

9.2

IBM MQ em contêineres

IBM

Nota

Antes de usar estas informações e o produto suportado por elas, leia as informações em [“Avisos” na página 161](#).

Esta edição aplica-se à versão 9 release 2 de IBM® MQ e a todos os lançamentos e modificações subsequentes até indicado de outra forma em novas edições.

Ao enviar informações para IBM, você concede à IBM um direito não exclusivo de usar ou distribuir as informações de qualquer maneira que acredite ser apropriado sem incorrer em qualquer obrigação para você.

© **Copyright International Business Machines Corporation 2007, 2025.**

Índice

IBM MQ em contêineres e IBM Cloud Pak for Integration.....	5
Planejando-se para o IBM MQ em contêineres.....	5
Escolhendo como você deseja usar o IBM MQ em contêineres.....	5
Suporte para IBM MQ Operator.....	6
Dependências para o IBM MQ Operator.....	10
Permissões com escopo de cluster necessárias pelo IBM MQ Operator.....	10
Considerações de armazenamento para o IBM MQ Operator.....	11
Suporte para construção de suas próprias imagens de contêiner do gerenciador de filas do IBM MQ.....	13
Alta disponibilidade para IBM MQ em contêineres.....	16
Recuperação de desastre para o IBM MQ em contêineres.....	18
Autenticação e autorização de usuário para o IBM MQ em contêineres.....	18
Planejando a escalabilidade e o desempenho do IBM MQ em contêineres.....	19
Usando IBM MQ em IBM Cloud Pak for Integration e Red Hat OpenShift.....	20
Histórico de liberação para o IBM MQ Operator.....	20
Migrando o IBM MQ para o IBM Cloud Pak for Integration.....	36
Instalando e desinstalando o IBM MQ Operator no Red Hat OpenShift.....	59
Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas.....	71
Implementando e configurando gerenciadores de filas usando o IBM MQ Operator.....	79
Operando o IBM MQ usando o IBM MQ Operator.....	115
Resolução de problemas com o IBM MQ Operator.....	125
Referência da API do IBM MQ Operator.....	126
Construindo seu próprio contêiner e código de implementação do IBM MQ.....	147
Planejando a sua própria imagem do gerenciador de filas do IBM MQ usando um contêiner.....	148
Construindo uma imagem de contêiner de gerenciador de filas do IBM MQ.....	148
Executando aplicativos de ligação local em contêineres separados.....	151
Criando o grupo de HA nativa se estiver criando seus próprios contêineres.....	153
Avisos.....	161
Informações sobre a Interface de Programação.....	162
Marcas comerciais.....	163

Multi IBM MQ em contêineres e IBM Cloud Pak for Integration

Os contêineres permitem que você compacte um gerenciador de filas do IBM MQ ou o aplicativo cliente do IBM MQ, com todas as suas dependências, em uma unidade padronizada para desenvolvimento de software.

É possível executar o IBM MQ usando o IBM MQ Operator no Red Hat® OpenShift®. Isso pode ser feito usando o IBM Cloud Pak for Integration, IBM MQ Advanced ou IBM MQ Advanced for Developers.

Também é possível executar o IBM MQ em um contêiner que você mesmo constrói.

 Para obter mais informações sobre o IBM MQ Operator, consulte os links a seguir.

Multi Planejando-se para o IBM MQ em contêineres

Ao planejar o IBM MQ em contêineres, considere o suporte fornecido pelo IBM MQ para diversas opções de arquitetura, como a forma como a alta disponibilidade é gerenciada e seus gerenciadores de filas são protegidos.

Sobre esta tarefa

Antes de planejar o seu IBM MQ em arquitetura de contêineres, deve-se se familiarizar com os conceitos básicos IBM MQ (ver [IBM MQ Visão geral técnica](#)), bem como conceitos básicos Kubernetes/Red Hat OpenShift (ver [Red Hat OpenShift Container Platform arquitetura](#)).

Procedimento

- [“Escolhendo como você deseja usar o IBM MQ em contêineres”](#) na página 5.
- [“Suporte para IBM MQ Operator”](#) na página 6.
- [“Suporte para construção de suas próprias imagens de contêiner do gerenciador de filas do IBM MQ”](#) na página 13.
- [“Considerações de armazenamento para o IBM MQ Operator”](#) na página 11.
- [“Alta disponibilidade para IBM MQ em contêineres”](#) na página 16.
- [“Recuperação de desastre para o IBM MQ em contêineres”](#) na página 18.
- [“Autenticação e autorização de usuário para o IBM MQ em contêineres”](#) na página 18.

Escolhendo como você deseja usar o IBM MQ em contêineres

Há várias opções para usar o IBM MQ em contêineres: é possível optar por usar o IBM MQ Operator, que usa imagens de contêiner pré-empacotadas ou construir suas próprias imagens e código de implementação.

Usando o IBM MQ Operator



Se você estiver planejando implementar no Red Hat OpenShift Container Platform, provavelmente desejará usar o IBM MQ Operator.

O IBM MQ Operator inclui um novo recurso personalizado `QueueManager` para Red Hat OpenShift Container Platform. O operador observa as novas definições do gerenciador de filas e, em seguida, as transforma em recursos de baixo nível necessários, como recursos `StatefulSet` e `Service`. No caso do Native HA, o operador também pode executar a atualização contínua complexa de instâncias do

gerenciador de filas. Consulte [“Considerações para executar a sua própria atualização contínua de um gerenciador de filas de alta disponibilidade nativo”](#) na página 155

Alguns recursos do IBM MQ não são suportados ao usar o IBM MQ Operator. Será necessário construir suas próprias imagens e seus próprios gráficos se você desejar fazer um dos seguintes:

- Usar as APIs de REST para a administração ou o sistema de mensagens
- Usar qualquer um dos componentes do MQ a seguir:
 - Agentes do Managed File Transfer e seus recursos. No entanto, é possível usar o IBM MQ Operator para fornecer um ou mais gerenciadores de filas de Coordenação, Comando ou Agente.
 - AMQP
 - IBM MQ Bridge to Salesforce
 - IBM MQ Bridge to blockchain (não suportado em contêineres)
 - IBM MQ Telemetry Transport (MQTT).
- Customize as opções usadas com `crtmqm`, `strmqm` e `endmqm`, como configurar as páginas do arquivo de log. A maioria das opções pode ser configurada usando um arquivo INI.

Note que o IBM MQ Operator e os contêineres estão evoluindo rapidamente e, portanto, não são suportados em liberações de Long Term Support.

O IBM MQ Operator inclui ambas as imagens de contêiner pré-construídas, bem como o código de implementação para execução no Red Hat OpenShift Container Platform. O IBM MQ Operator pode ser usado para implementar a imagem de contêiner do IBM MQ fornecida, ou uma imagem de contêiner em camadas em cima dessa, mas não pode ser usado para implementar imagens de contêiner customizadas do MQ construídas.

Construindo suas próprias imagens e código de implementação

Multi

Essa é a solução de contêiner mais flexível, mas requer que você tenha um bom conhecimento técnico sobre configuração de contêineres e que você seja o "proprietário" do contêiner resultante. Se você não estiver planejando usar o Red Hat OpenShift Container Platform, será necessário construir suas próprias imagens e seu próprio código de implementação.

Estão disponíveis amostras para a construção de suas próprias imagens. Consulte o [“Construindo seu próprio contêiner e código de implementação do IBM MQ”](#) na página 147.

Conceitos relacionados

[“Suporte para IBM MQ Operator”](#) na página 6

O IBM MQ Operator é suportado somente quando implementado no Red Hat OpenShift Container Platform.

[“Suporte para construção de suas próprias imagens de contêiner do gerenciador de filas do IBM MQ”](#) na página 13

O IBM MQ fornece código para construir um contêiner gerenciador de filas do IBM MQ no GitHub. Isso se baseia no processo que o IBM usa para construir seu próprio contêiner suportado e você pode usar este repositório GitHub para simplificar e acelerar a construção de suas próprias imagens de contêiner.

OpenShift CP4I CD EUS Suporte para IBM MQ Operator

O IBM MQ Operator é suportado somente quando implementado no Red Hat OpenShift Container Platform.

O IBM MQ Operator usa imagens baseadas em liberações IBM MQ Continuous Delivery (CD), embora uma liberação de Extended Update Support (EUS) esteja disponível com IBM Cloud Pak for Integration. As liberações de CD são suportadas por até um ano, ou por duas liberações de CD, o que for mais longo. As versões do Long Term Support de IBM MQ não estão disponíveis via IBM MQ Operator. IBM Cloud Pak for Integration 2020.4.1 é uma versão de Suporte de atualização estendida (EUS), que é suportada por

18 meses, se você utilizar uma versão de IBM MQ marcada como -eus. Caso contrário, o IBM MQ 9.2 é considerado uma liberação do Continuous Delivery com o IBM MQ Operator.

O IBM MQ Operator usa imagens de contêiner que fornecem uma instalação do IBM MQ em um Red Hat Universal Base Image (UBI), que inclui as principais bibliotecas e utilitários do Linux® usados pelo IBM MQ. A UBI é suportada pelo Red Hat quando executado no Red Hat OpenShift.

O IBM MQ Operator é suportado nas arquiteturas amd64 e s390x (z/Linux).

Conceitos relacionados

“[Suporte para construção de suas próprias imagens de contêiner do gerenciador de filas do IBM MQ](#)” na página 13

O IBM MQ fornece código para construir um contêiner gerenciador de filas do IBM MQ no GitHub. Isso se baseia no processo que o IBM usa para construir seu próprio contêiner suportado e você pode usar este repositório GitHub para simplificar e acelerar a construção de suas próprias imagens de contêiner.

OpenShift CP4I CD EUS Suporte de versão para o IBM MQ Operator

Um mapeamento entre versões suportadas de IBM MQ, Red Hat OpenShift Container Platform e IBM Cloud Pak for Integration.

- “[Versões disponíveis do IBM MQ](#)” na página 7
- “[Versões compatíveis do Red Hat OpenShift Container Platform](#)” na página 8
- “[Versões do IBM Cloud Pak for Integration](#)” na página 8
- “[Versões do IBM MQ disponíveis em operadores mais antigos](#)” na página 8
- “[Versões do Red Hat OpenShift Container Platform compatíveis para operadores mais antigos](#)” na página 9

Versões disponíveis do IBM MQ

Canal do operador	Versão do operador	Versões do IBM MQ							
		9.1.5	9.2.0 CD	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.6	1.6	⚠	⚠	→	⚠	●	●		
v1.7	1.7	⚠	⚠	→	⚠	●	●	●	
v1.8	1.8	⚠	⚠	→	⚠	⚠	●	●	●

Chave:



Suporte do Continuous Delivery disponível



Extended Update Support disponível



Somente disponível durante a migração do operando Extended Update Support para um operando Continuous Delivery.



Descontinuado. À medida que as liberações do IBM MQ ficam sem suporte, elas ainda podem ser configuráveis no operador, mas não são mais elegíveis para suporte e podem ser removidas em versões futuras.

Consulte “[Histórico de liberação para o IBM MQ Operator](#)” na página 20 para obter detalhes completos de cada versão, incluindo recursos detalhados, mudanças e correções em cada versão.

Versões compatíveis do Red Hat OpenShift Container Platform

Canal do operador	Versão do operador	Versões do Red Hat OpenShift Container Platform ¹				
		4.6	4.7 ²	4.8	4.9	4.10
v1.6	1.6	●	●	●	●	●
v1.7	1.7	●	●	●	●	●
v1.8	1.8	●	●	●	●	●

Chave:

- Suporte do Continuous Delivery disponível
- Extended Update Support disponível

Versões do IBM Cloud Pak for Integration

O IBM MQ Operator 1.8.x tem suporte para uso como parte do IBM Cloud Pak for Integration versão 2021.4.1 ou de independente.

O IBM MQ Operator 1.7.x é suportado para uso como parte do IBM Cloud Pak for Integration versão 2021.4.1 ou de forma independente.

O IBM MQ Operator 1.6.x é suportado para uso como parte do IBM Cloud Pak for Integration versão 2021.2.1, 2021.3.1 ou de forma independente.

O IBM MQ Operator 1.5.x não é mais suportado.

O IBM MQ Operator 1.4.x não é mais suportado.

O IBM MQ Operator 1.3.x não é mais suportado.

O IBM MQ Operator 1.2.x não é mais suportado.

Os IBM MQ Operators 1.1.x e 1.0.x não são mais suportados.

Versões do IBM MQ disponíveis em operadores mais antigos

A tabela a seguir se aplica a versões do IBM MQ Operator que já atingiram o "fim da vida útil".

Canal do operador	Versão do operador	Versões do IBM MQ							
		9.1.5	9.2.0 CD	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.0	1.0	⚠							
v1.1	1.1	⚠	⚠						
v1.2	1.2	⚠	⚠						
v1.3-eus	1.3	⚠	⚠	⚠					
v1.4	1.4	⚠	⚠	→	⚠				

¹ As versões do Red Hat OpenShift Container Platform estão sujeitas às suas próprias datas de suporte. Veja [Política de ciclo de vida do Red Hat OpenShift Container Platform](#) para obter mais informações.

² O IBM MQ Operator depende do IBM Cloud Pak foundational services. Se você deseja usar o Red Hat OpenShift Container Platform 4,7, deve-se primeiro atualizar a versão do IBM Cloud Pak foundational services.

Canal do operador	Versão do operador	Versões do IBM MQ							
		9.1.5	9.2.0 CD	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.5	1.5	⚠	⚠	→	⚠	⚠			

Chave:

→

Somente disponível durante a migração do operando Extended Update Support para um operando Continuous Delivery.

⚠

Descontinuado. À medida que as liberações do IBM MQ ficam sem suporte, elas ainda podem ser configuradas no IBM MQ Operator, mas não são mais qualificadas para suporte.

Consulte “Histórico de liberação para o IBM MQ Operator” na página 20 para obter detalhes completos de cada versão, incluindo recursos detalhados, mudanças e correções em cada versão.

Versões do Red Hat OpenShift Container Platform compatíveis para operadores mais antigos

A tabela a seguir se aplica a versões do IBM MQ Operator que já atingiram o "fim da vida útil".

Canal do operador	Versão do operador	Versões do Red Hat OpenShift Container Platform ³						
		4.4 ⁴	4.5 ⁵	4.6	4.7 ⁶	4.8	4.9	4.10
v1.0	1.0	⚠	⚠	⚠	⚠			
v1.1	1.1	⚠	⚠	⚠	⚠	⚠		
v1.2	1.2	⚠	⚠	⚠	⚠	⚠		
v1.3-eus	1.3			⚠	→	→	→	→
v1.4	1.4			⚠	⚠	⚠	⚠	
v1.5	1.5			⚠	⚠	⚠	⚠	⚠

Chave:

→

Somente disponível durante a migração do operando Extended Update Support para um operando Continuous Delivery.

⚠

A versão do IBM MQ Operator chegou ao "fim da vida", mas já estava disponível anteriormente nesta versão do Red Hat OpenShift Container Platform

³ As versões do Red Hat OpenShift Container Platform estão sujeitas às suas próprias datas de suporte. Veja [Política de ciclo de vida do Red Hat OpenShift Container Platform](#) para obter mais informações.

⁴ O Red Hat OpenShift Container Platform 4.4 chegou ao "fim da vida". Veja [Política de ciclo de vida do Red Hat OpenShift Container Platform](#) para obter mais informações.

⁵ O Red Hat OpenShift Container Platform 4.5 chegou ao "fim da vida". Veja [Política de ciclo de vida do Red Hat OpenShift Container Platform](#) para obter mais informações.

⁶ O IBM MQ Operator depende do IBM Cloud Pak foundational services. Se você deseja usar o Red Hat OpenShift Container Platform 4,7, deve-se primeiro atualizar a versão do IBM Cloud Pak foundational services.

O IBM MQ Operator depende do IBM Cloud Pak foundational services Operator, que também instala o IBM Operand Deployment Lifecycle Manager (ODLM) Operator. Esses operadores serão instalados automaticamente quando o IBM MQ Operator for instalado. Esses operadores dependentes têm um pequeno volume de CPU e memória e são usados para implementar recursos adicionais em algumas circunstâncias.

Ao criar um QueueManager, o IBM MQ Operator criará um OperandRequest para os serviços adicionais que precisar. O OperandRequest é preenchido pelo ODLM Operator e irá instalar e instanciar os serviços necessários, se necessário. Quais serviços são necessários é determinado com base no contrato de licença aceito ao implementar o gerenciador de filas e em quais componentes do gerenciador de filas são solicitados.

- Se você escolher uma licença IBM MQ Advanced ou IBM MQ Advanced for Developers, nenhum serviço adicional será solicitado. Por exemplo, no caso a seguir, os IBM Cloud Pak foundational services não são usados:

```
spec:
  license:
    accept: true
    license: L-APIG-BZDDDY
    use: "Production"
```

- Se você escolher uma licença IBM Cloud Pak for Integration e optar por habilitar o servidor da web, o IBM MQ Operator também instanciará o operador IBM Identity and Access Management (IAM) para habilitar a conexão única. O IAM Operator já estará disponível se você tiver instalado o IBM Cloud Pak for Integration Operator. Por exemplo:

```
spec:
  license:
    accept: true
    license: L-RJON-BUVMQX
    use: "Production"
```

No entanto, se você desativar o servidor da web, então nenhum IBM Cloud Pak foundational services será solicitado. Por exemplo:

```
spec:
  license:
    accept: true
    license: L-RJON-BUVMQX
    use: "Production"
  web:
    enabled: false
```

Versões mais antigas do IBM MQ Operator sempre solicitaram a instalação do IBM Licensing Operator (e suas dependências), para rastrear o uso de licença. Do IBM MQ Operator 1.5 em diante, o serviço de licenciamento não é solicitado, e é necessário solicitá-lo separadamente.

O IBM MQ Operator requer 1 núcleo de CPU e 1 GB de memória. Para um detalhamento dos requisitos de hardware e software para os operadores dependentes, consulte [Requisitos de hardware e recomendações para serviços de base](#).

É possível escolher a quantidade de CPU e de memória usada por seus gerenciadores de filas. Veja [“.spec.queueManager.resources” na página 136](#) para obter mais informações.

Referências relacionadas

[“Referência de licenciamento para mq.ibm.com/v1beta1” na página 126](#)

MQ Operator

O IBM MQ Operator requer permissões com escopo de cluster para gerenciar webhooks de admissão e amostras, além de ler informações de classe de armazenamento e de versão de cluster.

O IBM MQ Operator requer as permissões do cluster com escopo a seguir:

- Permissão para gerenciar webhooks de admissão. Isso permite a criação, recuperação e atualização de webhooks específicos que são usados no processo de criação e gerenciamento de contêineres fornecidos pelo Operador.
 - Grupos de API: **admissionregistration.k8s.io**
 - Recursos: **validatingwebhookconfigurations**
 - Verbos: **create, get, update**
- Permissão para criar e gerenciar recursos que são usados no console Red Hat OpenShift para fornecer amostras e trechos ao criar recursos customizados.
 - Grupos de API: **console.openshift.io**
 - Recursos: **consoleyamlsamples**
 - Verbos: **create, get, update, delete**
- Permissão para ler a versão do cluster. Isso permite que o Operador retorne todos os problemas com o ambiente em cluster.
 - Grupos de API: **config.openshift.io**
 - Recursos: **clusterversions**
 - Verbos: **get, list, watch**
- Permissão para ler classes de armazenamento no cluster. Isso permite que o Operador retorne todos os problemas com classes de armazenamento selecionadas em contêineres.
 - Grupos de API: **storage.k8s.io**
 - Recursos: **storageclasses**
 - Verbos: **get, list**

Considerações de armazenamento para o IBM MQ Operator

O IBM MQ Operator é executado em dois modos de armazenamento:

- O **armazenamento efêmero** é usado quando todas as informações de estado do contêiner podem ser descartadas quando o container reinicia. Isso é comumente usado quando os ambientes são criados para demonstração ou ao desenvolver com gerenciadores de filas independentes.
- O **armazenamento persistente** é a configuração comum para o IBM MQ e garante que, se o contêiner for reiniciado, a configuração, os logs e as mensagens persistentes existentes estarão disponíveis no contêiner reiniciado.

O IBM MQ Operator fornece a capacidade de customizar as características de armazenamento que podem diferir consideravelmente dependendo do ambiente e do modo de armazenamento desejado.

Armazenamento transitório

O IBM MQ é um aplicativo stateful e persiste este estado para o armazenamento para recuperação no caso de um reinício. Se estiver usando o armazenamento efêmero, todas as informações do estado do gerenciador de filas serão perdidas no reinício. Estão incluídas:

- Todas as mensagens
- Todos os estados de comunicação entre gerenciadores de filas (números de sequência de mensagens do canal)
- A identidade do Cluster do MQ do gerenciador de filas
- Todos os estados de transação
- Todas as configurações do gerenciador de filas

- Todos os dados de diagnóstico local

Por esta razão, é necessário considerar se o armazenamento efêmero é uma abordagem adequada para um cenário de produção, teste ou desenvolvimento. Por exemplo, quando todas as mensagens são conhecidas por serem não persistentes e o gerenciador de filas não é um membro de um cluster do MQ. No reinício, todos os estados de mensagens e a configuração do gerenciador de filas são descartados. Para ativar um contêiner completamente efêmero, a configuração do IBM MQ deve ser incluída na própria imagem do contêiner (para obter mais informações, consulte [“Construindo uma imagem com arquivos customizados de MQSC e INI, usando a CLI do Red Hat OpenShift”](#) na página 112). Se isso não for concluído, o IBM MQ precisará ser configurado cada vez que o contêiner for reiniciado.

