilha.

> **Note**
>
> Antes de implantar a pilha CDK, defina a `RABBITMQ_TEST_USER_NAME` variável de ambiente. Esse valor será usado como o Nome Comum (CN) no certificado do cliente e deve corresponder ao nome de usuário usado nas etapas do tutorial. Por exemplo: `export RABBITMQ_TEST_USER_NAME="myuser"`

Se você estiver configurando os recursos manualmente em vez de usar a pilha de CDK, certifique-se de ter a infraestrutura equivalente antes de configurar a autenticação de certificado SSL em seus corretores Amazon MQ para RabbitMQ.

Pré-requisito para configurar o Amazon MQ

AWS Versão CLI >= 2.28.23 para tornar opcional a adição de um nome de usuário e senha durante a criação do broker.

Configurando a autenticação de certificado SSL no RabbitMQ usando CLI AWS

Esse procedimento usa a AWS CLI para criar e configurar os recursos necessários. No procedimento a seguir, certifique-se de substituir os valores do espaço reservado, como configurationId e Revision, `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` e`<2>`, por seus valores reais.

1. Crie uma nova configuração usando o comando `create-configuration` AWS CLI, conforme mostrado no exemplo a seguir.

```
aws mq create-configuration \
  --name "rabbitmq-ssl-config" \
  --engine-type "RABBITMQ" \
```

```
  --engine-version "4.2"
```

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Crie um arquivo de configuração chamado `rabbitmq.conf` para usar a autenticação de certificado SSL, conforme mostrado no exemplo a seguir. Substitua todos os valores de espaço reservado no modelo (marcados com`${...}`) pelos valores reais das saídas de pilha de AWS CDK pré-requisitos implantadas ou da infraestrutura 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. Atualize a configuração usando o comando `update-configuration` AWS CLI, conforme mostrado no exemplo a seguir. Nesse comando, adicione o ID de configuração recebido na resposta da Etapa 1 do procedimento. Por exemplo, `.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)"
```

   Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Crie um broker com a configuração de autenticação do certificado SSL que você criou na Etapa 2 deste procedimento. Para fazer isso, use o comando `create-broker` AWS CLI conforme mostrado no exemplo a seguir. Nesse comando, informe o ID de configuração e o número da revisão obtidos nas respostas das etapas 1 e 2, respectivamente. Por exemplo, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` e `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"}]'
```

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Verifique se o status do broker muda de `CREATION_IN_PROGRESS` para`RUNNING`, usando o comando `describe-broker` AWS CLI, conforme mostrado no exemplo a seguir. Neste comando, forneça a ID do agente que você obteve no resultado da etapa anterior. Por exemplo, .`b-2a1b5133-a10c-49d2-879b-8c176c34cf73`

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

Esse comando retorna uma resposta semelhante ao exemplo a seguir. A resposta a seguir é uma versão abreviada da saída completa que o comando `describe-broker` retorna. Essa resposta mostra o status do agente e a estratégia de autenticação usada para proteger o agente. Nesse caso, a estratégia de `config_managed` autenticação indica que o agente usa o método de autenticação de certificado SSL.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
```

```
    "BrokerState": "RUNNING",
    ...
}
```

6. Verifique a autenticação do certificado SSL com o `ssl.sh` script a seguir.

   Use esse script bash para testar a conectividade com o agente do Amazon MQ para RabbitMQ. Esse script usa seu certificado de cliente para autenticação e verifica se a conexão foi configurada corretamente. Se for configurado com sucesso, você verá seu corretor publicar e consumir mensagens.

   Se você receber um `ACCESS_REFUSED` erro, poderá solucionar seus problemas de configuração usando os CloudWatch registros do seu corretor. Você pode encontrar o link para o grupo de CloudWatch registros do seu agente no console do Amazon MQ.

   Nesse script, é necessário fornecer os seguintes valores:

   - `USERNAME`: O nome comum (CN) do seu certificado de cliente.
   - `CLIENT_KEYSTORE`: caminho para o arquivo de armazenamento de chaves do seu cliente (PKCS12 formato). Se você usou a pilha CDK de pré-requisito, o caminho padrão é. `$(pwd)/certs/client-keystore.p12`
   - `KEYSTORE_PASSWORD`: Senha para o armazenamento de chaves do seu cliente. Se você usou a pilha CDK de pré-requisito, a senha padrão é. `changeit`
   - `BROKER_DNS`: Você pode encontrar esse valor em Conexões na página de detalhes do broker do console do 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
```

Usando mTLS para AMQP e endpoints de gerenciamento

Este tutorial descreve como configurar o TLS mútuo (mTLS) para conexões de clientes AMQP e a interface de gerenciamento do RabbitMQ usando uma autoridade de certificação privada.

> **Note**
>
> O uso de autoridades de certificação privadas para mTLS está disponível somente para Amazon MQ para RabbitMQ versão 4 e superior.

Nesta página

Pré-requisitos para configurar o mTLS

Você pode configurar os AWS recursos necessários neste tutorial implantando a pilha AWS CDK para a integração do Amazon MQ para RabbitMQ mTLS com.

Essa pilha de CDK cria automaticamente todos os AWS recursos necessários, incluindo autoridade de certificação, certificados de cliente e funções do IAM. Consulte o pacote README para obter uma lista completa dos recursos criados pela pilha.

Se você estiver configurando os recursos manualmente em vez de usar a pilha CDK, certifique-se de ter a infraestrutura equivalente antes de configurar o mTLS em seus corretores Amazon MQ para RabbitMQ.

Pré-requisito para configurar o Amazon MQ

AWS Versão CLI >= 2.28.23 para tornar opcional a adição de um nome de usuário e senha durante a criação do broker.

Configurando mTLS no RabbitMQ usando CLI AWS

Esse procedimento usa a AWS CLI para criar e configurar os recursos necessários. No procedimento a seguir, certifique-se de substituir os valores do espaço reservado, como ConfigurationId e Revision, `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` e`<2>`, por seus valores reais.

1. Crie uma nova configuração usando o comando `create-configuration` AWS CLI, conforme mostrado no exemplo a seguir.

```
aws mq create-configuration \
  --name "rabbitmq-mtls-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

   Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Crie um arquivo de configuração chamado `rabbitmq.conf` para configurar mTLS para AMQP e endpoints de gerenciamento, conforme mostrado no exemplo a seguir. Substitua todos os valores de espaço reservado no modelo (marcados com`${...}`) pelos valores reais das saídas de pilha de AWS CDK pré-requisitos implantadas ou da infraestrutura 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. Atualize a configuração usando o comando `update-configuration` AWS CLI, conforme mostrado no exemplo a seguir. Nesse comando, adicione o ID de configuração recebido na resposta da Etapa 1 do procedimento. Por exemplo, `.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)"
```

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Crie um broker com a configuração mTLS que você criou na Etapa 2 deste procedimento. Para fazer isso, use o comando `create-broker` AWS CLI conforme mostrado no exemplo a seguir. Nesse comando, informe o ID de configuração e o número da revisão obtidos nas respostas das etapas 1 e 2, respectivamente. Por exemplo, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` e 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"}]'
```

Esse comando retorna uma resposta semelhante ao exemplo a seguir.

```
{
    "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. Verifique se o status do broker muda de `CREATION_IN_PROGRESS` para`RUNNING`, usando o comando `describe-broker` AWS CLI, conforme mostrado no exemplo a seguir. Neste comando, forneça a ID do agente que você obteve no resultado da etapa anterior. Por exemplo, .`b-2a1b5133-a10c-49d2-879b-8c176c34cf73`

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

   Esse comando retorna uma resposta semelhante ao exemplo a seguir. A resposta a seguir é uma versão abreviada da saída completa que o comando `describe-broker` retorna.

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

6. Verifique a autenticação mTLS com o `mtls.sh` script a seguir.

   Use esse script bash para testar a conectividade com o agente do Amazon MQ para RabbitMQ. Esse script usa seu certificado de cliente para autenticar e verificar se a conexão foi configurada corretamente. Se for configurado com sucesso, você verá seu corretor publicar e consumir mensagens.

   Se você receber um `ACCESS_REFUSED` erro, poderá solucionar seus problemas de configuração usando os CloudWatch registros do seu broker. Você pode encontrar o link para o grupo de CloudWatch registros do seu agente no console do Amazon MQ.

Nesse script, é necessário fornecer os seguintes valores:

- `USERNAMEePASSWORD`: As credenciais de usuário do RabbitMQ que você criou com o corretor.
- `CLIENT_KEYSTORE`: caminho para o arquivo de armazenamento de chaves do seu cliente (PKCS12 formato). Se você usou a pilha CDK de pré-requisito, o caminho padrão é. `$(pwd)/certs/client-keystore.p12`
- `KEYSTORE_PASSWORD`: Senha para o armazenamento de chaves do seu cliente. Se você usou a pilha CDK de pré-requisito, a senha padrão é. `changeit`
- `BROKER_DNS`: Você pode encontrar esse valor em Conexões na página de detalhes do broker do console do 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
```

Conectando seu aplicativo JMS

Este tutorial mostra como conectar seu aplicativo JMS ao agente Amazon MQ for RabbitMQ usando o cliente RabbitMQ JMS. Você aprenderá como criar um produtor para enviar mensagens e um consumidor para receber mensagens das filas do RabbitMQ.

Antes de começar, adicione a dependência apropriada do RabbitMQ JMS ao seu projeto Maven:

Para JMS 1.1 e 2.0:

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>2.12.0</version>
  </dependency>

</dependencies>
```

Para o JMS 3.1:

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>3.5.0</version>
  </dependency>

</dependencies>
```

Crie um produtor

O exemplo de código a seguir mostra como gravar em uma fila do RabbitMQ usando 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);
```

Crie um consumidor

O exemplo de código a seguir mostra como ler de uma fila do RabbitMQ usando 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 (JMSException e) {
        System.err.printf("Error processing message: %s%n", e.getMessage());
    }
});

connection.start();
```

Segurança no Amazon MQ

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade que se aplicam ao Amazon MQ, consulte AWS Serviços no escopo do programa de conformidade AWS Serviços no escopo do programa .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação te ajuda a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon MQ. Os tópicos a seguir mostram como configurar o Amazon MQ para atender aos seus objetivos de segurança e de conformidade. Você também aprende a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos do Amazon MQ.

Tópicos

Proteção de dados no Amazon MQ

O modelo de responsabilidade AWS compartilhada de se aplica à proteção de dados no Amazon MQ. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para saber mais sobre a privacidade de dados, consulte as Data Privacy FAQ. Para saber mais sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com Centro de Identidade do AWS IAM ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sensíveis armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para saber mais sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sensíveis, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Amazon MQ ou outro Serviços da AWS usando o console, a API ou. AWS CLI AWS SDKs Quaisquer dados inseridos em tags ou em campos de texto de formato

livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse servidor.

Para os agentes do Amazon MQ para ActiveMQ e do Amazon MQ para RabbitMQ, não use qualquer informação de identificação pessoal (PII) ou outras informações confidenciais para os nomes de agente ou nomes de usuário ao criar recursos por meio do console da Web do agente ou da API do Amazon MQ. Os nomes de corretores e nomes de usuário podem ser acessados por outros AWS serviços, incluindo CloudWatch registros. Nomes de usuário do agente não devem ser usados para dados privados ou sigilosos.

> **Important**
>
> O TLS 1.3 não está disponível para agentes do RabbitMQ.

Criptografia

Os dados de usuário armazenados no Amazon MQ são criptografados em repouso. A criptografia em repouso do Amazon MQ fornece segurança aprimorada ao criptografar os seus dados usando chaves de criptografia armazenadas no AWS Key Management Service (KMS). Esse serviço ajuda a reduzir a carga e a complexidade operacionais necessárias para proteger dados confidenciais. Com a criptografia de dados em repouso, você pode criar aplicativos confidenciais que atendem a requisitos de conformidade e regulamentação de criptografia.

Todas as conexões entre os agentes do Amazon MQ usam Transport Layer Security (TLS) para fornecer a criptografia em trânsito.

O Amazon MQ criptografa mensagens em repouso e em trânsito usando chaves de criptografia que gerencia e armazena com segurança. Para obter mais informações, consulte o Guia do desenvolvedor do AWS Encryption SDK.

Criptografia em repouso

O Amazon MQ se integra ao AWS Key Management Service (KMS) para oferecer criptografia transparente no lado do servidor. O Amazon MQ sempre criptografa seus dados em repouso.

Ao criar um agente Amazon MQ para ActiveMQ ou um agente Amazon MQ para RabbitMQ, você pode especificar o que AWS KMS key deseja que o Amazon MQ use para criptografar seus dados

em repouso. Se você não especificar uma chave KMS, o Amazon MQ cria AWS uma chave KMS própria para você e a usa em seu nome. Atualmente, o Amazon MQ é compatível com chaves simétricas do KMS. Para obter mais informações sobre chaves do KMS, consulte AWS KMS keys.

Ao criar um agente, você pode configurar o que o Amazon MQ utiliza para a sua chave de criptografia ao selecionar uma das seguintes ações.

- Chave do KMS pertencente ao Amazon MQ (padrão): a chave pertence ao Amazon MQ e é gerenciada por ele e não está na sua conta.
- AWS chave KMS gerenciada — A chave KMS AWS gerenciada (`aws/mq`) é uma chave KMS em sua conta que é criada, gerenciada e usada em seu nome pelo Amazon MQ.
- Selecione uma chave KMS gerenciada pelo cliente — KMSs gerenciadas pelo cliente são criadas e gerenciadas por você no AWS Key Management Service (KMS).

⚠ Important

- A revogação de uma concessão não pode ser desfeita. Exclua o agente para revogar os direitos de acesso.
- Quanto a agentes do Amazon MQ para ActiveMQ que usam o Amazon Elastic File System (EFS) para armazenar dados de mensagens, pode levar várias horas para que as permissões de uso das chaves do KMS em sua conta sejam revogadas após a realização das ações necessárias.
- Com relação a agentes do Amazon MQ para RabbitMQ e agentes do Amazon MQ para ActiveMQ que usam o EBS para armazenar dados de mensagens, se você desabilitar, agendar para exclusão ou revogar a concessão que permite que o Amazon EBS use as chaves do KMS em sua conta, o Amazon MQ não poderá manter seu agente e ele poderá mudar para um estado degradado.
- Se você desativou ou programou a exclusão da chave, poderá reativá-la ou cancelar a exclusão e manter o agente.
- Pode levar várias horas para desativar uma chave ou revogar uma concessão depois de realizar as ações necessárias.
- Para criptografar ou descriptografar CloudWatch registros, você não pode configurar o que o Amazon MQ usa para sua chave de criptografia. CloudWatch os registros protegem os dados em repouso usando criptografia, e os grupos de registros são criptografados. O serviço de CloudWatch registros gerencia a criptografia do lado do servidor por padrão.

Para obter mais informações sobre como os grupos de registros são criptografados, consulte o Guia do usuário do Amazon CloudWatch Logs.

Ao criar um agente de instância única com uma chave do KMS para o RabbitMQ, você verá dois eventos `CreateGrant` conectados no AWS CloudTrail. O primeiro evento é o Amazon MQ criando uma concessão para a chave do KMS. O segundo evento é o EBS criando uma concessão para uso do EBS.

CreateGrant AWS CloudTrail entrada de registro: agente de instância 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

Você verá um evento para a criação de concessão do 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"
}
```

Ao criar uma implantação de cluster com uma chave do KMS para o RabbitMQ, você verá cinco eventos `CreateGrant` conectados no AWS CloudTrail. Os dois primeiros eventos são criações de concessão para o Amazon MQ. Os próximos três eventos são concessões criadas pelo EBS para uso do EBS.

CreateGrant AWS CloudTrail entrada de registro: implantação de cluster

mq_grant

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
                "accountId": "111122223333",
                "userName": "AmazonMqConsole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-23T18:59:10Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "mq.amazonaws.com"
```

```
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-east-1:316438333700:key/bdbe42ae-f825-4e78-
a8a1-828d411c4be2",
        "retiringPrincipal": "mq.amazonaws.com",
        "operations": [
            "CreateGrant",
            "Encrypt",
            "Decrypt",
            "ReEncryptFrom",
            "ReEncryptTo",
            "GenerateDataKey",
            "GenerateDataKeyWithoutPlaintext",
            "DescribeKey"
        ]
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
           "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
```

```
    "sessionCredentialFromConsole": "true"
}
```

mq_rabbit_grant

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
                "accountId": "111122223333",
                "userName": "AmazonMqConsole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-23T18:59:10Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "retiringPrincipal": "mq.amazonaws.com",
        "operations": [
            "DescribeKey"
        ],
```

```
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
           "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}
```

EBS grant creation

Você verá três eventos para a criação de concessão do 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 obter mais informações sobre como usar as chaves KMS, consulte AWS KMS keys o AWS Key Management Service Guia do desenvolvedor.

Criptografia em trânsito

Amazon MQ para ActiveMQ: o Amazon MQ para ActiveMQ exige Transport Layer Security (TLS) forte e criptografa dados em trânsito entre os agentes da implantação do Amazon MQ. Todos os dados transmitidos entre os agentes do Amazon MQ são criptografados usando Transport Layer Security (TLS) forte. Isso se aplica a todos os protocolos disponíveis.

Amazon MQ para RabbitMQ: o Amazon MQ para RabbitMQ exige uma criptografia forte de Transport Layer Security (TLS) para todas as conexões do cliente. O tráfego de replicação de cluster do RabbitMQ transita apenas pela VPC do seu broker e todo o tráfego de rede entre os AWS data centers é criptografado de forma transparente na camada física. Atualmente, os agentes em clusters do Amazon MQ para RabbitMQ não são compatíveis com a criptografia entre nós para replicação de clusters. Para saber mais sobre isso data-in-transit, consulte Criptografia Data-at-Rest e em trânsito.

Amazon MQ para protocolos do ActiveMQ

Você pode acessar seus agentes do ActiveMQ usando os seguintes protocolos com TLS habilitado:

- AMQP
- MQTT
- Acabou o MQTT WebSocket
- OpenWire
- STOMP
- STOMP over WebSocket

Pacotes de criptografia do TLS compatíveis com ActiveMQ.

O ActiveMQ no Amazon MQ é compatível com os seguintes pacotes de criptografia:

- TLS_ECDHE_RSA_COM_AES_256_GCM_ SHA384
- TLS_ECDHE_RSA_COM_AES_256_CBC_ SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_COM_AES_256_GCM_ SHA384
- TLS_DHE_RSA_COM_AES_256_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA

- TLS_RSA_COM_AES_256_GCM_ SHA384
- TLS_RSA_COM_AES_256_CBC_ SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_COM_AES_128_GCM_ SHA256
- TLS_ECDHE_RSA_COM_AES_128_CBC_ SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_COM_AES_128_GCM_ SHA256
- TLS_DHE_RSA_COM_AES_128_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_COM_AES_128_GCM_ SHA256
- TLS_RSA_COM_AES_128_CBC_ SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA

Amazon MQ para protocolos RabbitMQ

Você pode acessar seus agentes RabbitMQ usando os seguintes protocolos com TLS habilitado:

- AMQP (0-9-1)

Pacotes de criptografia do TLS compatíveis com RabbitMQ.

O RabbitMQ no Amazon MQ é compatível com os seguintes pacotes de criptografia:

- TLS_ECDHE_RSA_COM_AES_256_GCM_ SHA384
- TLS_ECDHE_RSA_COM_AES_128_GCM_ SHA256

Gerenciamento de identidade e acesso para o Amazon MQ

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) para usar os recursos da Amazon MQ. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte Solução de problemas de identidade e acesso da Amazon MQ).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como o Amazon MQ funciona com o IAM)
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte Exemplos de políticas baseadas em identidade do Amazon MQ)

Autenticação com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte Como fazer login em sua Conta da AWS no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte AWS Signature Version 4 para solicitações de API no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Usuários e grupos

Um usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do usuário do IAM.

Um grupo do IAM especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte Casos de uso de usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma perfil do IAM é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função mudando de um usuário para uma função do IAM (console) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte Métodos para assumir um perfil no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como

documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário especificar uma entidade principal em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o AWS WAF Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como o Amazon MQ funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon MQ, você deve entender quais recursos do IAM estão disponíveis para uso com a Amazon MQ. Para ter uma visão de alto nível de como o Amazon MQ e AWS outros serviços funcionam com o IAM, AWS consulte Serviços que funcionam com o IAM no Guia do usuário do IAM.

O Amazon MQ usa o IAM para operações de API do Amazon MQ para criar, atualizar, excluir e listar agentes. Para acesso do agente para publicar e assinar mensagens, o Amazon MQ para ActiveMQ

oferece suporte à autenticação ActiveMQ nativa e ao LDAP, enquanto o Amazon MQ para RabbitMQ suporta autenticação IAM e outros métodos. Para obter mais informações, consulte the section called “Autenticação e autorização do corretor”.

Tópicos

Políticas baseadas em identidade do Amazon MQ

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. O Amazon MQ é compatível com ações, chaves de condição e recursos específicos. Para conhecer todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

As ações de política no Amazon MQ usam o seguinte prefixo antes da ação: `mq:`. Por exemplo, para conceder permissão a alguém para executar uma instância do Amazon MQ com a operação da API `CreateBroker` do Amazon MQ, inclua a ação `mq:CreateBroker` na política da pessoa. As instruções de política devem incluir um elemento `Action` ou `NotAction`. O Amazon MQ define seu próprio conjunto de ações que descrevem as tarefas que você pode executar com esse serviço.

Para especificar várias ações em uma única instrução, separe-as com vírgulas, como segue:

```
"Action": [
      "mq:action1",
      "mq:action2"
```

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "mq:Describe*"
```

Para ver uma lista das ações do Amazon MQ, consulte Ações definidas pelo Amazon MQ no Manual do usuário IAM.

Recursos

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

No Amazon MQ, os principais AWS recursos são um agente de mensagens do Amazon MQ e sua configuração. Cada agente e configuração do Amazon MQ tem Amazon Resource Names (ARNs) exclusivos associados a eles, conforme mostrado na tabela a seguir.

Tipos de recursos	ARN	Chaves de condição
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 obter mais informações sobre o formato de ARNs, consulte Amazon Resource Names (ARNs) e AWS Service Namespaces.

Por exemplo, para especificar o agente denominado `MyBroker` com brokerId `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` em sua declaração, use o seguinte ARN:

```
"Resource": "arn:aws:mq:us-
east-1:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9"
```

Para especificar todos os agentes e configurações que pertencem a uma conta específica, use o caractere curinga (*):

```
"Resource": "arn:aws:mq:us-east-1:123456789012:*"
```

Algumas ações do Amazon MQ, como as de criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, é necessário utilizar o caractere curinga (*).

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

A ação da API `CreateTags` requer um agente e uma configuração. Para especificar vários recursos em uma única instrução, separe-os ARNs com vírgulas.

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

Para ver uma lista dos tipos de recursos do Amazon MQ e seus ARNs, consulte Recursos definidos pelo Amazon MQ no Guia do usuário do IAM. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo Amazon MQ.

Chaves de condição

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

O Amazon MQ não define nenhuma chave de condição específica ao serviço, mas é compatível com o uso de algumas chaves de condição globais. Para ver uma lista de chaves de condição do Amazon MQ, consulte a tabela abaixo ou Chaves de condição para o Amazon MQ no Manual do usuário do IAM. Para saber com quais ações e recursos você pode usar a chave de condição, consulte Ações definidas pelo Amazon MQ.

Chaves de condição	Descrição	Tipo
foi: RequestTag / $ {} TagKey	Filtra ações com base nas tags transmitidas na solicitaç ão.	String
foi: ResourceT ag /$ {} TagKey	Filtra as ações com base nas tags associadas ao recurso.	String
leis: TagKeys	Filtra ações com base nas chaves de tag transmitidas na solicitação.	String

Exemplos

Para visualizar exemplos de políticas baseadas em identidade do Amazon MQ, consulte Exemplos de políticas baseadas em identidade do Amazon MQ.

Políticas baseadas em recursos do Amazon MQ

Atualmente, o Amazon MQ não é compatível com a autenticação IAM que usam permissões baseadas em recursos ou políticas baseadas em recursos.

Autorização baseada em etiquetas do Amazon MQ

É possível anexar etiquetas aos recursos do Amazon MQ ou informar etiquetas em uma solicitação para o Amazon MQ. Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `mq:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

O Amazon MQ é compatível com políticas baseadas em etiquetas. Por exemplo, você pode negar acesso a todos os recursos do Amazon MQ que incluem uma etiqueta com a chave `environment` e o 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 vai `Deny` a capacidade de excluir ou reiniciar um agente do Amazon MQ que inclui a etiqueta `environment/production`.

Para obter mais informações sobre marcação, consulte:

- Adicionar tags aos recursos do Amazon MQ
- Controlar o acesso com tags do IAM

Funções do IAM do Amazon MQ

Uma função do IAM é uma entidade dentro da sua AWS conta que tem permissões específicas.

Usar credenciais temporárias com o Amazon MQ

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. Você obtém credenciais de segurança temporárias chamando operações de AWS STS API, como AssumeRoleou GetFederationToken.

A Amazon MQ é compatível com o uso de credenciais temporárias.

Perfis de serviço

Esse atributo permite que um serviço assuma um perfil de serviço em seu nome. O perfil permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. Os perfis de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso significa que um administrador do IAM pode alterar as permissões para esse perfil. Porém, fazer isso pode alterar a funcionalidade do serviço.

O Amazon MQ é compatível com as funções de serviço.

Exemplos de políticas baseadas em identidade do Amazon MQ

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do Amazon MQ. Eles também não podem realizar tarefas usando a AWS API Console de gerenciamento da AWS AWS CLI, ou. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e perfis permissão para executarem operações de API específicas nos recursos especificados de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Tópicos

- Práticas recomendadas de política
- Usar o console do Amazon MQ
- Permitir que os usuários visualizem suas próprias permissões

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos do Amazon MQ em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões

definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.

- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usar o console do Amazon MQ

Para acessar o console da Amazon MQ, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do Amazon MQ em sua AWS conta. Se você criar uma política baseada em identidade que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis do IAM) com essa política.

Para garantir que essas entidades ainda possam usar o console do Amazon MQ, anexe também a seguinte política AWS gerenciada às entidades. Para obter mais informações, consulte Adição de permissões a um usuário no Manual do usuário do IAM:

```
AmazonMQReadOnlyAccess
```

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente às ações que correspondem à operação da API que você está tentando executar.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
        }
    ]
}
```

Autorização e autenticação de API para o Amazon MQ

O Amazon MQ usa assinatura de AWS solicitação padrão para autenticação de API. Para obter mais informações, consulte Assinatura de solicitações da API AWS da no Referência geral da AWS.

> **Note**
>
> Atualmente, o Amazon MQ não é compatível com a autenticação IAM que usam permissões baseadas em recursos ou políticas baseadas em recursos.

Para autorizar AWS os usuários a trabalhar com corretores, configurações e usuários, você deve editar suas permissões de política do IAM.

Tópicos

- Permissões de IAM necessárias para criar um agente Amazon MQ
- Referência de permissões da API REST do Amazon MQ
- Referência de permissões adicionais do Amazon MQ
- Permissões no nível do recurso suportadas para ações de API do Amazon MQ

Permissões de IAM necessárias para criar um agente Amazon MQ

Para criar um agente, você deve usar a política do IAM `AmazonMQFullAccess` ou incluir as permissões do EC2 a seguir em sua política do IAM.

A seguinte política personalizada é composta de duas declarações (uma condicional) que concedem permissões para manipular os recursos que o Amazon MQ exige para criar um agente do ActiveMQ.

> **Important**
>
> - A ação `ec2:CreateNetworkInterface` é necessária para permitir que o Amazon MQ crie uma interface de rede elástica (ENI) em sua conta em seu nome.
> - A ação do `ec2:CreateNetworkInterfacePermission` autoriza o Amazon MQ a anexar a ENI para um agente do ActiveMQ.
> - A chave de condição `ec2:AuthorizedService` garante que as permissões de ENI possam ser concedidas apenas para contas de serviço do 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 obter mais informações, consulte Etapa 2: criar um usuário e obter suas AWS credenciais e Nunca modifique ou exclua a interface de rede elástica do Amazon MQ.

Referência de permissões da API REST do Amazon MQ

A tabela a seguir lista o Amazon MQ REST APIs e as permissões correspondentes do IAM.

Amazon MQ REST APIs e permissões necessárias

Amazon MQ REST APIs	Permissões obrigatórias
CreateBroker	mq:CreateBroker
CreateConfiguration	mq:CreateConfiguration
CreateTags	mq:CreateTags
CreateUser	mq:CreateUser
DeleteBroker	mq:DeleteBroker
DeleteUser	mq:DeleteUser
DescribeBroker	mq:DescribeBroker
DescribeConfiguration	mq:DescribeConfiguration
DescribeConfigurationRevision	mq:DescribeConfigurationRevision
DescribeUser	mq:DescribeUser
ListBrokers	mq:ListBrokers
ListConfigurationRevisions	mq:ListConfigurationRevisions
ListConfigurations	mq:ListConfigurations

Amazon MQ REST APIs	Permissões obrigatórias
ListTags	mq:ListTags
ListUsers	mq:ListUsers
RebootBroker	mq:RebootBroker
UpdateBroker	mq:UpdateBroker
UpdateConfiguration	mq:UpdateConfiguration
UpdateUser	mq:UpdateUser

Referência de permissões adicionais do Amazon MQ

A tabela a seguir lista a API do Amazon MQ e a permissão adicional do IAM necessária para recursos específicos, como a autenticação OAuth 2.0.

API REST do Amazon MQ	Permissão	Description
UpdateBroker	mq:UpdateBrokerAccessConfiguration	Você precisa dessa permissão para atualizar as opções de autenticação e autorização na configuração do agente associado. Para obter mais informações, consulte OAuth Autenticação e autorização 2.0 para Amazon MQ para RabbitMQ.

Permissões no nível do recurso suportadas para ações de API do Amazon MQ

O termo permissões no nível do recurso se refere à capacidade de especificar em quais recursos os usuários têm permissão para realizar ações. O Amazon MQ é compatível parcialmente com as permissões no nível do recurso. Para determinadas ações do Amazon MQ, você pode controlar

quando os usuários têm permissão para usar essas ações com base em condições que precisam ser concluídas, ou em recursos específicos que os usuários têm permissão para usar.

A tabela a seguir descreve as ações da API do Amazon MQ que atualmente oferecem suporte a permissões em nível de recurso, bem como os recursos ARNs, recursos e chaves de condição suportados para cada ação.

> **Important**
>
> Caso uma ação de API do Amazon MQ não esteja listada nessa tabela, isso significa que ela não é compatível com as permissões no nível do recurso. Se uma ação da API do Amazon MQ não for compatível com as permissões em nível de recurso, você poderá conceder aos usuários permissão para usar a ação, mas precisará especificar um curinga * para o elemento do recurso da declaração de política.

Ação API	Tipos de recursos (*necessários)
CreateConfiguration	configurações*
CreateTags	agentes, configurações
CreateUser	operadores*
DeleteBroker	operadores*
DeleteUser	operadores*
DescribeBroker	operadores*
DescribeConfiguration	configurações*
DescribeConfigurationRevision	configurações*
DescribeUser	operadores*
ListConfigurationRevisions	configurações*

Ação API	Tipos de recursos (*necessários)
ListConfigurationRevisions	configurações*
ListTags	agentes, configurações
ListUsers	operadores*
RebootBroker	operadores*
UpdateBroker	operadores*
UpdateConfiguration	configurações*
UpdateUser	operadores*

Autenticação e autorização do corretor

O Amazon MQ fornece diferentes métodos de autenticação e autorização, dependendo do tipo de mecanismo de sua corretora.

Autenticação e autorização para Amazon MQ for ActiveMQ

O Amazon MQ para ActiveMQ suporta os seguintes métodos de autenticação e autorização:

Autorização e autenticação simples

Nesse método, os usuários do broker são criados e gerenciados por meio do console ou da API do Amazon MQ. Os usuários podem ser configurados com permissões específicas para acessar filas, tópicos e o ActiveMQ Web Console. Para obter mais informações sobre esse método, consulte Criando um usuário do ActiveMQ broker.

Autenticação e autorização LDAP

Nesse método, os usuários do broker se autenticam por meio de credenciais armazenadas em seu servidor LDAP. Você pode adicionar, excluir e modificar usuários e atribuir permissões a tópicos e filas por meio do servidor LDAP, fornecendo autenticação e autorização centralizadas. Para obter mais informações sobre esse método, consulte Integrando os corretores ActiveMQ com o LDAP.

Autenticação e autorização do Amazon MQ for RabbitMQ

O Amazon MQ para RabbitMQ oferece suporte aos seguintes métodos de autenticação e autorização:

Autorização e autenticação simples

Nesse método, os usuários do agente são armazenados internamente no agente do RabbitMQ e gerenciados por meio do console Web ou da API de gerenciamento. As permissões para vhosts, trocas, filas e tópicos são configuradas diretamente no RabbitMQ. Esse é o método padrão. Para obter mais informações, consulte Autenticação e autorização simples.

OAuth 2.0 autenticação e autorização

Nesse método, os usuários do broker e suas permissões são gerenciados por um provedor de identidade (IdP) externo OAuth 2.0. A autenticação do usuário e as permissões de recursos para vhosts, trocas, filas e tópicos são centralizadas por meio do sistema de escopo do provedor OAuth 2.0. Isso simplifica o gerenciamento de usuários e permite a integração com os sistemas de identidade existentes. Para obter mais informações, consulte Autenticação e autorização OAuth 2.0.

Autenticação e autorização do IAM

Nesse método, os usuários do broker se autenticam usando credenciais AWS do IAM por meio da federação de saída do IAM. As credenciais do IAM são usadas para obter tokens JWT do AWS Security Token Service (STS), e esses tokens JWT servem como tokens OAuth 2.0 para autenticação. Esse método aproveita o suporte OAuth 2.0 existente no Amazon MQ para RabbitMQ, AWS onde atua como o provedor de identidade 2.0. OAuth A autenticação do usuário é gerenciada pelo AWS IAM, enquanto as permissões de recursos para vhosts, trocas, filas e tópicos são gerenciadas por meio de políticas do IAM e aliases de escopo configurados no RabbitMQ. Para obter mais informações, consulte Autenticação e autorização do IAM.

Autenticação e autorização LDAP

Nesse método, os usuários do broker e suas permissões são gerenciados por um serviço de diretório LDAP externo. A autenticação do usuário e as permissões de recursos são centralizadas por meio do servidor LDAP, permitindo que os usuários acessem o RabbitMQ usando suas credenciais de serviço de diretório existentes. Para obter mais informações, consulte Autenticação e autorização LDAP.

Autenticação e autorização HTTP

Nesse método, os usuários do broker e suas permissões são gerenciados por um servidor HTTP externo. A autenticação do usuário e as permissões de recursos são centralizadas por meio do servidor HTTP, permitindo que os usuários acessem o RabbitMQ usando seu próprio provedor de autenticação e autorização. Para obter mais informações sobre esse método, consulte Autenticação e autorização HTTP.

Autenticação de certificado SSL

O Amazon MQ oferece suporte a TLS mútuo (mTLS) para corretores RabbitMQ. O plug-in de autenticação SSL usa certificados de cliente de conexões mTLS para autenticar usuários. Nesse método, os usuários do broker são autenticados usando certificados de cliente X.509 em vez de credenciais de nome de usuário e senha. O certificado do cliente é validado em relação a uma Autoridade Certificadora (CA) confiável e o nome de usuário é extraído de um campo no certificado, como Nome comum (CN) ou Nome alternativo do assunto (SAN). Esse método fornece autenticação forte sem transmitir credenciais pela rede. Para obter mais informações, consulte Autenticação de certificado SSL.

> **Note**
>
> O RabbitMQ suporta vários métodos de autenticação e autorização para serem usados simultaneamente. Por exemplo, você pode ativar a autenticação OAuth 2.0 e a autenticação simples (interna). Para obter mais informações, consulte a seção do tutorial OAuth 2.0 sobre como habilitar a autenticação OAuth 2.0 e simples (interna) e a documentação de controle de acesso do RabbitMQ.
>
> O Amazon MQ recomenda criar um usuário interno ao testar as configurações de autenticação. Isso permite que a configuração de acesso seja validada usando a API de gerenciamento do RabbitMQ. Para obter mais informações, consulte Validação de acesso.

AWS políticas gerenciadas para o Amazon MQ

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque elas estão disponíveis para uso de todos os

AWS clientes. Recomendamos que você reduza ainda mais as permissões definindo as políticas gerenciadas pelo cliente que são específicas para seus casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) for lançada ou novas operações de API forem disponibilizadas para serviços existentes.

Para saber mais, consulte AWS Políticas gerenciadas pela no Guia do usuário do IAM.

O Amazon MQ oferece suporte às seguintes políticas AWS gerenciadas:

- AmazonMQApiFullAccess
- AmazonMQApiReadOnlyAccess
- Amazon MQFull Access
- AmazonMQReadOnlyAccess
- AmazonMQServiceRolePolicy

AWS política gerenciada: Amazon MQService RolePolicy

Não é possível anexar `AmazonMQServiceRolePolicy` às suas entidades do IAM. Essa política é anexada a uma função vinculada ao serviço que permite que o Amazon MQ realize ações em seu nome. Para obter mais informações sobre essa política de permissão e as ações que ela permite que o Amazon MQ execute, consulte the section called “Permissões de função vinculada ao serviço para o Amazon MQ”.

Atualizações do Amazon MQ para AWS políticas gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do Amazon MQ desde que esse serviço começou a monitorar essas alterações. Para alertas automáticos sobre mudanças nesta página, assine o RSS feed na página de histórico de documentos do Amazon MQ.

Alteração	Descrição	Data
O Amazon MQ passou a monitorar as alterações	O Amazon MQ começou a monitorar as alterações em suas políticas AWS gerenciadas.	5 de maio de 2021

Uso de funções vinculadas ao serviço para o Amazon MQ

O Amazon MQ usa funções vinculadas a serviços AWS Identity and Access Management (IAM). A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Amazon MQ. As funções vinculadas ao serviço são predefinidas pelo Amazon MQ e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Uma função vinculada ao serviço facilita a configuração do Amazon MQ porque você não precisa adicionar as permissões necessárias manualmente. O Amazon MQ define as permissões das funções vinculadas ao serviço e, exceto se definido de outra forma, somente o Amazon MQ pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus recursos do Amazon MQ, pois você não pode remover por engano as permissões para acessar os recursos.

Para obter informações sobre outros produtos que são compatíveis com às funções vinculadas a serviços, consulte Serviços da AWS compatíveis com o IAM e procure os serviços que contêm Yes (Sim) na coluna Service-Linked Role (Função vinculada a serviço). Escolha um Sim com um link para visualizar a documentação do perfil vinculado para esse serviço.

Permissões de função vinculada ao serviço para o Amazon MQ

O Amazon MQ usa a função vinculada ao serviço chamada MQ AWSServiceRoleForAmazon— O Amazon MQ usa essa função vinculada ao serviço para chamar serviços em seu nome. AWS

A função vinculada ao serviço AWSService RoleForAmazon MQ confia nos seguintes serviços para assumir a função:

- `mq.amazonaws.com`

O Amazon MQ usa a política de permissão `AmazonMQServiceRolePolicy`, que é anexada à função vinculada ao serviço AWSService RoleForAmazon MQ, para concluir as seguintes ações nos recursos especificados:

- Ação: `ec2:CreateVpcEndpoint` no recurso `vpc`.
- Ação: `ec2:CreateVpcEndpoint` no recurso `subnet`.
- Ação: `ec2:CreateVpcEndpoint` no recurso `security-group`.
- Ação: `ec2:CreateVpcEndpoint` no recurso `vpc-endpoint`.
- Ação: `ec2:DescribeVpcEndpoints` no recurso `vpc`.
- Ação: `ec2:DescribeVpcEndpoints` no recurso `subnet`.
- Ação: `ec2:CreateTags` no recurso `vpc-endpoint`.
- Ação: `logs:PutLogEvents` no recurso `log-group`.
- Ação: `logs:DescribeLogStreams` no recurso `log-group`.
- Ação: `logs:DescribeLogGroups` no recurso `log-group`.
- Ação: `CreateLogStream` no recurso `log-group`.
- Ação: `CreateLogGroup` no recurso `log-group`.

Quando você cria um Amazon MQ para agente RabbitMQ, a política de permissão `AmazonMQServiceRolePolicy` do Amazon MQ realize as seguintes tarefas em seu nome.