  Por exemplo, para configurar o IBM MQ com armazenamento efêmero, o tipo de armazenamento do QueueManager deverá incluir o seguinte:

```
queueManager:
  storage:
    queueManager:
      type: ephemeral
```

Armazenamento persistente

O IBM MQ normalmente é executado com armazenamento de persistência para assegurar que o gerenciador de filas retenha suas mensagens persistentes e a configuração após um reinício. Portanto, esse é o comportamento padrão. Devido aos vários provedores de armazenamento e diferentes capacidades que cada um suporta, muitas vezes, isso significa que a customização da configuração é necessária. O exemplo abaixo descreve os campos comuns que customizam a configuração de armazenamento do MQ na API do v1beta1:

- `spec.queueManager.availability` controla o modo de disponibilidade. Se você estiver usando `SingleInstance`, somente o armazenamento `ReadWriteOnce` será necessário. Para `MultiInstance`, no entanto, uma classe de armazenamento que suporte `ReadWriteMany` com as características de bloqueio de arquivo corretas será necessária. O IBM MQ fornece uma [instrução de suporte](#) e uma [instrução de teste](#). O modo de disponibilidade também influencia o layout de volume persistente. Para obter mais informações, consulte [“Alta disponibilidade para IBM MQ em contêineres”](#) na página 16.
- `spec.queueManager.storage` controla as configurações de armazenamento individuais. Um gerenciador de filas pode ser configurado para usar entre um e quatro volumes persistentes

O exemplo a seguir mostra um trecho de uma configuração simples usando um gerenciador de filas de uma única instância:

```
spec:
  queueManager:
    storage:
      queueManager:
        enabled: true
```

O exemplo a seguir mostra um trecho de uma configuração do gerenciador de filas de várias instâncias, com uma classe de armazenamento não padrão e com armazenamento de arquivo que requer grupos complementares:

```
spec:
  queueManager:
    availability:
      type: MultiInstance
    storage:
      queueManager:
        class: ibmc-file-gold-gid
      persistedData:
        enabled: true
        class: ibmc-file-gold-gid
      recoveryLogs:
        enabled: true
        class: ibmc-file-gold-gid
```

```
securityContext:  
  supplementalGroups: [99]
```

Nota: Também é possível configurar grupos suplementares com gerenciadores de filas de instância única.

Nota: Você não requer sistemas de arquivos compartilhados se usar HA nativa (consulte [“Alta disponibilidade para IBM MQ em contêineres”](#) na página 16). Especificamente, você não deve usar NFSv3.

Linux Suporte para construção de suas próprias imagens de contêiner do gerenciador de filas do IBM MQ

O IBM MQ fornece código para construir um contêiner gerenciador de filas do IBM MQ no GitHub. Isso se baseia no processo que o IBM usa para construir seu próprio contêiner suportado e você pode usar este repositório GitHub para simplificar e acelerar a construção de suas próprias imagens de contêiner.

O código é fornecido no repositório mq-container GitHub aqui: <https://github.com/ibm-messaging/mq-container>. Isso é fornecido sob uma licença Apache 2.0, com suporte fornecido pela comunidade.

O repositório não usa os pacotes rpm padrão Linux ; ele usa o pacote compactado para implementações de contêiner. O benefício dessa abordagem é que você pode executar em ambientes de contêiner mais seguros sem a necessidade de permissões escalonadas. No entanto, isso afeta as opções de segurança disponíveis, porque o IBM MQ usa tradicionalmente permissões escalonadas para autenticação baseada no sistema operacional. Para uma implementação de contêiner, o uso de autenticação baseada em sistema operacional normalmente não é uma boa prática; em vez disso, é possível usar a autenticação mútua TLS ou LDAP. Com o IBM MQ Advanced for Developers, também é possível usar a autenticação baseada em arquivo, permitindo que seus usuários comecem rapidamente.

O replicated data queue manager (RDQM) não é suportado em um ambiente de contêiner. É possível obter recursos semelhantes ao RDQM usando o [“HA nativa”](#) na página 92.

Conceitos relacionados

[“Suporte para IBM MQ Operator”](#) na página 6

O IBM MQ Operator é suportado somente quando implementado no Red Hat OpenShift Container Platform.

[IBM MQ imagens de não instalação](#)

Linux Anotações de licença ao construir a sua imagem de contêiner do IBM MQ

As anotações de licença permitem que você acompanhe o uso com base nos limites definidos no contêiner, em vez de na máquina subjacente. Configure os seus clientes para implementar o container com anotações específicas que o IBM License Service usa para rastrear o uso.

Ao implementar uma imagem de contêiner do IBM MQ construída automaticamente, há duas abordagens comuns para o licenciamento:

- Licenciamento da máquina inteira executando o contêiner.
- Licenciamento do contêiner com base nos limites associados.

Ambas as opções estão disponíveis para os clientes e mais detalhes podem ser encontrados na página [Licenças do IBM Container em Passport Advantage](#).

Se o contêiner IBM MQ for licenciado com base nos limites de contêiner, o IBM License Service precisará estar instalado para rastrear o uso. Outras informações relativas aos ambientes suportados e instruções de instalação podem ser encontradas na página [ibm-licensing-operator](#) no GitHub.

O IBM License Service é instalado no cluster Kubernetes em que o contêiner IBM MQ é implementado, e anotações de pod são usadas para rastrear o uso. Por isso, os clientes precisam implementar o pod com anotações específicas que o IBM License Service usará. Com base na autorização e na capacidade implementadas no contêiner, use uma ou mais das anotações a seguir:

- [“Contêiner do IBM MQ Advanced” na página 14](#)
- [“Contêiner do IBM MQ Advanced High Availability Replica” na página 14](#)
- [“Contêiner do IBM MQ Base” na página 14](#)
- [“Contêiner do IBM MQ Base High Availability Replica” na página 14](#)
- [“Contêiner do IBM MQ Advanced for Developers” na página 14](#)
- [“Contêiner do IBM MQ Advanced com uma autorização CP4I \(produção\)” na página 14](#)
- [“Contêiner do IBM MQ Advanced High Availability Replica com uma autorização CP4I \(produção\)” na página 15](#)
- [“Contêiner do IBM MQ Advanced com uma autorização CP4I \(não produção\)” na página 15](#)
- [“Contêiner do IBM MQ Advanced High Availability Replica com uma autorização CP4I \(não produção\)” na página 15](#)
- [“IBM MQ Base com uma autorização CP4I \(produção\)” na página 15](#)
- [“IBM MQ Base High Availability Replica com uma autorização CP4I \(produção\)” na página 15](#)
- [“IBM MQ Base com uma autorização CP4I \(não produção\)” na página 15](#)
- [“IBM MQ Base High Availability Replica com uma autorização CP4I \(não produção\)” na página 16](#)

Contêiner do IBM MQ Advanced

```
productName: "IBM MQ Advanced"
productID: "208423bb063c43288328b1d788745b0c"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

Contêiner do IBM MQ Advanced High Availability Replica

```
productName: "IBM MQ Advanced High Availability Replica"
productID: "546cb719714942c18748137ddd8d5659"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

Contêiner do IBM MQ Base

```
productName: "IBM MQ"
productID: "c661609261d5471fb4ff8970a36bccea"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

Contêiner do IBM MQ Base High Availability Replica

```
productName: "IBM MQ High Availability Replica"
productID: "2a2a8e0511c849969d2f286670ea125e"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

Contêiner do IBM MQ Advanced for Developers

```
productName: "IBM MQ Advanced for Developers"
productID: "2f886a3eefbe4ccb89b2adb97c78b9cb"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "FREE"
```

Contêiner do IBM MQ Advanced com uma autorização CP4I (produção)

```
productName: "IBM MQ Advanced with CP4I License"
productID: "208423bb063c43288328b1d788745b0c"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

```
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "2:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

Contêiner do IBM MQ Advanced High Availability Replica com uma autorização CP4I (produção)

```
productName: "IBM MQ Advanced High Availability Replica with CP4I License"  
productID: "546cb719714942c18748137ddd8d5659"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "10:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

Contêiner do IBM MQ Advanced com uma autorização CP4I (não produção)

```
productName: "IBM MQ Advanced for Non-Production with CP4I License"  
productID: "21dfe9a0f00f444f888756d835334909"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "4:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

Contêiner do IBM MQ Advanced High Availability Replica com uma autorização CP4I (não produção)

```
productName: "IBM MQ Advanced High Availability Replica for Non-Production with CP4I License"  
productID: "b3f8f984007d47fb981221589cc50081"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "20:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Base com uma autorização CP4I (produção)

```
productName: "IBM MQ with CP4I License"  
productID: "c661609261d5471fb4ff8970a36bceca"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "4:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Base High Availability Replica com uma autorização CP4I (produção)

```
productName: "IBM MQ High Availability Replica with CP4I License"  
productID: "2a2a8e0511c849969d2f286670ea125e"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "20:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Base com uma autorização CP4I (não produção)

```
productName: "IBM MQ with CP4I License Non-Production"  
productID: "151bec68564a4a47a14e6fa99266deff"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "8:1"
```

```
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Base High Availability Replica com uma autorização CP4I (não produção)

```
productName: "IBM MQ High Availability Replica with CP4I License Non-Production"
productID: "f5d0e21c013c4d4b8b9b2ce701f31928"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "40:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

OpenShift CP4I Kubernetes Alta disponibilidade para IBM MQ em contêineres

Há três opções para alta disponibilidade com o IBM MQ Operator: **gerenciador de filas Native HA** (que tem uma réplica ativa e duas réplicas de espera), **gerenciador de filas de várias instâncias** (que é um par ativo/espera usando um sistema de arquivos compartilhado em rede) ou **gerenciador de filas resiliente único** (que oferece uma abordagem simples para HA usando armazenamento em rede). Estes dois últimos contam com o sistema de arquivos para assegurar a disponibilidade dos dados recuperáveis, no entanto, o Native HA não. Portanto, ao não usar o Native HA, a disponibilidade do sistema de arquivos é crítica para a disponibilidade do gerenciador de filas. Quando a recuperação de dados é importante, o sistema de arquivos deve assegurar a redundância por meio da replicação.

É necessário considerar a disponibilidade de **mensagem** e de **serviço** separadamente. Com o IBM MQ for Multiplatforms, uma mensagem é armazenada em exatamente um gerenciador de filas. Portanto, se esse gerenciador de filas se tornar indisponível, você perderá temporariamente o acesso às mensagens que ele contém. Para alcançar a alta disponibilidade de message, é necessário ser capaz de recuperar um gerenciador de filas o mais rapidamente possível. É possível alcançar a disponibilidade de service tendo várias instâncias de filas para uso dos aplicativos clientes, por exemplo, usando um cluster uniforme do IBM MQ.

Um gerenciador de filas pode ser considerado em duas partes: os dados armazenados no disco e os processos em execução que permitem acesso aos dados. Qualquer gerenciador de filas pode ser movido para um Nó diferente do Kubernetes, desde que mantenha os mesmos dados (fornecidos pelos Volumes persistentes do Kubernetes) e continuará endereçável na rede pelos aplicativos clientes. No Kubernetes, um Serviço é usado para fornecer uma identidade de rede consistente.

O IBM MQ conta com a disponibilidade dos dados nos volumes persistentes. Portanto, a disponibilidade do armazenamento que fornece os volumes persistentes é crítica para disponibilidade do gerenciador de filas, porque o IBM MQ não pode estar mais disponível do que o armazenamento que ele está usando. Se desejar tolerar uma indisponibilidade de uma zona de disponibilidade inteira, será necessário usar um provedor de volume que replique gravações de disco para outra zona.

Gerenciador de filas Native HA

CP4I V 9.2.3

Os gerenciadores de filas Native HA estão disponíveis no IBM Cloud Pak for Integration 2021.2.1, usando o IBM MQ Operator 1.6 ou superior, com o IBM MQ 9.2.3 ou superior.

Os gerenciadores de filas de HA nativa envolvem um Pod **ativo** e dois Pods de **réplica** do Kubernetes, que são executados como parte de um StatefulSet do Kubernetes com exatamente três réplicas cada com seu próprio conjunto de Volumes Persistentes do Kubernetes. Os requisitos do IBM MQ para sistemas de arquivos compartilhados também se aplicam ao usar um gerenciador de filas de HA nativa (exceto o bloqueio baseado em lease), mas você não precisa usar um sistema de arquivos compartilhado. É possível usar o armazenamento de bloco com um sistema de arquivos adequado na parte superior. Por exemplo, *xfs* ou *ext4*. Os tempos de recuperação de um gerenciador de filas de HA nativa são controlados pelos fatores a seguir:

1. Quanto tempo as instâncias de réplica levam para detectar a falha na instância ativa. Ela é configurável.
2. Quanto tempo leva para a análise de prontidão do Pod do Kubernetes detectar que o contêiner pronto mudou e redirecionar o tráfego de rede. Ela é configurável.
3. Quanto tempo leva para os clientes do IBM MQ se reconectarem.

Para obter informações adicionais, consulte [“HA nativa” na página 92](#)

Gerenciador de filas de várias instâncias



Os gerenciadores de filas de várias instâncias envolvem um Pod **ativo** e um **em espera** do Kubernetes, que são executados como parte de um Conjunto stateful do Kubernetes com exatamente duas réplicas e um conjunto de Volumes persistentes do Kubernetes. Os logs de transações e os dados do gerenciador de filas são mantidos em dois volumes persistentes usando um sistema de arquivo compartilhado.

Os gerenciadores de filas de várias instâncias requerem que os Pods **ativo** e **em espera** tenham acesso simultâneo ao volume persistente. Para configurar isso, use os Volumes persistentes do Kubernetes com **access mode** configurado como `ReadWriteMany`. Os volumes também devem atender aos [IBM MQ requisitos para sistemas de arquivos compartilhados](#) porque o IBM MQ conta com a liberação automática de travas de arquivo para instigar um failover do gerenciador de filas. O IBM MQ produz uma [lista de sistemas de arquivos testados](#).

Os tempos de recuperação de um gerenciador de filas de várias instâncias são controlados pelos fatores a seguir:

1. Quanto tempo leva depois de ocorrer uma falha para o sistema de arquivo compartilhado liberar os bloqueios originalmente obtidos pela instância ativa.
2. Quanto tempo leva para a instância em espera adquirir os bloqueios e, em seguida, iniciar.
3. Quanto tempo leva para a análise de prontidão do Pod do Kubernetes detectar que o contêiner pronto mudou e redirecionar o tráfego de rede. Ela é configurável.
4. Quanto tempo leva para os clientes do IBM MQ se reconectarem.

Gerenciador de filas resiliente único



Um gerenciador de filas resiliente único é uma única instância de um gerenciador de filas em execução em um único Pod do Kubernetes, em que o Kubernetes monitora o gerenciador de filas e substitui o Pod conforme necessário.

Os [IBM MQ requisitos para sistemas de arquivos compartilhados](#) também se aplicam ao usar um único gerenciador de filas resiliente (exceto o bloqueio baseado em lease), mas não é necessário usar um sistema de arquivos compartilhado. É possível usar o armazenamento de bloco com um sistema de arquivos adequado na parte superior. Por exemplo, `xfs` ou `ext4`.

Os tempos de recuperação de um gerenciador de filas resiliente único são controlados pelos fatores a seguir:

1. Quanto tempo leva para que a análise de atividade seja executada e quantas falhas são toleradas. Ela é configurável.
2. Quanto tempo o Planejador do Kubernetes leva para reagendar o Pod com falha para um novo Nó.
3. Quanto tempo leva para fazer download da imagem do contêiner no novo Nó. Se você usar um valor **imagePullPolicy** de `IfNotPresent`, a imagem talvez já esteja disponível nesse Nó.
4. Quanto tempo leva para a nova instância do gerenciador de filas iniciar.
5. Quanto tempo leva para a análise de prontidão do Pod do Kubernetes detectar se o contêiner está pronto. Ela é configurável.
6. Quanto tempo leva para os clientes do IBM MQ se reconectarem.

Importante:

Embora o padrão do gerenciador de filas resiliente único ofereça alguns benefícios, você precisa entender se será possível alcançar seus objetivos de disponibilidade com as limitações em torno das falhas do Nó.

No Kubernetes, em geral, um Pod com falha é recuperado rapidamente, mas a falha de um Nó inteiro é manipulada de forma diferente. Ao usar uma carga de trabalho stateful como IBM MQ com um Kubernetes StatefulSet, se um Kubernetes Nó principal perder contato com um nó do trabalhador, ele não poderá determinar se o nó falhou ou se simplesmente perdeu a conectividade de rede. Portanto, o Kubernetes **não executará nenhuma ação** nesse caso até que um dos eventos a seguir ocorra:

1. O nó é recuperado para um estado em que o Nó principal do Kubernetes possa se comunicar com ele.
2. Uma ação administrativa é executada para excluir explicitamente o Pod no Nó principal do Kubernetes. Isso não para necessariamente a execução do Pod, apenas o exclui do armazenamento do Kubernetes. Essa ação administrativa deve, portanto, ser executada com muito cuidado.

Tarefas relacionadas

[“Configurando alta disponibilidade para gerenciadores de filas usando o IBM MQ Operator” na página 92](#)

Referências relacionadas

[Configurações de alta disponibilidade](#)

OpenShift CP4I Kubernetes Recuperação de desastre para o IBM MQ em contêineres

É necessário considerar para qual tipo de desastre você está se preparando. Em ambientes de nuvem, o uso de zonas de disponibilidade fornece certo nível de tolerância para desastres, e são muito mais fáceis de usar. Se você tiver um número ímpar de data centers (para quorum) e um link de rede de baixa latência, será possível executar um único cluster Red Hat OpenShift Container Platform ou Kubernetes com várias zonas de disponibilidade, cada uma em um local físico separado. Este tópico discute as considerações de recuperação de desastre para os casos nos quais esses critérios não podem ser atendidos: ou seja, um número par de data centers ou um link de rede de alta latência.

Para recuperação de desastre, é necessário considerar o seguinte:

- Replicação de dados IBM MQ (mantidos em um ou mais recursos PersistentVolume) para a localização de recuperação de desastres
- Recriação do gerenciador de filas usando os dados replicados
- O ID de rede do gerenciador de filas que é visível para aplicativos clientes do IBM MQ e outros gerenciadores de filas. Esse ID poderia ser uma entrada de DNS, por exemplo.

Os dados persistentes precisam ser replicados, de forma síncrona ou assíncrona, para o site de recuperação de desastre. Isso geralmente é específico para o provedor de armazenamento, mas também pode ser feito usando um VolumeSnapshot. Consulte [capturas instantâneas de volume do CSI](#) para obter mais informações sobre capturas instantâneas de volume.

Ao recuperar-se de um desastre, será necessário recriar a instância do gerenciador de filas no novo cluster do Kubernetes, usando os dados replicados. Se você estiver usando o IBM MQ Operator, será necessário o YAML QueueManager, assim como o YAML para outros recursos de suporte como um ConfigMap ou Secret.

Informações relacionadas

[ha_for_ctr.dita](#)

Autenticação e autorização de usuário para o IBM MQ em contêineres

O IBM MQ pode ser configurado para usar usuários e grupos LDAP. Como alternativa, é possível usar usuários e grupos do sistema operacional local dentro da imagem de contêiner. O IBM MQ Operator não permite o uso de usuários e grupos do sistema operacional devido a preocupações de segurança.

Em um ambiente containerizado de diversos locatários, as restrições de segurança geralmente são colocadas em prática para evitar problemas de segurança em potencial, por exemplo:

- **Evitando o uso do usuário "raiz" dentro de um contêiner**
- **Forçando o uso de um ID do Usuário aleatório.** Por exemplo, no Red Hat OpenShift Container Platform, o padrão SecurityContextConstraints (chamado restricted) usa um ID de usuário escolhido a esmo para cada contêiner.
- **Evitando o uso da escalada de privilégio.** O IBM MQ no Linux usa a escalada de privilégio para verificar as senhas dos usuários. Ele usa um programa "setuid" para se tornar o usuário "raiz" para fazer isso.

  Para assegurar a conformidade com essas medidas de segurança, o IBM MQ Operator não permite o uso de IDs que são definidos nas bibliotecas do sistema operacional dentro de um contêiner. Não há nenhum ID do usuário ou grupo mqm definido no contêiner. Ao usar o IBM MQ em IBM Cloud Pak for Integration e Red Hat OpenShift, é necessário configurar o seu gerenciador de filas para usar o LDAP para autenticação e autorização do usuário. Para obter informações sobre como configurar o IBM MQ para fazer isso, consulte [Autenticação de conexão: repositórios de usuários e Autorização LDAP](#)

Planejando a escalabilidade e o desempenho do IBM MQ em contêineres

Na maioria dos casos, o ajuste de escala e o desempenho do IBM MQ em contêineres são iguais ao IBM MQ para Multiplataformas. No entanto, há alguns limites adicionais que podem ser impostos pela plataforma de contêiner

Sobre esta tarefa

Ao planejar a escalabilidade e o desempenho do IBM MQ em contêineres, considere as opções a seguir:

Procedimento

- **Limite o número de encadeamentos e processos.**

IBM MQ usa encadeamentos para gerenciar simultaneidade. No Linux, os encadeamentos são implementados como processos, portanto, é possível encontrar limites impostos pela plataforma de contêiner ou sistema operacional no número máximo de processos. No Red Hat OpenShift Container Platform, há um limite padrão de 4096 processos por contêiner (1024 processos até OpenShift 4.11). Embora isso seja adequado para a grande maioria dos cenários, pode haver casos em que isso possa impactar o número de conexões do cliente para um gerenciador de filas

O limite de processo no Kubernetes pode ser definido por um administrador de cluster usando a definição de configuração do kubelet **podPidsLimit** Consulte [Limites e reservas do ID do processo](#) no Kubernetes documentação. No Red Hat OpenShift Container Platform, também é possível [criar um recurso customizado ContainerRuntimeConfig](#) para editar CRI-O parâmetros.

Em sua configuração do IBM MQ, também é possível configurar o número máximo de conexões do cliente para um gerenciador de fila Consulte [Limites do Canal de Conexão do Servidor](#) para aplicar limites a um canal de conexão do servidor individual e o [atributo MAXCHANNELS INI](#) para aplicar limites ao gerenciador de filas inteiro.

- **Limite o número de volumes.**

Em sistemas de nuvem e de contêiner, os volumes de armazenamento conectados à rede são comumente usados. Há limites para o número de volumes que podem ser anexados aos Nós do Linux. Por exemplo, [AWS EC2 limita a não mais de 30 volumes por VM](#). O Red Hat OpenShift Container Platform [tem um limite semelhante](#), assim como Microsoft Azure e Google Cloud Platform.

Um gerenciador de filas de HA nativa requer um volume para cada uma das três instâncias e força as instâncias a serem difundidas entre os Nós. No entanto, é possível configurar o gerenciador de filas para usar três volumes por instância (dados do gerenciador de filas, logs de recuperação e dados persistidos)...

- **Use IBM MQ técnicas de ajuste de escala.**

Em vez de um pequeno número de gerenciadores de fila grandes, pode ser benéfico usar técnicas de ajuste de escala do IBM MQ, como clusters uniformes do IBM MQ para executar vários gerenciadores de fila com a mesma configuração. Isso tem o benefício adicional de que o impacto de uma reinicialização de contêiner único (por exemplo, como parte da manutenção da plataforma de contêiner) é diminuído.

OpenShift > CP4I > CD > EUS Usando IBM MQ em IBM Cloud Pak for Integration e Red Hat OpenShift

O IBM MQ Operator implementa e gerencia o IBM MQ como parte do IBM Cloud Pak for Integration, ou independente no Red Hat OpenShift Container Platform.

Procedimento

- [“Histórico de liberação para o IBM MQ Operator”](#) na página 20.
- [“Migrando o IBM MQ para o IBM Cloud Pak for Integration”](#) na página 36.
- [“Instalando e desinstalando o IBM MQ Operator no Red Hat OpenShift”](#) na página 59.
- [“Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas”](#) na página 71.
- [“Implementando e configurando gerenciadores de filas usando o IBM MQ Operator”](#) na página 79.
- [“Operando o IBM MQ usando o IBM MQ Operator”](#) na página 115.
- [“Referência da API do IBM MQ Operator”](#) na página 126.

OpenShift > CP4I > CD > EUS Histórico de liberação para o IBM MQ Operator

IBM MQ Operator

IBM MQ Operator 1.8.2



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

Canal do operador

v1.8

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 e superior

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 e posterior (canal v3)

O Quê Há de Novo

- Atualização somente de segurança construída no [IBM MQ Operator 1.8.0](#).
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 1.8.1



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

Canal do operador

v1.8

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, [9.2.5.0-r2](#)

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 e superior

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 e posterior (canal v3)

O Quê Há de Novo

- Atualização somente de segurança construída no [IBM MQ Operator 1.8.0](#).
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 1.8.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

Canal do operador

v1.8

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, [9.2.5.0-r1](#)

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 e superior

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 e posterior (canal v3)

O Quê Há de Novo

- Inclui condições de status para versões do IBM MQ descontinuadas.

O que mudou

- Imagens movidas do Docker Hub para IBM Container Registry.
 - Os clientes com regras de firewall podem precisar ajustá-las para acessar as imagens no IBM Container Registry.
 - Os clientes com isolamento físico experimentam uma reinicialização de nó ao fazer upgrade para o IBM MQ Operator 1.8.0.
- Versões descontinuadas: IBM MQ 9.1.5, 9.2.0 CD, 9.2.1, 9.2.2. Essas versões podem não ser reconciliadas por versões futuras do IBM MQ Operator.
- Mudanças na lógica da licença: os clientes que fizerem upgrade do IBM MQ 9.2.5, poderão usar apenas as licenças especificadas para trabalhar com o IBM MQ 9.2.5. Consulte [“Referência de licenciamento para mq.ibm.com/v1beta1”](#) na página 126.
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 1.7.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

Canal do operador

v1.7

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 e superior

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 e posterior (canal v3)

O Quê Há de Novo

- Inclui IBM MQ 9.2.4 como uma liberação de entrega contínua

IBM MQ Operator 1.6.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.2.1

Canal do operador

v1.6

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 e superior

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.7 e posterior (canal v3)

O Quê Há de Novo

- Inclui o IBM MQ 9.2.3 como uma versão de entrega contínua (amd64 apenas para IBM Cloud Pak for Integration 2021.2.1; amd64 ou s390x ao usar uma licença IBM MQ)
- Novo tipo de disponibilidade para gerenciadores de filas: [Native HA](#). Disponível para uso de produção, como parte do IBM Cloud Pak for Integration 2021.2.1.

O que mudou

- IBM MQ Operator 1.6 e superior, use o IBM Container Registry em vez do Docker Hub. Isso significa que será necessário usar um `catalogsource` de `icr.io`. Consulte [“Instalando e desinstalando o IBM MQ Operator no Red Hat OpenShift”](#) na página 59.
- A atualização contínua do Native HA não espera mais que uma réplica esteja em sincronização antes de mover-se para a próxima réplica.
- Correção de problema com a afinidade do Native HA no OCP 4.7 e acima.
- Correção de problema ao usar certificados assinados por CA com Native HA.

IBM MQ Operator 1.5.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.1.1

Canal do operador

v1.5

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 e superior

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.7 e posterior (canal v3)

O Quê Há de Novo

- Inclui o IBM MQ 9.2.2 como uma versão de entrega contínua (amd64 apenas para IBM Cloud Pak for Integration 2021.1.1; amd64 ou s390x ao usar uma licença IBM MQ)
- Novo tipo de disponibilidade para gerenciadores de filas: Native HA. Disponível apenas para propósitos de avaliação, como parte do IBM Cloud Pak for Integration 2021.1.1.
- Integração com Monitoramento de cluster do Red Hat OpenShift Container Platform para métricas do Prometheus fornecendo um recurso ServiceMonitor

O que mudou

- O IBM Licensing Operator não é mais criado por padrão quando você cria um gerenciador de filas
- As atualizações para os gerenciadores de filas de várias instâncias agora são manipuladas em uma ordem contínua. Como parte dessa mudança, foi introduzida uma análise de inicialização do Kubernetes que afeta os valores usados ao configurar a análise de vivacidade. A análise de inicialização começa imediatamente, em seguida, aguarda até que o gerenciador de filas seja iniciado com sucesso. Se a análise de inicialização passar a qualquer momento dentro desse período de espera, as análises de prontidão serão iniciadas. Anteriormente, se você tivesse um gerenciador de filas que demorava para iniciar, era possível aumentar a configuração de `initialDelaySeconds` na análise de vivacidade. Se você fez isso, agora será necessário reverter `initialDelaySeconds` para a configuração anterior.
- O `CustomResourceDefinition` sofre um upgrade de `apiextensions.k8s.io/v1beta1` para `apiextensions.k8s.io/v1`

Problemas Conhecidos e Limitações

- Requer o IBM Cloud Pak foundational services 3.7, que contém uma mudança incompatível no componente Identity and Access Management (IAM). Se você tiver algum gerenciador de filas que use uma licença do IBM Cloud Pak for Integration, após este upgrade, será necessário uma reinicialização do gerenciador de filas para acessar o console da web e você também verá outros erros ao efetuar login no console da web. É possível corrigir esses erros fazendo o upgrade para o valor mais recente de `.spec.version` para a sua versão escolhida do IBM MQ, após o upgrade do operador ser concluído.
- A atualização contínua não será iniciada automaticamente se você estiver fazendo upgrade da versão do MQ. É necessário excluir os pods manualmente.

IBM MQ Operator 1.4.0

CD

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1 (o IBM MQ Operator 1.4.0 é uma liberação de CD e não está elegível para Suporte de atualização estendida)

Canal do operador

v1.4

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.1.0-r1

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 e superior

O Quê Há de Novo

- Inclui o IBM MQ 9.2.1 como uma liberação de entrega contínua

- Agora, é possível evitar a criação do gerenciador de filas padrão Route configurando `.spec.queueManager.route.enabled` para `false`

Problemas Conhecidos e Limitações

- Ao atualizar um `QueueManager` com um tipo de disponibilidade de `MultiInstance`, ambos os Pods serão excluídos imediatamente. Ambos devem ser reiniciados rapidamente pelo Red Hat OpenShift Container Platform.

IBM MQ Operator 1.3.8 (EUS)



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal do operador

v1.3-eus

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus

Versões do Red Hat OpenShift Container Platform

Apenas Red Hat OpenShift Container Platform 4.6

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canal `stable-v1`)

O Quê Há de Novo

- Inclui a nova versão do operando 9.2.0.6-r3-eus
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 1.3.7 (EUS)



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal do operador

v1.3-eus

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus

Versões do Red Hat OpenShift Container Platform

Apenas Red Hat OpenShift Container Platform 4.6

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canal `stable-v1`)

O Quê Há de Novo

- Inclui nova versão do operando 9.2.0.6-r2-eus.
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 1.3.6 (EUS)



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal do operador

v1.3-eus

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, [9.2.0.6-r1-eus](#)

Versões do Red Hat OpenShift Container Platform

Apenas Red Hat OpenShift Container Platform 4.6

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canal `stable-v1`)

O Quê Há de Novo

- Inclui nova versão do operando [9.2.0.6-r1-eus](#).
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 1.3.5 (EUS)



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal do operador

v1.3-eus

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, [9.2.0.5-r3-eus](#)

Versões do Red Hat OpenShift Container Platform

Apenas Red Hat OpenShift Container Platform 4.6

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canal `stable-v1`)

O Quê Há de Novo

- Inclui nova versão do operando [9.2.0.5-r3-eus](#)
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 1.3.4 (EUS)



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal do operador

v1.3-eus

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, [9.2.0.5-r2-eus](#)

Versões do Red Hat OpenShift Container Platform

Apenas Red Hat OpenShift Container Platform 4.6

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canal `stable-v1`)

O Quê Há de Novo

- Inclui nova versão do operando [9.2.0.5-r2-eus](#)
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 1.3.3 (EUS).



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal do operador

v1.3-eus

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, [9.2.0.5-r1-eus](#)

Versões do Red Hat OpenShift Container Platform

Apenas Red Hat OpenShift Container Platform 4.6

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canal `stable-v1`)

O Quê Há de Novo

- Inclui nova versão do operando [9.2.0.5-r1-eus](#)
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 1.3.2 (EUS)



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal do operador

v1.3-eus

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, [9.2.0.4-r1-eus](#)

Versões do Red Hat OpenShift Container Platform

Apenas Red Hat OpenShift Container Platform 4.6

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canal `stable-v1`)

O Quê Há de Novo

- Inclui nova versão do operando [9.2.0.4-r1-eus](#)

IBM MQ Operator 1.3.1 (EUS)



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal do operador

v1.3-eus

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, [9.2.0.2-r1-eus](#)

Versões do Red Hat OpenShift Container Platform

Apenas Red Hat OpenShift Container Platform 4.6

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canal `stable-v1`)

O Quê Há de Novo

- Inclui uma nova versão do operando [9.2.0.2-r1-eus](#)

IBM MQ Operator 1.3.0 (EUS).



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal do operador

v1.3-eus

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, [9.2.0.1-r1-eus](#)

Versões do Red Hat OpenShift Container Platform

Apenas Red Hat OpenShift Container Platform 4.6

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canal `stable-v1`)

O Quê Há de Novo

- O Suporte de atualização estendida (EUS) é oferecido para os campos `.spec.version` que terminam com `-eus`, ao usar uma licença IBM Cloud Pak for Integration
- Inclui uma nova maneira de configurar rótulos e anotações sobre o recurso `QueueManager` usando `.spec.labels` e `.spec.annotations`

O que mudou

- Melhora a manipulação de erros ao tentar mudar de única instância para multi-instância
- Melhorias na forma como as propriedades `QueueManager` são renderizadas no IBM Cloud Pak for Integration Platform Navigator e o "Form View" do console da web do Red Hat OpenShift Container Platform
- Corrige a métrica de licença padrão ao usar uma licença IBM Cloud Pak for Integration, para ser `VirtualProcessorCore`
- Corrige a guia **Recursos** para `QueueManager` no console da web do Red Hat OpenShift Container Platform, que agora mostra corretamente os recursos gerenciados pelo IBM MQ Operator para esse gerenciador de filas

Problemas Conhecidos e Limitações

- Ao atualizar um `QueueManager` com um tipo de disponibilidade de `MultiInstance`, ambos os Pods serão excluídos imediatamente. Ambos devem ser reiniciados rapidamente pelo Red Hat OpenShift Container Platform.

IBM MQ Operator 1.2.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.3.1

Canal do operador

v1.2

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, [9.2.0.0-r2](#)

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.4 e superior

O Quê Há de Novo

- Inclui suporte para z/Linux
- Inclui condições de status mais detalhadas para o recurso `QueueManager`. Para obter mais informações, consulte [“Condições de status para o QueueManager \(mq.ibm.com/v1beta1\)”](#) na página 145
- Inclui verificações de tempo de execução adicionais para evitar o uso de classes de armazenamento inválidas. Para obter mais informações, consulte [“Desativando as verificações de webhook de tempo de execução”](#) na página 114.

- Simplifica a experiência para os gerenciadores de filas de várias instâncias: isso agora pode ser escolhido com apenas uma propriedade (`.spec.queueManager.availability.type`) no recurso `QueueManager`
- Simplifica a escolha de uma Classe de armazenamento não padrão, introduzindo a propriedade `.spec.queueManager.storage.defaultClass` no recurso `QueueManager`

O que mudou

- Melhorias na forma como as propriedades `QueueManager` são renderizadas no IBM Cloud Pak for Integration Platform Navigator e o "Form View" do console da web do Red Hat OpenShift Container Platform
- Se uma versão do gerenciador de filas da qual foi feito upgrade estiver disponível, ela agora será sinalizada no IBM Cloud Pak for Integration Platform Navigator

IBM MQ Operator 1.1.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.2.1

Canal do operador

v1.1

Valores permitidos para `.spec.version`

9.1.5.0-r2, [9.2.0.0-r1](#)

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.4 e superior

O Quê Há de Novo

- Inclui o IBM MQ Advanced 9.2.0 como uma liberação de entrega contínua
- Inclui um recurso para especificar informações de INI e de MQSC em um ConfigMap ou Segredo
- Ativa o navegador do esquema ao usar o console da web do Red Hat OpenShift Container Platform

O que mudou

- Corrige o problema com a política de rede, impactando o Red Hat OpenShift no IBM Cloud
- Melhorias no webhook de validação, para evitar combinações inválidas de configurações em recursos `QueueManager`

IBM MQ Operator 1.0.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.2.1

Canal do operador

v1.0

Valores permitidos para `.spec.version`

[9.1.5.0-r2](#)

Versões do Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.4 e superior

O Quê Há de Novo

- Versão inicial do operador, introduzindo a API `mq.ibm.com/v1beta1`

Imagens de contêiner do gerenciador de filas para uso com o IBM MQ Operator

9.2.5.0-r3

CD

Versão do operador necessária

1.8.2 ou superior

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r3
- icr.io/ibm-messaging/mq:9.2.5.0-r3

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.2.5](#)

O que mudou

- [O que mudou no IBM MQ 9.2.5](#)
- Com base em [Red Hat Universal Base Image 8.6-751](#)

9.2.5.0-r2

CD

Versão do operador necessária

1.8.1 ou superior

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r2
- icr.io/ibm-messaging/mq:9.2.5.0-r2

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.2.5](#)

O que mudou

- [O que mudou no IBM MQ 9.2.5](#)
- Baseado em [Red Hat Universal Base Image 8.5-240.1648458092](#)

9.2.5.0-r1

CD

Versão do operador necessária

1.8.0 ou posterior

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r1
- icr.io/ibm-messaging/mq:9.2.5.0-r1

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.2.5](#)

O que mudou

- [O que mudou no IBM MQ 9.2.5](#)
- Opção Gerenciadores de filas remotas inválida agora removida de IBM MQ Console
- Baseado no [Red Hat Universal Base Image 8.5-240](#)

9.2.4.0-r1



Versão do operador necessária

[1.7.0](#) ou superior

Arquiteturas suportadas

amd64, s390x

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.4.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.4.0-r1](#)
- [docker.io/ibmcom/mq:9.2.4.0-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.2.4](#)

O que mudou

- [O que mudou no IBM MQ 9.2.4](#)
- Baseado no [Red Hat Universal Base Image 8.5-204](#)

9.2.3.0-r1



Versão do operador necessária

[1.6.0](#) ou superior

Arquiteturas suportadas

amd64, s390x

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.3.0-r1](#) (apenas amd64)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.3.0-r1](#)
- [docker.io/ibmcom/mq:9.2.3.0-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.2.3](#)
- Suporte para o MQ Native HA para uso de produção, quando usado com uma licença do IBM Cloud Pak for Integration. Note que os gerenciadores de filas que usam o Native HA sob uma licença de avaliação com o IBM MQ 9.2.2 não podem ser submetidos a upgrade para o 9.2.3. O período de avaliação terminou.

O que mudou

- [O que mudou no IBM MQ 9.2.3](#)
- Baseado no [Red Hat Universal Base Image 8.4-205](#)

9.2.2.0-r1

CD

Versão do operador necessária

[1.5.0](#) ou mais recente

Arquiteturas suportadas

amd64, s390x

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.2.0-r1](#) (apenas amd64)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.2.0-r1](#)
- [docker.io/ibmcom/mq:9.2.2.0-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.2.2](#)
- Suporte à [HA nativa](#) do MQ para propósitos de avaliação, ao usar com uma licença do IBM Cloud Pak for Integration

O que mudou

- [O que mudou no IBM MQ 9.2.2](#)
- Corrigido o problema que causa FDC ao encerrar um gerenciador de filas IBM MQ Advanced for Developers
- Baseado no [Red Hat Universal Base Image 8.3-291](#)

9.2.1.0-r2

CD

Versão do operador necessária

[1.5.0](#) ou mais recente

Arquiteturas suportadas

amd64, s390x

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.1.0-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.1.0-r2](#)
- [docker.io/ibmcom/mq:9.2.1.0-r2](#)

O que mudou

- Correção de problema em relação à conexão única com o IBM Cloud Pak foundational services 3.7 e superior.
- Baseado no [Red Hat Universal Base Image 8.3-291](#)

9.2.1.0-r1

CD

Versão do operador necessária

[1.4.0](#) ou mais recente

Arquiteturas suportadas

amd64, s390x

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.1.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.1.0-r1](#)

- docker.io/ibmcom/mq:9.2.1.0-r1

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.2.1](#)
- Informações de conexão para a Rota padrão estão disponíveis no console da web do MQ

O que mudou

- [O que mudou no IBM MQ 9.2.1](#)
- Baseado no [Red Hat Universal Base Image 8.3-230](#)

9.2.0.6-r3-eus



Versão do operador necessária

[1.3.8](#) e fix packs futuros

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r3-eus

O que mudou

- Inclui IBM MQ 9.2.0 Fix Pack 6. Para obter mais informações, consulte [Lista de Correções para o IBM MQ Versão 9.2 LTS](#).
- Baseado em [Red Hat Universal Base Image 8.6-941](#).

9.2.0.6-r2-eus



Versão do operador necessária

[1.3.7](#) e fix packs futuros

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r2-eus

O que mudou

- Inclui IBM MQ 9.2.0 Fix Pack 6. Para obter mais informações, consulte [Lista de Correções para o IBM MQ Versão 9.2 LTS](#).
- Baseado em [Red Hat Universal Base Image 8.6-902](#).

9.2.0.6-r1-eus



Versão do operador necessária

[1.3.6](#) e fix packs futuros

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r1-eus

O que mudou

- Inclui IBM MQ 9.2.0 Fix Pack 6. Para obter mais informações, consulte [Lista de Correções para o IBM MQ Versão 9.2 LTS](#).
- Baseado em [Red Hat Universal Base Image 8.6-854](#).

9.2.0.5-r3-eus



Versão do operador necessária

[1.3.5](#) e fix packs futuros

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r3-eus

O que mudou

- Inclui IBM MQ 9.2.0 Fix Pack 5. Para obter mais informações, consulte [O que mudou em IBM MQ 9.2.0 Fix Pack 5](#) e [Lista de correções para IBM MQ Versão 9.2 LTS](#).
- Com base em [Red Hat Universal Base Image 8.6-751.1655117800](#)

9.2.0.5-r2-eus



Versão do operador necessária

[1.3.4](#) e fix packs futuros

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r2-eus

O que mudou

- Inclui IBM MQ 9.2.0 Fix Pack 5. Para obter mais informações, consulte [O que mudou no IBM MQ 9.2.0 Fix Pack 5](#) e [Lista de correção para IBM MQ Versão 9.2 LTS](#)
- Com base em [Red Hat Universal Base Image 8.6-751](#)

9.2.0.5-r1-eus



Versão do operador necessária

[1.3.3](#) e fix packs futuros

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r1-eus

O que mudou

- Inclui IBM MQ 9.2.0 Fix Pack 5. Para obter mais informações, consulte [O que mudou no IBM MQ 9.2.0 Fix Pack 5](#) e [Lista de correção para IBM MQ Versão 9.2 LTS](#)
- Baseado em [Red Hat Universal Base Image 8.5-240.1648458092](#)

9.2.0.4-r1-eus

EUS

Versão do operador necessária

[1.3.2](#) e futuros fix packs

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.4-r1-eus

O que mudou

- Inclui IBM MQ 9.2.0 Fix Pack 4. Para obter mais informações, consulte [O que mudou no IBM MQ 9.2.0 Fix Pack 4](#) e [Lista de correção para IBM MQ Versão 9.2 LTS](#)
- Baseado no [Red Hat Universal Base Image 8.5-204](#)

9.2.0.2-r2-eus

EUS

Versão do operador necessária

[1.6.0](#) ou superior

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.2-r2-eus

O que mudou

- Correção de problema em relação à conexão única com o IBM Cloud Pak foundational services 3.7 e superior, que é necessária somente ao migrar de uma liberação de EUS para uma liberação de CD.
- Baseado no [Red Hat Universal Base Image 8.4-200.1622548483](#)

9.2.0.2-r1-eus

EUS

Versão do operador necessária

[1.3.1](#) e futuros fix packs ; [1.6.0](#) ou superior

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.2-r1-eus

O que mudou

- A integração do Operations Dashboard usa a versão 1.0.8 do agente e coletor de rastreo
- Inclui IBM MQ 9.2.0 Fix Pack 2. Para obter mais informações, consulte [O que mudou no IBM MQ 9.2.0 Fix Pack 2](#) e [Lista de correção para IBM MQ Versão 9.2 LTS](#)
- Baseado no [Red Hat Universal Base Image 8.4-200.1622548483](#)

9.2.0.1-r1-eus

EUS

Versão do operador necessária

[1.3.0](#) ou mais recente

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.1-r1-eus

O Quê Há de Novo

- Disponível somente ao usar uma licença do IBM Cloud Pak for Integration
- O Suporte de atualização estendida (EUS) está disponível ao usar o IBM MQ Operator 1.3.x e o IBM Common Services 3.6 no Red Hat OpenShift Container Platform 4.6

O que mudou

- Inclui IBM MQ 9.2.0 Fix Pack 1. Para obter mais informações, consulte [O que mudou no IBM MQ 9.2.0 Fix Pack 1](#) e [Lista de correção para IBM MQ Versão 9.2 LTS](#)
- Baseado no [Red Hat Universal Base Image 8.3-201](#)
- Corrige o problema com a análise de vivacidade (chkmqhealthy) e a análise de prontidão (chkmqready) ao executar no SecurityContextConstraints que permite a escalada de privilégios.

9.2.0.0-r3



Versão do operador necessária

[1.5.0](#) ou mais recente

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r3
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r3
- docker.io/ibmcom/mq:9.2.0.0-r3

O que mudou

- Baseado no [Red Hat Universal Base Image 8.3-291](#)

9.2.0.0-r2



Versão do operador necessária

[1.2.0](#) ou mais recente

Arquiteturas suportadas

amd64, s390x

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r2
- docker.io/ibmcom/mq:9.2.0.0-r2

O Quê Há de Novo

- Agora disponível no z/Linux

O que mudou

- Baseado no [Red Hat Universal Base Image 8.2-349](#)

9.2.0.0-r1

CD

Versão do operador necessária

[1.1.0](#) ou mais recente

Arquiteturas suportadas

amd64

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r1-amd64](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r1-amd64](#)
- [docker.io/ibmcom/mq:9.2.0.0-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.2.0](#)

O que mudou

- [O que mudou no IBM MQ 9.2.0](#)
- Utiliza argumento `-ic` para `crtmqm` para aplicar automaticamente arquivos MQSC. Substitui o uso anterior de comandos `runmqsc`
- Baseado no [Red Hat Universal Base Image 8.2-301.1593113563](#)

9.1.5.0-r2

CD

Versão do operador necessária

[1.0.0](#) ou mais recente

Arquiteturas suportadas

amd64

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.1.5.0-r2-amd64](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.1.5.0-r2-amd64](#)
- [docker.io/ibmcom/mq:9.1.5.0-r2](#)

O que mudou

- Baseado no [Red Hat Universal Base Image 8.2-267](#)

OpenShift

V 9.2.1

CD

EUS

Migrando o IBM MQ para o IBM Cloud

Pak for Integration

Este conjunto de tópicos descreve as principais etapas para migrar um gerenciador de filas do IBM MQ existente para um ambiente de contêiner usando o IBM MQ Operator no IBM Cloud Pak for Integration.

Sobre esta tarefa

Os clientes que implementam o IBM MQ no Red Hat OpenShift podem ser separados nos seguintes cenários:

1. Criação de uma nova implementação do IBM MQ em Red Hat OpenShift para novos aplicativos.
2. Estendendo uma rede IBM MQ em Red Hat OpenShift para novos aplicativos em Red Hat OpenShift.
3. Movendo uma implementação do IBM MQ em Red Hat OpenShift para continuar a suportar os aplicativos existentes.

Somente para o cenário 3 é necessário migrar sua configuração do IBM MQ. Os outros cenários são considerados novas implementações.

Este conjunto de tópicos se concentra no cenário 3 e descreve as principais etapas para migrar um gerenciador de filas do IBM MQ existente para um ambiente de contêiner usando o IBM MQ Operator. Devido à flexibilidade e ao uso extensivo do IBM MQ, há várias etapas opcionais. Cada uma delas inclui uma seção "Eu preciso fazer isso?". Verificar a sua necessidade deve economizar tempo durante a sua migração.

Também é necessário considerar quais dados migrar:

1. Migrar o IBM MQ com a mesma configuração mas sem nenhuma mensagem enfileirada existente.
2. Migrar o IBM MQ com a mesma configuração e as mensagens existentes.

Uma versão típica para a migração de versão pode usar qualquer abordagem. Em um gerenciador de filas típico do IBM MQ, no ponto de migração, há poucas (ou nenhuma) mensagens armazenadas em filas, o que torna a opção 1 apropriada para muitos casos. No caso da migração para uma plataforma de contêiner, é ainda mais comum usar a opção 1, para reduzir a complexidade da migração e permitir uma implementação azul-verde. Portanto, as instruções se concentram nesse cenário.

O objetivo desse cenário é criar um gerenciador de filas no ambiente de contêiner que corresponda à definição do gerenciador de filas existente. Isso permite que os aplicativos conectados à rede existente simplesmente sejam reconfigurados para apontar para o novo gerenciador de filas, sem mudar nenhuma outra lógica de configuração ou aplicativo.

Ao longo dessa migração, você gera vários arquivos de configuração a serem aplicados no novo gerenciador de filas. Para simplificar o gerenciamento desses arquivos, é necessário criar um diretório e gerá-los nele.

Procedimento

1. [“Verificando se as funções necessárias estão disponíveis” na página 37](#)
2. [“Extraindo a configuração do gerenciador de filas” na página 38](#)
3. Opcional: [“Opcional: extraindo e adquirindo as chaves e os certificados do gerenciador de filas” na página 39](#)
4. Opcional: [“Opcional: configurando o LDAP” na página 41](#)
5. Opcional: [“Opcional: mudando os endereços IP e nomes de host na configuração do IBM MQ” na página 47](#)
6. [“Atualizando a configuração do gerenciador de filas para um ambiente de contêiner” na página 49](#)
7. [“Selecionando a arquitetura de HA de destino para o IBM MQ em execução em contêineres” na página 51](#)
8. [“Criando os recursos para o gerenciador de filas” na página 52](#)
9. [“Criando o novo gerenciador de filas no Red Hat OpenShift” na página 53](#)
10. [“Verificando a nova implementação de contêiner” na página 57](#)



Verificando se as funções necessárias estão disponíveis

O IBM MQ Operator não inclui todos os recursos disponíveis no IBM MQ Advanced e deve-se verificar a necessidade deles. Outros recursos são parcialmente suportados e podem ser reconfigurados para corresponder ao que está disponível no contêiner.

Antes de começar

Esta é a primeira etapa no [“Migrando o IBM MQ para o IBM Cloud Pak for Integration” na página 36.](#)

Procedimento

1. Verifique se a imagem do contêiner de destino inclui todas as funções necessárias.

Para obter as informações mais recentes, consulte [“Escolhendo como você deseja usar o IBM MQ em contêineres”](#) na página 5.

2. O IBM MQ Operator tem uma única porta de tráfego do IBM MQ, conhecida como listener. Se você tiver vários listeners, simplifique isso para usar um único listener no contêiner. Como este não é um cenário comum, essa modificação não é documentada em detalhes.
3. Se saídas do IBM MQ forem usadas, migre-as para o contêiner criando camadas nos binários de saída do IBM MQ. Este é um cenário de migração avançada e, portanto, não incluído aqui. Para obter um esboço das etapas, consulte [“Construindo uma imagem com arquivos customizados de MQSC e INI, usando a CLI do Red Hat OpenShift”](#) na página 112.
4. Se o seu sistema IBM MQ incluir Alta Disponibilidade, revise as opções disponíveis.
Consulte o [“Alta disponibilidade para IBM MQ em contêineres”](#) na página 16.

Como proceder a seguir

Agora você está pronto para [extrair a configuração do gerenciador de filas](#).



Extraindo a configuração do gerenciador de filas

A maioria da configuração é móvel entre os gerenciadores de filas. Por exemplo, as coisas com as quais os aplicativos interagem, como definições de filas, tópicos e canais. Use esta tarefa para extrair a configuração do gerenciador de filas existente do IBM MQ.

Antes de começar

Esta tarefa supõe que você tenha [verificado se as funções necessárias estão disponíveis](#).

Procedimento

1. Efetue login na máquina com a instalação existente do IBM MQ.
2. Faça backup da configuração.

Execute o seguinte comando:

```
dmpmqcfg -m QMGR_NAME > /tmp/backup.mqsc
```

Notas de uso para este comando:

- Este comando armazena o backup no diretório tmp. É possível armazenar o backup em outro local, mas este cenário assume o diretório tmp para comandos subsequentes.
- Substitua *QMGR_NAME* pelo nome do gerenciador de filas de seu ambiente. Se você não tiver certeza do valor, execute o comando **dspmqr** para visualizar os gerenciadores de filas disponíveis na máquina. Veja a seguir a saída de comando **dspmqr** de amostra para um gerenciador de filas denominado qm1:

```
QMNAME(qm1)                STATUS(Running)
```

O comando **dspmqr** requer que o gerenciador de filas do IBM MQ seja iniciado, caso contrário, você receberá o erro a seguir:

```
AMQ8146E: IBM MQ queue manager not available.
```

Se necessário, inicie o gerenciador de filas executando o comando a seguir:

```
strmqm QMGR_NAME
```

Como proceder a seguir

Agora você está pronto para [extrair e adquirir as chaves e certificados do gerenciador de filas](#).

OpenShift V 9.2.1 CD EUS Opcional: extraindo e adquirindo as chaves e os certificados do gerenciador de filas

O IBM MQ pode ser configurado usando TLS para criptografar o tráfego para o gerenciador de filas. Use esta tarefa para verificar se o seu gerenciador de filas está usando TLS, para extrair chaves e certificados e para configurar o TLS no gerenciador de filas migrado.

Antes de começar

Esta tarefa assume que você [extraiu a configuração do gerenciador de filas](#).

Sobre esta tarefa

Eu preciso fazer isso?

O IBM MQ pode ser configurado para criptografar o tráfego no gerenciador de filas. Essa criptografia é concluída usando um repositório de chaves configurado no gerenciador de filas. Em seguida, os canais do IBM MQ ativam a comunicação TLS. Se você não tiver certeza se ele está configurado em seu ambiente, execute o comando a seguir para verificar:

```
grep 'SECCOMM(ALL\|SECCOMM(ANON\|SSLCIPH' backup.mqsc
```

Se nenhum resultado for localizado, o TLS não estará sendo usado. No entanto, isso não significa que o TLS não deve ser configurado no gerenciador de filas migrado. Há várias motivos para mudar esse comportamento:

- A abordagem de segurança no ambiente Red Hat OpenShift deve ser aprimorada em comparação com o ambiente anterior.
- Se você precisar acessar o gerenciador de filas migrado de fora do ambiente Red Hat OpenShift, o TLS é necessário para passar pelo Rota Red Hat OpenShift.

Procedimento

1. Extraia quaisquer certificados confiáveis do armazenamento existente.

Se o TLS estiver atualmente em uso no gerenciador de filas, o gerenciador de filas poderá ter um número de certificados confiáveis armazenados. Eles precisam ser extraídos e copiados para o novo gerenciador de filas. Conclua uma das etapas opcionais a seguir:

- Para aperfeiçoar a extração dos certificados, execute o script a seguir no sistema local:

```
#!/bin/bash
keyr=$(grep SSLKEYR $1)
if [ -n "$keyr" ]; then
  keyrlocation=$(sed -n "s/^\.*\(\.*\)'.*\$/\1/ p" <<< $keyr)
  mapfile -t runmqckmResult <<(runmqckm -cert -list -db $keyrlocation.kdb -stashed)
  cert=1
  for i in "${runmqckmResult[@]:1}"
  do
    certlabel=$(echo $i | xargs)
    echo Extracting certificate $certlabel to $cert.cert
    runmqckm -cert -extract -db $keyrlocation.kdb -label "$certlabel" -target $
  {cert}.cert -stashed
  cert=${cert+1}
  done
fi
```

Ao executar o script, especifique o local do backup do IBM MQ como um argumento e os certificados são extraídos. Por exemplo, se o script for chamado `extractCert.sh` e o backup do IBM MQ estiver localizado em `/tmp/backup.mqsc`, execute o comando a seguir:

```
extractCert.sh /tmp/backup.mqsc
```

- Como alternativa, execute os comandos a seguir na ordem mostrada:

- a. Identifique o local do armazenamento TLS:

```
grep SSLKEYR /tmp/backup.mqsc
```

Saída de amostra:

```
SSLKEYR('/run/runmqserver/tls/key') +
```

em que o armazenamento de chaves está localizado em `/run/runmqserver/tls/key.kdb`

- b. Com base nas informações desse local, consulte o armazenamento de chaves para determinar os certificados armazenados:

```
runmqckm -cert -list -db /run/runmqserver/tls/key.kdb -stashed
```

Saída de amostra:

```
Certificates in database /run/runmqserver/tls/key.kdb:  
  default  
  CN=cs-ca-certificate,0=cert-manager
```

- c. Extraia cada um dos certificados listados. Isso é feito executando-se o seguinte comando:

```
runmqckm -cert -extract -db KEYSTORE_LOCATION -label "LABEL_NAME" -target OUTPUT_FILE  
-stashed
```

Nas amostras mostradas anteriormente, isso é igual ao seguinte:

```
runmqckm -cert -extract -db /run/runmqserver/tls/key.kdb -label "CN=cs-ca-  
certificate,0=cert-manager" -target /tmp/cert-manager.crt -stashed  
runmqckm -cert -extract -db /run/runmqserver/tls/key.kdb -label "default" -target /tmp/  
default.crt -stashed
```

2. Adquira uma nova chave e certificado para o gerenciador de filas

Para configurar o TLS no gerenciador de filas migrado, você gera uma nova chave e certificado. Em seguida, isso é usado durante a implementação. Em muitas organizações, isso significa entrar em contato com sua equipe de segurança para solicitar uma chave e um certificado. Em algumas organizações, essa opção não está disponível e são usados certificados autoassinados.

O exemplo a seguir gera um certificado autoassinado no qual a expiração está configurada para 10 anos:

```
openssl req \  
  -newkey rsa:2048 -nodes -keyout qmgr.key \  
  -subj "/CN=mq queuemanager/OU=ibm mq" \  
  -x509 -days 3650 -out qmgr.crt
```

Dois novos arquivos são criados:

- `qmgr.key` é a chave privada para o gerenciador de filas
- `qmgr.crt` é o certificado público

Como proceder a seguir

Agora você está pronto para [configurar o LDAP](#).

O IBM MQ Operator pode ser configurado para usar várias abordagens de segurança diferentes. Geralmente, o LDAP é o mais eficaz para uma implementação corporativa e o LDAP é usado para esse cenário de migração.

Antes de começar

Esta tarefa assume que você extraiu e adquiriu as chaves e os certificados do gerenciador de filas.

Sobre esta tarefa

Eu preciso fazer isso?

Se você já estiver usando LDAP para autenticação e autorização, nenhuma mudança será necessária.

Se você não tiver certeza se o LDAP está sendo usado, execute o comando a seguir:

```
connauthname="$(grep CONNAUTH backup.mqsc | cut -d "(" -f2 | cut -d ")" -f1"; grep -A 20 AUTHINFO\($connauthname\) backup.mqsc
```

Saída de amostra:

```
DEFINE AUTHINFO('USE.LDAP') +
  AUTHTYPE(IDPWLDAP) +
  ADOPTCTX(YES) +
  CONNAME('ldap-service.ldap(389)') +
  CHCKCLNT(REQUIRED) +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
*  LDAPPWD('*****') +
  SHORTUSR('uid') +
  GRPFIELD('cn') +
  USRFIELD('uid') +
  AUTHORMD(SEARCHGRP) +
*  ALTDAT(2020-11-26) +
*  ALTTIME(15.44.38) +
  REPLACE
```

Há dois atributos na saída que são de interesse específico:

AUTHTYPE

Quando tem o valor IDPWLDAP, você está usando o LDAP para autenticação.

Se o valor estiver em branco ou for diferente, o LDAP não será configurado. Neste caso, verifique o atributo AUTHORMD para ver se os usuários do LDAP estão sendo usados para autorização.

AUTHORMD

Quando tem o valor OS, você não está usando o LDAP para autorização.

Para modificar a autorização e a autenticação para usar o LDAP, conclua as tarefas a seguir:

Procedimento

1. Atualize o backup do IBM MQ para o servidor LDAP.
2. Atualize o backup do IBM MQ para as informações de autorização do LDAP.

Uma descrição abrangente de como configurar o LDAP está fora do escopo deste cenário. Este tópico fornece um resumo do processo, uma amostra e referências a informações adicionais.

Antes de começar

Esta tarefa assume que você extraiu e adquiriu as chaves e os certificados do gerenciador de filas.

Sobre esta tarefa

Eu preciso fazer isso?

Se você já estiver usando LDAP para autenticação e autorização, nenhuma mudança será necessária. Se você não tiver certeza se o LDAP está sendo usado, consulte [“Opcional: configurando o LDAP”](#) na página 41.

Há duas partes para configurar o servidor LDAP:

1. [Defina uma configuração do LDAP.](#)
2. [Associe a configuração do LDAP com a definição do gerenciador de filas.](#)

Mais informações para ajudá-lo com esta configuração:

- [Visão geral do repositório do usuário](#)
- [Guia de referência para o comando AUTHINFO](#)

Procedimento

1. Defina uma configuração do LDAP.

Edite o arquivo `backup.mqsc` para definir um novo objeto **AUTHINFO** para o sistema LDAP. Por exemplo:

```
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember')
REPLACE
```

em que

- **CONNAME** é o nome do host e a porta correspondentes ao servidor LDAP. Se houver vários endereços para resiliência, eles poderão ser configurados usando uma lista separada por vírgula.
- **LDAPUSER** é o nome distinto correspondente ao usuário que o IBM MQ usa ao se conectar ao LDAP para consultar registros do usuário.
- **LDAPPWD** é a senha que corresponde ao usuário **LDAPUSER**.
- **SECCOM** especifica se a comunicação para o servidor LDAP deve usar TLS. Valores possíveis:
 - YES: o TLS é usado e um certificado é apresentado pelo servidor IBM MQ.
 - ANON: o TLS é usado sem que um certificado seja apresentado pelo servidor IBM MQ.
 - NO: o TLS não é usado durante a conexão.
- **USRFIELD** especifica o campo no registro LDAP com relação ao qual o nome do usuário apresentado é correspondido.
- **SHORTUSR** é um campo dentro do registro LDAP que não excede 12 caracteres de comprimento. O valor neste campo representará a identidade declarada se a autenticação tiver sido bem-sucedida.
- **BASEDNU** é o DN base que deve ser usado para procura do LDAP.
- **BASEDNG** é o DN base para grupos dentro do LDAP.

- **AUTHORMD** define o mecanismo usado para resolver a associação ao grupo para o usuário. Existem quatro opções:
 - OS: consultar o sistema operacional para os grupos associados com o nome abreviado.
 - SEARCHGRP: procurar as entradas de grupo no LDAP para o usuário autenticado.
 - SEARCHUSR: procurar o registro do usuário autenticado para informações de associação do grupo.
 - SRCHGRPSN: procurar as entradas do grupo no LDAP para o nome do usuário abreviado de usuários autenticados (definido pelo campo SHORTUSR).
- **GRPFIELD** é o atributo dentro do registro do grupo LDAP que corresponde a um nome simples. Se especificado, poderá ser usado para a definição de registros de autorização.
- **CLASSUSR** é a classe de objeto LDAP que corresponde a um usuário.
- **CLASSGRP** é a classe de objeto LDAP que corresponde a um grupo.
- **FINDGRP** é o atributo dentro do registro LDAP que corresponde à associação ao grupo.

A nova entrada pode ser colocada em qualquer lugar do arquivo, no entanto, pode ser útil colocar novas entradas no início do arquivo:

```

*****
* Script generated on 2020-10-21   at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQ
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dpmmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.11) +

```

2. Associe a configuração do LDAP com a definição do gerenciador de filas.

É necessário associar a configuração do LDAP com a definição do gerenciador de filas. Imediatamente abaixo da entrada DEFINE AUTHINFO há uma entrada ALTER QMGR. Modifique a entrada CONNAUTH para corresponder ao nome de AUTHINFO recém-criado. Por exemplo, no exemplo

anterior AUTHINFO(USE.LDAP) foi definido, significando que o nome é USE.LDAP. Portanto, mude CONNAUTH('SYSTEM.DEFAULT.AUTHINFO.IDPWOS') para CONNAUTH('USE.LDAP'):

```
Open [icon]
backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'l
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSID(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDO(SYSTEM.ADMIN.COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +
..
```

Para fazer com que a comutação para o LDAP ocorra imediatamente, chame um comando REFRESH SECURITY, incluindo uma linha imediatamente após o comando ALTER QMGR:

```

*backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQKJ5UH'
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfc -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSID(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +
* CRDATE(2020-10-26) +
* CRTIME(11.43.11) +
* QMID(qm1_2020-10-26_11.43.11) +
  SSLCRYP(' ') +
  SSLKEYR('/run/runmqserver/tls/key') +
  SUITEB(NONE) +
* VERSION(09020000) +
  FORCE
REFRESH SECURITY

```

Como proceder a seguir

Agora você está pronto para [atualizar o backup do IBM MQ para informações de autorização do LDAP](#).

OpenShift V 9.2.1 CD EUS **LDAP parte 2: atualizando o backup do IBM MQ para informações de autorização do LDAP**

O IBM MQ fornece regras de autorização refinadas que controlam o acesso aos objetos do IBM MQ. Se você tiver mudado a autenticação e a autorização para o LDAP, as regras de autorização poderão ser inválidas e requerer atualização.

Antes de começar

Esta tarefa assume que você atualizou o backup para o servidor LDAP.

Sobre esta tarefa

Eu preciso fazer isso?

Se você já estiver usando LDAP para autenticação e autorização, nenhuma mudança será necessária. Se você não tiver certeza se o LDAP está sendo usado, consulte [“Opcional: configurando o LDAP” na página 41](#).

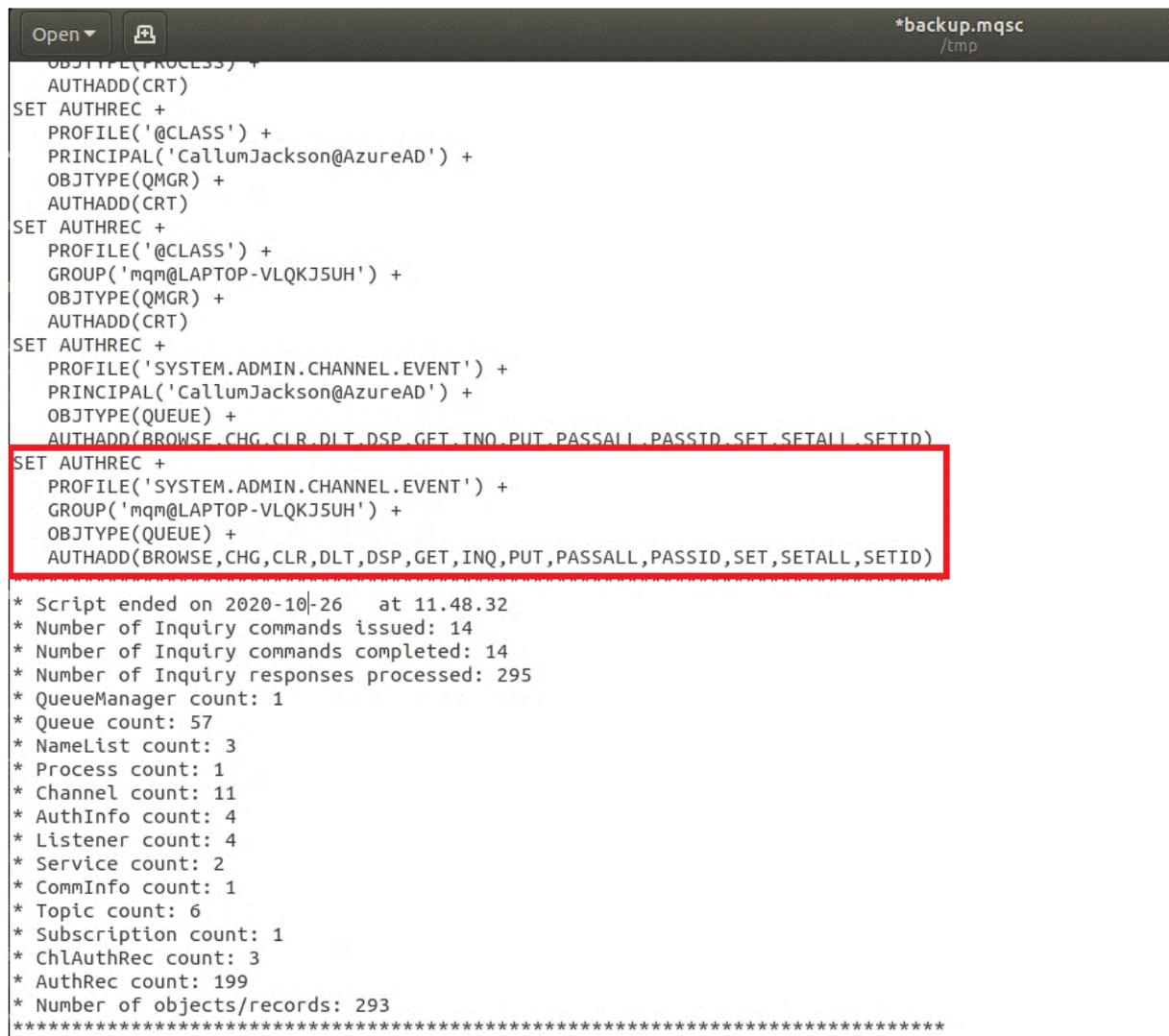
Há duas partes para atualizar as informações de autorização do LDAP:

1. Remova toda a autorização existente do arquivo.
2. Defina novas informações de autorização para o LDAP.

Procedimento

1. Remova toda a autorização existente do arquivo.

Próximo ao final do arquivo de backup, devem ser exibidas várias entradas que começam com SET AUTHREC:



```
Open [icon] *backup.mqsc /tmp
OBJTYPE(PROCESS) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('@CLASS') +
PRINCIPAL('CallumJackson@AzureAD') +
OBJTYPE(QMGR) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('@CLASS') +
GROUP('mqm@LAPTOP-VLQKJ5UH') +
OBJTYPE(QMGR) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('SYSTEM.ADMIN.CHANNEL.EVENT') +
PRINCIPAL('CallumJackson@AzureAD') +
OBJTYPE(Queue) +
AUTHADD(BROWSE,CHG,CLR,DLT,DSP,GET,INQ,PUT,PASSALL,PASSID,SET,SETALL,SETID)
SET AUTHREC +
PROFILE('SYSTEM.ADMIN.CHANNEL.EVENT') +
GROUP('mqm@LAPTOP-VLQKJ5UH') +
OBJTYPE(Queue) +
AUTHADD(BROWSE,CHG,CLR,DLT,DSP,GET,INQ,PUT,PASSALL,PASSID,SET,SETALL,SETID)

* Script ended on 2020-10-26 at 11.48.32
* Number of Inquiry commands issued: 14
* Number of Inquiry commands completed: 14
* Number of Inquiry responses processed: 295
* QueueManager count: 1
* Queue count: 57
* NameList count: 3
* Process count: 1
* Channel count: 11
* AuthInfo count: 4
* Listener count: 4
* Service count: 2
* CommInfo count: 1
* Topic count: 6
* Subscription count: 1
* ChlAuthRec count: 3
* AuthRec count: 199
* Number of objects/records: 293
*****
```

Localize as entradas existentes e exclua-as. A abordagem mais direta é remover todas as regras de SET AUTHREC existentes, em seguida, criar novas entradas com base nas entradas LDAP.

2. Defina novas informações de autorização para o LDAP

Dependendo da configuração de seu gerenciador de filas, e do número de recursos e grupos, essa pode ser uma atividade demorada ou direta. O exemplo a seguir assume que o seu gerenciador de filas tem apenas uma única fila chamada Q1 e você deseja permitir que o grupo LDAP apps tenha acesso.

```
SET AUTHREC GROUP('apps') OBJTYPE(QMGR) AUTHADD(ALL)
SET AUTHREC PROFILE('Q1') GROUP('apps') OBJTYPE(Queue) AUTHADD(ALL)
```

O primeiro comando AUTHREC inclui permissão para acessar o gerenciador de filas e o segundo fornece acesso à fila. Se o acesso a uma segunda fila for necessário, um terceiro comando AUTHREC será necessário, a menos que você tenha decidido usar curingas para fornecer acesso mais genérico.

A seguir está outro exemplo. Se um grupo de administradores (chamado admins) precisar de acesso total ao gerenciador de filas, inclua os comandos a seguir:

```
SET AUTHREC PROFILE('*') OBJTYPE(Queue) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Topic) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Channel) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(CLNTCONN) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(AUTHINFO) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Listener) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Namelist) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Process) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Service) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(QMGR) GROUP('admins') AUTHADD(ALL)
```

Como proceder a seguir

Agora você está pronto para [mudar os endereços IP e os nomes de host na configuração do IBM MQ](#).

Opcional: mudando os endereços IP e nomes de host na configuração do IBM MQ

A configuração do IBM MQ pode ter endereços IP e nomes de host especificados. Em algumas situações, eles podem permanecer, enquanto em outras situações, eles precisam ser atualizados.

Antes de começar

Esta tarefa assume que você [configurou o LDAP](#).

Sobre esta tarefa

Eu preciso fazer isso?

Primeiro, determine se você tem algum endereço IP ou nome de host especificado, além da configuração do LDAP definida na seção anterior. Para fazer isso, execute o seguinte comando:

```
grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc
```

Saída de amostra:

```
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
--
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.IDPWLDAP') +
  AUTHTYPE(IDPWLDAP) +
  ADOPTCTX(YES) +
  CONNAME(' ') +
--
REPLACE
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.CRLLDAP') +
  AUTHTYPE(CRLLDAP) +
  CONNAME(' ') +
```

Neste exemplo, a procura retorna três resultados. Um resultado corresponde à configuração do LDAP definida anteriormente. Isso pode ser ignorado, porque o nome do host do servidor LDAP está permanecendo o mesmo. Os outros dois resultados são entradas de conexão vazias, portanto, eles também podem ser ignorados. Se você não tiver nenhuma entrada adicional, será possível ignorar o restante deste tópico.

Procedimento

1. Entenda as entradas retornadas.

O IBM MQ pode incluir endereços IP, nomes de host e portas dentro de muitos aspectos da configuração. Nós podemos classificá-los em duas categorias:

- a. **Local deste gerenciador de filas:** informações do local que este gerenciador de filas usa ou publica, que outros gerenciadores de filas ou aplicativos dentro de uma rede IBM MQ podem usar para conectividade.
- b. **Localização das dependências do gerenciador de filas:** os locais de outros gerenciadores de filas ou sistemas dos quais este gerenciador de filas precisa ter conhecimento.

Como esse cenário é focado apenas nas mudanças para essa configuração do gerenciador de filas, nós manipulamos apenas as atualizações de configuração para a categoria (a). No entanto, se esse local do gerenciador de filas for referenciado por outros gerenciadores de filas ou aplicativos, as configurações deles poderão precisar de atualização para corresponder ao novo local.

Há dois objetos chave que podem conter informações que precisam ser atualizadas:

- Listeners: eles representam o endereço de rede no qual o IBM MQ está atendendo.
 - Canal CLUSTER RECEIVER: se o gerenciador de filas faz parte de um cluster do IBM MQ, esse objeto existe. Ele especifica o endereço de rede ao qual outros gerenciadores de filas podem se conectar.
2. Na saída original do comando `grep 'CONNNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc`, identifique se algum canal CLUSTER RECEIVER está definido. Em caso afirmativo, atualize os endereços IP.

Para identificar se todos os canais CLUSTER RECEIVER estão definidos, localize quaisquer entradas com `CHLTYPE(CLUSRCVR)` na saída original:

```
DEFINE CHANNEL(ANY_NAME) +  
  CHLTYPE(CLUSRCVR) +
```

Se existirem entradas, atualize o `CONNNAME` com a rota do IBM MQ Red Hat OpenShift. Este valor é baseado no ambiente Red Hat OpenShift e usa uma sintaxe previsível:

```
queue_manager_resource_name-ibm-mq-qm-openshift_project_name.openshift_app_route_hostname
```

Por exemplo, se a implementação do gerenciador de filas é denominada `qm1` dentro do namespace `cp4i` e o `openshift_app_route_hostname` é `apps.callumj.icp4i.com`, então a URL da rota é esta:

```
qm1-ibm-mq-qm-cp4i.apps.callumj.icp4i.com
```

O número da porta para a rota geralmente é 443. A menos que o seu Administrador Red Hat OpenShift fale algo diferente disso, este é normalmente o valor correto. Usando essas informações, atualize os campos `CONNNAME`. Por exemplo:

```
CONNNAME('qm1-ibm-mq-qm-cp4i.apps.callumj.icp4i.com(443)')
```

Na saída original do comando `grep 'CONNNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc`, verifique se há alguma entrada para `LOCLADDR` ou `IPADDRV`. Se existir, exclua-as. Elas não são relevantes em um ambiente de contêiner.

Como proceder a seguir

Agora você está pronto para atualizar a configuração do gerenciador de filas para um ambiente de contêiner.

Atualizando a configuração do gerenciador de filas para um ambiente de contêiner

Ao executar em um contêiner, certos aspectos de configuração são definidos pelo contêiner e podem entrar em conflito com a configuração exportada.

Antes de começar

Esta tarefa assume que você mudou a configuração do IBM MQ de endereços IP e nomes de host.

Sobre esta tarefa

Os aspectos de configuração a seguir são definidos pelo contêiner:

- As definições do listener (que correspondem às portas expostas).
- O local de qualquer armazenamento TLS em potencial.

Portanto, é necessário atualizar a configuração exportada:

1. Remova quaisquer definições de listener.
2. Defina o local do repositório de chaves TLS.

Procedimento

1. Remova quaisquer definições de listener.

Na configuração de backup, procure por `DEFINE LISTENER`. Isso deve estar entre as definições `AUTHINFO` e `SERVICE`. Destaque a área e exclua-a.

```

*backup.mqsc
** ALTDATA(2020-11-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.CRLLDAP') +
  AUTHTYPE(CRLLDAP) +
  CONNAME(' ') +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.LU62') +
  TRPTYPE(LU62) +
  CONTROL(MANUAL) +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.NETBIOS') +
  TRPTYPE(NETBIOS) +
  CONTROL(MANUAL) +
  LOCLNAME(' ') +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.SPX') +
  TRPTYPE(SPX) +
  CONTROL(MANUAL) +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.TCP') +
  TRPTYPE(TCP) +
  CONTROL(MANUAL) +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE SERVICE('SYSTEM.AMQP.SERVICE') +
  CONTROL(QMGR) +
  SERVTYPE(SERVER) +
  STARTCMD('+MQ_INSTALL_PATH+\bin\amqp.bat') +
  STARTARG('start -m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\.'
  STOPCMD('+MQ_INSTALL_PATH+\bin64\endmqsd.exe') +

```

2. Defina o local do repositório de chaves TLS.

O backup do gerenciador de filas contém a configuração do TLS para o ambiente original. Isso é diferente do ambiente de contêiner e, portanto, são necessárias algumas atualizações:

- Mude a entrada **CERTLABL** para default
- Mude o local do repositório de chaves TLS (**SSLKEYR**) para: /run/runmqserver/tls/key

Para localizar o local do atributo **SSLKEYR** no arquivo, procure por **SSLKEYR**. Geralmente, apenas uma entrada é localizada. Se várias entradas forem localizadas, verifique se você está editando o objeto **QMGR**, conforme mostrado na ilustração a seguir:

```

*backup.mqsc
*****
* Script generated on 2020-10-21   at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQKJ5UH'
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSTD(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +
* CRDATE(2020-10-26) +
* CRTIME(11.43.11) +
* QMID(qm1_2020-10-26_11.43.11) +
  SSLCRYP(' ') +
  SSLKEYR('/run/runmqserver/tls/key') +
  SUITEB(NONE) +
* VERSION(09020000) +
  FORCE
REFRESH SECURITY

```

Como proceder a seguir

Agora você está pronto para selecionar a arquitetura de destino para o IBM MQ em execução em contêineres.

OpenShift V 9.2.1 CD EUS Selecionando a arquitetura de HA de destino para o IBM MQ em execução em contêineres

Escolha entre a instância única (um único Pod do Kubernetes) e multi-instância (dois Pods) para atender aos seus requisitos de alta disponibilidade.

Antes de começar

Esta tarefa assume que você [atualizou a configuração do gerenciador de filas para um ambiente de contêiner](#).

Sobre esta tarefa

O IBM MQ Operator fornece duas opções de alta disponibilidade:

- **Instância única:** um único contêiner (Pod) é iniciado e é de responsabilidade de Red Hat OpenShift reiniciar em caso de falha. Devido às características de um conjunto stateful dentro do Kubernetes, há várias situações nas quais esse failover pode levar um período de tempo estendido ou requerer uma ação administrativa.
- **Multi-instância:** dois contêineres (cada um em um Pod separado) são iniciados, um no modo ativo e outro em espera. Essa topologia permite um failover muito mais rápido. Ela requer um sistema de arquivos Read Write Many que atende aos requisitos do IBM MQ.

Nesta tarefa você escolhe apenas a arquitetura de HA de destino. As etapas para configurar sua arquitetura escolhida são descritas em uma tarefa subsequente neste cenário ([“Criando o novo gerenciador de filas no Red Hat OpenShift” na página 53](#)).

Procedimento

1. Revise as duas opções.

Para obter uma descrição abrangente dessas duas opções, consulte [“Alta disponibilidade para IBM MQ em contêineres” na página 16](#).

2. Selecione a arquitetura de HA de destino.

Se você não tiver certeza de qual opção escolher, comece com a opção **Instância única** e verifique se ela atende aos seus requisitos de alta disponibilidade.

Como proceder a seguir

Agora você está pronto para [criar os recursos do gerenciador de filas](#).

Criando os recursos para o gerenciador de filas

Importe a configuração do IBM MQ, os certificados e as chaves TLS para o ambiente do Red Hat OpenShift.

Antes de começar

Esta tarefa assume que você [selecionou a arquitetura de destino para o IBM MQ em execução em contêineres](#).

Sobre esta tarefa

Nas seções anteriores, você extraiu, atualizou e definiu dois recursos:

- Configuração do IBM MQ
- Certificados e chaves TLS

É necessário importar esses recursos para o ambiente Red Hat OpenShift antes que o gerenciador de filas seja implementado.

Procedimento

1. Importe a configuração do IBM MQ em Red Hat OpenShift.

As instruções a seguir assumem que você tem a configuração do IBM MQ no diretório atual, em um arquivo chamado `backup.mqsc`. Caso contrário, é necessário customizar o nome do arquivo com base em seu ambiente.

- a) Efetue login em seu cluster usando `oc login`.
- b) Carregue a configuração do IBM MQ para um `configmap`.

Execute o seguinte comando:

```
oc create configmap my-mqsc-migrated --from-file=backup.mqsc
```

- c) Verifique se o arquivo foi carregado com sucesso.

Execute o seguinte comando:

```
oc describe configmap my-mqsc-migrated
```

2. Importe os recursos do TLS do IBM MQ

Conforme discutido em [“Opcional: extraindo e adquirindo as chaves e os certificados do gerenciador de filas”](#) na página 39, o TLS pode ser necessário para a implementação do gerenciador de filas. Se sim, você já deve ter um número de arquivos terminando com `.crt` e `.key`. É necessário incluí-los nos segredos do Kubernetes para o gerenciador de filas para referência no momento da implementação.

Por exemplo, se você tinha uma chave e um certificado para o gerenciador de filas, é possível que eles tenham os nomes a seguir:

- `qmgr.crt`
- `qmgr.key`

Para importar esses arquivos, execute o comando a seguir:

```
oc create secret tls my-tls-migration --cert=qmgr.crt --key=qmgr.key
```

O Kubernetes fornece este utilitário útil quando você está importando uma chave pública e privada correspondente. Se você tiver certificados adicionais para incluir, por exemplo, no armazenamento confiável do gerenciador de filas, execute o comando a seguir:

```
oc create secret generic my-extra-tls-migration --from-file=comma_separated_list_of_files
```

Por exemplo, se os arquivos a serem importados são `trust1.crt`, `trust2.crt` e `trust3.crt`, o comando é este:

```
oc create secret generic my-extra-tls-migration --from-file=trust1.crt,trust2.crt,trust3.crt
```

Como proceder a seguir

Agora você está pronto para [criar o novo gerenciador de filas no Red Hat OpenShift](#).

Criando o novo gerenciador de filas no Red Hat OpenShift

Implemente um gerenciador de filas de instância única ou de várias instâncias no Red Hat OpenShift.

Antes de começar

Esta tarefa supõe que você [criou os recursos do gerenciador de filas](#) e [instalou o IBM MQ Operator para o Red Hat OpenShift](#).

Sobre esta tarefa

Conforme descrito em [“Selecionando a arquitetura de HA de destino para o IBM MQ em execução em contêineres”](#) na página 51, há duas topologias de implementação possíveis. Portanto, este tópico fornece dois modelos diferentes:

- [Implementar um gerenciador de filas de instância única.](#)
- [Implementar um gerenciador de filas de várias instâncias.](#)

Importante: Apenas conclua um dos dois modelos, com base em sua topologia preferencial.

Procedimento

- Implemente um gerenciador de filas de instância única.

O gerenciador de filas migrado é implementado para o Red Hat OpenShift usando um arquivo YAML. Veja a seguir uma amostra, com base nos nomes usados em tópicos anteriores:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
spec:
  version: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJ0N-C7QG3S
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
          items:
            - tls.key
            - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
  mqsc:
    - configMap:
        name: my-mqsc-migrated
        items:
          - backup.mqsc
```

Dependendo das etapas que você executou, o YAML anterior pode precisar ser customizado. Para ajudá-lo com isso, veja a seguir uma explicação sobre este YAML:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
```

Isso define o objeto do Kubernetes, o tipo e o nome. O único campo que exige customização é o campo name.