- Cria um endpoint da Amazon VPC para o agente usando o Amazon VPC, a sub-rede e o grupo de segurança que você fornece. Você pode usar o endpoint criado para que seu agente se conecte ao agente por meio do console de gerenciamento RabbitMQ, da API de gerenciamento ou de forma programática.
- Crie grupos de registros e publique registros de agentes no 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/*"
            ]
        }
    ]
}
```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado a serviço. Para saber mais, consulte Permissões de Função Vinculadas ao Serviço no Guia do Usuário do IAM.

Criação de uma função vinculada ao serviço para Amazon MQ

Não é necessário criar manualmente um perfil vinculado ao serviço. Quando você cria um agente pela primeira vez, o Amazon MQ cria uma função vinculada a serviços para chamar AWS serviços

em seu nome. Todos os agentes subsequentes que você criar usarão a mesma função e nenhuma nova função será criada.

> **Important**
>
> Esse perfil vinculado ao serviço pode aparecer em sua conta se você concluiu uma ação em outro serviço que usa os atributos compatíveis com esse perfil. Para saber mais, consulte Uma Nova Função Apareceu na minha Conta do IAM.

Se excluir essa função vinculada ao serviço e precisar criá-la novamente, você pode usar esse mesmo processo para recriar a função na sua conta.

Você também pode usar o console do IAM para criar uma função vinculada ao serviço com o caso de uso do Amazon MQ. Na AWS CLI ou na AWS API, crie uma função vinculada ao serviço com o nome do `mq.amazonaws.com` serviço. Para saber mais, consulte Criar um perfil vinculado a serviço no Guia do usuário do IAM. Se você excluir essa função vinculada ao serviço, será possível usar esse mesmo processo para criar a função novamente.

> **Important**
>
> As funções vinculadas ao serviço são criadas somente para o Amazon MQ para RabbitMQ.

Edição de uma função vinculada ao serviço do Amazon MQ

O Amazon MQ não permite que você edite a função vinculada ao serviço AWSService RoleForAmazon MQ. No entanto, você poderá editar a descrição do perfil usando o IAM. Para saber mais, consulte Editar um perfil vinculado ao serviço no Guia do usuário do IAM.

Exclusão de uma função vinculada ao serviço do Amazon MQ

Se você não precisar mais usar um recurso ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. No entanto, você deve limpar os recursos de seu perfil vinculado ao serviço antes de excluí-lo manualmente.

> **Note**
>
> Se o serviço do Amazon MQ estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente.

Para excluir recursos do Amazon MQ usados pelo MQ AWSService RoleForAmazon

- Exclua seus corretores do Amazon MQ usando a CLI do Console de gerenciamento da AWS Amazon MQ ou a API do Amazon MQ. Para obter mais informações sobre como excluir um agente, consulte ???.

Como excluir manualmente o perfil vinculado ao serviço usando o IAM

Use o console do IAM AWS CLI, o ou a AWS API para excluir a função vinculada ao serviço AWSService RoleForAmazon MQ. Para saber mais, consulte Excluir um perfil vinculado ao serviço no Guia do usuário do IAM.

Regiões compatíveis com as funções vinculadas a serviços do Amazon MQ

O Amazon MQ é compatível com as funções vinculadas a serviços em todas as regiões em que o serviço está disponível. Para mais informações, consulte Regiões e endpoints da AWS.

Nome da região	Identidade da região	Compatível com o Amazon MQ
Leste dos EUA (Norte da Virgínia)	us-east-1	Sim
Leste dos EUA (Ohio)	us-east-2	Sim
Oeste dos EUA (N. da Califórnia)	us-west-1	Sim
Oeste dos EUA (Oregon)	us-west-2	Sim
Ásia-Pacífico (Mumbai)	ap-south-1	Sim
Ásia Pacifico (Osaka)	ap-northeast-3	Sim

Nome da região	Identidade da região	Compatível com o Amazon MQ
Ásia-Pacífico (Seul)	ap-northeast-2	Sim
Ásia-Pacífico (Singapura)	ap-southeast-1	Sim
Ásia-Pacífico (Sydney)	ap-southeast-2	Sim
Ásia-Pacífico (Tóquio)	ap-northeast-1	Sim
Canadá (Central)	ca-central-1	Sim
Europa (Frankfurt)	eu-central-1	Sim
Europa (Irlanda)	eu-west-1	Sim
Europa (Londres)	eu-west-2	Sim
Europa (Paris)	eu-west-3	Sim
América do Sul (São Paulo)	sa-east-1	Sim
AWS GovCloud (US)	us-gov-west-1	Não

Solução de problemas de identidade e acesso da Amazon MQ

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que você possa encontrar ao trabalhar com a Amazon MQ e o IAM.

Tópicos

- Não tenho autorização para executar uma ação na Amazon MQ
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha AWS conta acessem meus recursos do Amazon MQ

Não tenho autorização para executar uma ação na Amazon MQ

Se isso Console de gerenciamento da AWS indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. Caso seu administrador seja a pessoa que forneceu suas credenciais de início de sessão.

O exemplo de erro a seguir ocorre quando o `mateojackson` usuário tenta usar o console para ver detalhes sobre um*`widget`*, mas não tem `mq:`*`GetWidget`* permissões.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mq:GetWidget on resource: my-example-widget
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso *`my-example-widget`* usando a ação `mq:`*`GetWidget`*.

Não estou autorizado a realizar iam: PassRole

Se receber uma mensagem de erro informando que você não tem autorização para executar a ação `iam:PassRole`, suas políticas devem ser atualizadas para permitir a transmissão de um perfil ao Amazon MQ.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro exemplificado a seguir ocorre quando um usuário do IAM chamado `marymajor` tenta usar o console para executar uma ação na Amazon MQ. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha AWS conta acessem meus recursos do Amazon MQ

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amazon MQ é compatível com esses recursos, consulte Como o Amazon MQ funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Validação de conformidade para o Amazon MQ

Auditores terceirizados avaliam a segurança e a conformidade do Amazon MQ como parte de AWS vários programas de conformidade. Isso inclui SOC, PCI, HIPAA e outros.

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis

e regulamentações aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a documentação AWS de segurança.

Resiliência no Amazon MQ

A infraestrutura global da AWS é criada com base em Regiões e Zonas de Disponibilidade AWS da AWS. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, conectadas com baixa latência, throughput elevado e redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre regiões e zonas de disponibilidade da AWS, consulte Infraestrutura global da AWS.

Segurança da infraestrutura no Amazon MQ

Como um serviço gerenciado, o Amazon MQ é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte AWS Cloud Security. Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte Proteção de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar o Amazon MQ pela rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Práticas recomendadas de segurança para o Amazon MQ

Os padrões de design a seguir podem melhorar a segurança de seu agente do Amazon MQ.

Tópicos

- Preferir agentes sem acessibilidade pública
- Sempre configurar um mapa de autorização
- Bloquear protocolos desnecessários com os grupos de segurança da VPC

Para obter mais informações sobre como o Amazon MQ criptografa seus dados, bem como uma lista de protocolos compatíveis, consulte Proteção de dados.

Preferir agentes sem acessibilidade pública

Agentes criados sem acessibilidade pública não podem ser acessados de fora de sua VPC. Isso reduz muito a suscetibilidade do seu corretor a ataques distribuídos de negação de serviço (DDoS) da Internet pública. Para obter mais informações, consulte Como ajudar a se preparar para ataques DDo S reduzindo sua superfície de ataque no blog AWS de segurança.

Sempre configurar um mapa de autorização

Como o ActiveMQ não tem uma mapa de autorização configurado por padrão, qualquer usuário autenticado pode executar qualquer ação no agente. Portanto, uma prática recomendada é restringir as permissões por grupo. Para obter mais informações, consulte `authorizationEntry`.

> **Important**
>
> Se você especificar um mapa de autorização que não inclua o `activemq-webconsole`, você não poderá usar o Console da Web do ActiveMQ porque o grupo não estará autorizado a enviar mensagens ou receber mensagens do agente do Amazon MQ.

Bloquear protocolos desnecessários com os grupos de segurança da VPC

Para aprimorar a segurança dos agentes privados, você deve restringir as conexões de protocolos desnecessários configurando adequadamente o grupo de segurança do Amazon VPC. Por exemplo, para restringir o acesso à maioria dos protocolos OpenWire e, ao mesmo tempo, permitir o acesso ao console web, você pode permitir o acesso somente a 61617 e 8162. Isso limita sua exposição bloqueando protocolos que você não está usando OpenWire e, ao mesmo tempo, permitindo que o console web funcione normalmente.

Permita somente as portas de protocolos que estão sendo usados.

- AMQP: 5671
- MQTT: 8883
- OpenWire: 61617
- STOMP: 61614
- WebSocket: 61619

Para obter mais informações, consulte:

- Grupos de segurança para sua VPC
- Grupo de segurança padrão para sua VPC
- Como trabalhar com grupos de segurança

Monitoramento e registro em agentes do Amazon MQ

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de suas AWS soluções. Você deve coletar dados de monitoramento de todas as partes da sua AWS solução para poder depurar com mais facilidade uma falha multiponto, caso ocorra. AWS fornece várias ferramentas para monitorar seus recursos do Amazon MQ e responder a possíveis incidentes:

Você pode usar CloudWatch para visualizar e analisar métricas para seu corretor Amazon MQ. Você pode visualizar e analisar as métricas do seu corretor no CloudWatch console AWS CLI, no ou no CloudWatch AWS CLI. CloudWatch as métricas do Amazon MQ são automaticamente pesquisadas pela corretora e, em seguida, enviadas para CloudWatch cada minuto. Para corretores ActiveMQ CloudWatch , monitora somente os primeiros 1000 destinos. Para os corretores RabbitMQ, CloudWatch monitora apenas os primeiros 500 destinos, ordenados por número de consumidores.

Para obter uma lista completa das métricas do Amazon MQ, consulte CloudWatch Métricas disponíveis Amazon MQ para corretores ActiveMQ.

Para obter informações sobre como criar um CloudWatch alarme para uma métrica, consulte Criar ou editar um CloudWatch alarme no Guia CloudWatch do usuário da Amazon.

Acessando CloudWatch métricas para o Amazon MQ

Você pode acessar CloudWatch as métricas usando a API Console de gerenciamento da AWS AWS CLI, e.

Talvez você queira acessar CloudWatch as métricas sem usar Console de gerenciamento da AWS o.

Para acessar as métricas do Amazon MQ usando o AWS CLI, use o `get-metric-statistics` comando. Para obter mais informações, consulte Obter estatísticas de uma métrica no Guia CloudWatch do usuário da Amazon.

Para acessar as métricas do Amazon MQ usando a CloudWatch API, use a `GetMetricStatistics` ação. Para obter mais informações, consulte Obter estatísticas de uma métrica no Guia CloudWatch do usuário da Amazon.

Acessando CloudWatch métricas usando o Console de gerenciamento da AWS

O exemplo a seguir mostra como acessar CloudWatch métricas do Amazon MQ usando Console de gerenciamento da AWS o.Se você já estiver conectado ao console do Amazon MQ, na página de detalhes do agente, escolha Ações, Visualizar métricas. CloudWatch

1. Faça login no console do CloudWatch.
2. No painel de navegação, selecione Métricas.
3. Selecione o namespace de métrica do AmazonMQ.
4. Selecione uma das seguintes dimensões de métricas:

   - Broker Metrics (Métricas do agente)
   - Métricas de fila por operador
   - Métricas de tópico por operador

   Neste exemplo, está selecionado Broker Metrics (Métricas do operador).
5. Agora você pode examinar as métricas do Amazon MQ:

   - Para classificar a métrica, use o cabeçalho da coluna.
   - Para criar o gráfico de uma métrica, marque a caixa de seleção ao lado da métrica.
   - Para filtrar por métrica, selecione o nome da métrica e, em seguida, escolha Adicionar à pesquisa.

CloudWatch Métricas disponíveis Amazon MQ para corretores ActiveMQ

Métricas do Amazon MQ para ActiveMQ

Métrica	Unidade	Description
`AmqpMaximumConnections`	Contagem	O número máximo de clientes que podem ser conectados ao

Métrica	Unidade	Description
		seu agente via AMQP. Para obter mais informações sobre cotas de conexão, consulte Quotas in Amazon MQ.
`BurstBalance`	Percentual	A porcentagem de créditos de intermitência restantes no volume do Amazon EBS usada para persistir dados de mensagens para agentes otimizados para taxa de transferência. Se esse saldo atingir zero, as IOPS fornecidas pelo volume do Amazon EBS diminuirão até que o Saldo de intermitência seja reabastecido. Para obter mais informações sobre como funcionam os Saldos de intermitência no Amazon EBS, consulte: Créditos de E/S e performance de intermitência.

Métrica	Unidade	Description
`CpuCreditBalance`	Créditos (minutos de vCPU)	**Important** Essa métrica está disponível somente para os tipos de instância de agente `mq.t2.micro` . As métricas de crédito de CPU estão disponíveis somente em intervalos de cinco minutos. O número de créditos ganhos de CPU que uma instância acumulou desde que foi executada ou iniciada (incluindo o número de créditos de execução). O saldo de créditos são disponibilizados para que a instância do agente gaste em intermitências com uma utilização de CPU acima da linha de base. Os créditos são acumulados no saldo de créditos após terem sido ganhos e são removidos do saldo de créditos após serem gastos. O saldo de créditos tem um limite máximo. Depois que o limite é atingido, todos os créditos ganhos mais

Métrica	Unidade	Description
		recentemente são descartados.
`CpuUtilization`	Percentual	O percentual de unidades de processamento do Amazon EC2 alocadas que o agente utiliza no momento.
`CurrentConnectionsCount`	Contagem	O número atual de conexões ativas no agente atual.
`EstablishedConnectionsCount`	Contagem	O número total de conexões, ativas e inativas, que foram estabelecidas com o operador.
`HeapUsage`	Percentual	A porcentagem do limite de memória do ActiveMQ JVM que o agente usa atualmente.
`InactiveDurableTopicSubscribersCount`	Contagem	O número de inscritos inativos no tópico durável, até um máximo de 2000.
`JobSchedulerStorePercentUsage`	Percentual	A porcentagem de espaço em disco usada pelo armazenamento do agendador de tarefas.
`JournalFilesForFastRecovery`	Contagem	O número de arquivos de diário que serão reproduzidos novamente após um desligamento normal.
`JournalFilesForFullRecovery`	Contagem	O número de arquivos de diário que serão reproduzidos novamente após um desligamento inesperado.

Métrica	Unidade	Description
`MqttMaximumConnections`	Contagem	O número máximo de clientes que podem ser conectados ao seu agente via MQTT. Para obter mais informações sobre cotas de conexão, consulte Quotas in Amazon MQ.
`NetworkConnectorConnectionCount`	Contagem	O número de nós conectados ao corretor em uma rede de corretores usando NetworkConnector.
`NetworkIn`	Bytes	O volume de tráfego de entrada para o operador.
`NetworkOut`	Bytes	O volume de tráfego de saída para o operador.
`OpenTransactionCount`	Contagem	O número total de transações em andamento.
`OpenwireMaximumConnections`	Contagem	O número máximo de clientes que você pode conectar à sua corretora usando OpenWire. Para obter mais informações sobre cotas de conexão, consulte Quotas in Amazon MQ.
`StompMaximumConnections`	Contagem	O número máximo de clientes que podem ser conectados ao seu agente via STOMP. Para obter mais informações sobre cotas de conexão, consulte Quotas in Amazon MQ.

Métrica	Unidade	Description
`StorePercentUsage`	Percentual	A porcentagem usada pelo limite de armazenamento. Se ela chegar a 100, o agente recusará mensagens.
`TempPercentUsage`	Percentual	A porcentagem de armazenamento temporári o disponível usada por mensagens não persistentes.
`TotalConsumerCount`	Contagem	O número de consumidores de mensagens inscritos em destinos no agente atual.
`TotalMessageCount`	Contagem	O número de mensagens armazenadas no operador.
`TotalProducerCount`	Contagem	O número de produtores de mensagens ativos em destinos no agente atual.
`VolumeReadOps`	Contagem	O número de operações de leitura executadas no volume do Amazon EBS.
`VolumeWriteOps`	Contagem	O número de operações de gravação executadas no volume do Amazon EBS.
`WsMaximumConnections`	Contagem	O número máximo de clientes que você pode conectar à sua corretora usando WebSocket . Para obter mais informaçõ es sobre cotas de conexão, consulte Quotas in Amazon MQ.

Dimensões para métricas do agente ActiveMQ

Dimensão	Description
`Broker`	O nome do agente **Note** Um agente de instância única tem o sufixo -1. Um active/standby corretor de alta disponibilidade tem os sufixos -1 e -2 para seu par redundante.

Métricas de destino do ActiveMQ (fila e tópico)

Important

As métricas a seguir incluem contagens por minuto para o período da CloudWatch pesquisa.

- `EnqueueCount`
- `ExpiredCount`
- `DequeueCount`
- `DispatchCount`
- `InFlightCount`

Por exemplo, em um período do CloudWatch de cinco minutos, `EnqueueCount` tem cinco valores de contagem, cada um para uma parte de um minuto do período. As estatísticas `Minimum` e `Maximum` fornecem o valor mais baixo e mais alto por minuto durante o período especificado.

Métrica	Unidade	Description
`ConsumerCount`	Contagem	O número de consumidores que se inscreveram para o destino.
`EnqueueCount`	Contagem	O número de mensagens enviadas ao destino por minuto.
`EnqueueTime`	Tempo (milissegundos)	A end-to-end latência desde o momento em que uma mensagem chega a uma corretora até ser entregue ao consumidor. **Note** `EnqueueTime` não mede a end-to-end latência de quando uma mensagem é enviada por um produtor até chegar ao corretor, nem a latência de quando uma mensagem é recebida por um corretor até ser confirmada pelo corretor. Em vez disso, `EnqueueTime` é o número de milissegundos a partir do momento em que uma mensagem é recebida pelo agente

Métrica	Unidade	Description
		até ser entregue com sucesso a um consumidor.
`ExpiredCount`	Contagem	O número de mensagens que não puderam ser entregues porque expiraram, por minuto.
`DispatchCount`	Contagem	O número de mensagens enviadas a consumidores por minuto.
`DequeueCount`	Contagem	O número de mensagens confirmadas por consumidores por minuto.
`InFlightCount`	Contagem	O número de mensagens enviadas para os consumidores que não foram reconhecidos.
`ReceiveCount`	Contagem	O número de mensagens que foram recebidas do agente remoto por um conector de rede duplex.
`MemoryUsage`	Percentual	A porcentagem do limite de memória que o destino usa atualmente.
`ProducerCount`	Contagem	O número de produtores para o destino.

Métrica	Unidade	Description
`QueueSize`	Contagem	O número de mensagens na fila. **Important** Esta métrica aplica-se apenas às filas.
`TotalEnqueueCount`	Contagem	O número total de mensagens que foram enviadas para o agente.
`TotalDequeueCount`	Contagem	O número total de mensagens que foram consumidas pelos clientes.

Note

As métricas `TotalEnqueueCount` e `TotalDequeueCount` incluem mensagens para tópicos de aviso. Para obter mais informações sobre mensagens de tópico de aviso, consulte a documentação do ActiveMQ.

Dimensões para métricas de destino do ActiveMQ (fila e tópico)

Dimensão	Description
`Broker`	O nome do operador. **Note** Um agente de instância única tem o sufixo `-1`. Um active/standby corretor

Dimensão	Description
	de alta disponibilidade tem os sufixos `-1` e seu `-2` par redundante.
`Topic` ou `Queue`	O nome do tópico ou da fila.
`NetworkConnector`	O nome do conector de rede.

CloudWatch Métricas disponíveis para Amazon MQ para corretores RabbitMQ

Métricas do agente RabbitMQ

Métrica	Unidade	Description
`ExchangeCount`	Contagem	O número total de trocas configuradas com o agente.
`QueueCount`	Contagem	O número total de filas configuradas com o agente.
`ConnectionCount`	Contagem	O número total de conexões estabelecidas com o agente.
`ChannelCount`	Contagem	O número total de canais estabelecidas com o agente.
`ConsumerCount`	Contagem	O número total de consumido res conectados com o agente.
`MessageCount`	Contagem	O número de mensagens nas filas.

Métrica	Unidade	Description
		Note O número produzido é a soma total de mensagens prontas e não reconhecidas no agente.
`MessageReadyCount`	Contagem	O número total de mensagens prontas nas filas.
`MessageUnacknowledgedCount`	Contagem	O número total de mensagens não reconhecidas nas filas.
`PublishRate`	Contagem	A taxa na qual as mensagens são publicadas para o agente. O número produzido represent a o número de mensagens por segundo no momento da amostragem.
`ConfirmRate`	Contagem	A taxa com a qual o servidor RabbitMQ está confirmando mensagens publicadas. Você pode comparar essa métrica com `PublishRate` para entender melhor a performan ce do seu agente. O número produzido represent a o número de mensagens por segundo no momento da amostragem.

Métrica	Unidade	Description
`AckRate`	Contagem	A taxa em que as mensagens estão sendo reconhecidas pelos consumidores. O número produzido represent a o número de mensagens por segundo no momento da amostragem.
`SystemCpuUtilization`	Percentual	O percentual de unidades de processamento do Amazon EC2 alocadas que o agente utiliza no momento. Para implantações de cluster, esse valor representa o agregado dos três valores métricos correspondentes dos três nós RabbitMQ.
`RabbitMQMemLimit`	Bytes	O limite de RAM para um agente RabbitMQ. Para implantações de cluster, esse valor representa o agregado dos três valores métricos correspondentes dos três nós RabbitMQ.
`RabbitMQMemUsed`	Bytes	O volume de RAM usado por um agente RabbitMQ. Para implantações de cluster, esse valor representa o agregado dos três valores métricos correspondentes dos três nós RabbitMQ.

Métrica	Unidade	Description
`RabbitMQDiskFreeLimit`	Bytes	O limite de disco para um agente RabbitMQ. Para implantações de cluster, esse valor representa o agregado dos três valores métricos correspondentes dos três nós RabbitMQ. Esta métrica é diferente por tamanho de instância.
`RabbitMQDiskFree`	Bytes	O volume total de espaço livre em disco disponível em um agente RabbitMQ. Quando o uso do disco ultrapassa seu limite, o cluster bloqueará todas as conexões do produtor. Para implantações de cluster, esse valor representa o agregado dos três valores métricos correspondentes dos três nós RabbitMQ.
`RabbitMQFdUsed`	Contagem	Número de descritores de arquivos usados. Para implantações de cluster, esse valor representa o agregado dos três valores métricos correspondentes dos três nós RabbitMQ.

Métrica	Unidade	Description
`RabbitMQIOReadAverageTime`	Contagem	O tempo médio (em milissegundos) para o RabbitMQ realizar uma operação de leitura. O valor é proporcional ao tamanho da mensagem.
`RabbitMQIOWriteAverageTime`	Contagem	O tempo médio (em milissegundos) para o RabbitMQ realizar uma operação de gravação. O valor é proporcional ao tamanho da mensagem.

Dimensões para métricas de agente RabbitMQ

Dimensão	Description
`Broker`	O nome do operador.

Métricas do nó RabbitMQ

Métrica	Unidade	Description
`SystemCpuUtilization`	Percentual	O percentual de unidades de processamento do Amazon EC2 alocadas que o agente utiliza no momento.
`RabbitMQMemLimit`	Bytes	O limite de RAM para um nó RabbitMQ.
`RabbitMQMemUsed`	Bytes	O volume de RAM usado por um nó RabbitMQ. Quando o uso da memória ultrapassa

Métrica	Unidade	Description
		o limite, o cluster bloqueará todas as conexões do produtor.
`RabbitMQDiskFreeLimit`	Bytes	O limite de disco para um nó RabbitMQ. Esta métrica é diferente por tamanho de instância.
`RabbitMQDiskFree`	Bytes	O volume total de espaço livre em disco disponível em um nó RabbitMQ. Quando o uso do disco ultrapassa seu limite, o cluster bloqueará todas as conexões do produtor.
`RabbitMQFdUsed`	Contagem	Número de descritores de arquivos usados.

Dimensões para métricas de nó RabbitMQ

Dimensão	Description
`Node`	Nome do nó. **Note** Um nome de nó consiste em duas partes: um prefixo (normalmente `rabbit`) e um nome de host. Por exemplo, `rabbit@ip-10-0-0-230.us-west-2.compute.internal` é um nome de nó com o prefixo `rabbit` e o nome

Dimensão	Description
	do host `ip-10-0-0-230.us-west-2.compute.internal` .
`Broker`	O nome do operador.

Métricas de fila RabbitMQ

Métrica	Unidade	Description
`ConsumerCount`	Contagem	O número de consumidores que se inscreveram para a fila.
`MessageReadyCount`	Contagem	O número de mensagens que estão atualmente disponíveis para serem entregues.
`MessageUnacknowledgedCount`	Contagem	O número de mensagens para as quais o servidor está aguardando confirmação.
`MessageCount`	Contagem	O número total de `MessageReadyCount` e `MessageUnacknowledgedCount` (também conhecida como profundidade de fila).

Dimensões para métricas de fila RabbitMQ

> **Note**
>
> O Amazon MQ para RabbitMQ não publicará métricas para hosts virtuais e filas com nomes que contenham espaços em branco, guias ou outros caracteres não ASCII.

Para obter mais informações sobre nomes de dimensões, consulte Dimension na Amazon CloudWatch API Reference.

Dimensão	Description
`Queue`	O nome da fila do .
`VirtualHost`	O nome do host virtual.
`Broker`	O nome do operador.

Métricas de rede do RabbitMQ

Métrica	Unidade	Description
`NetworkOut`	Bytes	A quantidade de bytes enviados em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de rede de saída de uma única instância. O número relatado é o número de bytes enviados durante o período. Se você estiver usando o monitoramento básico (5 minutos) e a estatística for Sum (Soma), divida esse número por 300 para encontrar o número de bytes/segundo. Se você tiver o monitoramento detalhado (1 minuto) e a estatística for Sum (soma), divida o número por 60. Você também pode usar a função matemática CloudWatch métrica `DIFF_TIME` para encontrar os bytes por segundo. Por exemplo, se você tiver representado graficamente NetworkOut CloudWatch comom1, a fórmula matemática métrica `m1/(DIFF_TIME(m1))` retornará a métrica em bytes/segundo. Para obter mais informações sobre `DIFF_TIME` e outras as funções matemáticas de métrica consulte Uso de matemática de métrica. Estatísticas significativas: Soma, Média, Mínimo, Máximo

Métrica	Unidade	Description
`NetworkIn`	Bytes	A quantidade de bytes recebidos em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de rede de entrada para uma única instância. O número relatado é o número de bytes recebidos durante o período. Se você estiver usando o monitoramento básico (5 minutos) e a estatística for Sum (Soma), divida esse número por 300 para encontrar o número de bytes/seg undo. Se você tiver o monitoramento detalhado (1 minuto) e a estatística for Sum (soma), divida o número por 60. Você também pode usar a função matemática CloudWatch métrica `DIFF_TIME` para encontrar os bytes por segundo. Por exemplo, se você tiver representado graficamente NetworkIn CloudWatch com`m1`, a fórmula matemática métrica `m1/(DIFF_TIME(m1))` retornará a métrica em bytes/segundo. Para obter mais informações sobre `DIFF_TIME` e outras as funções matemáticas de métrica consulte Uso de matemática de métrica. Estatísticas significativas: Soma, Média, Mínimo, Máximo

Dimensões para agentes do RabbitMQ

Dimensão	Description
`BrokerId`	ID do agente

Configurar logs do Amazon MQ for RabbitMQ

Quando você ativa o CloudWatch registro em log para seus corretores RabbitMQ, o Amazon MQ usa uma função vinculada ao serviço para publicar registros gerais. CloudWatch Se nenhuma função vinculada ao serviço do Amazon MQ existir quando você criar um agente pela primeira vez, o Amazon MQ criará um agente automaticamente. Todos os corretores subsequentes do RabbitMQ usarão a mesma função vinculada ao serviço para publicar registros. CloudWatch

Para obter mais informações sobre perfis vinculados ao serviço, consulte Using service-linked roles no Guia do usuário do AWS Identity and Access Management . Para obter mais informações sobre o Amazon MQ usa as funções vinculadas ao serviço, consulte the section called “Uso de perfis vinculados ao serviço”.

Registro de chamadas de API do Amazon MQ usando AWS CloudTrail

O Amazon MQ é integrado com AWS CloudTrail, um serviço que fornece um registro das chamadas do Amazon MQ que um usuário, AWS função ou serviço faz. CloudTrail captura chamadas de API relacionadas aos agentes e configurações do Amazon MQ como eventos, incluindo chamadas do console do Amazon MQ e chamadas de código do Amazon MQ. APIs Para obter mais informações sobre CloudTrail, consulte o Guia AWS CloudTrail do usuário.

> **Note**
>
> CloudTrail não registra chamadas de API relacionadas às operações do ActiveMQ (por exemplo, envio e recebimento de mensagens) ou ao ActiveMQ Web Console. Para registrar informações relacionadas às operações do ActiveMQ, você pode configurar o Amazon MQ para publicar registros gerais e de auditoria no Amazon Logs. CloudWatch

Usando as informações CloudTrail coletadas, você pode identificar uma solicitação específica para uma API do Amazon MQ, o endereço IP do solicitante, a identidade do solicitante, a data e a hora da solicitação e assim por diante. Se você configurar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3. Se você não configurar uma trilha, poderá ver os eventos mais recentes no histórico de eventos no CloudTrail console. Para mais informações, consulte Visão geral da criação de uma trilha no Guia do usuário do AWS CloudTrail.

Informações sobre o Amazon MQ em CloudTrail

Quando você cria sua AWS conta, CloudTrail está habilitado. Quando ocorre uma atividade de evento compatível do Amazon MQ, ela é registrada em um CloudTrail evento com outros eventos de AWS serviço no histórico de eventos. Você pode visualizar, pesquisar e fazer download de eventos recentes para a sua conta da AWS . Para obter mais informações, consulte Visualização de CloudTrail eventos com histórico de eventos no Guia AWS CloudTrail do usuário.

Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Você pode criar uma trilha para manter um registro contínuo dos eventos em sua AWS conta. Por padrão, quando você cria uma trilha usando o Console de gerenciamento da AWS, a trilha se aplica a todas as AWS regiões. A trilha registra eventos de todas as AWS regiões e entrega arquivos de log para o bucket especificado do Amazon S3. Você também pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para saber mais, consulte os seguintes tópicos no Manual do usuário do AWS CloudTrail :

- CloudTrail Serviços e integrações compatíveis
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões
- Recebendo arquivos de CloudTrail log de várias contas

O Amazon MQ suporta o registro dos parâmetros de solicitação e das respostas para o seguinte APIs como eventos em arquivos de CloudTrail log:

- `CreateConfiguration`
- `DeleteBroker`
- `DeleteUser`
- `RebootBroker`
- `UpdateBroker`

> **Note**
>
> RebootBroker os arquivos de log são registrados quando você reinicia o broker. Durante a janela de manutenção, o serviço é reinicializado automaticamente e os arquivos de RebootBroker log não são registrados.

> **Important**
>
> Para os `GET` métodos a seguir APIs, os parâmetros da solicitação são registrados, mas as respostas são editadas:
>
> - `DescribeBroker`

- DescribeConfiguration
- DescribeConfigurationRevision
- DescribeUser
- ListBrokers
- ListConfigurationRevisions
- ListConfigurations
- ListUsers

Para o seguinte APIs, os parâmetros de `password` solicitação `data` e estão ocultos por asteriscos ()`***`:

- CreateBroker (POST)
- CreateUser (POST)
- UpdateConfiguration (PUT)
- UpdateUser (PUT)

Cada evento ou entrada de log contém informações sobre o solicitante. As informações ajudam a identificar:

- A solicitação foi feita com credenciais de usuário raiz ou do ?
- A solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado?
- A solicitação foi feita por outro AWS serviço?

Para obter mais informações, consulte CloudTrailUserIdentity Element no Guia do AWS CloudTrail usuário.

Exemplo de entrada do arquivo de log do Amazon MQ

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para o bucket especificado do Amazon S3. CloudTrail os arquivos de log contêm uma ou mais entradas de log.

Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a solicitação para uma API do Amazon MQ, o endereço IP do solicitante, a identidade do solicitante, a data e a hora da solicitação e assim por diante.

O exemplo a seguir mostra uma entrada de CloudTrail registro para uma chamada de `CreateBroker`API.

> **Note**
>
> Como os arquivos de CloudTrail log não são um rastreamento de pilha ordenado do público APIs, eles não listam as informações em nenhuma ordem específica.


```
{
    "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"
}
```

Configurar logs do Amazon MQ for ActiveMQ

Para permitir que o Amazon MQ publique registros no CloudWatch Logs, você deve adicionar uma permissão ao seu usuário do Amazon MQ e também configurar uma política baseada em recursos para o Amazon MQ antes de criar ou reiniciar o agente.

> **Note**
>
> Quando você ativa os registros e publica mensagens do console web ActiveMQ, o conteúdo da mensagem é enviado e exibido CloudWatch nos registros.

A seguir, descrevemos as etapas para configurar CloudWatch registros para seus corretores ActiveMQ.

Tópicos

Entendendo a estrutura do registro em CloudWatch Logs

Você pode habilitar o registro geral e de auditoria quando configurar definições avançadas do agente na criação ou edição de um agente.

O registro geral ativa o nível de `INFO` registro padrão (o `DEBUG` registro não é suportado) e publica `activemq.log` em um grupo de registros em sua CloudWatch conta. O grupo de logs tem um formato semelhante ao seguinte:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/general
```

O registro de auditoria permite o registro de ações de gerenciamento realizadas usando o JMX ou o ActiveMQ Web Console e publica em um grupo de registros em `audit.log` sua conta. CloudWatch O grupo de logs tem um formato semelhante ao seguinte:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/audit
```

Dependendo se você tem um agente de instância única ou um agente ativo/em espera, o Amazon MQ cria uma ou duas transmissões de log dentro de cada grupo de logs. Os fluxos de log têm um formato semelhante ao seguinte:

```
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.log
               activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-2.log
```

Os sufixos `-1` e `-2` denotam instâncias individuais do agente. Para obter mais informações, consulte Como trabalhar com grupos de registros e fluxos de registros no Guia do usuário do Amazon CloudWatch Logs.

Adicionar a permissão **`CreateLogGroup`** ao seu usuário do Amazon MQ

Para permitir que o Amazon MQ crie um grupo de CloudWatch logs de registros, você deve garantir que o usuário que cria ou reinicializa o agente tenha a permissão. `logs:CreateLogGroup`

> **Important**
>
> Se você não adicionar a permissão `CreateLogGroup` ao seu usuário do Amazon MQ antes que ele crie ou reinicialize o agente, o Amazon MQ não criará o grupo de logs.

O exemplo a seguir Política baseada no IAM concede permissão para `logs:CreateLogGroup` para usuários aos quais esta política está anexada.

JSON

```
{
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "logs:CreateLogGroup",
                            "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*"
                        }
                    ]
                    }
```

> **Note**
>
> Aqui, o termo usuário se refere a Usuários e não a Usuários do Amazon MQ, que são criados quando um novo agente é configurado. Para obter mais informações sobre a configuração de usuários e de políticas do IAM, consulte a seção Visão geral do gerenciamento de identidade do Guia do usuário do IAM.

Para obter mais informações, consulte `CreateLogGroup` a Referência da API Amazon CloudWatch Logs.

Configure uma política baseada em recursos para o Amazon MQ

> **Important**
>
> Se você não configurar uma política baseada em recursos para o Amazon MQ, o agente não poderá publicar os registros no Logs. CloudWatch

Para permitir que o Amazon MQ publique registros em seu grupo de registros de CloudWatch registros, configure uma política baseada em recursos para dar ao Amazon MQ acesso às seguintes ações da API de registros: CloudWatch

- `CreateLogStream`— Cria um fluxo de CloudWatch registros para o grupo de registros especificado.
- `PutLogEvents`— Entrega eventos para o fluxo de registro de CloudWatch registros especificado.

A política baseada em recursos a seguir concede permissão para `logs:CreateLogStream` e `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/*"
                                }
                            ]
                        }
```

Essa política baseada em recursos deve ser configurada usando o, AWS CLI conforme mostrado no comando a seguir. No exemplo, substitua *us-east-1* com suas próprias informações.

```
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 esse exemplo usa o `/aws/amazonmq/` prefixo, você precisa configurar a política baseada em recursos somente uma vez por AWS conta, por região.

Prevenção contra o ataque do “substituto confuso” em todos os serviços

“Confused deputy” é um problema de segurança no qual uma entidade sem permissão para executar uma ação pode coagir uma entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar no problema confuso do deputado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com diretores de serviços que receberam acesso aos recursos em sua conta.

Recomendamos usar as `aws:SourceArn` chaves de contexto de condição `aws:SourceAccount` global em sua política baseada em recursos do Amazon MQ para limitar o acesso aos CloudWatch registros a um ou mais corretores específicos.

> **Note**
>
> Se você utilizar ambas as chaves de contexto de condição global, o valor `aws:SourceAccount` e a conta `aws:SourceArn` no valor deverão utilizar o mesmo ID de conta quando utilizados na mesma instrução de política.

O exemplo a seguir demonstra uma política baseada em recursos que limita o acesso aos CloudWatch registros a um único agente do 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"
                                }
                            }
                            }
                        ]
                        }
```

Você também pode configurar sua política baseada em recursos para limitar o acesso aos CloudWatch registros a todos os corretores em uma conta, conforme mostrado a seguir.

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 obter mais informações sobre o problema de segurança de representante confuso, consulte O problema do representante confuso, no Guia do usuário.

Solução de problemas na configuração de CloudWatch registros com o Amazon MQ

Em alguns casos, CloudWatch os registros nem sempre se comportam conforme o esperado. Esta seção fornece uma visão geral dos problemas comuns e mostra como resolvê-los.

Grupos de registros não aparecem em CloudWatch

Adicione a permissão `CreateLogGroup` ao seu usuário do Amazon MQ e reinicialize o agente. Isso permite que o Amazon MQ crie o grupo de logs.

Os fluxos de registros não aparecem nos grupos de CloudWatch registros

Configure uma política baseada em recursos para o Amazon MQ. Isso permite que seu agente publique seus logs.

Cotas no Amazon MQ

Este tópico lista os limites do Amazon MQ. Muitos dos limites a seguir podem ser alterados para AWS contas específicas. Para solicitar o aumento de um limite, consulte Cotas de serviço da AWS na Referência geral da Amazon Web Services. Os limites atualizados não estarão visíveis mesmo após a aplicação do aumento do limite. Para obter mais informações sobre a visualização dos limites de conexão atuais na Amazon CloudWatch, consulte Monitoramento de agentes do Amazon MQ usando a Amazon. CloudWatch

Tópicos

Operadores

A tabela a seguir lista as cotas relacionadas aos agente do Amazon MQ.

Limite	Description
Nome do agente	• Deve ser exclusivo em sua AWS conta. • Deve ter entre 1 e 50 caracteres. • Deve conter somente caracteres especific ados no Conjunto de caracteres imprimíveis ASCII. • Pode conter somente caracteres alfanumér icos, traços, pontos, sublinhados e tils (- . _ ~).

Limite	Description
Número de agentes, por região	200
Conexões de nível de fio por protocolo para agente menor	**Important** Não se aplica aos agentes do RabbitMQ. 300 para agentes de tipo de instância `mq.*.micro` .
Conexões de nível de fio por protocolo para agente maior	**Important** Não se aplica aos agentes do RabbitMQ. 2 mil para agentes de tipo de instância `mq.*.*large` .
Grupos de segurança por agente	5
Destinos do ActiveMQ (filas e tópicos) monitorados em CloudWatch	CloudWatch monitora somente os primeiros 1000 destinos.
Destinos (filas) do RabbitMQ monitorados em CloudWatch	CloudWatch monitora apenas os primeiros 500 destinos, ordenados por número de consumidores.
Etiquetas por agente	50

Configurações

A tabela a seguir lista as cotas relacionadas às configurações do Amazon MQ.

Limite	Description
Nome da configuração	• Deve ter entre 1 e 150 caracteres. • Deve conter somente caracteres especific ados no Conjunto de caracteres imprimíveis ASCII. • Pode conter somente caracteres alfanumér icos, traços, pontos, sublinhados e tils (- . _ ~).
Revisões por configuração	300

Usuários

A tabela a seguir lista as cotas relacionadas aos usuários do Amazon MQ ActiveMQ.

Limite	Description
Nome de usuário	• Deve ter entre 1 e 100 caracteres. • Deve conter somente caracteres especific ados no Conjunto de caracteres imprimíveis ASCII. • Pode conter somente caracteres alfanumér icos, traços, pontos, sublinhados e tils (- . _ ~). • Não deve conter vírgulas (,).
Senha	• Deve ter entre 12 e 250 caracteres. •

Limite	Description
	Deve conter somente caracteres especific ados no Conjunto de caracteres imprimíveis ASCII. • Deve conter pelo menos 4 caracteres únicos. • Não deve conter vírgulas (`,`).
Usuários por agente (autenticação simples)	250
Grupos por usuário (autenticação simples)	20

Armazenamento de dados

A tabela a seguir lista as cotas relacionadas ao armazenamento de dados do Amazon MQ.

Limite	Description
Capacidade de armazenamento por agente	20 GB para agentes de tipo de instância `mq.*.micro` . Para obter mais informações sobre os tipos de instância do Amazon MQ, consulte Broker instance types.
Capacidade de armazenamento por agente	200 GB para agentes de tipo de instância `mq.m5.*`. Para obter mais informações sobre os tipos de instância do Amazon MQ, consulte Broker instance types.
Limite de uso do programador de trabalhos por agente com o suporte do Amazon EBS	**Important** Não se aplica aos agentes do RabbitMQ.

Limite	Description
	50 GB. Para obter mais informações sobre o uso do programador de trabalhos, consulte `JobSchedulerUsage` no Documentação da API Apache ActiveMQ.
Capacidade de armazenamento temporário por intermediário menor.	**Important** Não se aplica aos agentes do RabbitMQ. 5 GB para agentes de tipo de instância `mq.*.micro` .
Capacidade de armazenamento temporário por intermediário maior.	**Important** Não se aplica aos agentes do RabbitMQ. 50 GB para agentes de tipo de instância `mq.m5.*`.

Controle de utilização de API

As seguintes cotas de limitação são agregadas por AWS conta, em todo o Amazon MQ, para manter a largura de banda do serviço. APIs Para obter mais informações sobre o Amazon MQ APIs, consulte a Referência da API REST do Amazon MQ.

> **Important**
>
> Essas cotas não se aplicam às mensagens do agente Amazon MQ para ActiveMQ ou Amazon MQ para RabbitMQ. APIs Por exemplo, o Amazon MQ não limita o envio nem o recebimento de mensagens.

Limite de expansão da API	Limite da taxa de API
100	15

Solução de problemas do Amazon MQ

Esta seção descreve problemas comuns que você pode encontrar ao usar os agentes do Amazon MQ e as etapas que você pode implementar para resolvê-los. Para solução geral de problemas, consultethe section called “Solução de problemas comuns: Amazon MQ”. Para solucionar problemas da versão do mecanismo específico, consulte as seções a seguir.

Solução de problemas do ActiveMQ no Amazon MQ

Tópico da solução de problemas	Description
Solução de problemas gerais	Use as informações desta seção para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados quando se trabalha com os agentes do ActiveMQ no Amazon MQ.
BROKER_ENI_EXCLUÍDO	O ActiveMQ no Amazon MQ emitirá um alarme `BROKER_ENI_DELETED` quando você excluir a interface de rede elástica (ENI) de um agente.
BROKER_OOM	O ActiveMQ no Amazon MQ emitirá um alarme BROKER_OOM quando o agente passar por um ciclo de reinicialização devido à capacidade insuficiente de memória

Solução de problemas: do RabbitMQ no Amazon MQ

Tópico da solução de problemas	Description
Solução de problemas gerais	Identifique problemas comuns que podem ser encontrados quando se trabalha com agentes do RabbitMQ.

Tópico da solução de problemas	Description
RABBITMQ_MEMORY_ALARM	O RabbitMQ emitirá um alarme de alta memória quando o uso de memória do corretor, identific ado pela CloudWatch métrica`RabbitMQMemUsed` , exceder o limite de memória identificado por. `RabbitMQM emLimit`
RABBITMQ_INVALID_KMS_KEY	O RabbitMQ no Amazon MQ gerará um código obrigatór io de ação crítica INVALID_K MS_KEY quando um agente criado com uma solução gerenciada pelo cliente AWS KMS key(CMK) detectar que a chave (KMS) está desativad a. AWS Key Management Service
RABBITMQ_INVALID_ASSUME ROLE	O RabbitMQ no Amazon MQ gerará um código obrigatór io de ação crítica INVALID_A SSUMEROLE quando o ARN da função do IAM especificado em não puder ser assumido pelo Amazon MQ. `aws.arns. assume_role_arn`

Tópico da solução de problemas	Description
RABBITMQ_INVALID_ARN_LDAP	O RabbitMQ no Amazon MQ gerará um código obrigatório de ação crítica INVALID_ARN_LDAP quando a senha ARN da conta de serviço LDAP for inválida ou inacessível.
RABBITMQ_INVALID_ARN_HTTP	O RabbitMQ no Amazon MQ gerará um código obrigatório de ação crítica INVALID_ARN_HTTP quando um ou mais certificados SSL ou arquivo de chave para HTTP auth_backend forem ARNs inválidos ou inacessíveis.
RABBITMQ_INVALID_ARN_SSL	O RabbitMQ no Amazon MQ gerará um código de ação crítica INVALID_ARN_SSL exigido quando um ou mais ARNs repositórios confiáveis de certificados da CA para EXTERNAL auth_mechanism forem inválidos ou inacessíveis.
RABBITMQ_INVALID_ARN	O RabbitMQ no Amazon MQ gerará um código de ação crítica INVALID_ARN necessário quando um ou mais ARNs na configuração do agente forem inválidos ou inacessíveis.

Tópico da solução de problemas	Description
RABBITMQ_DISK_ALARM	O alarme de limite de disco é uma indicação de que o volume de disco usado por um nó do RabbitMQ diminuiu devido ao alto número de mensagens não consumidas enquanto novas mensagens foram adicionadas.

Solução de problemas comuns: Amazon MQ

Use as informações desta seção para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com agentes do Amazon MQ, como problemas de conexão com seu o agente e reinicializações do agente.

Sumário

Não consigo me conectar ao console da Web ou endpoints do agente.

Se você estiver enfrentando problemas para se conectar ao seu agente usando o console da Web ou endpoints em nível de fio, recomendamos as etapas a seguir.

1. Confira se você está tentando se conectar ao seu agente por trás de um firewall. Pode ser necessário configurar o firewall para permitir o acesso ao agente.
2. Confira se você está tentando se conectar ao seu agente usando um endpoint FIPS. O Amazon MQ só oferece suporte a endpoints FIPS ao usar operações de API, mas não para conexões de nível de conexão com a própria instância do agente.

3. Confira se a Acessibilidade pública para o seu agente está definida como Yes (Sim). Se esta opção estiver definida como No (Não), confira as regras da Lista de controle de acesso (ACL) da sua sub-rede. Se você criou uma rede personalizada ACLs, talvez seja necessário alterar as regras de ACL da rede para fornecer acesso ao seu corretor. Para obter mais informações sobre a rede da Amazon VPC, consulte Habilitação do acesso à Internet no Manual do usuário da Amazon VPC.
4. Confira as regras do Grupo de Segurança do seu agente. Confira se você está permitindo conexões com as seguintes portas:

> **Note**
>
> As portas a seguir são agrupadas de acordo com os tipos de mecanismo, pois o ActiveMQ no Amazon MQ e o RabbitMQ no Amazon MQ usam portas diferentes para conexões.

ActiveMQ no Amazon MQ

- Console da Web — Porta 8162
- OpenWire — Porto 61617
- AMQP — Porta 5671
- STOMP: porta 61614
- MQTT — Porta 8883
- WSS — Porta 61619

RabbitMQ no Amazon MQ

- Console da Web e API de gerenciamento — Porta 443 e 15671
- AMQP — Porta 5671

5. Execute os seguintes testes de conectividade de rede para o tipo de mecanismo do agente.

> **Note**
>
> Para agentes sem acessibilidade pública, execute os testes de uma instância do Amazon EC2 na mesma Amazon VPC que o seu agente do Amazon MQ e avalie as respostas.

ActiveMQ on Amazon MQ

Para testar a conectividade de rede do agente do ActiveMQ no Amazon MQ

1. Abra uma nova janela de terminal ou de linha de comando.
2. Execute o seguinte comando `nslookup` para consultar o registro DNS do seu agente. Para implantações ativas/em espera, teste os endpoints ativos e em espera. Os active/standby endpoints são identificados com um sufixo `-1` ou `-2` adicionados ao ID exclusivo do corretor. Substitua o endpoint com as suas informações.

```
$ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Se a consulta for bem-sucedida, você verá um resultado semelhante a este.