```
spec:
  version: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJ0N-C7QG3S
    use: "Production"
```

Isso corresponde à versão e às informações de licença para a implementação. Se for necessário fazer a customização, use as informações fornecidas em [“Referência de licenciamento para mq.ibm.com/v1beta1”](#) na página 126.

```
pki:
  keys:
```

```

- name: default
  secret:
    secretName: my-tls-migration
    items:
      - tls.key
      - tls.crt

```

Para que o gerenciador de filas seja configurado para usar TLS, ele deve referenciar os certificados e chaves relevantes. O campo `secretName` referencia o segredo Kubernetes criado na seção [Importar os recursos TLS IBM MQ](#) e a lista de itens (`tls.key` e `tls.crt`) são os nomes padrão Kubernetes atribuídos ao usar a sintaxe `oc create secret tls`. Se você tiver certificados adicionais para incluir no armazenamento confiável, eles poderão ser incluídos de uma maneira semelhante, mas os itens são os nomes de arquivo correspondentes usados durante a importação. Por exemplo, o código a seguir pode ser usado para criar os certificados de armazenamento confiável:

```
oc create secret generic my-extra-tls-migration --from-file=trust1.crt,trust2.crt,trust3.crt
```

```

pki:
  trust:
    - name: default
      secret:
        secretName: my-extra-tls-migration
        items:
          - trust1.crt
          - trust2.crt
          - trust3.crt

```

Importante: Se o TLS não for necessário, exclua a seção TLS do YAML.

```

web:
  enabled: true

```

Isso ativa o console da web para a implementação

```

queueManager:
  name: QM1

```

Isso define o nome do gerenciador de filas como QM1. O gerenciador de filas é customizado com base em seus requisitos, por exemplo, qual era o nome do gerenciador de filas original.

```

mqsc:
  - configMap:
      name: my-mqsc-migrated
      items:
        - backup.mqsc

```

O código anterior faz pull na configuração do gerenciador de filas que foi importada na seção [Importar a configuração do IBM MQ](#). Se você usou nomes diferentes, será necessário modificar `my-mqsc-migrated` e `backup.mqsc`.

Observe que o YAML de amostra assume que a classe de armazenamento padrão para o ambiente do Red Hat OpenShift é definida como uma classe de armazenamento RWX ou RWO. Se um padrão não for definido dentro de seu ambiente, será necessário especificar a classe de armazenamento a ser usada. É possível fazer isso estendendo o YAML como a seguir:

```

queueManager:
  name: QM1
  storage:
    defaultClass: my_storage_class
  queueManager:
    type: persistent-claim

```

Inclua o texto destacado, com o atributo de classe customizado para corresponder ao seu ambiente. Para descobrir os nomes de classes de armazenamento dentro de seu ambiente, execute o comando a seguir:

```
oc get storageclass
```

Veja a seguir a saída de amostra retornada por este comando:

NAME	PROVISIONER	RECLAIMPOLICY
aws-efs	openshift.org/aws-efs	Delete
gp2 (default)	kubernetes.io/aws-efs	Delete

O código a seguir mostra como referenciar a configuração do IBM MQ que foi importada na seção [Importar a configuração do IBM MQ](#). Se você usou nomes diferentes, será necessário modificar `my-mqsc-migrated` e `backup.mqsc`.

```
mqsc:
  - configMap:
      name: my-mqsc-migrated
      items:
        - backup.mqsc
```

Você implementou seu gerenciador de filas de instância única. Isso conclui o modelo. Agora você está pronto para [verificar a implementação do novo contêiner](#).

- Implemente um gerenciador de filas de várias instâncias.

O gerenciador de filas migrado é implementado para o Red Hat OpenShift usando um arquivo YAML. A amostra a seguir é baseada nos nomes usados em seções anteriores.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1mi
spec:
  version: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
          items:
            - tls.key
            - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    availability: MultiInstance
  storage:
    defaultClass: aws-efs
    persistedData:
      enabled: true
    queueManager:
      enabled: true
    recoveryLogs:
      enabled: true
  mqsc:
    - configMap:
        name: my-mqsc-migrated
        items:
          - backup.mqsc
```

Veja a seguir uma explicação sobre este YAML. A maioria da configuração segue a mesma abordagem que a implementação de um gerenciador de filas de instância única, portanto, apenas os aspectos de disponibilidade e armazenamento do gerenciador de filas são explicados aqui.

```
queueManager:  
  name: QM1  
  availability: MultiInstance
```

Isso especifica o nome do gerenciador de filas como QM1 e configura a implementação para ser MultiInstance em vez da instância única padrão.

```
storage:  
  defaultClass: aws-efs  
  persistedData:  
    enabled: true  
  queueManager:  
    enabled: true  
  recoveryLogs:  
    enabled: true
```

Um gerenciador de filas de várias instâncias do IBM MQ depende do armazenamento RWX. Por padrão, um gerenciador de filas é implementado no modo de instância única e, portanto, opções de armazenamento adicionais são necessárias ao mudar para o modo de várias instâncias. Na amostra anterior do YAML, foram definidos três volumes persistentes de armazenamento e uma classe de volume persistido. Essa classe de volume persistido precisa ser uma classe de armazenamento RWX. Se você não tiver certeza dos nomes de classe de armazenamento em seu ambiente, execute o comando a seguir para descobrir:

```
oc get storageclass
```

Veja a seguir a saída de amostra retornada por este comando:

NAME	PROVISIONER	RECLAIMPOLICY
aws-efs	openshift.org/aws-efs	Delete
gp2 (default)	kubernetes.io/aws-efs	Delete

O código a seguir mostra como referenciar a configuração do IBM MQ que foi importada na seção [Importar a configuração do IBM MQ](#). Se você usou nomes diferentes, será necessário modificar `my-mqsc-migrated` e `backup.mqsc`.

```
mqsc:  
  - configMap:  
    name: my-mqsc-migrated  
    items:  
      - backup.mqsc
```

Você implementou o seu gerenciador de filas de várias instâncias. Isso conclui o modelo. Agora você está pronto para [verificar a implementação do novo contêiner](#).

Verificando a nova implementação de contêiner

Agora que o IBM MQ está implementado no Red Hat OpenShift, é possível verificar o ambiente usando as amostras do IBM MQ.

Antes de começar

Esta tarefa assume que você [criou o novo gerenciador de filas no Red Hat OpenShift](#).

Importante: Esta tarefa assume que o TLS não está ativado no gerenciador de filas.

Sobre esta tarefa

Nesta tarefa, você executa as amostras do IBM MQ de dentro do contêiner do gerenciador de filas migrado. No entanto, é possível que você prefira usar seus próprios aplicativos em execução em outro ambiente.

Você precisa das seguintes informações:

- Nome de Usuário do LDAP
- Senha do LDAP
- Nome de canal do IBM MQ
- Nome da fila

Este código de exemplo usa as configurações a seguir. Observe que suas configurações serão diferentes.

- Nome do usuário do LDAP: mqapp
- Senha do LDAP: mqapp
- Nome do canal do IBM MQ: DEV.APP.SVRCONN
- Nome da fila: Q1

Procedimento

1. Executável para o contêiner do IBM MQ em execução.

Utilize o seguinte comando:

```
oc exec -it qm1-ibm-mq-0 /bin/bash
```

em que `qm1-ibm-mq-0` é o Pod que implementamos no [“Criando o novo gerenciador de filas no Red Hat OpenShift”](#) na página 53. Se você tiver dado um nome diferente à implementação, customize esse valor.

2. Envie uma mensagem.

Execute os comandos a seguir:

```
cd /opt/mqm/samp/bin
export IBM MQSAMP_USER_ID=mqapp
export IBM MQSERVÉR=DEV.APP.SVRCONN/TCP/'localhost(1414) '
./amqsputc Q1 QM1
```

É solicitado que você forneça uma senha, em seguida, é possível enviar uma mensagem.

3. Verifique se a mensagem foi recebida com sucesso.

Execute a amostra GET:

```
./amqsgetc Q1 QM1
```

Resultados

Você concluiu o [“Migrando o IBM MQ para o IBM Cloud Pak for Integration”](#) na página 36.

Como proceder a seguir

Use as informações a seguir para ajudá-lo com cenários de migração mais complexos:

Migrando mensagens enfileiradas

Para migrar mensagens enfileiradas existentes, siga a orientação no tópico a seguir para exportar e importar mensagens depois que o novo gerenciador de filas estiver em vigor: [Usando o utilitário dmpmqmsg entre dois sistemas.](#)

Conectando-se a IBM MQ de fora do ambiente Red Hat OpenShift

O gerenciador de filas implementado pode ser exposto a clientes e gerenciadores de filas do IBM MQ fora do ambiente Red Hat OpenShift. O processo depende da versão de IBM MQ conectando-se ao ambiente Red Hat OpenShift. Consulte [“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift”](#) na página 108.

Instalando e desinstalando o IBM MQ Operator no Red Hat OpenShift

O IBM MQ Operator pode ser instalado no Red Hat OpenShift usando o Operator Hub.

Procedimento

- [“Dependências para o IBM MQ Operator”](#) na página 10.
- [“Permissões com escopo de cluster necessárias pelo IBM MQ Operator”](#) na página 10.
- [“Instalando o IBM MQ Operator usando o console da web Red Hat OpenShift”](#) na página 59.
- [“Instalando o IBM MQ Operator usando a CLI do Red Hat OpenShift”](#) na página 61.
- [“Instalando o IBM MQ Operator em um ambiente de isolamento físico”](#) na página 65.

Tarefas relacionadas

[“Desinstalando o IBM MQ Operator usando o console da web do Red Hat OpenShift”](#) na página 61

É possível usar o console da web do Red Hat OpenShift para desinstalar o IBM MQ Operator de Red Hat OpenShift.

[“Desinstalando o IBM MQ Operator usando a CLI do Red Hat OpenShift”](#) na página 63

É possível usar a CLI Red Hat OpenShift para desinstalar o IBM MQ Operator de Red Hat OpenShift. Há diferenças no processo de desinstalação, que dependem se o IBM MQ Operator está instalado em um único namespace ou se ele está instalado e disponível para todos os namespaces no cluster.

Instalando o IBM MQ Operator usando o console da web Red Hat OpenShift

O IBM MQ Operator pode ser instalado no Red Hat OpenShift usando o Operator Hub.

Antes de começar

Faça login no seu console da web do cluster Red Hat OpenShift.

Procedimento

1.  `EUS`

Opcional: Inclua os IBM Common Services Operators na lista de operadores instaláveis.

Nota:

Esta etapa se aplica às liberações do IBM MQ Operator 1.5 e anteriores. A etapa inclui um catálogo separado de Serviços Comuns. Para liberações mais recentes do Operator, os Serviços Comuns estão incluídos no catálogo da IBM.

- a) Clique no ícone de mais na parte superior direita da tela. É exibida a caixa de diálogo **Importar YAML**.
- b) Cole a definição de recurso a seguir na caixa de diálogo.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: opencloud-operators
  namespace: openshift-marketplace
spec:
  displayName: IBMCS Operators
```

```
publisher: IBM
sourceType: grpc
image: icr.io/cpopen/ibm-common-service-catalog:latest
updateStrategy:
  registryPoll:
    interval: 45m
```

- c) Clique em **Criar**.
2. Inclua os operadores IBM na lista de operadores instaláveis
 - a) Clique no ícone de mais na parte superior direita da tela. É exibida a caixa de diálogo **Importar YAML**.
 - b) Cole a definição de recurso a seguir na caixa de diálogo.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  image: icr.io/cpopen/ibm-operator-catalog:latest
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

- c) Clique em **Criar**.
3. Crie um namespace para usar no IBM MQ Operator

O IBM MQ Operator pode ser instalado com escopo em um único namespace ou em todos os namespaces. Esta etapa será necessária apenas para realizar a instalação em um determinado namespace que ainda não exista.

 - a) Na área de janela de navegação, clique em **Início > Projetos**.
A página de Projetos é exibida.
 - b) Clique em **Criar projeto**. Uma área Criar projeto é exibida.
 - c) Insira detalhes do namespace que você está criando. Por exemplo, é possível especificar "ibm-mq" como o nome.
 - d) Clique em **Criar**. O namespace do seu IBM MQ Operator é criado.
4. Instale o IBM MQ Operator.
 - a) Na área de janela de navegação, clique em **Operadores > OperatorHub**.
A página OperatorHub é exibida.
 - b) No campo **Todos os itens**, insira "IBM MQ".
A entrada do catálogo do IBM MQ é exibida.
 - c) Selecione **IBM MQ**.
A janela do IBM MQ é exibida.
 - d) Clique em **Instalar**.
Você vê a página Criar assinatura do Operator.
 - e) Revise [“Suporte de versão para o IBM MQ Operator”](#) na [página 7](#) para determinar qual canal do operador escolher.
 - f) Configure o Modo de Instalação para o namespace específico que você criou ou para o escopo do cluster.

A escolha do escopo de todo o cluster é recomendada, pois a instalação de diferentes versões de um Operador em diferentes namespaces pode levar a problemas. Os operadores são projetados para serem extensões do plano de controle.
 - g) Clique em **Subscrever**.
Você verá o IBM MQ na página Operadores instalados.

- h) Verifique o status do Operador na página Operadores instalados; ele mudará para Bem-sucedido quando a instalação estiver concluída.

Como proceder a seguir

“Preparando o seu projeto Red Hat OpenShift para IBM MQ usando o console da web do Red Hat OpenShift” na página 80

Desinstalando o IBM MQ Operator usando o console da web do Red Hat OpenShift

É possível usar o console da web do Red Hat OpenShift para desinstalar o IBM MQ Operator de Red Hat OpenShift.

Antes de começar

Faça login no console da web do seu cluster do Red Hat OpenShift.

Se o IBM MQ Operator for instalado em todos os projetos/namespaces no cluster, repita as etapas de 1 a 5 do procedimento a seguir em todos os projetos nos quais deseja excluir os gerenciadores de filas.

Procedimento

1. Selecione **Operadores > Operadores instalados**.
2. Na lista suspensa **Projeto**, selecione um projeto.
3. Clique no operador **IBM MQ**.
4. Clique na guia **Gerenciadores de filas** para visualizar os gerenciadores de filas gerenciados por este IBM MQ Operator.
5. Exclua um ou mais gerenciadores de filas.

Observe que, embora esses gerenciadores de filas continuem sendo executados, eles podem não funcionar como esperado sem um IBM MQ Operator.

6. Opcional: Se for apropriado, repita as etapas de 1 a 5 em todos os projetos em que deseja excluir os gerenciadores de filas.
7. Retorne para **Operadores > Operadores instalados**.
8. Ao lado do operador **IBM MQ**, clique no menu de três pontos e selecione **Desinstalar operador**.
9. Se você estiver usando o Red Hat OpenShift Container Platform 4.7, será necessário excluir manualmente o web hook validando a partir da linha de comandos:

```
oc delete validatingwebhookconfiguration namespace.validator.queuemanagers.mq.ibm.com
```

Instalando o IBM MQ Operator usando a CLI do Red Hat OpenShift

O IBM MQ Operator pode ser instalado no Red Hat OpenShift usando o Operator Hub.

Antes de começar

Faça login na interface da linha de comandos do Red Hat OpenShift (CLI) usando **oc login**. Para estas etapas, será necessário que você seja um administrador de cluster.

Procedimento

1.  Opcional: Crie um **CatalogSource** para o IBM Common Services Operators.

Nota:

Esta etapa se aplica às liberações do IBM MQ Operator 1.5 e anteriores. A etapa inclui um catálogo separado de Serviços Comuns. Para liberações mais recentes do Operator, os Serviços Comuns estão incluídos no catálogo da IBM.

- a) Crie um arquivo YAML definindo o recurso **CatalogSource**.

Crie um arquivo chamado "operator-source-cs.yaml" com o conteúdo a seguir:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: opencloud-operators
  namespace: openshift-marketplace
spec:
  displayName: IBMCS Operators
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-common-service-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

- b) Aplique o **CatalogSource** no servidor.

```
oc apply -f operator-source-cs.yaml -n openshift-marketplace
```

2. Criar um **CatalogSource** para o IBM Operators

- a) Crie um arquivo YAML definindo o recurso **CatalogSource**

Crie um arquivo chamado "operator-source-ibm.yaml" com o conteúdo a seguir:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  image: icr.io/cpopen/ibm-operator-catalog:latest
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

- b) Aplique o **CatalogSource** no servidor.

```
oc apply -f operator-source-ibm.yaml -n openshift-marketplace
```

3. Crie um namespace para usar no IBM MQ Operator

O IBM MQ Operator pode ser instalado com escopo em um único namespace ou em todos os namespaces. Esta etapa será necessária apenas para realizar a instalação em um determinado namespace que ainda não exista.

```
oc new-project ibm-mq
```

4. Visualize a lista de Operadores disponíveis para o cluster por meio do OperatorHub

```
oc get packagemanifests -n openshift-marketplace
```

5. Inspecionar o IBM MQ Operator para verificar os InstallModes suportados e Canais disponíveis

```
oc describe packagemanifests ibm-mq -n openshift-marketplace
```

6. Criar um arquivo YAML de objeto **OperatorGroup**

Um **OperatorGroup** é um recurso OLM que seleciona namespaces de destino nos quais gerar o acesso RBAC necessário para todos os Operadores no mesmo namespace do **OperatorGroup**.

O namespace para o qual você assina o Operador deve ter um **OperatorGroup** que corresponda ao **InstallMode** do Operador, seja o modo AllNamespaces ou SingleNamespace. Se o Operador

que você pretende instalar usar AllNamespaces, o namespace openshift-operators já terá um **OperatorGroup** apropriado.

No entanto, se o Operador utilizar o modo SingleNamespace e você ainda não tiver um **OperatorGroup** apropriado, será necessário criar um.

- a) Crie um arquivo chamado "mq-operator-group.yaml" com o conteúdo a seguir:

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <operatorgroup_name>
  namespace: <namespace_name>
spec:
  targetNamespaces:
  - <namespace_name>
```

- b) Criar o objeto **OperatorGroup**

```
oc apply -f mq-operator-group.yaml
```

7. Crie um arquivo YAML de objeto **Subscription** para assinar um namespace para o IBM MQ Operator

- a) Revise “Suporte de versão para o IBM MQ Operator” na [página 7](#) para determinar qual canal do operador escolher.
- b) Crie um arquivo chamado "mq-sub.yaml" com o conteúdo a seguir, mas alterando **channel** para corresponder ao canal da versão do IBM MQ Operator que você deseja instalar.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-mq
  namespace: openshift-operators
spec:
  channel: <ibm-mq-operator-channel>
  name: ibm-mq
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
```

Para uso do **InstallMode** AllNamespaces, especifique **openshift-operators** no namespace. Caso contrário, especifique o namespace único relevante para o uso SingleNamespace **InstallMode**. Note que é necessário alterar apenas o campo **namespace**, deixando o campo **sourceNamespace** como está.

- c) Criar o objeto **Subscription**

```
oc apply -f mq-sub.yaml
```

8. Verificar o status do Operador

Depois que a instalação do Operador for bem-sucedida, o status do pod será exibido como *Em execução*. Para o uso do **InstallMode** AllNamespaces, especifique **openshift-operators** como o namespace. Caso contrário, especifique o namespace único relevante para o uso SingleNamespace **InstallMode**.