```
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
```

O endereço IP resolvido deve corresponder aos endereços IP fornecidos no console do Amazon MQ. Isso indica que o nome do domínio está resolvendo corretamente no servidor DNS e você pode passar para a próxima etapa.

3. Execute o seguinte comando `telnet` para testar o caminho de rede para o seu agente. Substitua o endpoint com as suas informações. *`port`*Substitua `8162` pelo número da porta do console web ou por outras portas de nível de fio para testar protocolos adicionais conforme necessário.

> **Note**
>
> Para active/standby implantações, você receberá uma mensagem de `Connect failed` erro se executar `telnet` com o endpoint em espera. Isso é esperado, pois a própria instância em espera está sendo executada, mas o processo do ActiveMQ não está sendo executado e não tem acesso ao volume de

armazenamento do Amazon EFS do agente. Execute o comando para ambos os endpoints `-1` e `-2` para garantir que você teste as instâncias ativas e em espera.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Para a instância ativa, você verá um resultado semelhante ao seguinte.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

4. Execute um destes procedimentos:
   - Se o comando `telnet` tiver êxito, confira a métrica `EstablishedConnectionsCount` e confirme que o agente não tenha atingido o limite máximo de conexões com fio. Você também pode confirmar se o limite foi atingido revisando os logs `General` do agente. Se essa métrica for maior que zero, há pelo menos um cliente conectado ao agente no momento. Se a métrica não mostrar nenhuma conexão, execute o teste de caminho `telnet` novamente e aguarde pelo menos um minuto antes de desconectar, pois as métricas do agente são publicadas a cada minuto.
   - Se o comando `telnet` falhar, confira o status da interface de rede elástica do agente e confirme se o status é `in-use`. Crie um log de fluxo da Amazon VPC para a interface de rede de cada instância e revise os logs de fluxo gerados. Procure os endereços IP do agente quando você executou o comando `telnet` e confirme se os pacotes de conexão estão `ACCEPTED`, incluindo um pacote de devolução. Para obter mais informações e para ver um exemplo de log de fluxo, consulte Exemplos de registro de logs de fluxo no Guia de Desenvolvedores da Amazon VPC.
5. Execute o seguinte comando `curl` para conferir a conectividade com o console da Web de administração do ActiveMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com:8162/index.html
```

Se o comando for bem-sucedido, o resultado deve ser um documento HTML semelhante a este.

```
<!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 testar a conectividade de rede do agente do RabbitMQ no Amazon MQ

1. Abra uma nova janela de terminal ou de linha de comando.
2. Execute o seguinte comando `nslookup` para consultar o registro DNS do seu agente. Substitua o endpoint com as suas informações.

   ```
   $ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
   ```

   Se a consulta for bem-sucedida, você verá um resultado semelhante a este.

   ```
   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. Execute o seguinte comando `telnet` para testar o caminho de rede para o seu agente. Substitua o endpoint com as suas informações. Você pode *`port`* substituir por uma porta `443` para o console web e testar `5671` a conexão AMQP em nível de fio.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Se o comando for bem-sucedido, você verá um resultado semelhante a este.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

> **Note**
>
> A conexão telnet será fechada automaticamente após alguns segundos.

4. Execute um destes procedimentos:

   - Se o comando `telnet` for bem-sucedido, confira a métrica `ConnectionCount` e confirme que o agente não atingiu o valor definido na política padrão `max-connections`. Você também pode confirmar se o limite foi atingido revisando o grupo de logs do agente `Connection.log`. Se essa métrica for maior que zero, há pelo menos um cliente conectado ao agente no momento. Se a métrica não mostrar nenhuma conexão, execute o teste de caminho `telnet` novamente. Talvez seja necessário repetir esse processo se a conexão for fechada antes que seu agente publique novas métricas de conexão para CloudWatch. As métricas são publicadas a cada minuto.
   - Para agentes não acessíveis publicamente, se o comando `telnet` falhar, verifique o status das interfaces de rede elástica do agente e confirme se o status é `in-use`. Crie um log de fluxo da Amazon VPC para cada interface de rede e revise os logs de fluxo gerados. Procure os endereços IP privados do agente quando o comando `telnet` foi evocado e confirme se os pacotes de conexão estão `ACCEPTED`, incluindo um pacote de devolução. Para obter mais informações e para ver um exemplo de log de fluxo, consulte Exemplos de registro de logs de fluxo no Guia de Desenvolvedores da Amazon VPC.

> **Note**
>
> Essa etapa não se aplica aos do RabbitMQ no Amazon MQ com acessibilidade pública.

5. Execute o seguinte comando `curl` para conferir a conectividade com o console da Web de administração do RabbitMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com:443/index.html
```

Se o comando for bem-sucedido, o resultado deve ser um documento HTML semelhante a este.

```
<!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>
        ...
```

Meu agente está sendo executado e posso verificar a conectividade usando **`telnet`**, mas meus clientes não conseguem estabelecer conexão e estão retornando exceções SSL.

Seu certificado de endpoint do agente pode ter sido atualizado durante a janela de manutenção do agente. Os certificados de agente do Amazon MQ são alternados periodicamente para garantir a disponibilidade contínua e a segurança dos agentes.

Recomendamos o uso da autoridade de certificação (CA) raiz da Amazon no Amazon Trust Services para autenticar no armazenamento de confiança de seus clientes. Todos os certificados do agente do Amazon MQ são assinados com essa CA raiz. Ao usar uma CA raiz da Amazon, você não precisará mais baixar o novo certificado de agente do Amazon MQ sempre que um certificado for atualizado no agente.

Criei um agente, mas a criação falhou.

Se o seu agente estiver em um status `CREATION_FAILED`, faça o seguinte.

- Confira as suas permissões do IAM. Para criar um agente, você deve usar a política AWS gerenciada do IAM `AmazonMQFullAccess` ou ter o conjunto correto de permissões do Amazon EC2 em sua política personalizada do IAM. Para saber mais sobre as permissões necessárias do Amazon EC2 de que precisa, consulte as Permissões do IAM necessárias para criar um agente Amazon MQ.
- Confira se a sub-rede que você está escolhendo para o seu agente está em uma Amazon Virtual Private Cloud (VPC) compartilhada. Para criar um agente do Amazon MQ em uma Amazon VPC compartilhada, você deve criá-lo na conta que possui a Amazon VPC.

Meu agente reiniciou e não sei por quê.

Se o seu agente foi reiniciado automaticamente, pode ter sido por um dos motivos a seguir.

- Seu agente pode ter sido reiniciado devido a uma janela de manutenção semanal agendada. Periodicamente, o Amazon MQ realiza a manutenção do hardware, do sistema operacional ou do software do mecanismo de um agente de mensagens. A duração da manutenção varia, mas pode durar até duas horas, dependendo das operações agendadas para o agente de mensagens. Os agentes podem reiniciar a qualquer momento durante a janela de manutenção de duas horas. Para obter mais informações sobre janelas de manutenção do agente, consulte the section called “Agendar a manutenção do agente”.
- O tipo de instância do seu agente pode não ser adequado ao workload da sua aplicação. Por exemplo, executar um workload de produção em um `mq.t3.micro` pode resultar na falta de recursos do agente. Alta utilização da CPU ou alta utilização da memória do agente pode fazer com que este reinicie inesperadamente. Para ver quanta CPU e memória estão sendo utilizadas pelo seu corretor, use as seguintes CloudWatch métricas para seu tipo de mecanismo.
  - ActiveMQ no Amazon MQ — Confira em `CpuUtilization` o percentual de unidades de computação do Amazon EC2 alocadas que o agente utiliza no momento. Confira em `HeapUsage` a porcentagem do limite de memória do ActiveMQ JVM que o agente usa atualmente.
  - RabbitMQ no Amazon MQ – Confira em `SystemCpuUtilization` o percentual de unidades de computação do Amazon EC2 alocadas que o agente utiliza atualmente. Confira em

`RabbitMQMemUsed` o volume de RAM usado em Bytes, e divida por `RabbitMQMemLimit` para a porcentagem de memória usada pelo nó RabbitMQ.

Para obter mais informações sobre tipos de instância do agente e como escolher o tipo de instância correto para seu workload, consulte Broker instance types.

Solução de problemas do ActiveMQ no Amazon MQ

Use as informações desta seção para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados quando se trabalha com os agentes do ActiveMQ no Amazon MQ.

Sumário

Não consigo ver os registros gerais ou de auditoria do meu corretor no CloudWatch Logs, embora eu tenha ativado o registro.

Se você não conseguir visualizar os registros do seu corretor em CloudWatch Logs, faça o seguinte.

1. Confira se o usuário que cria ou reinicializa o agente tem a permissão `logs:CreateLogGroup`. Se você não adicionar a permissão `CreateLogGroup` a um usuário antes que este crie ou reinicialize o agente, o Amazon MQ não criará o grupo de logs.
2. Verifique se você configurou uma política baseada em recursos para permitir que o Amazon MQ publique registros no Logs. CloudWatch Para permitir que o Amazon MQ publique registros em seu grupo de registros de CloudWatch registros, configure uma política baseada em recursos para dar ao Amazon MQ acesso às seguintes ações da API de registros: CloudWatch
   - `CreateLogStream`— Cria um fluxo de CloudWatch registros para o grupo de registros especificado.

- `PutLogEvents`— Entrega eventos para o fluxo de registro de CloudWatch registros especificado.

Para obter mais informações sobre como configurar o ActiveMQ no Amazon MQ para publicar CloudWatch logs em Logs, consulte Configurando registros.

Após a reinicialização do agente ou da janela de manutenção, não consigo me conectar ao meu agente, embora o status seja **`RUNNING`**. Por quê?

Você pode estar enfrentando problemas de conexão após a reinicialização de um agente, após a conclusão de uma janela de manutenção programada ou em um evento de falha, em que a instância em espera é ativada. De qualquer forma, problemas de conexão após a reinicialização de um agente provavelmente são causados por uma quantidade muito grande de mensagens que persistiram no volume de armazenamento do Amazon EFS ou do Amazon EBS do agente. Durante uma reinicialização, o Amazon MQ move mensagens persistentes do armazenamento para a memória do agente. Para confirmar esse diagnóstico, você pode monitorar as seguintes métricas CloudWatch para seu agente Amazon MQ for ActiveMQ:

- **`StoragePercentUsage`** — Grandes porcentagens em ou perto de 100% podem fazer com que o agente recuse conexões.
- **`JournalFilesForFullRecovery`** — Indica o número de arquivos do diário que serão reproduzidos após desligamento e reinicialização não planejados. Um valor crescente, ou consistentemente maior que um, indica transações não resolvidas que podem causar problemas de conexão após a reinicialização.
- **`OpenTransactionCount`** — Um número maior que zero após uma reinicialização indica que o agente tentará armazenar mensagens consumidas anteriormente, causando assim problemas de conexão.

Para resolver esse problema, recomendamos resolver suas transações XA com `rollback()` ou `commit()`. Para obter mais informações e ver um exemplo de código para resolver transações XA usando `rollback()`, consulte Recuperar Transações XA.

Vejo alguns dos meus clientes se conectando ao agente, enquanto outros não conseguem se conectar.

Se o seu agente está no status `RUNNING` e alguns clientes são capazes de se conectar ao agente com sucesso, enquanto outros não conseguem fazê-lo, você pode ter chegado ao limite de Conexões em nível de fio para o agente. Para verificar se você atingiu o limite de conexões em nível de fio, faça o seguinte:

- Verifique os registros gerais do seu agente ActiveMQ no Amazon MQ em Logs. CloudWatch Se o limite tiver sido atingido, você verá `Reached Maximum Connections` nos logs do agente. Para obter mais informações sobre CloudWatch os registros do ActiveMQ em corretores do Amazon MQ, consulte. the section called “Entendendo a estrutura do registro em CloudWatch Logs”

Quando o limite de conexões em nível de fio for atingido, o agente recusará ativamente novas conexões de entrada. Para resolver esse problema, recomendamos atualizar o tipo de instância do seu agente. Para obter mais informações sobre como escolher o melhor tipo de instância para seu workload, consulte Broker instance types.

Se você confirmou que o número de conexões em nível de fio é menor que o limite de conexão do agente, o problema pode estar relacionado à reinicialização de clientes. Verifique, nos logs do agente, entradas numerosas e frequentes de `... Inactive for longer than 600000 ms - removing ...`. A entrada de log indica reinicialização de clientes ou problemas de conectividade. Esse efeito é mais evidente quando os clientes se conectam ao agente por meio de um Network Load Balancer (NLB) com clientes que frequentemente se desconectam e se reconectam ao agente. Isso geralmente é observado em clientes baseados em contêiner.

Para obter mais detalhes, verifique seus logs no lado do cliente. O agente limpará conexões TCP inativas após 600000 ms e liberará o soquete da conexão.

Estou vendo a exceção **`org.apache.jasper.JasperException: An exception occurred processing JSP page`** no console do ActiveMQ ao executar operações.

Se você estiver usando autenticação e configuração simples de `AuthorizationPlugin` para autorização de fila e tópico, use o elemento `AuthorizationEntries` em seu arquivo de configuração XML e conceda a permissão de grupo `activemq-webconsole` para todas as filas

e tópicos. Isso garante que o console da Web do ActiveMQ possa se comunicar com o agente do ActiveMQ.

O exemplo `AuthorizationEntry` a seguir concede permissões de leitura e gravação para todas as filas e tópicos para o grupo `activemq-webconsole`.

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

Da mesma forma, ao integrar seu agente ao LDAP, certifique-se de conceder permissão para o grupo `amazonmq-console-admins`. Para ter mais informações sobre LDAP, consulte the section called “Como funciona a integração com LDAP”.

Solução de problemas: RabbitMQ no Amazon MQ

Use as informações desta seção para ajudar a diagnosticar e resolver problemas comuns que podem ser encontrados quando se trabalha com os agentes do RabbitMQ no Amazon MQ.

Sumário

Não consigo ver as métricas das minhas filas ou dos meus hosts virtuais em CloudWatch.

Se você não conseguir visualizar as métricas de suas filas ou hosts virtuais em CloudWatch, verifique se os nomes da fila ou do host virtual contêm espaços em branco, guias ou outros caracteres não ASCII.

O Amazon MQ não pode publicar métricas para hosts virtuais e filas com nomes que contenham espaços em branco, “tabs” ou outros caracteres “não ASCII”.

Para obter mais informações sobre nomes de dimensões, consulte Dimension na Amazon CloudWatch API Reference.

Como faço para habilitar plugins no RabbitMQ no Amazon MQ?

Atualmente, o RabbitMQ no Amazon MQ oferece suporte apenas ao plugin de gerenciamento, shovel, federação, troca de hash consistente do RabbitMQ, que estão habilitados por padrão. Para obter mais informações sobre como usar plugins compatíveis, consulte o the section called “Plugins”.

Não consigo alterar a configuração da Amazon VPC para o agente.

O Amazon MQ não é compatível com a alteração da configuração da Amazon VPC após a criação do agente. Observe que você precisará criar um novo agente com a nova configuração da Amazon VPC e atualizar a URL de conexão do cliente com a nova URL de conexão do agente.

As implantações de cluster pausaram as sincronizações de fila.

Ao solucionar alarmes de alta memória do RabbitMQ, você pode constatar que as mensagens em uma ou várias filas não podem ser consumidas. Essas filas podem estar no processo de sincronização de mensagens entre nós, durante o qual as respectivas filas ficam indisponíveis para publicação e consumo. As sincronizações de filas podem ficar pausadas devido ao alarme de alta memória e até mesmo contribuir para o alarme de memória.

Para saber mais sobre como interromper e repetir sincronizações de filas pausadas, consulte the section called “Resolvendo a sincronização de fila pausada”.

Meu agente de instância única do Amazon MQ para RabbitMQ está em um loop de reinicialização.

Um agente de instância única do Amazon MQ para RabbitMQ que gera um alarme de alta memória corre o risco de se tornar indisponível se for reiniciado e não tiver memória suficiente para a inicialização. Isso pode fazer com que o RabbitMQ entre em um loop de reinicialização e evite interações adicionais com o agente até que o problema seja resolvido. Se o agente estiver em um loop de reinicialização, não será possível aplicar as práticas recomendadas do Amazon MQ para resolver o alarme de alta memória.

Para recuperar o agente, recomendamos fazer upgrade para um tipo de instância maior com mais memória. Ao contrário de implantações de cluster, você pode atualizar um agente de instância única enquanto ele está enfrentando um alarme de alta memória, pois não há sincronizações de filas a serem executadas entre os nós durante uma reinicialização.

Eu perdi o acesso a todas as contas de administrador do meu corretor.

Você pode recuperar o acesso usando a autenticação do IAM. Ative a federação externa de identidade da web para sua AWS conta, crie uma função do IAM com permissões para obter tokens de identidade da web, configure seu agente para aceitar a autenticação do IAM via OAuth 2.0 e, em seguida, use as credenciais do IAM para obter um token JWT e criar um novo usuário administrador. Para obter instruções detalhadas, consulte the section called “Usando autenticação e autorização do IAM”.

ActiveMQ no Amazon MQ: alarme de interface de rede elástica excluído

O ActiveMQ no Amazon MQ emitirá um alarme BROKER_ENI_DELETED quando você excluir a interface de rede elástica (ENI) de um agente. Quando você cria um agente do Amazon MQ pela primeira vez, o Amazon MQ provisiona uma interface de rede elástica na Virtual Private Cloud (VPC) em sua conta e, por isso, requer uma série de permissões do EC2.

Você não deve modificar ou excluir essa interface de rede. Modificar ou excluir a interface de rede pode causar uma perda permanente de conexão entre a VPC e o operador. Se você quiser excluir a interface de rede, exclua primeiro o agente.

ActiveMQ no Amazon MQ: alarme de falta de memória do agente

O ActiveMQ no Amazon MQ emitirá um alarme BROKER_OOM quando o agente passar por um ciclo de reinicialização devido à capacidade insuficiente de memória. Quando um agente está em um ciclo de reinicialização, também chamado de ciclo de rejeição, o agente inicia repetidas tentativas de recuperação em um curto espaço de tempo. Agentes que não conseguem concluir a inicialização devido à capacidade de memória insuficiente podem entrar em um ciclo de reinicialização, no qual as interações com o agente ficam limitadas.

O Amazon MQ habilita métricas para o seu agente por padrão. Você pode visualizar as métricas do seu corretor acessando o CloudWatch console da Amazon ou usando a CloudWatch API. As seguintes métricas são úteis ao diagnosticar o alarme BROKER_OOM do ActiveMQ:

Métrica do Amazon MQ CloudWatch	Motivo do alto uso de memória
`TotalMessageCount`	Mensagens são armazenadas na memória até que sejam consumidas ou descartadas. Uma alta contagem de mensagens pode indicar o excesso de uso de recursos e pode resultar em um alarme de alta memória.
`HeapUsage`	A porcentagem do limite de memória do ActiveMQ JVM que o agente usa atualmente. Uma porcentagem maior indica que o agente está usando recursos significativos e pode levar a um alarme OOM.
`ConnectionCount`	Conexões de clientes usam memória, e muitas conexões simultâneas podem resultar em um alarme de alta memória.
`CpuUtilization`	O percentual de unidades alocadas de processamento do EC2 que o operador utiliza no momento.
`TotalConsumerCount`	Para cada consumidor conectado ao agente, um número definido de mensagens é carregado do armazenamento na memória

Métrica do Amazon MQ CloudWatch	Motivo do alto uso de memória	
	antes de ser entregue ao consumidor. Um alto número de conexões de consumido r pode causar alto uso de memória e resultar em um alarme de alta memória.	

Para evitar ciclos de reinicialização e evitar o alarme BROKER_OOM, as mensagens devem ser consumidas rapidamente. É possível fazer isso escolhendo o tipo de instância de agente mais eficaz e limpando a fila de mensagens não entregues para descartar mensagens não entregues ou expiradas. Você pode saber mais sobre como garantir um desempenho eficaz em Práticas recomendadas para o ActiveMQ no Amazon MQ.

Amazon MQ for RabbitMQ: Alarme de alta memória

O Amazon MQ para RabbitMQ emitirá um alarme de memória alta quando o uso de memória do agente, identificado por CloudWatch métrica`RabbitMQMemUsed`, exceder o limite de memória identificado por. `RabbitMQMemLimit`

Um agente do RabbitMQ que tiver gerado um alarme de alta memória bloqueará todos os clientes que estiverem publicando mensagens. O agente pode entrar em um loop de reinicialização, ter uma sincronização de filas pausada ou outros problemas que complicam o diagnóstico e a resolução do alarme.

Para diagnosticar e resolver o alarme de alta memória, primeiro siga todas as práticas recomendadas do RabbitMQ e, em seguida, conclua as etapas a seguir.

> **Important**
>
> - `RabbitMQMemLimit` é definido pelo Amazon MQ e é ajustado especificamente considerando-se a memória disponível para cada tipo de instância de host.

- O Amazon MQ não reiniciará um agente com alarme de alta memória e retornará uma exceção para operações da API `RebootBroker`, desde que o agente continue a gerar esse alarme.

Etapa 1: Diagnosticar o alarme de alta memória

Há duas maneiras de diagnosticar alarmes de alta memória no agente do Amazon MQ para RabbitMQ. Recomendamos que você verifique o console web do RabbitMQ e as métricas do Amazon MQ em. CloudWatch

Diagnosticar alarme de alta memória usando-se o console da Web do RabbitMQ

O console da Web do RabbitMQ pode gerar e exibir informações detalhadas de uso de memória para cada nó. Para encontrar essas informações, faça o seguinte:

1. Faça login Console de gerenciamento da AWS e abra o console web RabbitMQ do seu corretor.
2. No console do RabbitMQ, na página Overview (Visão geral), escolha o nome de um nó na lista Nodes (Nós).
3. Na página de detalhes do nó, escolha Memory details (Detalhes da memória) para expandir a seção e visualizar as informações de uso de memória do nó.

As informações de uso de memória fornecidas pelo RabbitMQ no console da Web podem ajudar você a determinar quais recursos podem estar consumindo muita memória e contribuindo para o alarme de alta memória. Para obter mais informações sobre os detalhes de uso da memória disponíveis no console da Web do RabbitMQ, consulte o tópico de Considerações sobre o uso da memória no site de documentação do RabbitMQ Server.

Diagnosticar o alarme de alta memória usando-se métricas do Amazon MQ

O Amazon MQ habilita métricas para o seu agente por padrão. Você pode visualizar as métricas do seu corretor acessando o CloudWatch console ou usando a CloudWatch API. As seguintes métricas são úteis ao diagnosticar o alarme de alta memória do RabbitMQ.

Métrica do Amazon MQ CloudWatch	Motivo do alto uso de memória	
`MessageCount`	Mensagens são armazenadas na memória até que sejam consumidas ou descartadas. Uma alta contagem de mensagens pode indicar o excesso de uso de recursos e pode resultar em um alarme de alta memória.	
`QueueCount`	Filas são armazenadas na memória, e um grande número de filas pode resultar em um alarme de alta memória.	
`ConnectionCount`	Conexões de clientes usam memória, e muitas conexões simultâneas podem resultar em um alarme de alta memória.	
`ChannelCount`	De maneira semelhante a conexões, canais estabelecidos usando cada conexão também são armazenados na memória do nó, e um alto número de canais pode resultar em um alarme de alta memória.	
`ConsumerCount`	Para cada consumidor conectado ao agente, um número definido de mensagens é carregado do	

Métrica do Amazon MQ CloudWatch	Motivo do alto uso de memória	
	armazenamento na memória antes de ser entregue ao consumidor. Um alto número de conexões de consumido r pode causar alto uso de memória e resultar em um alarme de alta memória.	
`PublishRate`	A publicação de mensagens utiliza a memória do agente. Se a taxa na qual as mensagens são publicada s no agente for muito alta e ultrapassar significativamente a taxa na qual o agente entrega mensagens aos consumidores, o agente poderá gerar um alarme de alta memória.	

Etapa 2: Resolver e evitar o alarme de alta memória

> **Note**
>
> Depois que você realiza as ações necessárias, pode levar várias horas para que o status RABBITMQ_MEMORY_ALARM seja apagado.

Siga todas as práticas recomendadas do RabbitMQ como um método geral de prevenção. Para cada colaborador especificado identificado, convém seguir o conjunto de ações a seguir para abordar e evitar alarmes de alta memória do RabbitMQ.

Origem do alto uso de memória	Recomendação do Amazon MQ para abordar	Recomendação do Amazon MQ para evitar
Número de mensagens enviadas	Consuma mensagens publicadas nas filas, limpe mensagens das filas ou exclua as filas do agente.	Ative filas lentas e defina ou reduza o limite de profundidade da fila.
Número de filas	Reduza o número dessas filas.	Defina ou reduza o limite de contagem de filas.
Número de conexões	Reduza o número de conexões.	Defina ou reduza o limite de contagem de conexões.
Número de canais	Reduza o número de canais.	Defina um número máximo de canais por conexão em aplicações cliente.
Número de consumidores	Reduza o número de consumidores conectados ao agente.	Defina um pequeno limite de pré-busca de consumidores.
Taxa de publicação de mensagens	Reduza a taxa na qual os publicadores enviam mensagens ao agente.	Ative confirmações do publicador.
Taxa de tentativas de conexão do cliente	Reduza a frequência na qual os clientes tentam se conectar ao agente para publicar ou consumir mensagens ou configure o agente.	Use conexões de maior duração para reduzir o número e a frequência de tentativas de conexão.

Depois que o alarme de alta memória do agente for resolvido, você poderá fazer upgrade do tipo de instância do host para uma instância com recursos adicionais. Para obter informações sobre como atualizar o tipo de instância do agente, consulte `UpdateBrokerInput` na Referência da API REST do Amazon MQ.

> **Note**
>
> Não é possível fazer downgrade de um agente de um tipo de instância `mq.m5.x` para um tipo de instância `mq.t3.micro`. Se quiser fazer downgrade, você deverá excluir o agente e criar outro.

RabbitMQ no Amazon MQ: chave inválida AWS Key Management Service

O RabbitMQ no Amazon MQ gerará um código obrigatório de ação crítica INVALID_KMS_KEY quando um agente criado com uma solução gerenciada pelo cliente AWS KMS key(CMK) detectar que a chave (KMS) está desativada. AWS Key Management Service Um agente do RabbitMQ com uma CMK verifica periodicamente se a chave KMS está ativada e se o corretor tem todas as concessões necessárias. Se o RabbitMQ não puder verificar se a chave está ativada, o agente será colocado em quarentena e o RabbitMQ retornará INVALID_KMS_KEY.

Sem uma chave do KMS ativa, o agente não tem permissões básicas para chaves do KMS gerenciadas pelo cliente. O agente não pode realizar operações criptográficas usando sua chave até que você reative-a e o agente reinicie. Um agente do RabbitMQ com uma chave do KMS desativada é colocado em quarentena para evitar a deterioração. Depois que o RabbitMQ determinar que a chave do KMS está ativa novamente, o corretor será removido da quarentena. O Amazon MQ não reinicia um agente com uma chave do KMS desativada e retorna uma exceção para operações de API `RebootBroker`, desde que o agente continue a ter uma chave do KMS inválida.

Diagnosticar e solucionar INVALID_KMS_KEY

Para diagnosticar e endereçar o código necessário da ação INVALID_KMS_KEY, você deve usar a Interface de AWS Linha de Comando (CLI) e o console. AWS Key Management Service

Para reativar a chave do KMS

1. Chame o método `DescribeBroker` para recuperar o `kmsKeyId` para o agente da CMK.
2. Faça login no AWS Key Management Service console.
3. Na página Chaves gerenciadas pelo cliente, localize o ID da chave do KMS do agente problemático e verifique se o status é Ativado.

4. Se a chave do KMS tiver sido desativada, reative-a selecionando Ações da chave e Ativar. Depois que a chave for reativada, você deverá esperar que o RabbitMQ remova o agente da quarentena.

Para verificar se as concessões necessárias ainda estão associadas à chave KMS do corretor, chame o `ListGrant` ListGrant método para verificar se elas `mq_rabbit_grant` `mq_grant` estão presentes. Se a concessão ou chave do KMS tiver sido excluída, você deverá excluir o agente e criar outro com todas as concessões necessárias. Para ver as etapas de exclusão de um agente, consulte Excluir um agente.

Para evitar o código obrigatório de ação crítica INVALID_KMS_KEY, não exclua nem desabilite manualmente uma chave do KMS ou uma concessão de CMK. Se você quiser excluir a chave, exclua primeiro o agente.

RabbitMQ no Amazon MQ: alarme de limite de disco

O alarme de limite de disco é uma indicação de que o volume de disco usado por um nó do RabbitMQ diminuiu devido ao alto número de mensagens não consumidas enquanto novas mensagens foram adicionadas. O RabbitMQ emitirá um alarme de limite de disco quando o espaço livre em disco do corretor, identificado pela CloudWatch métrica da Amazon`RabbitMQDiskFree`, atingir o limite de disco, identificado por. `RabbitMQDiskFreeLimit` `RabbitMQDiskFreeLimité` definido pelo Amazon MQ e foi definido considerando o espaço em disco disponível para cada tipo de instância do broker.

Um agente do RabbitMQ no Amazon MQ que tiver gerado um alarme de limite de disco ficará indisponível para novas mensagens que estiverem sendo publicadas. Se você tiver um publicador e um consumidor na mesma conexão, o consumidor também não estará disponível para receber mensagens. Ao executar o RabbitMQ em um cluster, o alarme de disco abrange todo o cluster. Se um nó ficar abaixo do limite, todos os outros nós bloquearão as mensagens recebidas. Devido à falta de espaço em disco, o agente pode ter também outros problemas que complicam o diagnóstico e a resolução do alarme.

O Amazon MQ não reiniciará um agente com alarme de disco e retornará uma exceção para operações da API `RebootBroker`, desde que o agente continue a gerar esse alarme.

> **Note**
>
> Não é possível fazer downgrade de um agente de um tipo de instância `mq.m5` para um tipo de instância `mq.t3.micro`. Se quiser fazer downgrade, você deverá excluir o agente e criar outro.

Diagnostico e solução do alarme de limite de disco

O Amazon MQ habilita métricas para o seu agente por padrão. Você pode visualizar as métricas do seu corretor acessando o CloudWatch console da Amazon ou usando a CloudWatch API. `MessageCount`é uma métrica útil ao diagnosticar o alarme de limite de disco do RabbitMQ. Mensagens são armazenadas na memória até que sejam consumidas ou descartadas. Uma alta contagem de mensagens indica o uso em excesso de armazenamento em disco e pode levar a um alarme de disco.

Para diagnosticar o alarme de limite de disco, use o Console de Gerenciamento do Amazon MQ para:

- Crie uma conexão para consumir as mensagens publicadas nas filas.
- Limpe mensagens das filas.
- Exclua as filas do seu agente.

> **Note**
>
> Pode levar várias horas para que o status RABBITMQ_DISK_ALARM seja apagado depois que você realiza as ações necessárias.

Para evitar que o alarme de limite de disco seja gerado novamente, é possível fazer upgrade do tipo de instância do host para uma instância com recursos adicionais. Para obter informações sobre como atualizar o tipo de instância do agente, consulte `UpdateBrokerInput`, na Referência da API REST do Amazon MQ. Também recomendamos manter publicadores e consumidores em conexões diferentes.

Amazon MQ para RabbitMQ: alarme de alteração de tipo de instância

`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` é uma indicação de que uma alteração solicitada no tipo de instância de agente não pode continuar devido ao alto uso do disco no nó atual do RabbitMQ. O Amazon MQ para RabbitMQ emitirá esse alarme quando o uso atual do disco exceder o que estaria disponível no tipo de instância solicitado, conforme identificado pela métrica. CloudWatch `RabbitMQDiskFree`

Os agentes do RabbitMQ que entrarem no estado `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` continuarão disponíveis para as aplicações, mas a alteração do tipo de instância solicitada não prosseguirá. O Amazon MQ permite a reinicialização do agente nesse estado. No entanto, não é possível alterar o tipo de instância enquanto o uso do disco permanecer acima do limite para o tipo de instância solicitado. O agente retornará uma exceção para operações de API de `ModifyBroker` que tentarem alterar o tipo de instância enquanto estiverem nesse estado.

Diagnostico e abordagem do alarme de alteração do tipo de instância

O Amazon MQ habilita métricas para o seu agente por padrão. Você pode visualizar as métricas do seu corretor acessando o CloudWatch console ou usando a CloudWatch API. `MessageCounte` `RabbitMQDiskFree` métricas podem ser usadas para diagnosticar`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`.

Para resolver o estado de quarentena e permitir que a alteração do tipo de instância continue, use o Console de Gerenciamento do Amazon MQ para:

- Crie uma conexão para consumir as mensagens publicadas nas filas.
- Limpe mensagens das filas.
- Exclua as filas do seu agente.

> **Note**
>
> Pode-se levar várias horas para que o status `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` seja limpo depois que você realiza as ações necessárias.

RabbitMQ no Amazon MQ: função inválida de assumir o IAM

O RabbitMQ no Amazon MQ gerará um código obrigatório de ação crítica INVALID_ASSUMEROLE quando o ARN da função do IAM especificado em for inválido ou não puder ser assumido pelo Amazon MQ. `aws.arns.assume_role_arn` Isso pode ocorrer quando a função não existe, está em uma AWS conta diferente da do corretor ou não tem a relação de confiança necessária com mq.amazonaws.com.

Um agente na quarentena do RABBITMQ_INVALID_ASSUMEROLE não pode recuperar as credenciais ou certificados necessários para a autenticação LDAP, tornando a autenticação LDAP indisponível. Se o LDAP for o único método de autenticação configurado, os usuários não conseguirão se conectar ao agente. A função do IAM é exigida pelo Amazon MQ para acessar AWS recursos referenciados ARNs na configuração do agente, como AWS Secrets Manager segredos ou objetos do Amazon S3 usados para autenticação LDAP.

Diagnosticando e abordando RABBITMQ_INVALID_ASSUMEROLE

Para diagnosticar e resolver o código necessário da ação RABBITMQ_INVALID_ASSUMEROLE, você deve usar o Amazon Logs e o console. CloudWatch AWS Identity and Access Management

Para resolver o problema de assumir função inválida

1. Navegue até o Amazon CloudWatch Logs Insights e execute a seguinte consulta no grupo de registros do seu corretor`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Procure mensagens de erro semelhantes a:

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

3. Verifique a configuração da função do IAM e corrija quaisquer problemas, como:

- Certifique-se de que a função exista na mesma AWS conta do corretor
- Verifique se a política de confiança permite que mq.amazonaws.com assuma a função
- Confirme se a função tem as permissões apropriadas para acessar os AWS recursos necessários

4. Valide a correção usando o endpoint da API de validação de acesso ARN antes de atualizar a configuração do broker.
5. Atualize a configuração do broker e reinicialize o broker.

RabbitMQ no Amazon MQ: ARN LDAP inválido

O RabbitMQ no Amazon MQ gerará um código obrigatório de ação crítica INVALID_ARN_LDAP quando o ARN configurado para a senha da conta de serviço LDAP for inválido ou inacessível. Isso se aplica ao ARNs especificado em `aws.arns.auth_ldap.dn_lookup_bind.password` ou`aws.arns.auth_ldap.other_bind.password`, que deve fazer referência a AWS Secrets Manager segredos contendo senhas em texto simples.

Um agente na quarentena do RABBITMQ_INVALID_ARN_LDAP não pode se autenticar com a conta de serviço LDAP, tornando a autenticação LDAP indisponível. Se o LDAP for o único método de autenticação configurado, os usuários não conseguirão se conectar ao agente. A inválida ARNs pode ser causada por uma sintaxe de ARN malformada, referências a segredos inexistentes, segredos localizados em uma AWS região diferente da corretora ou permissões insuficientes de secretsmanager: na função do IAM. GetSecretValue

Diagnosticando e abordando RABBITMQ_INVALID_ARN_LDAP

Para diagnosticar e resolver o código necessário da ação RABBITMQ_INVALID_ARN_LDAP, você deve usar o Amazon Logs e o console. CloudWatch

Para resolver o problema de ARN LDAP inválido

1. Navegue até o Amazon CloudWatch Logs Insights e execute a seguinte consulta no grupo de registros do seu corretor`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
```

```
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Procure mensagens de erro semelhantes a:

```
[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. Verifique o segredo do Secrets Manager e corrija quaisquer problemas, como:
   - Verifique se o segredo existe na mesma AWS região do corretor
   - Confirme se a sintaxe do ARN está correta
   - Certifique-se de que a função do IAM tenha secretsmanager: permissões GetSecretValue
4. Valide a correção usando o endpoint da API de validação de acesso ARN antes de atualizar a configuração do broker.
5. Atualize a configuração do broker e reinicialize o broker.

RabbitMQ no Amazon MQ: ARN HTTP inválido

O RabbitMQ no Amazon MQ gerará um código obrigatório de ação crítica INVALID_ARN_HTTP quando um ou mais certificados SSL ou arquivo de chave para HTTP auth_backend forem ARNs inválidos ou inacessíveis. Isso se aplica aos ARNS especificados em`aws.arns.auth_http.ssl_options.cacertfile`, `aws.arns.auth_http.ssl_options.certfile` ou`aws.arns.auth_http.ssl_options.keyfile`, que devem fazer referência a objetos AWS Secrets Manager e segredos do Amazon S3 contendo certificados e chave privada.

Um agente na quarentena do RABBITMQ_INVALID_ARN_HTTP não pode se autenticar por meio do servidor HTTP. Se o HTTP for o único método de autenticação configurado, os usuários não conseguirão se conectar ao agente. A inválida ARNs pode ser causada por uma sintaxe de ARN malformada, referências a segredos inexistentes, segredos localizados em uma AWS região diferente da corretora ou permissões s3: /secretsmanager: GetObject insuficientes na função do IAM. GetSecretValue

Diagnosticando e abordando RABBITMQ_INVALID_ARN_HTTP

Para diagnosticar e resolver o código necessário da ação RABBITMQ_INVALID_ARN_HTTP, você deve usar o Amazon Logs e o console. CloudWatch

Para resolver o problema de ARN HTTP inválido

1. Navegue até o Amazon CloudWatch Logs Insights e execute a seguinte consulta no grupo de registros do seu corretor`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Procure mensagens de erro semelhantes a:

```
[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. Verifique o segredo do S3 Object/Secrets Manager e corrija quaisquer problemas, como:
   - Verifique se o recurso existe na mesma AWS região do corretor
   - Confirme se a sintaxe do ARN está correta
   - Certifique-se de que a função do IAM tenha as permissões s3: GetObject e secretsmanager: GetSecretValue
4. Valide a correção usando o endpoint da API de validação de acesso ARN antes de atualizar a configuração do broker.
5. Atualize a configuração do broker e reinicialize o broker.

RabbitMQ no Amazon MQ: ARN SSL inválido

O RabbitMQ no Amazon MQ gerará um código de ação crítica INVALID_ARN_SSL exigido quando um ou mais ARNs repositórios confiáveis de certificados da CA para EXTERNAL

auth_mechanism forem inválidos ou inacessíveis. Isso se aplica ao ARNS especificado em `aws.arns.ssl_options.cacertfile` ou`aws.arns.management.ssl.cacertfile`, que deve fazer referência ao objeto Amazon S3 ou ACM PCA contendo o certificado.

Um agente na quarentena do RABBITMQ_INVALID_ARN_SSL não pode autenticar certificados de clientes durante apertos de mão TLS mútuos porque nenhum armazenamento confiável válido está configurado. Se o mecanismo de autenticação EXTERNO for o único método de autenticação configurado, os usuários não conseguirão se conectar ao agente. A inválida ARNs pode ser causada por uma sintaxe de ARN malformada, referências a objetos S3 inexistentes, objetos S3 localizados em uma AWS região diferente da corretora ou permissões s3: /acm-pca: insuficientes na função do IAM. GetObject GetCertificateAuthorityCertificate

Diagnosticando e endereçando RABBITMQ_INVALID_ARN_SSL

Para diagnosticar e resolver o código obrigatório da ação RABBITMQ_INVALID_ARN_SSL, você deve usar o Amazon Logs e o console. CloudWatch

Para resolver o problema de SSL ARN inválido

1. Navegue até o Amazon CloudWatch Logs Insights e execute a seguinte consulta no grupo de registros do seu corretor`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Procure mensagens de erro semelhantes a:

```
[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. Verifique o objeto S3/ACM-PCA e corrija quaisquer problemas, como:

   - Verifique se o segredo existe na mesma AWS região do corretor
   - Confirme se a sintaxe do ARN está correta

- Certifique-se de que a função do IAM tenha permissões s3: GetObject /acm-pca: GetCertificateAuthorityCertificate

4. Valide a correção usando o endpoint da API de validação de acesso ARN antes de atualizar a configuração do broker.
5. Atualize a configuração do broker e reinicialize o broker.

RabbitMQ no Amazon MQ: ARN inválido

O RabbitMQ no Amazon MQ gerará um código de ação crítica INVALID_ARN necessário quando um ou mais ARNs configurados no broker forem inválidos ou inacessíveis. Isso se aplica ao ARNs uso de certificados SSL, AWS Secrets Manager segredos, objetos do Amazon S3 ou AWS outras referências de recursos não cobertas por códigos de quarentena mais específicos, como RABBITMQ_INVALID_ARN_LDAP ou RABBITMQ_INVALID_ASSUMEROLE.

Um corretor na quarentena do RABBITMQ_INVALID_ARN pode ter uma funcionalidade degradada dependendo de quais são inválidos. ARNs Os recursos que dependem dos recursos inacessíveis não estarão disponíveis e o agente registrará erros indicando qual ARN não conseguiu resolver. O impacto na disponibilidade da corretora depende se o ARN inválido é necessário para operações críticas da corretora.

Diagnosticando e abordando RABBITMQ_INVALID_ARN

Para diagnosticar e resolver o código de ação obrigatório do RABBITMQ_INVALID_ARN, você deve usar o Amazon CloudWatch Logs e o console de serviço apropriado para o recurso afetado. AWS