```
oc get pods -n <namespace_name>
```

Como proceder a seguir

“Preparando o seu projeto Red Hat OpenShift para IBM MQ usando a CLI do Red Hat OpenShift” na [página 80](#)

Desinstalando o IBM MQ Operator usando a CLI do Red Hat OpenShift

É possível usar a CLI Red Hat OpenShift para desinstalar o IBM MQ Operator de Red Hat OpenShift. Há diferenças no processo de desinstalação, que dependem se o IBM MQ Operator está instalado em um único namespace ou se ele está instalado e disponível para todos os namespaces no cluster.

Antes de começar

Efetue login no cluster do Red Hat OpenShift usando o `login`.

Procedimento

- Se o IBM MQ Operator for instalado em um único namespace, conclua as subetapas a seguir:

- a) Assegure-se de estar no projeto correto:

```
oc project <project_name>
```

- b) Visualize os gerenciadores de filas instalados no projeto:

```
oc get qmgr
```

- c) Exclua um ou mais gerenciadores de filas:

```
oc delete qmgr <qmgr_name>
```

Observe que, embora esses gerenciadores de filas continuem sendo executados, eles podem não funcionar como esperado sem um IBM MQ Operator.

- d) Visualize as instâncias **ClusterServiceVersion**:

```
oc get csv
```

- e) Exclua o IBM MQ **ClusterServiceVersion**:

```
oc delete csv <ibm_mq_csv_name>
```

- f) Visualize as assinaturas:

```
oc get subscription
```

- g) Exclua todas as assinaturas:

```
oc delete subscription <ibm_mq_subscription_name>
```

- h) Opcional: Se nada mais estiver usando serviços comuns, talvez você deseje desinstalar o operador de serviços comuns e excluir o grupo do operador:

- a. Desinstale o operador de serviços comuns seguindo as instruções em [Desinstalando serviços comuns](#) na documentação do produto IBM Cloud Pak foundational services.

- b. Visualize o grupo do operador:

```
oc get operatorgroup
```

- c. Exclua o grupo do operador:

```
oc delete OperatorGroup <operator_group_name>
```

- Se o IBM MQ Operator estiver instalado e disponível para todos os namespaces no cluster, conclua as subetapas a seguir:

- a) Visualize todos os gerenciadores de filas instalados:

```
oc get qmgr -A
```

- b) Exclua um ou mais gerenciadores de filas:

```
oc delete qmgr <qmgr_name> -n <namespace_name>
```

Observe que, embora esses gerenciadores de filas continuem sendo executados, eles podem não funcionar como esperado sem um IBM MQ Operator.

- c) Visualize as instâncias **ClusterServiceVersion**:

```
oc get csv -A
```

d) Exclua o IBM MQ **ClusterServiceVersion** do cluster:

```
oc delete csv <ibm_mq_csv_name> -n openshift-operators
```

e) Visualize as assinaturas:

```
oc get subscription -n openshift-operators
```

f) Exclua as assinaturas:

```
oc delete subscription <ibm_mq_subscription_name> -n openshift-operators
```

g) Se você estiver usando Red Hat OpenShift Container Platform 4.7, pode ser necessário excluir manualmente o web hook validada:

```
oc delete validatingwebhookconfiguration namespace.validator.queuemanagers.mq.ibm.com
```

h) Opcional: Se nada mais estiver usando serviços comuns, talvez você queira desinstalar o operador de serviços comuns:

Siga as instruções em [Desinstalando serviços comuns](#) na documentação do produto IBM Cloud Pak foundational services.

Instalando o IBM MQ Operator em um ambiente de isolamento físico

Este tutorial orienta você na instalação do IBM MQ Operator em um cluster do Red Hat OpenShift que não possui conectividade de internet. É possível instalar o IBM MQ Operator em um ambiente de isolamento físico usando um dispositivo de armazenamento móvel ou usando uma máquina bastion.

Instalando o IBM MQ Operator em um ambiente de isolamento físico usando um dispositivo de armazenamento móvel

Para obter as etapas para concluir a instalação, consulte [Espelhando imagens com um dispositivo de armazenamento móvel](#) na documentação do IBM Cloud Pak for Integration. Se você estiver instalando apenas IBM MQ, substitua todas as ocorrências das variáveis de ambiente a seguir pelos valores fornecidos aqui:

```
export CASE_NAME=ibm-mq
export CASE_ARCHIVE_VERSION=version_number
export CASE_INVENTORY_SETUP=ibmMQOperator
```

em que *version_number* é a versão do caso que você deseja usar para fazer a instalação do isolamento físico. Para obter uma lista de versões de casos disponíveis, consulte <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>. Revise “Suporte de versão para o IBM MQ Operator” na página 7 para determinar qual canal do operador escolher.

Instalando o IBM MQ Operator em um ambiente de isolamento físico usando uma máquina bastion

1. [“Pré-requisitos”](#) na página 66
2. [“Preparar um registro do Docker”](#) na página 66
3. [“Prepare um host bastion”](#) na página 67
4. [“Crie variáveis de ambiente para o instalador e inventário de imagem”](#) na página 68
5. [“Faça download do instalador do IBM MQ e do inventário de imagens”](#) na página 68
6. [“Efetue login no cluster do Red Hat OpenShift Container Platform como um administrador de cluster”](#) na página 68

7. [“Crie um namespace do Kubernetes para o IBM MQ Operator” na página 68](#)
8. [“Espelhe as imagens e configure o cluster” na página 68](#)
9. [“Instale o IBM MQ Operator.” na página 70](#)
10. [“Implementar o IBM MQ Queue Manager” na página 71](#)

Pré-requisitos

1. Um cluster do Red Hat OpenShift Container Platform deve ser instalado. Para as versões do Red Hat OpenShift Container Platform suportadas, consulte [“Suporte de versão para o IBM MQ Operator” na página 7](#).
2. Um registro do Docker deve estar disponível. Para obter mais informações, consulte [“Preparar um registro do Docker” na página 66](#).
3. Um servidor de bastion deve ser configurado. Para obter informações adicionais, consulte [“Prepare um host bastion” na página 67](#).

Preparar um registro do Docker

Um registro do Docker local é usado para armazenar todas as imagens em seu ambiente local. Deve-se criar tal registro e deve-se assegurar que ele atenda aos seguintes requisitos:

- Suporta o [Docker Manifest V2, Esquema 2](#).
- Suporta imagens de multiarquitetura.
- É acessível tanto do servidor de base quanto de seus nós do cluster do Red Hat OpenShift Container Platform.
- Possui o nome de usuário e a senha de um usuário que pode gravar no registro de destino a partir do host bastion.
- Tem o nome de usuário e a senha de um usuário que pode ler a partir do registro de destino que está nos nós do cluster do Red Hat OpenShift.
- Permite separadores de caminho no nome da imagem.

Depois de criar o registro do Docker, deve-se configurar o registro:

1. Criar namespaces do registro

- `ibmcom` - Espaço de nomes para armazenar todas as imagens a partir do namespace `dockerhub.io/ibmcom`.

O namespace `ibmcom` destina-se a todas as imagens do IBM disponíveis publicamente e que não requerem credenciais para a extração.

- `cp` - Namespace no qual armazenar as imagens do IBM do repositório `cp.icr.io/cp`.

O namespace `cp` destina-se às imagens no IBM Entitled Registry que requerem uma chave de autorização de produto e credenciais para a extração. Para obter sua chave de autorização, efetue login no MyIBM Container Software Library com o IBMid e a senha que estão associados ao software autorizado. Na seção **Chaves de autorização**, selecione **Copiar chave** para copiar a chave de autorização para a área de transferência, em seguida, salve-a para uso nas etapas a seguir.

- `opencloudio` - Espaço de nomes para armazenar as imagens a partir de `quay.io/opencloudio`.

O namespace `opencloudio` é para selecionar imagens do componente de software livre do IBM que estão disponíveis no [quay.io](#). As imagens do IBM Cloud Pak foundational services são hospedadas em `opencloudio`.

2. Verifique se cada um dos namespaces atende aos requisitos a seguir:

- Suporta criação de repositório automático.
- Possui credenciais de um usuário que pode gravar e criar repositórios. O host bastion usa essas credenciais.

- Possui credenciais de um usuário que pode ler todos os repositórios. O cluster do Red Hat OpenShift Container Platform usa essas credenciais.

Prepare um host bastion

Prepare um host bastion que possa acessar o cluster Red Hat OpenShift Container Platform, o registro do Docker local e a Internet. O host bastion deve estar em uma plataforma Linux for x86-64 com qualquer sistema operacional que a CLI do IBM Cloud Pak e o suporte da CLI do Red Hat OpenShift Container Platform suporta.

Conclua estas etapas em seu nó bastion:

1. Instale o OpenSSL versão 1.11.1 ou mais recente.
2. Instale o Docker ou o Podman no nó bastion.

- Para instalar o Docker, execute estes comandos:

```
yum check-update
yum install docker
```

- Para instalar o Podman, consulte [Instruções de instalação do Podman](#)

3. Instale o skopeo versão 1.x.x no nó bastion. Para instalar o skopeo, execute estes comandos:

```
yum check-update
yum install skopeo
```

4. Instale a CLI do IBM Cloud Pak. Instale a versão mais recente do arquivo binário para sua plataforma. Para obter mais informações, consulte [cloud-pak-cli](#).

- a. Faça o download do arquivo binário.

```
wget https://github.com/IBM/cloud-pak-cli/releases/download/vversion-number/binary-file-name
```

Por exemplo:

```
wget https://github.com/IBM/cloud-pak-cli/releases/latest/download/cloudctl-linux-amd64.tar.gz
```

- b. Extraia o arquivo binário.

```
tar -xf binary-file-name
```

- c. Execute os comandos a seguir para modificar e mover o arquivo

```
chmod 755 file-name
mv file-name /usr/local/bin/cloudctl
```

- d. Confirme se `cloudctl` está instalado:

```
cloudctl --help
```

5. Instale a ferramenta oc Red Hat OpenShift Container Platform da CLI.

Para obter mais informações, consulte [Ferramentas da CLI do Red Hat OpenShift Container Platform](#)

6. Crie um diretório para servir como armazenamento off-line.

A seguir, está um diretório de exemplo. Este exemplo é usado nas etapas subsequentes.

```
mkdir $HOME/offline
```

Nota: esse armazenamento off-line deve ser persistente para evitar a transferência de dados mais de uma vez. A persistência também ajuda a executar o processo de espelhamento várias vezes ou de acordo com um planejamento.

Crie variáveis de ambiente para o instalador e inventário de imagem

Crie as seguintes variáveis de ambiente com o nome de imagem do instalador e o inventário de imagem:

```
export CASE_ARCHIVE_VERSION=version_number
export CASE_ARCHIVE=ibm-mq-$CASE_ARCHIVE_VERSION.tgz
export CASE_INVENTORY=ibmMQOperator
```

em que *version_number* é a versão do caso que você deseja usar para fazer a instalação do isolamento físico. Para obter uma lista de versões de casos disponíveis, consulte <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>. Revise [Suporte de versão para o IBM MQ Operator](#) para determinar qual canal do operador escolher.

Faça download do instalador do IBM MQ e do inventário de imagens

Faça o download do instalador do `ibm-mq` e do inventário de imagem para o host de bastidor:

```
cloudctl case save \  
  --case https://github.com/IBM/cloud-pak/raw/master/repo/case/ibm-mq/$CASE_ARCHIVE_VERSION/  
  $CASE_ARCHIVE \  
  --outputdir $HOME/offline/
```

Efetue login no cluster do Red Hat OpenShift Container Platform como um administrador de cluster

A seguir, está um comando de exemplo para efetuar login no cluster Red Hat OpenShift Container Platform:

```
oc login cluster_host:port --username=cluster_admin_user --password=cluster_admin_password
```

Crie um namespace do Kubernetes para o IBM MQ Operator

Crie uma variável de ambiente com um namespace para instalar o IBM MQ Operator, em seguida, crie o namespace:

```
export NAMESPACE=ibm-mq-test
oc create namespace }${NAMESPACE}
```

Espelhe as imagens e configure o cluster

Conclua estas etapas para espelhar as imagens e configurar seu cluster:

Nota: Não use o til dentro das aspas duplas em nenhum comando. Por exemplo, não use `args "--registry registry --user registry_userid --pass registry_password --inputDir ~/offline"`. O til não se expande e seus comandos podem falhar.

1. Armazene credenciais de autenticação para todos os registros do Docker de origem.

Todos os IBM Cloud Platform Common Services, a imagem do IBM MQ Operator e a imagem do IBM MQ Advanced Developer são armazenados em registros públicos que não requerem autenticação. No entanto, o IBM MQ Advanced Server (não desenvolvedor), outros produtos e componentes de terceiros requerem um ou mais registros autenticados. Os seguintes registros requerem autenticação:

- `cp.icr.io`
- `registry.redhat.io`
- `registry.access.redhat.com`

Para obter mais informações sobre esses registros, consulte [Crie espaços de nomes de registro](#).

Você deve executar o comando a seguir para configurar credenciais para todos os registros que requerem autenticação. Execute o comando separadamente para cada registro desse tipo:

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-creds-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry registry --user registry_userid --pass registry_password --inputDir $HOME/  
offline"
```

O comando armazena e cachê as credenciais de registro em um arquivo em seu sistema de arquivos no local `$HOME/.airgap/secrets`.

2. Crie variáveis de ambiente com as informações de conexão de registro do Docker local.

```
export LOCAL_DOCKER_REGISTRY=IP_or_FQDN_of_local_docker_registry  
export LOCAL_DOCKER_USER=username  
export LOCAL_DOCKER_PASSWORD=password
```

Nota: o registro do Docker usa portas padrão, como 80 ou 443. Se o seu registro do Docker usar uma porta não padrão, especifique-a usando a sintaxe `host:port`. Por exemplo:

```
export LOCAL_DOCKER_REGISTRY=myregistry.local:5000
```

3. Configure um segredo de autenticação para o registro do Docker local.

Nota: esta etapa precisa ser executada apenas uma vez.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-creds-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $  
{LOCAL_DOCKER_PASSWORD}"
```

O comando armazena e cachê as credenciais de registro em um arquivo em seu sistema de arquivos no local `$HOME/.airgap/secrets`.

4. Configure um segredo de extração de imagem global e **ImageContentSourcePolicy**.

- a. Verifique se um reinício de nó é necessário.

- Na Red Hat OpenShift Container Platform versão 4.4 e acima, e em uma nova instalação do IBM MQ Operator usando isolamento físico, esta etapa reinicia todos os nós do cluster. Os recursos de cluster podem estar indisponíveis até o momento em que o novo segredo de extração for aplicado.
- No IBM MQ Operator 1.8, o CASE é atualizado para incluir uma origem de espelhamento adicional para imagens.. Portanto, quando você faz upgrade de versões anteriores do IBM MQ Operator para a versão 1.8 ou superior, uma reinicialização do nó é acionada
- Para verificar se esta etapa precisa de um restart de nó, adicione a opção `--dry-run` ao código para esta etapa. Isso gera o mais recente **ImageContentSourcePolicy** e o exibe na janela do console (**stdout**). Se este **ImageContentSourcePolicy** difere do cluster configurado **ImageContentSourcePolicy**, ocorre um reinício.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-cluster-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $  
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline --dryRun"
```

- b. Para configurar o segredo de pull da imagem global e **ImageContentSourcePolicy**, execute o código para esta etapa sem a opção `--dry-run` :

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-cluster-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $  
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

5. Verifique se o recurso **ImageContentSourcePolicy** foi criado.

```
oc get imageContentSourcePolicy
```

6. Opcional: se você estiver usando um registro não seguro, o registro local deverá ser incluído na lista **insecureRegistries** do cluster.

```
oc patch image.config.openshift.io/cluster --type=merge -p '{"spec":{"registrySources":  
{"insecureRegistries":["${LOCAL_DOCKER_REGISTRY}"]}}'
```

7. Verifique o status do nó do cluster.

```
oc get nodes
```

Após o segredo de extração de imagem **imageContentsourcePolicy** e global estarem aplicados, pode-se consultar o status do nó como **Ready**, **Scheduling** ou **Disabled**. Aguarde até que todos os nós mostrem um status **Ready** .

8. Espelhe as imagens para o registro local.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action mirror-images \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $  
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

Instale o IBM MQ Operator.

1. Faça login no seu console da web do cluster Red Hat OpenShift.
2. Crie uma origem de catálogo. Use o mesmo terminal que executou as etapas anteriores.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action install-catalog \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --recursive"
```

3. Verifique se o **CatalogSource** foi criado para o Common Services Installer Operator.

```
oc get pods -n openshift-marketplace  
oc get catalogsource -n openshift-marketplace
```

4. Instale o IBM MQ Operator usando o OLM.

- a. Na área de janela de navegação, clique em **Operadores > OperatorHub**.

A página **OperatorHub** é exibida.

- b. No campo **Todos os itens**, insira IBM MQ.

A entrada do catálogo IBM MQ é exibida.

- c. Selecione **IBM MQ**.

A janela do **IBM MQ** é exibida.

- d. Clique em **Instalar**.

A página **Criar assinatura do Operator** é exibida.

- e. Revise “Suporte de versão para o IBM MQ Operator” na página 7 para determinar qual canal do operador escolher.
- f. Configure **Modo de instalação** para o namespace específico que você criou ou o escopo de abrangência do cluster.
- g. Clique em **Subscrever**.
O **IBM MQ** é incluído na página **Operadores instalados**.
- h. Verifique o status do operador na página **Operadores instalados**. O status muda para **Succeeded** quando a instalação é concluída.

Implementar o IBM MQ Queue Manager

Para criar um novo gerenciador de filas no operador instalado, consulte [“Implementando e configurando gerenciadores de filas usando o IBM MQ Operator”](#) na página 79.

Tarefas relacionadas

[“Preparando-se para fazer upgrade do IBM MQ Operator ou do gerenciador de filas em um ambiente de isolamento físico”](#) na página 71

Em um cluster do Red Hat OpenShift que não tem conectividade com a Internet, há etapas preparatórias que você precisa executar antes de fazer upgrade do IBM MQ Operator

Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas

Fazer upgrade do IBM MQ Operator permitirá que você faça o upgrade dos seus gerenciadores de filas.

Procedimento

- [“Fazendo upgrade do IBM MQ Operator usando o console da web do Red Hat OpenShift”](#) na página 74.
- [“Fazendo upgrade do IBM MQ Operator usando a CLI do Red Hat OpenShift”](#) na página 76.
- [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando o console da web do Red Hat OpenShift”](#) na página 77.
- [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando a CLI do Red Hat OpenShift”](#) na página 78.

Preparando-se para fazer upgrade do IBM MQ Operator ou do gerenciador de filas em um ambiente de isolamento físico

Em um cluster do Red Hat OpenShift que não tem conectividade com a Internet, há etapas preparatórias que você precisa executar antes de fazer upgrade do IBM MQ Operator

Antes de começar

Este tópico supõe que você já tenha configurado um registro de imagem local no qual as imagens anteriores do IBM Cloud Pak for Integration liberadas são espelhadas

Sobre esta tarefa

Antes de fazer o upgrade do IBM MQ Operator ou do gerenciador de filas em um ambiente de airgap, você deve espelhar as imagens mais recentes IBM Cloud Pak for Integration .

Observe que as quatro primeiras etapas nesta tarefa são as mesmas que as etapas executadas quando [“Instalando o IBM MQ Operator em um ambiente de isolamento físico”](#) na página 65.

Procedimento

1. Crie variáveis de ambiente para o instalador e inventário de imagem.

Crie as seguintes variáveis de ambiente com o nome de imagem do instalador e o inventário de imagem:

```
export CASE_ARCHIVE_VERSION=version_number
export CASE_ARCHIVE=ibm-mq-CASE_ARCHIVE_VERSION.tgz
export CASE_INVENTORY=ibmMQOperator
```

em que *version_number* é a versão do caso que você deseja usar para fazer a instalação do isolamento físico. Para obter uma lista de versões de casos disponíveis, consulte <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>. Revise [Suporte de versão para o IBM MQ Operator](#) para determinar qual canal do operador escolher.

2. Faça o download do instalador do IBM MQ e do inventário de imagem

Faça o download do instalador do `ibm-mq` e do inventário de imagem para o host de bastidor:

```
cloudctl case save \  
  --case https://github.com/IBM/cloud-pak/raw/master/repo/case/ibm-mq/  
  CASE_ARCHIVE_VERSION/CASE_ARCHIVE \  
  --outputdir $HOME/offline/
```

3. Faça login no cluster do Red Hat OpenShift Container Platform como um administrador de cluster.

A seguir, está um comando de exemplo para efetuar login no cluster Red Hat OpenShift Container Platform:

```
oc login cluster_host:port --username=cluster_admin_user --password=cluster_admin_password
```

4. Espelhe as imagens e configure o cluster.

Conclua estas etapas para espelhar as imagens e configurar seu cluster:

Nota: Não use o til dentro das aspas duplas em nenhum comando. Por exemplo, não use `args "--registry registry --user registry_userid --pass registry_password --inputDir ~/offline"`. O til não se expande e seus comandos podem falhar.

- a. Armazene credenciais de autenticação para todos os registros do Docker de origem.

Todos os IBM Cloud Platform Common Services, a imagem do IBM MQ Operator e a imagem do IBM MQ Advanced Developer são armazenados em registros públicos que não requerem autenticação. No entanto, o IBM MQ Advanced Server (não desenvolvedor), outros produtos e componentes de terceiros requerem um ou mais registros autenticados. Os seguintes registros requerem autenticação:

- `cp.icr.io`
- `registry.redhat.io`
- `registry.access.redhat.com`

Para obter mais informações sobre esses registros, consulte [Crie espaços de nomes de registro](#).

Você deve executar o comando a seguir para configurar credenciais para todos os registros que requerem autenticação. Execute o comando separadamente para cada registro desse tipo:

```
cloudctl case launch \  
  --case $HOME/offline/CASE_ARCHIVE \  
  --inventory CASE_INVENTORY \  
  --action configure-creds-airgap \  
  --namespace NAMESPACE \  
  --args "--registry registry --user registry_userid --pass registry_password --inputDir  
  $HOME/offline"
```

O comando armazena e cachê as credenciais de registro em um arquivo em seu sistema de arquivos no local `$HOME/.airgap/secrets`.

b. Crie variáveis de ambiente com as informações de conexão de registro do Docker local.