Para resolver o problema de ARN inválido

1. Navegue até o Amazon CloudWatch Logs Insights e execute a seguinte consulta no grupo de registros do seu corretor`/aws/amazonmq/broker/<broker-id>/general`:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Procure mensagens de erro semelhantes a:

```
[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. Verifique o AWS recurso e corrija quaisquer problemas, como:
   - Verifique se o recurso existe na mesma AWS região do corretor
   - Confirme se a sintaxe do ARN está correta
   - Certifique-se de que a função do IAM tenha as permissões apropriadas para acessar o recurso
4. Valide a correção usando o endpoint da API de validação de acesso ARN antes de atualizar a configuração do broker.
5. Atualize a configuração do broker e reinicialize o broker.

Recursos relacionados

Recursos do Amazon MQ

A tabela a seguir lista recursos úteis para trabalhar com o Amazon MQ .

Recurso	Descrição
Referência da API REST do Amazon MQ	Descrições de recursos REST, solicitações de exemplo, métodos HTTP, esquemas, parâmetros e erros que o serviço retorna.
Amazon MQ na Referência de comandos da AWS CLI	Descrições dos comandos da AWS CLI que você pode usar para trabalhar com agentes de mensagem.
Amazon MQ no Guia do usuário do AWS CloudFormation	O recurso `AWS::Amazon MQ::Broker` permite criar agentes do Amazon MQ, adicionar alterações de configuração ou modificar usuários para o agente especificado, retornar informações sobre o agente especificado e excluir o agente especificado. O recurso `AWS::Amazon MQ::Configuration` permite que você crie configurações do Amazon MQ, adicione alterações de configuração ou modifique usuários e retorne informações sobre a configuração especificada.
Regiões e endpoints da	Informações sobre regiões e endpoints do Amazon MQ
Página do produto	A principal página da Web para obter informações sobre o Amazon MQ.

Recurso	Descrição
Fórum de discussão	Um fórum comunitário para que os desenvolvedores discutam questões técnicas relacionadas ao Amazon MQ.
AWS Informações sobre o Premium Support	A principal página da Web para obter informações sobre o AWS Premium Support, um canal de suporte de resposta rápida e com atendimento individual, para ajudá-lo a desenvolver e executar aplicações nos serviços de infraestrutura da AWS

Recursos do Amazon MQ para ActiveMQ

A tabela a seguir lista os recursos úteis para trabalhar com o Apache ActiveMQ.

Recurso	Descrição
Guia de conceitos básicos do Apache ActiveMQ	A documentação oficial do Apache ActiveMQ.
ActiveMQ em ação	Um guia para o Apache ActiveMQ que abrange a anatomia das mensagens JMS, conectores, persistência de mensagem, autenticação e autorização.
Clientes interlinguagem	Uma lista de linguagens de programação e bibliotecas Apache ActiveMQ correspondentes. Consulte também Cliente ActiveMQ e Cliente QpidJMS.

Recursos do Amazon MQ para RabbitMQ

A tabela a seguir lista os recursos úteis para trabalhar com o RabbitMQ.

Recurso	Descrição
The RabbitMQ Getting Started Guide	A documentação oficial do RabbitMQ.
RabbitMQ Client Libraries and Developer Tools	Um guia para as bibliotecas clientes oficialmente compatíveis e ferramentas de desenvolvedor para trabalhar com RabbitMQ usando uma variedade de linguagens e plataformas de programação.
RabbitMQ Best Practices	O guia do CloudAMQP sobre práticas recomendadas e recomendações para trabalhar com o RabbitMQ.

Notas de lançamento do Amazon MQ

A tabela a seguir relaciona as versões e as melhorias de recursos do Amazon MQ.

Data	Atualização da documentação
19 de fevereiro de 2026	Amazon MQ O Amazon MQ agora oferece suporte ao ActiveMQ 5.19, uma nova versão secundária do mecanismo. Para obter mais informações, consulte . • Página de lançamento do ActiveMQ 5.19 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Atualizando uma versão do mecanismo de agente do Amazon MQ • Usar arquivos de configuração XML do Spring
22 de janeiro de 2026	O Amazon MQ agora oferece suporte ao plug-in de troca de tópicos JMS para corretores no RabbitMQ 4.2 e versões posteriores. Você pode usar o cliente JMS oficial do RabbitMQ para executar cargas de trabalho JMS no Amazon MQ for RabbitMQ broker. Ele suporta JMS 1.1, 2.0 e 3.1. Para obter mais informações, consulte . • Especificação oficial do JMS 2.0 (compatível com versões anteriores e JMS 1.1 estendido) • Especificação oficial do JMS 3.1 • Limitação do cliente RabbitMQ JMS • Conectando seu aplicativo JMS ao Amazon MQ for RabbitMQ broker
8 de janeiro de 2026	O Amazon MQ agora oferece suporte à autenticação de certificado SSL para corretores no RabbitMQ 4.2 e superior usando certificados de cliente X.509 e configuração de TLS mútuo (mTLS). Você pode configurar a autenticação de certificados SSL e mTLS por meio de Console de gerenciamento da AWS,, AWS CloudFormation AWS CLI, ou AWS CDK em todos os Regiões da AWS lugares onde o Amazon MQ está disponível.

Data	Atualização da documentação
	Para obter mais informações, consulte Autenticação de certificado SSL e Configuração do mTLS.
6 de janeiro de 2026	O Amazon MQ agora oferece suporte à autenticação e autorização HTTP para corretores no RabbitMQ 4.2 e superior com servidores HTTP externos. Você pode configurar a autenticação HTTP por meio de Console de gerenciamento da AWS, AWS CloudFormation AWS CLI, ou AWS CDK em todos os Regiões da AWS lugares onde o Amazon MQ está disponível. Para obter mais informações, consulte Autenticação e autorização HTTP.
20 de novembro de 2025	O Amazon MQ agora oferece suporte ao RabbitMQ 4.2, uma nova versão principal que introduz suporte nativo para o protocolo AMQP 1.0, um novo repositório de metadados Khepri baseado em Raft, escavadeiras locais e prioridades de mensagens para filas de quórum. O RabbitMQ 4.2 também inclui várias correções de erros e melhorias de desempenho para gerenciam ento de produtividade e memória. Embora esta versão introduza novos recursos, há algumas mudanças importantes. Para obter mais informações, consulte . • RabbitMQ 4 • Notas de lançamento do RabbitMQ de código aberto • Configurando limites de recursos • Protocolos suportados • Atualizações de versão do Amazon MQ
18 de novembro de 2024	O Amazon MQ agora oferece suporte a instâncias m7g baseadas em Graviton3 para RabbitMQ em uma variedade de tamanhos, de médio a 16 vezes maior, na África (Cidade do Cabo). Para obter mais informações, consulte Tipos de instância do agente do Amazon MQ para RabbitMQ.

Data	Atualização da documentação
17 de novembro de 2025	O Amazon MQ agora oferece suporte à autenticação e autorização LDAP para agentes RabbitMQ com serviços de diretório LDAP externos. Você pode configurar o LDAP por meio de Console de gerenciamento da AWS, AWS CloudFormation AWS CLI, ou AWS CDK em todos os Regiões da AWS lugares onde o Amazon MQ está disponível. Para obter mais informações, consulte Autenticação e autorização LDAP para Amazon MQ para RabbitMQ.
22 de outubro de 2025	Agora, o Amazon MQ está disponível na região da Ásia-Pacífico (Nova Zelândia). Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .
3 de setembro de 2025	O Amazon MQ agora oferece suporte à autenticação e autorização OAuth 2.0 para corretores RabbitMQ com provedores de identidade pública (). IdPs Você pode configurar OAuth 2.0 por meio de Console de gerenciamento da AWS, AWS CloudFormation AWS CLI, ou AWS CDK em todos os Regiões da AWS lugares onde o Amazon MQ está disponível. Para obter mais informações, consulte OAuth Autenticação e autorização 2.0 para Amazon MQ para RabbitMQ.

Data	Atualização da documentação
22 de julho de 2025	Agora, o Amazon MQ oferece suporte a instâncias de `m7g` baseadas em Graviton3 para RabbitMQ em uma variedade de tamanhos, desde médio a 16 vezes maior. Os clusters do RabbitMQ executados nas instância do `m7g` oferecem capacidade de workload até 50% maior e melhorias no throughput de até 85% em relação aos clusters comparáveis do Amazon MQ para RabbitMQ executados em instâncias do `m5`. As instâncias do `M7g` também têm tamanhos de volume de disco otimizados que variam de acordo com o tamanho da instância. Para obter mais informações, consulte Broker instance types. As instâncias de `M7g` no Amazon MQ já estão disponíveis em todas as regiões geralmente disponíveis, exceto as regiões África (Cidade do Cabo), Oeste do Canadá (Calgary) e Europa (Milão).
8 de julho de 2025	Agora, o Amazon MQ está disponível na região da Ásia-Pacífico (Taipei). Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .
22 de abril de 2025	Você pode excluir as configurações do agente do Amazon MQ usando a API do `DeleteConfiguration` . Para obter mais informações, consulte Configurações na Referência da API do Amazon MQ.
16 de abril de 2025	O Amazon MQ para RabbitMQ agora oferece suporte ao uso de endpoints de pilha dupla (IPv4 e IPv6) para se conectar a corretores públicos e privados. Para obter mais informações, consulte Connecting to Amazon MQ e Configuring a private Amazon MQ broker.
7 de abril de 2025	Agora, o Amazon MQ está disponível nas regiões da Ásia-Pacífico (Tailândia) e México (Centro). Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .

Data	Atualização da documentação
13 de fevereiro de 2025	Os endpoints FIPS da API do Amazon MQ já estão disponíveis nas regiões Canadá (Central) e Oeste do Canadá (Calgary). Para saber mais sobre o uso de endpoints FIPS com a API do Amazon MQ, consulte. Connecting to Amazon MQ Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .
12 de fevereiro de 2025	O Amazon MQ está anunciando as seguintes datas de fim do suporte a tipo de instância: Broker instance types • `mq.t2.micro` do ActiveMQ: 12 de maio de 2025 • `mq.m4.large` do ActiveMQ: 12 de maio de 2025 Não é possível criar agentes em `mq.t2.micro` ou `mq.m4.large` após 17 de março de 2025.
10 de dezembro de 2024	O Amazon MQ agora suporta o uso AWS PrivateLink para se conectar entre suas nuvens privadas virtuais (VPCs) e a API do Amazon MQ sem expor seu tráfego à Internet pública. Para obter mais informações, consulte the section called “Conectar-se ao Amazon MQ usando o AWS PrivateLink”.
18 de novembro de 2024	Agora, o Amazon MQ está disponível na região Ásia-Pacífico (Malásia). Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .

Data	Atualização da documentação
14 de novembro de 2024	O Amazon MQ está anunciando as seguintes datas de fim do suporte à versão do mecanismo: Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • ActiveMQ 5.17:16 de junho de 2025 Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ • RabbitMQ 3.11:17 de fevereiro de 2025 • RabbitMQ 3.12:17 de março de 2025 Para obter mais informações sobre como atualizar para a versão mais recente, consulte Atualizando uma versão do mecanismo de agente do Amazon MQ
13 de novembro de 2024	O Amazon MQ agora oferece suporte a endpoints de serviço de pilha dupla aos quais você pode se conectar usando um ou. IPv4 IPv6 Os endpoints dos serviços regionais de pilha dupla do Amazon MQ podem ser resolvidos com os dois registros de DNS. A e AAAA. Para obter mais informações, consulte ???.
25 de julho de 2024	O Amazon MQ agora é compatível com o ActiveMQ 5.18, uma nova versão secundária do mecanismo. Para saber mais, consulte: • Página da versão do ActiveMQ 5.18 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Atualizando uma versão do mecanismo de agente do Amazon MQ • Usar arquivos de configuração XML do Spring

Data	Atualização da documentação
22 de julho de 2024	O Amazon MQ agora permite filas de quórum somente em agentes que usam as versões 3.13 e posteriores. As filas de quórum são um tipo de fila FIFO replicada que usa o algoritmo de consenso Raft para manter a consistência de dados. As filas de quórum oferecem tratamento de mensagens mal-intencionadas, o que pode ajudar você a gerenciar mensagens não processadas. Para começar a usar as filas de quórum, consulte Filas de quórum do RabbitMQ no Amazon MQ.
2 de julho de 2024	O Amazon MQ para RabbitMQ agora é compatível com a versão secundária 3.13 do RabbitMQ. Para todos os agentes que usam a versão 3.13 e posterior do mecanismo, o Amazon MQ gerencia as atualizações para a versão de patch mais recente compatível durante a janela de manutenção. Para obter mais informações, consulte Atualizando uma versão do mecanismo de agente do Amazon MQ. As Diretrizes de dimensionamento do Amazon MQ para RabbitMQ foram atualizadas para incluir novos limites para filas, consumidores por canal e shovels para agentes que usam a versão 3.13 do mecanismo. Para obter mais informações sobre as correções e os recursos desta versão, consulte as notas de lançamento do RabbitMQ 3.13 no repositório do servidor RabbitMQ. GitHub Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.
10 de junho de 2024	O Amazon MQ já está disponível na região Oeste do Canadá (Calgary). Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .

Data	Atualização da documentação
10 de maio de 2024	O calendário de suporte da versão do Amazon MQ indica quando uma versão do mecanismo do agente chega ao fim do suporte. Quando uma versão do mecanismo chega ao fim do suporte, o Amazon MQ atualiza automatic amente todos os agentes dessa versão para a próxima versão secundária compatível. O Amazon MQ avisa com pelo menos noventa dias de antecedên cia quando uma versão do mecanismo chegará ao fim do suporte. Para ver o calendário de suporte da versão e o fim do suporte, consulte o seguinte: • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ Você também pode habilitar atualizações automáticas de versões secundári as para que o agente atualize para a próxima versão de patch durante uma janela de manutenção. Para obter mais informações, consulte Atualizando uma versão do mecanismo de agente do Amazon MQ.

Data	Atualização da documentação
9 de maio de 2024	O Amazon MQ para RabbitMQ agora é compatível com a versão secundári a 3.12 do RabbitMQ. Todos os corretores em 3.12.13 e superiores usam Classic Queues versão 2 (CQv2), e todas as filas em 3.12.13 e superiores se comportam como filas preguiçosas. Recomendamos que os corretores nas versões anteriores à 3.12.13 habilitem CQv2 e lazy queues ou atualizem para a versão mais recente do Amazon MQ para RabbitMQ. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.12 no repositório do servidor RabbitMQ. GitHub Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.
4 de março de 2024	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ 3.11.28. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.11.28 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.

Data	Atualização da documentação
19 de janeiro de 2024	O Amazon MQ para RabbitMQ não permite o nome de usuário “convidado” e excluirá a conta de convidado padrão quando você criar um agente. O Amazon MQ também excluirá periodicamente qualquer conta de “convidado” criada pelo cliente.
15 de dezembro de 2023	O Amazon MQ agora está disponível na região de Israel (Tel Aviv). Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .
11 de dezembro de 2023	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ 3.10.25. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.10.25 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.
26 de outubro de 2023	O Amazon MQ lançou as versões secundárias mais recentes do ActiveMQ 5.15.16, 5.16.7, 5.17.6 com uma atualização importante. Descontinuamos as versões secundárias mais antigas do ActiveMQ e atualizaremos todos os agentes em todas as versões de 5.15 a 5.15.16 ou de 5.16 a 5.16.7 e de 5.17 a 5.17.6. Para receber mais informações sobre como atualizar o agente do ActiveMQ, consulte Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ.

Data	Atualização da documentação
27 de setembro de 2023	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ 3.11.20. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.11.20 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.
27 de julho de 2023	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ 3.11.16 Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.11.16 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.
27 de julho de 2023	O Amazon MQ para RabbitMQ agora permite a criação e a aplicação de configurações ao agente do RabbitMQ. Para obter mais informações sobre como adicionar configurações ao agente, consulte RabbitMQ Broker Configurations. Para obter mais informações sobre esse atributo, consulte: • Políticas do operador • Mudanças nas políticas do operador

Data	Atualização da documentação
23 de junho de 2023	O Amazon MQ agora é compatível com o ActiveMQ 5.17.3, uma nova versão secundária do mecanismo. Esta versão é compatível com o novo atributo de replicação de dados entre regiões (CRDR) do Amazon MQ. Para saber mais, consulte: • Para começar a usar a CRDR, consulte Replicação de dados entre regiões para o Amazon MQ for ActiveMQ no Guia do desenvolvedor. • Página da versão do ActiveMQ 5.17.3 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Atualizando uma versão do mecanismo de agente do Amazon MQ • Usar arquivos de configuração XML do Spring
21 de junho de 2023	O Amazon MQ for ActiveMQ agora oferece um recurso de replicação de dados entre regiões (CRDR) que permite a replicação assíncrona de mensagens do agente principal em uma região primária para o agente de réplica em uma região de réplica. AWS Se o agente primário na região primária falhar, você poderá promover o agente de réplica na região secundária para primário iniciando uma transição ou um failover. Para começar a usar a CRDR, consulte Replicação de dados entre regiões para o Amazon MQ for ActiveMQ no Guia do desenvolvedor.
18 de maio de 2023	O Amazon MQ já está disponível nas seguintes regiões: • Ásia-Pacífico (Melbourne) • Ásia-Pacífico (Hyderabad) • Europa (Espanha) • Europa (Zurique) Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .

Data	Atualização da documentação
14 de abril de 2023	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.9.27. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.9.27 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.
14 de abril de 2023	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.10.20. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.10.20 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.

Data	Atualização da documentação
31 de março de 2023	O Amazon MQ para RabbitMQ desativou o RabbitMQ versão de mecanismo 3.10.17 A equipe do Amazon MQ para RabbitMQ e os mantenedores de código aberto do RabbitMQ identificaram um problema com o console de gerenciamento do RabbitMQ na versão 3.10.17. O Amazon MQ retirou essa versão. Para mitigar os impactos desse problema, crie outros agentes com a versão 3.10.10 enquanto trabalhamos para oferecer suporte a uma nova versão de patch do RabbitMQ. Recomendamos ativar a opção de atualização de versão para obter automaticamente as últimas correções de erros, atualizações de segurança e aprimoramentos de desempenho. Para obter mais informações sobre as versões disponíveis do Amazon MQ para RabbitMQ, consulte Versões de mecanismo do Amazon MQ para RabbitMQ.
1 de março de 2023	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.10.17. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.10.17 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.

Data	Atualização da documentação
21 de fevereiro de 2023	O Amazon MQ para RabbitMQ agora se integra ao AWS Key Management Service (KMS) para oferecer criptografia no lado do servidor. Agora você pode selecionar sua própria CMK gerenciada pelo cliente ou usar uma chave KMS AWS gerenciada em sua AWS KMS conta. Para obter mais informações, consulte Criptografia em repouso. O Amazon MQ oferece suporte ao uso de AWS KMS chaves das seguintes formas. • Chave do KMS pertencente ao Amazon MQ (padrão): a chave pertence ao Amazon MQ e é gerenciada por ele e não está na sua conta. • AWS chave KMS gerenciada — A chave KMS AWS gerenciada (`aws/mq`) é uma chave KMS em sua conta que é criada, gerenciada e usada em seu nome pelo Amazon MQ. • Selecione uma chave KMS gerenciada pelo cliente — KMSs gerenciadas pelo cliente são criadas e gerenciadas por você no AWS Key Management Service (KMS).
13 de janeiro de 2023	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.8.34. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.8.34 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.

Data	Atualização da documentação
15 de dezembro de 2022	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.9.24. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.9.24 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.
13 de dezembro de 2022	O Amazon MQ já está disponível na região Oriente Médio (EAU). Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .
14 de novembro de 2022	O Amazon MQ para RabbitMQ agora é compatível com a versão 3.10, uma versão principal do mecanismo. Agora você pode habilitar as filas clássicas versão 2 (CQv2) em suas filas do RabbitMQ. Não é possível fazer atualizaç ões diretas da versão 3.8 para a 3.10. Para saber mais, consulte: • Notas de lançamento do RabbitMQ 3.10.10 • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.

Data	Atualização da documentação
9 de novembro de 2022	O Amazon MQ agora é compatível com o ActiveMQ 5.17.2, uma nova versão secundária do mecanismo. Para saber mais, consulte: • Página da versão do ActiveMQ 5.17.2 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Atualizando uma versão do mecanismo de agente do Amazon MQ • Usar arquivos de configuração XML do Spring
17 de agosto de 2022	O Amazon MQ agora é compatível com o ActiveMQ 5.17.1, uma nova versão principal do mecanismo. Para saber mais, consulte: • Página da versão do ActiveMQ 5.17.1 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Atualizando uma versão do mecanismo de agente do Amazon MQ • Usar arquivos de configuração XML do Spring
14 de julho de 2022	O Amazon MQ agora é compatível com o ActiveMQ 5.16.5, uma versão secundária do mecanismo. Para saber mais, consulte: • Página da versão do ActiveMQ 5.16.5 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Usar arquivos de configuração XML do Spring • Atualizando uma versão do mecanismo de agente do Amazon MQ
4 de maio de 2022	O Amazon MQ contém linguagem inclusiva para o elemento `networkConnector` na configuração do agente. • Criar e configurar uma rede de agentes do Amazon MQ

Data	Atualização da documentação
25 de abril de 2022	Amazon MQ Esta versão inclui o estado do agente `CRITICAL_ACTION_REQUIRED` e a propriedade `ActionRequired` da API. `CRITICAL_ACTION_REQUIRED` informa quando seu agente está degradado. `ActionRequired` fornece um código que você pode usar para encontrar instruções no Guia do desenvolvedor sobre como resolver o problema. • Solução de problemas • Documentação de `ActionRequired` na Referência da API do Amazon MQ.
20 de abril de 2022	O Amazon MQ agora é compatível com o ActiveMQ 5.16.4, uma nova versão secundária do mecanismo. Para saber mais, consulte: • Página da versão do ActiveMQ 5.16.4 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Usar arquivos de configuração XML do Spring • Atualizando uma versão do mecanismo de agente do Amazon MQ
1º de março de 2022	O Amazon MQ agora está disponível na região Ásia-Pacífico (Jacarta). Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .
25 de fevereiro de 2022	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.8.27. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.8.27 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.

Data	Atualização da documentação
16 de fevereiro de 2022	O Amazon MQ agora está disponível na Região da África (Cidade do Cabo). Para obter informações sobre as regiões disponíveis, consulte Regiões e endpoints da AWS na Guia de Referência geral da AWS .
14 de fevereiro de 2022	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.9.13. Atualizações da versão secundária automáticas não podem ser usadas para atualizar do Rabbit 3.8 para 3.9. Para isso,atualize manualmente seu agente. Para obter mais informações sobre os novos recursos introduzidos no RabbitMQ 3.9, consulte a página de notas de lançamento da versão 3.9.0 no site. GitHub **Note** Atualmente, o Amazon MQ não é compatível com transmissões, ou usando registro estruturado em JSON, introduzido no RabbitMQ 3.9. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.9.13 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.

Data	Atualização da documentação
07 de fevereiro de 2022	O Amazon MQ for RabbitMQ apresenta novas métricas de agente, permitind o que você monitore a utilização média de recursos em todos os três nós em uma implantação de cluster. Para saber mais, consulte: • the section called “Métricas para o RabbitMQ”
18 de janeiro de 2022	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.8.26. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.8.26 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.
13 de janeiro de 2022	O Amazon MQ introduz o código de status `RABBITMQ_MEMORY_ALARM` para informar você quando seu agente gerou um alarme de alta memória e se encontra em estado não íntegro. O Amazon MQ fornece informações detalhadas e recomendações para ajudar você a diagnosticar, solucionar e evitar alarmes de alta memória. Para obter mais informações, consulte: • the section called “ `RABBITMQ_MEMORY_ALARM` ”

Data	Atualização da documentação
6 de janeiro de 2022	Quando você configura o CloudWatch Logs for Amazon MQ para agentes do ActiveMQ, o Amazon MQ suporta o uso das chaves de contexto de condição global `aws:SourceAccount` e `aws:SourceArn` das políticas baseadas em recursos do IAM para evitar o confuso problema adjunto. Para obter mais informações, consulte. • the section called “Prevenção contra o ataque do “substituto confuso” em todos os serviços”
20 de dezembro de 2021	O Amazon MQ for ActiveMQ introduz um conjunto de novas métricas, permitindo que você monitore o número máximo de conexões que pode fazer com seu agente usando diferentes protocolos de transporte com suporte, bem como uma nova métrica adicional que permite monitorar o número de nós conectados ao seu agente em uma rede de agentes. Para obter mais informações, consulte: • the section called “Métricas para o ActiveMQ”
16 de novembro de 2021	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.8.23. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.8.23 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ.

Data	Atualização da documentação
12 de outubro de 2021	O Amazon MQ agora é compatível com o ActiveMQ 5.16.3, uma nova versão secundária do mecanismo. Para saber mais, consulte: • Página da versão do ActiveMQ 5.16.3 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Atualizando uma versão do mecanismo de agente do Amazon MQ • Usar arquivos de configuração XML do Spring
8 de setembro de 2021	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.8.22. Esta versão inclui uma correção para um problema com filas usando TTL por mensagem (vida útil), identificado na versão anteriormente compatível, RabbitMQ 3.8.17. Recomendamos atualizar seus agentes existentes para a versão 3.8.22. Para obter mais informações sobre as correções e recursos nesta versão, consulte: • Notas de lançamento do RabbitMQ 3.8.22 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ Para obter mais informações sobre as versões compatíveis do Amazon MQ for RabbitMQ e atualizações de agentes, consulte Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ
25 de agosto de 2021	O Amazon MQ para RabbitMQ desativou temporariamente a versão 3.8.17 do mecanismo RabbitMQ devido a um problema identificado com filas usando por mensagem (TTL). time-to-live Recomendamos usar a versão 3.8.11.

Data	Atualização da documentação
29 de julho de 2021	O Amazon MQ para RabbitMQ agora é compatível com o RabbitMQ versão 3.8.17. Para obter mais informações sobre as correções e recursos contidos nesta atualização, consulte: • Notas de lançamento do RabbitMQ 3.8.17 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ • Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ
16 de julho de 2021	Agora você pode ajustar a janela de manutenção de um agente do Amazon MQ usando o Console de gerenciamento da AWS, AWS CLI, ou a API do Amazon MQ. Para saber mais sobre as janelas de manutenção do agente, consulte o seguinte. • Agendar a janela de manutenção para um agente do Amazon MQ
6 de julho de 2021	O Amazon MQ para RabbitMQ oferece compatibilidade com o tipo de troca Consistent Hash. As trocas de hash consistentes fazem o roteamento de mensagens para filas com base em um valor de hash calculado a partir da routing key (chave de roteamento) de uma mensagem. Para saber mais, consulte: • Plugin de troca de hash consistente • Tipo de troca de hash consistente do RabbitMQ no repositório do RabbitMQ GitHub
7 de junho de 2021	O Amazon MQ agora é compatível com o ActiveMQ 5.16.2, uma nova versão principal do mecanismo. Para saber mais, consulte: • Página da versão do ActiveMQ 5.16.2 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Atualizando uma versão do mecanismo de agente do Amazon MQ • Usar arquivos de configuração XML do Spring

Data	Atualização da documentação
26 de maio de 2021	O Amazon MQ para RabbitMQ agora está disponível nas regiões China (Pequim) e China (Ningxia). Para obter informações sobre regiões disponíveis, consulte AWS Regiões e endpoints.
18 de maio de 2021	O Amazon MQ para RabbitMQ implementa os padrões do agente. Quando você cria um agente pela primeira vez, o Amazon MQ cria um conjunto de políticas de agente e limites de vhost com base no tipo de instância e no modo de implantação escolhidos, a fim de otimizar a performance do agente. Para obter mais informações, consulte o seguinte: https://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html
5 de maio de 2021	O Amazon MQ agora é compatível com o ActiveMQ 5.15.15. Para saber mais, consulte: • Página da versão do ActiveMQ 5.15.15 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Usar arquivos de configuração XML do Spring
5 de maio de 2021	O Amazon MQ começou a monitorar as alterações nas políticas AWS gerenciadas. Para saber mais, consulte: • the section called “AWS políticas gerenciadas”
14 de abril de 2021	O Amazon MQ agora está disponível nas regiões China (Pequim) e China (Ningxia). Para obter informações sobre regiões disponíveis, consulte AWS Regiões e endpoints.

Data	Atualização da documentação
7 de abril de 2021	O Amazon MQ agora é compatível com o RabbitMQ 3.8.11. Para obter mais informações sobre as correções e recursos contidos nesta atualização, consulte: • Notas de lançamento do RabbitMQ 3.8.11 no repositório do servidor RabbitMQ GitHub • Log de alterações do RabbitMQ • Gerenciando o Amazon MQ para versões do mecanismo RabbitMQ
1.º de abril de 2021	O Amazon MQ agora está disponível na região Ásia-Pacífico (Osaka). Para obter mais informações sobre as regiões e os endpoints disponíveis consulte Regiões e endpoints do Amazon MQ.
21 de dezembro de 2020	O Amazon MQ agora é compatível com o ActiveMQ 5.15.14. Para saber mais, consulte: • Notas de lançamento do ActiveMQ 5.15.14 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Usar arquivos de configuração XML do Spring • **Important** Devido a um problema conhecido do Apache ActiveMQ nesta versão, o novo botão Pausar Fila no console da Web ActiveMQ não pode ser usado com o Amazon MQ para agentes ActiveMQ. Para obter mais informações sobre esse problema, consulte AMQ-8104.

Data	Atualização da documentação
4 de novembro de 2020	O Amazon MQ agora é compatível com o RabbitMQ, o popular agente de mensagens de código aberto. Isso permite que você migre seus agentes de mensagens existentes do RabbitMQ AWS sem precisar reescrever o código. O Amazon MQ para RabbitMQ gerencia agentes de mensagens individua is e em cluster e lida com tarefas como provisionamento da infraestrutura, configuração do agente e atualização do software. • O Amazon MQ é compatível com o RabbitMQ 3.8.6. Para obter mais informações sobre as versões de engine compatíveis, consulte the section called “Gerenciamento de versão”. • O AWS Nível gratuito inclui até 750 horas de uma única instância de agente `mq.t3.micro` e até 20GB de armazenamento por mês durante um ano. Para obter mais informações sobre os tipos de instâncias compatíveis, consulte Broker instance types. • Com o Amazon MQ para RabbitMQ, você pode acessar seus agentes usando AMQP 0-9-1 com qualquer linguagem compatível com as bibliotec as de cliente do RabbitMQ. Para obter mais informações sobre protocolos e portas compatíveis, consulte the section called “Amazon MQ para protocolo s RabbitMQ”. • O Amazon MQ para RabbitMQ está disponível em todas as regiões em que o Amazon MQ está disponível no momento. Para saber mais sobre todas as regiões disponíveis, consulte a AWS Tabela de região. Para começar a usar o Amazon MQ, criar um agente e conectar uma aplicação baseada em JVM ao seu agente RabbitMQ, consulte Conceitos básicos: criar e conectar a um agente do RabbitMQ.
22 de outubro de 2020	O Amazon MQ é compatível com o ActiveMQ 5.15.13. Para saber mais, consulte: • Notas de lançamento do ActiveMQ 5.15.13 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Usar arquivos de configuração XML do Spring

Data	Atualização da documentação
30 de setembro de 2020	O Amazon MQ agora está disponível na Região da Europa (Milão). Para obter mais informações sobre as regiões e os endpoints disponíveis consulte Regiões e endpoints do Amazon MQ.
27 de julho de 2020	Você pode autenticar usuários do Amazon MQ usando as credenciais armazenadas no seu Active Directory ou em outro servidor LDAP. Você também pode adicionar, excluir e modificar usuários do Amazon MQ e atribuir permissões a tópicos e filas. Para obter mais informações, consulte Integrar LDAP com ActiveMQ.
17 de julho de 2020	O Amazon MQ agora é compatível com o tipo de instância `mq.t3.micro` . Para obter mais informações, consulte Broker instance types.
30 de junho de 2020	O Amazon MQ é compatível com o ActiveMQ 5.15.12. Para obter mais informações, consulte as informações a seguir: • Notas de lançamento do ActiveMQ 5.15.12 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Usar arquivos de configuração XML do Spring

Data	Atualização da documentação
30 de abril de 2020	O Amazon MQ é compatível com um novo elemento da coleção filho, `systemUsage` , no elemento `broker`. Para obter mais informações, consulte systemUsage. O Amazon MQ também é compatível com três novos atributos no `kahaDB` elemento filho. • `journalDiskSyncInterval` — intervalo (ms) para quando executar uma sincronização de disco se `journalDiskSyncStrategy=periodic` . • `journalDiskSyncStrategy` — configura a política de sincronização de disco. • `preallocationStrategy` — configura como o agente tentará pré-alocar os arquivos do diário quando um novo arquivo do diário for necessário. Para obter mais informações, consulte Atributos.
3 de março de 2020	O Amazon MQ oferece suporte a duas novas métricas CloudWatch • `TempPercentUsage` — a porcentagem de armazenamento temporário disponível usada por mensagens não persistentes. • `JobSchedulerStorePercentUsage` — A porcentagem de espaço em disco usada pelo armazenamento do programador de trabalhos. Para obter mais informações, consulte Monitoring and logging Amazon MQ brokers.
4 de fevereiro de 2020	O Amazon MQ está disponível nas regiões Ásia-Pacífico (Hong Kong) e Oriente Médio (Bahrein) Para obter informações sobre regiões disponíveis, consulte AWS Regiões e endpoints.

Data	Atualização da documentação
22 de janeiro de 2020	O Amazon MQ é compatível com o ActiveMQ 5.15.10. Para saber mais, consulte: • Notas de lançamento do ActiveMQ 5.15.10 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Usar arquivos de configuração XML do Spring
19 de dezembro de 2019	O Amazon MQ está disponível nas regiões Europa (Estocolmo) e América do Sul (São Paulo) Para obter informações sobre regiões disponíveis, consulte AWS Regiões e endpoints.

Data	Atualização da documentação
16 de dezembro de 2019	O Amazon MQ é compatível com a criação de agentes otimizados para taxa de transferência usando o Amazon Elastic Block Store (EBS) — em vez do padrão Amazon Elastic File System (Amazon EFS) — para armazenam ento de agentes. Para aproveitar a alta durabilidade e a replicação em várias zonas de disponibilidade, use o Amazon EFS. Para aproveitar a baixa latência e alta taxa de transferência, use o Amazon EBS. **Important** • Você pode usar o Amazon EBS somente com a família `mq.m5` de tipo de instância de agente. • Embora você possa alterar o tipo de instância de agente, você não pode alterar o tipo de armazenamento do agente depois de criar o agente. • O Amazon EBS replica dados em uma única zona de disponibi lidade e não é compatível com o modo de implantação ativo/em espera do ActiveMQ. Para saber mais, consulte: • Storage • Escolha o tipo de armazenamento de agente correto para obter a melhor taxa de transferência • A propriedade `storageType` dos recursos `broker-instance-options` na Referência da API REST do Amazon MQ • As métricas `BurstBalance` , `VolumeReadOps` e `VolumeWriteOps` na seção Monitoring and logging Amazon MQ brokers.
18 de outubro de 2019	Duas CloudWatch métricas da Amazon estão disponíveis: `TotalEnqu eueCount` `TotalDequeueCount` e. Para obter mais informações, consulte Monitoring and logging Amazon MQ brokers.

Data	Atualização da documentação
11 de outubro de 2019	Agora o Amazon MQ é compatível com endpoints em conformidade com o Federal Information Processing Standard 140-2 (FIPS) em regiões comerciais dos EUA. Para obter mais informações, consulte: • Federal Information Processing Standard (FIPS) 140-2 • Regiões e endpoints do Amazon MQ
30 de setembro de 2019	Agora o Amazon MQ inclui a capacidade de escalar os agentes alterando o tipo de instância do host. Para obter mais informações, consulte a proprieda de `hostInstanceType` de `UpdateBrokerInput` e a propriedade `pendingHostInstanceType` de `DescribeBrokerOutput`.
30 de agosto de 2019	Agora é possível atualizar os grupos de segurança associados a um agente, tanto no console quanto no `UpdateBrokerInput`.
22 de julho de 2019	O Amazon MQ se integra ao AWS Key Management Service (KMS) para oferecer criptografia no lado do servidor. Agora você pode selecionar sua própria CMK gerenciada pelo cliente ou usar uma chave KMS AWS gerenciad a em sua AWS KMS conta. Para obter mais informações, consulte Criptogra fia em repouso. O Amazon MQ oferece suporte ao uso de AWS KMS chaves das seguintes formas. • AWS chave KMS de propriedade — A chave é de propriedade do Amazon MQ e não está na sua conta. • AWS chave KMS gerenciada — A chave KMS AWS gerenciada (aws/mq) é uma chave KMS em sua conta que é criada, gerenciada e usada em seu nome pelo Amazon MQ. • Selecione a CMK existente gerenciada pelo cliente — CMKs Os gerenciad os pelo cliente são criados e gerenciados por você em AWS Key Management Service (KMS).

Data	Atualização da documentação
19 de junho de 2019	O Amazon MQ está disponível nas regiões Europa (Paris) e Ásia-Pacífico (Mumbai). Para obter informações sobre regiões disponíveis, consulte AWS Regiões e endpoints.
12 de junho de 2019	O Amazon MQ está disponível na região Canadá (Central). Para obter informações sobre regiões disponíveis, consulte AWS Regiões e endpoints.
3 de junho de 2019	Duas novas CloudWatch métricas da Amazon estão disponíveis: `EstablishedConnectionsCount` `InactiveDurableSubscribers` e. Para saber mais, consulte: • Monitoring and logging Amazon MQ brokers • Monitoring and logging Amazon MQ brokers
10 de maio de 2019	O armazenamento de dados para novos tipos de instância `mq.t2.micro` é limitado a 20 GB. Para saber mais, consulte: • the section called “Armazenamento de dados” • Broker instance types
29 de abril de 2019	Agora você pode usar políticas baseadas em tags e permissões no nível de recursos. Para saber mais, consulte: • Como o Amazon MQ funciona com o IAM • Permissões no nível do recurso suportadas para ações de API do Amazon MQ
16 de abril de 2019	Agora você pode recuperar informações sobre opções de instâncias e mecanismo de agente usando a API REST. Para saber mais, consulte: • Opções de instâncias de agente • Tipos de mecanismo de agente

Data	Atualização da documentação
8 de abril de 2019	O Amazon MQ é compatível com o ActiveMQ 5.15.9. Para saber mais, consulte: • Notas de lançamento do ActiveMQ 5.15.9 • Gerenciar as versões do mecanismo do Amazon MQ para ActiveMQ • Usar arquivos de configuração XML do Spring
4 de março de 2019	A documentação para configurar o failover dinâmico e o rebalanceamento de clientes para uma rede de agentes foi aprimorada. Habilite o failover dinâmico configurando as opções de configuração do `transportConnectors` com `networkConnectors` . Para saber mais, consulte: • Failover dinâmico com conectores de transporte • Rede de agentes do Amazon MQ • Amazon MQ Broker Configuration Parameters
27 de fevereiro de 2019	O Amazon MQ está disponível na região Europa (Londres), além das seguintes regiões: • Ásia-Pacífico (Singapura) • Leste dos EUA (Ohio) • Leste dos EUA (Norte da Virgínia) • Oeste dos EUA (N. da Califórnia) • Oeste dos EUA (Oregon) • Ásia-Pacífico (Tóquio) • Ásia-Pacífico (Seul) • Ásia-Pacífico (Sydney) • Europa (Frankfurt) • Europa (Irlanda)
24 de janeiro de 2019	Agora, a configuração padrão inclui uma política para eliminar destinos inativos.

Data	Atualização da documentação
17 de janeiro de 2019	Agora os tipos de instância `mq.t2.micro` do Amazon MQ são compatíveis somente com 100 conexões por protocolo de nível de conexão. Para obter mais informações, consulte Quotas in Amazon MQ.
19 de dezembro de 2018	Você pode configurar uma série de agentes do Amazon MQ em uma rede de agentes. Para obter mais informações, consulte as seções a seguir: • Rede de agentes do Amazon MQ • Creating and Configuring a Network of Brokers • Configurar sua rede de agentes corretamente • `networkConnector` • `networkConnectionStartAssíncrono`

11 de dezembro de 2018	O Amazon MQ é compatível com o ActiveMQ 5.15.8, 5.15.6 e 5.15.0.
	• Bugs resolvidos e melhorias no ActiveMQ:
	  • Notas de lançamento do ActiveMQ 5.15.8
	  • Notas de lançamento do ActiveMQ 5.15.7

5 de dezembro de 2018	AWS oferece suporte à marcação de recursos para ajudar a monitorar sua alocação de custos. É possível etiquetar recursos ao criá-los ou visualizando os detalhes do recurso. Para obter mais informações, consulte Etiquetar recursos.
19 de novembro de 2018	AWS expandiu seu programa de conformidade com o SOC para incluir o Amazon MQ como um serviço compatível com o SOC.
15 de outubro de 2018	• O número máximo de grupos por usuário é 20. Para obter mais informações, consulte Usuários. • O número máximo de conexões por agente, por protocolos de nível de conexão é 1.000. Para obter mais informações, consulte Operadores.
2 de outubro de 2018	AWS expandiu seu programa de conformidade com a HIPAA para incluir o Amazon MQ como um serviço qualificado para a HIPAA.

Data	Atualização da documentação

27 de setembro de 2018	O Amazon MQ é compatível com ActiveMQ 5.15.6, além do 5.15.0. Para saber mais, consulte:
	• Conceitos básicos: criar e conectar a um agente do ActiveMQ
	• Bugs resolvidos e melhorias na documentação do ActiveMQ:
	  • Notas de lançamento do ActiveMQ 5.15.6
	  • Notas de lançamento do ActiveMQ 5.15.5
	  • Notas de lançamento do ActiveMQ 5.15.4
	  • Notas de lançamento do ActiveMQ 5.15.3
	  • Notas de lançamento do ActiveMQ 5.15.2
	  • Notas de lançamento do ActiveMQ 5.15.1
	• Cliente ActiveMQ 5.15.6

31 de agosto de 2018	• As seguintes métricas estão disponíveis:
	  • `CurrentConnectionsCount`
	  • `TotalConsumerCount`
	  • `TotalProducerCount`
	  Para obter mais informações, consulte a seção Monitoring and logging Amazon MQ brokers.
	• O endereço IP do agente é exibido na página Detalhes.
	  Note
	  Para agentes com acessibilidade pública desabilitada, o endereço IP interno é exibido.

Data	Atualização da documentação
30 de agosto de 2018	O Amazon MQ está disponível na região Ásia-Pacífico (Singapura), além das seguintes regiões: • Leste dos EUA (Ohio) • Leste dos EUA (Norte da Virgínia) • Oeste dos EUA (N. da Califórnia) • Oeste dos EUA (Oregon) • Ásia-Pacífico (Tóquio) • Ásia-Pacífico (Seul) • Ásia-Pacífico (Sydney) • Europa (Frankfurt) • Europa (Irlanda)
30 de julho de 2018	Você pode configurar o Amazon MQ para publicar registros gerais e de auditoria no Amazon CloudWatch Logs. Para obter mais informações, consulte Monitoring and logging Amazon MQ brokers.
25 de julho de 2018	O Amazon MQ está disponível nas regiões Ásia-Pacífico (Tóquio) e Ásia-Pacífico (Seul), além das seguintes regiões: • Leste dos EUA (Ohio) • Leste dos EUA (Norte da Virgínia) • Oeste dos EUA (N. da Califórnia) • Oeste dos EUA (Oregon) • Ásia-Pacífico (Sydney) • Europa (Frankfurt) • Europa (Irlanda)
19 de julho de 2018	Você pode usar AWS CloudTrail para registrar chamadas de API do Amazon MQ. Para obter mais informações, consulte Logging Amazon MQ API calls using CloudTrail.

Data	Atualização da documentação
29 de junho de 2018	Além de `mq.t2.micro` e `mq.m4.large`, os seguintes tipos de instância de agente estão disponíveis para workloads normais de desenvolvimento, teste e produção que exigem uma alta taxa de transferência: • `mq.m5.large` • `mq.m5.xlarge` • `mq.m5.2xlarge` • `mq.m5.4xlarge` Para obter mais informações, consulte Broker instance types.
27 de junho de 2018	O Amazon MQ está disponível na região Oeste dos EUA (Norte da Califórnia), além das seguintes regiões: • Leste dos EUA (Ohio) • Leste dos EUA (N. da Virgínia) • Oeste dos EUA (Oregon) • Ásia-Pacífico (Sydney) • Europa (Frankfurt) • Europa (Irlanda)

Data	Atualização da documentação

14 de junho de 2018	• Você pode usar o `AWS::Amazon MQ::Broker` AWS CloudFormation recurso para realizar as seguintes ações:
	  • Criar um agente.
	  • Adicionar alterações de configuração ou modificar usuários para o agente.
	  • Retornar informações sobre o agente especificado.
	  • Excluir o agente especificado.
	
	  Note
	  Quando você altera qualquer propriedade do tipo de propriedade Amazon MQ Broker ConfigurationId ou Amazon MQ Broker User, o broker é reiniciado imediatamente.
	• Você pode usar o `AWS::Amazon MQ::Configuration` AWS CloudFormation recurso para realizar as seguintes ações:
	  • Criar uma configuração.
	  • Atualizar a configuração especificada.
	  • Retornar informações sobre a configuração especificada.
	
	  Note
	  Você pode usar CloudFormation para modificar, mas não excluir, uma configuração do Amazon MQ.

7 de junho de 2018	O console do Amazon MQ é compatível com alemão, português do Brasil, espanhol, italiano e chinês tradicional.
17 de maio de 2018	O limite do número de usuários por agente é de 250. Para obter mais informações, consulte Usuários.
13 de março de 2018	A criação de um agente leva cerca de 15 minutos. Para obter mais informaçõ es, consulte Concluir a criação do agente.

Data	Atualização da documentação
1º de março de 2018	• Você pode configurar o armazenamento e a expedição simultâneos para o Apache KahaDB usando o atributo `concurrentStoreAndDispatchQueues`. • A `CpuCreditBalance` CloudWatch métrica > está disponível para o tipo de instância do `mq.t2.micro` broker.
10 de janeiro de 2018	As alterações a seguir afetam o console do Amazon MQ: • Na lista de agentes, a coluna Creation (Criação) é oculta por padrão. Para personalizar o tamanho da página e as colunas, selecione  . • Na ***MyBroker***página, na seção Conexões, escolha o nome do seu grupo de segurança ou  abra o console EC2 (em vez do console VPC). O console do EC2 permite configurações mais intuitivas das regras de entrada e de saída. Para obter mais informações, consulte a seção Connecting a Java application to your broker atualizada.
9 de janeiro de 2018	• A permissão para o ID da operação REST `UpdateBroker` está listada corretamente como `mq:UpdateBroker` no console do IAM. • A permissão errada `mq:DescribeEngine` foi removida do console do IAM.

Data	Atualização da documentação

28 de novembro de 2017	Esta é a versão inicial do Amazon MQ e do Guia do Desenvolvedor do Amazon MQ.
	• O Amazon MQ está disponível nas seguintes regiões:
	  • Leste dos EUA (Ohio)
	  • Leste dos EUA (N. da Virgínia)
	  • Oeste dos EUA (Oregon)
	  • Ásia-Pacífico (Sydney)
	  • Europa (Frankfurt)
	  • Europa (Irlanda)
	  O uso do tipo de instância `mq.t2.micro` está sujeito a créditos de CPU e performance de linha de base, com a capacidade de expandir acima do nível da linha de base (para obter mais informações, consulte a métrica do `CpuCreditBalance`). Se a sua aplicação exigir performance fixa, considere usar um tipo de instância `mq.m5.large`.
	• Você pode criar agentes `mq.m4.large` e `mq.t2.micro`.
	  O uso do tipo de instância `mq.t2.micro` está sujeito a créditos de CPU e performance de linha de base, com a capacidade de expandir acima do nível da linha de base (para obter mais informações, consulte a métrica do `CpuCreditBalance`). Se a sua aplicação exigir performance fixa, considere usar um tipo de instância `mq.m5.large`.
	• Você pode usar o mecanismo de agente ActiveMQ 5.15.0.
	• Você também pode criar e gerenciar corretores de forma programática usando a API REST do Amazon MQ e. AWS SDKs
	• Você pode acessar seus agentes usando qualquer linguagem de programação compatível com o ActiveMQ e habilitando o TLS explicitamente para os seguintes protocolos:
	  • AMQP
	  • MQTT
	  • Acabou o MQTT WebSocket
	  • OpenWire

Data	Atualização da documentação
	• STOMP • STOMP over WebSocket • Você pode se conectar a agentes do ActiveMQ usando vários clientes de ActiveMQ. Recomendamos usar o Cliente ActiveMQ. Para obter mais informações, consulte Connecting a Java application to your broker. • Seu agente pode enviar e receber mensagens de qualquer tamanho.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.