```
export LOCAL_DOCKER_REGISTRY=IP_or_FQDN_of_local_docker_registry
export LOCAL_DOCKER_USER=username
export LOCAL_DOCKER_PASSWORD=password
```

Nota: o registro do Docker usa portas padrão, como 80 ou 443. Se o seu registro do Docker usar uma porta não padrão, especifique-a usando a sintaxe `host:port`. Por exemplo:

```
export LOCAL_DOCKER_REGISTRY=myregistry.local:5000
```

c. Configure um segredo de autenticação para o registro do Docker local.

Nota: esta etapa precisa ser executada apenas uma vez.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-creds-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass ${LOCAL_DOCKER_PASSWORD}"
```

O comando armazena e cachê as credenciais de registro em um arquivo em seu sistema de arquivos no local `$HOME/.airgap/secrets`.

d. Configure um segredo de extração de imagem global e **ImageContentSourcePolicy**.

i) Verifique se um reinício de nó é necessário.

- Na Red Hat OpenShift Container Platform versão 4.4 e acima, e em uma nova instalação do IBM MQ Operator usando isolamento físico, esta etapa reinicia todos os nós do cluster. Os recursos de cluster podem estar indisponíveis até o momento em que o novo segredo de extração for aplicado.
- No IBM MQ Operator 1.8, o CASE é atualizado para incluir uma origem de espelhamento adicional para imagens.. Portanto, quando você faz upgrade de versões anteriores do IBM MQ Operator para a versão 1.8 ou superior, uma reinicialização do nó é acionada
- Para verificar se esta etapa precisa de um restart de nó, adicione a opção `--dry-run` ao código para esta etapa. Isso gera o mais recente **ImageContentSourcePolicy** e o exibe na janela do console (**stdout**). Se este **ImageContentSourcePolicy** difere do cluster configurado **ImageContentSourcePolicy**, ocorre um reinício.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-cluster-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass ${LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline --dryRun"
```

ii) Para configurar o segredo de pull da imagem global e **ImageContentSourcePolicy**, execute o código para esta etapa sem a opção `--dry-run`:

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-cluster-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass ${LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

e. Verifique se o recurso **ImageContentSourcePolicy** foi criado.

```
oc get imageContentSourcePolicy
```

- f. Opcional: se você estiver usando um registro não seguro, o registro local deverá ser incluído na lista **insecureRegistries** do cluster.

```
oc patch image.config.openshift.io/cluster --type=merge -p '{"spec":{"registrySources":{"insecureRegistries":["${LOCAL_DOCKER_REGISTRY}"]}}}'
```

- g. Verifique o status do nó do cluster.

```
oc get nodes
```

Após o segredo de extração de imagem **imageContentsourcePolicy** e global estarem aplicados, pode-se consultar o status do nó como **Ready**, **Scheduling** ou **Disabled**. Aguarde até que todos os nós mostrem um status **Ready**.

- h. Espelhe as imagens para o registro local.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action mirror-images \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $\  
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

5. Faça upgrade da origem do catálogos

Use o mesmo terminal que executou as etapas anteriores.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action install-catalog \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --recursive"
```

Como proceder a seguir

Agora você está pronto para fazer upgrade do IBM MQ Operator e do gerenciador de filas, concluindo uma das tarefas a seguir:

- [“Fazendo upgrade do IBM MQ Operator usando o console da web do Red Hat OpenShift” na página 74](#)
- [“Fazendo upgrade do IBM MQ Operator usando a CLI do Red Hat OpenShift” na página 76](#)
- [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando o console da web do Red Hat OpenShift” na página 77](#)
- [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando a CLI do Red Hat OpenShift” na página 78](#)
- [“Fazendo upgrade de um gerenciador de filas do IBM MQ no Red Hat OpenShift usando o Platform Navigator” na página 79](#)

Fazendo upgrade do IBM MQ Operator usando o console da web do Red Hat OpenShift

Pode ser feito upgrade do IBM MQ Operator usando o Operator Hub.

Antes de começar

Faça login no seu console da web do cluster Red Hat OpenShift.

Antes de fazer o upgrade do IBM MQ Operator em um ambiente de airgap, você deve espelhar as últimas imagens IBM Cloud Pak for Integration. Veja [Preparando-se para atualizar o IBM MQ Operator ou gerenciador de filas em um ambiente de airgap](#).

Procedimento

1. Revise [“Suporte de versão para o IBM MQ Operator”](#) na página 7 para determinar para qual canal do operador fazer upgrade.
2. Opcional: Se você estiver fazendo o upgrade de uma versão do IBM MQ Operator anterior à 1.5 para o IBM MQ Operator 1.5 ou posterior, é necessário primeiramente atualizar a versão do IBM Cloud Pak foundational services.

Para obter informações adicionais, consulte [“Fazendo upgrade do IBM Cloud Pak foundational services usando o console da web do Red Hat OpenShift”](#) na página 75.

3. Atualize o IBM MQ Operator. Novas versões principais ou secundárias do IBM MQ Operator são entregues por meio de novos Canais de assinatura. Para fazer upgrade do seu Operador para uma nova versão principal ou secundária, será necessário atualizar o canal selecionado em sua Assinatura do IBM MQ Operator.
 - a) Na área de janela de navegação, clique em **Operadores > Operadores instalados**.
Todos os Operadores instalados no projeto especificado serão exibidos.
 - b) Selecione o **IBM MQ Operator**
 - c) Navegue até a guia **Assinatura**
 - d) Clique no **Canal**
A janela **Mudar canal de atualização de assinatura** é exibida.
 - e) Selecione o canal desejado e clique em **Salvar**.
O upgrade do operador será feito para a versão mais recente disponível para o novo canal. Consulte o [“Suporte de versão para o IBM MQ Operator”](#) na página 7.

Como proceder a seguir

Se você tiver feito upgrade para o IBM Cloud Pak foundational services 3.7, será necessário atualizar ou reiniciar quaisquer gerenciadores de filas que usem uma licença do IBM Cloud Pak for Integration. Para mais informações sobre como fazer isso, consulte [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando o console da web do Red Hat OpenShift”](#) na página 77.

Fazendo upgrade do IBM Cloud Pak foundational services usando o console da web do Red Hat OpenShift

Se você estiver fazendo o upgrade de uma versão do IBM MQ Operator anterior à 1.5 para o IBM MQ Operator 1.5 ou posterior, é necessário primeiramente atualizar a versão do IBM Cloud Pak foundational services.

Antes de começar

Nota: É necessário concluir esta tarefa apenas se você estiver fazendo o upgrade de uma versão do IBM MQ Operator anterior a 1.5 para o IBM MQ Operator 1.5 ou posterior.

 Se você tiver algum gerenciador de filas que use uma licença do IBM Cloud Pak for Integration, após este upgrade, será necessário uma reinicialização do gerenciador de filas para acessar o console da web e você também verá outros erros ao efetuar login no console da web. É possível corrigir esses erros, fazendo o upgrade para o valor mais recente de `.spec.version` na sua versão escolhida de IBM MQ, após a atualização do operador ser concluída.

 Se você tiver os gerenciadores de filas existentes e usar o IBM Cloud Pak for Integration Operations Dashboard, veja [“Implementando ou fazendo upgrade da integração do IBM MQ 9.2.2 ou 9.2.3 com Operations Dashboard no IBM Cloud Pak for Integration 2021.4”](#) na página 111 antes de fazer upgrade.

Procedimento

1. Faça login no seu console da web do cluster Red Hat OpenShift.

2. Na área de janela de navegação, clique em **Operadores > Operadores instalados**.
Todos os Operadores instalados no projeto especificado serão exibidos.
3. Selecione o **IBM Cloud Pak foundational services Operator**. Observe que antes da versão 3.7, ele era chamado de **IBM Common Services Operator**
4. Navegue até a guia **Assinatura**.
5. Clique em **Canal**.
A janela **Mudar canal de atualização de assinatura** é exibida.
6. Selecione o canal **v3** e clique em **Salvar**.
O operador IBM Cloud Pak foundational services faz upgrade para a versão mais recente disponível do novo canal. Consulte [“Suporte de versão para o IBM MQ Operator”](#) na página 7.

Como proceder a seguir

Você agora está pronto para [Fazer upgrade do IBM MQ Operator](#).

Fazendo upgrade do IBM MQ Operator usando a CLI do Red Hat OpenShift

O IBM MQ Operator pode ser atualizado a partir da linha de comandos.

Antes de começar

Efetue login em seu cluster usando **cloudctl login** (para o IBM Cloud Pak for Integration) ou **oc login**.

Antes de fazer o upgrade do IBM MQ Operator em um ambiente de airgap, você deve espelhar as últimas imagens IBM Cloud Pak for Integration . Veja [Preparando-se para atualizar o IBM MQ Operator ou gerenciador de filas em um ambiente de airgap](#).

Procedimento

1. Revise [“Suporte de versão para o IBM MQ Operator”](#) na página 7 para determinar para qual canal do operador fazer upgrade.
2. Opcional: Se você estiver fazendo o upgrade de uma versão do IBM MQ Operator anterior à 1.5 para o IBM MQ Operator 1.5 ou posterior, é necessário primeiramente atualizar a versão do IBM Cloud Pak foundational services.

Para obter informações adicionais, consulte [“Fazendo upgrade do IBM Cloud Pak foundational services usando a CLI Red Hat OpenShift”](#) na página 77.

3. Atualize o IBM MQ Operator. Novas versões principais/secundárias do IBM MQ Operator são entregues por meio de novos Canais de Assinatura. Para fazer upgrade do seu Operador para uma nova versão principal/secundária, será necessário atualizar o canal selecionado na sua Assinatura do IBM MQ Operator.
 - a) Certifique-se de que o Canal de upgrade IBM MQ Operator necessário esteja disponível.

```
oc get packagemanifest ibm-mq -o=jsonpath='{.status.channels[*].name}'
```

- b) Corrija o Subscription para mover para o canal de atualização desejado (em que vX.Y é o canal de atualização desejado identificado na etapa anterior).

```
oc patch subscription ibm-mq --patch '{"spec":{"channel":"vX.Y"}}' --type=merge
```

Como proceder a seguir

Se você tiver feito upgrade para o IBM Cloud Pak foundational services 3.7, será necessário atualizar ou reiniciar quaisquer gerenciadores de filas que usem uma licença do IBM Cloud Pak for Integration. Para

mais informações sobre como fazer isso, consulte [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando a CLI do Red Hat OpenShift”](#) na página 78.

Fazendo upgrade do IBM Cloud Pak foundational services usando a CLI Red Hat OpenShift

Se você estiver fazendo o upgrade de uma versão do IBM MQ Operator anterior à 1.5 para o IBM MQ Operator 1.5 ou posterior, é necessário primeiramente atualizar a versão do IBM Cloud Pak foundational services.

Antes de começar

Nota: É necessário concluir esta tarefa apenas se você estiver fazendo o upgrade de uma versão do IBM MQ Operator anterior a 1.5 para o IBM MQ Operator 1.5 ou posterior.

 Se você tiver gerenciadores de filas que usem uma licença do IBM Cloud Pak for Integration, será necessário reiniciá-los após este upgrade para que seja possível acessar o console da web. Neste caso, ao efetuar login no console da web, você também verá [outros erros](#). É possível corrigir esses erros, fazendo o upgrade para o valor mais recente de `.spec.version` na sua versão escolhida de IBM MQ, após a atualização do operador ser concluída.

 Se você tiver os gerenciadores de filas existentes e usar o IBM Cloud Pak for Integration Operations Dashboard, veja [“Implementando ou fazendo upgrade da integração do IBM MQ 9.2.2 ou 9.2.3 com Operations Dashboard no IBM Cloud Pak for Integration 2021.4”](#) na página 111 antes de fazer upgrade.

Procedimento

1. Efetue login em seu cluster usando **cloudctl login** (para o IBM Cloud Pak for Integration) ou **oc login**.
2. Certifique-se de que v3 IBM Cloud Pak foundational services Upgrade Channel esteja disponível.

```
oc get packagemanifest -n ibm-common-services ibm-common-service-operator  
-o=jsonpath='{.status.channels[*].name}'
```

3. Corrija o Subscription para mover para o canal de atualização desejado: v3

```
oc patch subscription ibm-common-service-operator --patch '{"spec":{"channel":"v3"}}' --  
type=merge
```

Como proceder a seguir

Você agora está pronto para [Fazer upgrade do IBM MQ Operator](#).

Fazendo upgrade de um gerenciador de filas do IBM MQ usando o console da web do Red Hat OpenShift

Um gerenciador de filas do IBM MQ, implementado usando o IBM MQ Operator, pode ser atualizado em Red Hat OpenShift usando o Operator Hub.

Antes de começar

- Faça login no seu console da web do cluster Red Hat OpenShift.
- Assegure-se de que o seu IBM MQ Operator esteja usando o Canal de atualização desejado. Consulte [“Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas”](#) na página 71.

Antes de fazer o upgrade do gerenciador de filas em um ambiente de airgap, você deve espelhar as últimas imagens IBM Cloud Pak for Integration. Veja [Preparando-se para atualizar o IBM MQ Operator ou gerenciador de filas em um ambiente de airgap](#).

Procedimento

1. Na área de janela de navegação, clique em **Operadores > Operadores instalados**.
Todos os Operadores instalados no projeto especificado serão exibidos.
2. Selecione o **IBM MQ Operator**.
A janela **IBM MQ Operator** é exibida.
3. Navegue até a guia **Gerenciador de filas**.
A janela **Detalhes do gerenciador de filas** é exibida.
4. Selecione o gerenciador de filas do qual você deseja fazer upgrade.
5. Navegue até a guia YAML.
6. Atualize os campos a seguir, quando necessário, para que eles correspondem ao upgrade da versão do gerenciador de filas IBM MQ desejado.
 - spec.version
 - spec.license.licence

Consulte “Suporte de versão para o IBM MQ Operator” na página 7 para obter um mapeamento de canais para versões do IBM MQ Operator e versões do gerenciador de filas IBM MQ.
7. Salve o YAML do gerenciador de filas atualizado.

Fazendo upgrade de um gerenciador de filas do IBM MQ usando a CLI do Red Hat OpenShift

Um gerenciador de filas do IBM MQ, implementado usando o IBM MQ Operator, pode ser atualizado em Red Hat OpenShift usando a linha de comandos.

Antes de começar

É necessário ser um administrador de cluster para concluir essas etapas.

- Faça login na interface da linha de comandos (CLI) do Red Hat OpenShift usando `oc login`.
- Assegure-se de que o seu IBM MQ Operator esteja usando o Canal de atualização desejado. Consulte “Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas” na página 71.

Antes de fazer o upgrade do gerenciador de filas em um ambiente de airgap, você deve espelhar as últimas imagens IBM Cloud Pak for Integration. Veja [Preparando-se para atualizar o IBM MQ Operator ou gerenciador de filas em um ambiente de airgap](#).

Procedimento

Edite o recurso **QueueManager** para atualizar os campos a seguir, quando necessário, para que eles correspondam ao upgrade de versão do gerenciador de filas IBM MQ desejado.

- spec.version
- spec.license.licence

Consulte “Suporte de versão para o IBM MQ Operator” na página 7 para obter um mapeamento de canais para versões do IBM MQ Operator e versões do gerenciador de filas IBM MQ.

Use o comando a seguir:

```
oc edit queuemanager my_qmgr
```

em que `my_qmgr` é o nome do recurso do QueueManager do qual você deseja fazer upgrade.

Fazendo upgrade de um gerenciador de filas do IBM MQ no Red Hat OpenShift usando o Platform Navigator

Um gerenciador de filas do IBM MQ, implementado usando o IBM MQ Operator, pode ser atualizado em Red Hat OpenShift usando o IBM Cloud Pak for Integration Platform Navigator.

Antes de começar

- Efetue login no IBM Cloud Pak for Integration Platform Navigator no namespace que contém o gerenciador de filas do qual você deseja fazer upgrade.
- Assegure-se de que o seu IBM MQ Operator esteja usando o Canal de atualização desejado. Consulte [“Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas”](#) na página 71.

Antes de fazer o upgrade do gerenciador de filas em um ambiente de airgap, você deve espelhar as últimas imagens IBM Cloud Pak for Integration . Veja [Preparando-se para atualizar o IBM MQ Operator ou gerenciador de filas em um ambiente de airgap](#).

Procedimento

1. Na página inicial do IBM Cloud Pak for Integration Platform Navigator, clique na guia **Tempos de execução**.
2. Os gerenciadores de filas com upgrades disponíveis têm um **i** azul ao lado da **Versão**. Clique no **i** para mostrar a **Nova versão disponível**.
3. Clique nos três pontos na extrema direita do gerenciador de filas do qual você deseja fazer upgrade e clique em **Mudar versão**.
4. Em **Selecionar um novo canal ou versão**, selecione a versão de upgrade necessária.
5. Clique em **Mudar versão**.

Resultados

Foi feito upgrade do gerenciador de filas.

Implementando e configurando gerenciadores de filas usando o IBM MQ Operator

IBM MQ 9.1.5 e mais recente são implementados para o Red Hat OpenShift usando o IBM MQ Operator.

Sobre esta tarefa

Procedimento

- [“Preparando seu projeto Red Hat OpenShift para IBM MQ”](#) na página 79.
- [“Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform”](#) na página 81.

Preparando seu projeto Red Hat OpenShift para IBM MQ

Prepare o seu cluster do Red Hat OpenShift Container Platform para que ele esteja pronto para implementar um gerenciador de filas.

Procedimento

- [“Preparando o seu projeto Red Hat OpenShift para IBM MQ usando o console da web do Red Hat OpenShift”](#) na página 80.
- [“Preparando o seu projeto Red Hat OpenShift para IBM MQ usando a CLI do Red Hat OpenShift”](#) na página 80.

Tarefas relacionadas

“Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform” na [página 81](#)

Use o recurso customizado QueueManager para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform.

Preparando o seu projeto Red Hat OpenShift para IBM MQ usando o console da web do Red Hat OpenShift

Prepare o seu cluster do Red Hat OpenShift Container Platform para que ele esteja pronto para implementar um gerenciador de filas usando o IBM MQ Operator. Esta tarefa deve ser concluída por um administrador do projeto.

Antes de começar

Nota: Se você estiver planejando usar o IBM MQ em um projeto com outros componentes do IBM Cloud Pak for Integration já instalados, poderá não ser necessário seguir estas instruções.

Faça login no seu console da web do cluster Red Hat OpenShift.

Sobre esta tarefa

As imagens do IBM MQ Operator são extraídas de um registro de contêiner que executa uma verificação de titularidade de licença. Essa verificação requer uma chave de autorização que é armazenada em um segredo de extração de `docker-registry`. Se você ainda não tiver uma chave de autorização, siga estas instruções para obter uma chave de autorização e criar um segredo de extração.

Procedimento

1. Obtenha a chave de autorização designada para o seu ID.
 - a) Efetue login na [MyIBM Container Software Library](#) com o IBMid e a senha que estão associados ao software autorizado.
 - b) Na seção **Chaves de autorização**, selecione **Copiar chave** para copiar a chave de autorização para a área de transferência.
2. Crie um segredo contendo sua chave de autorização no projeto no qual você deseja implementar o seu gerenciador de filas.
 - a) Na área de janela de navegação, clique em **Cargas de trabalho > Segredo**.
A página Segredos é exibida.
 - b) No menu suspenso **Projeto**, selecione o projeto no qual você deseja instalar o IBM MQ
 - c) Clique no botão **Criar** e selecione **Segredo de extração de imagem**
 - d) No campo **Nome**, digite `ibm-entitlement-key`
 - e) No campo **Endereço do servidor de registro**, insira `cp.icr.io`
 - f) No campo **Nome do usuário**, digite `cp`
 - g) No campo **Senha**, insira a chave de autorização que você copiou na etapa anterior
 - h) No campo **E-mail**, insira o ID do IBM associado ao software autorizado

Como proceder a seguir

“Implementando um gerenciador de filas usando o console da web do Red Hat OpenShift” na [página 83](#)

Preparando o seu projeto Red Hat OpenShift para IBM MQ usando a CLI do Red Hat OpenShift

Prepare o seu cluster do Red Hat OpenShift Container Platform para que ele esteja pronto para implementar um gerenciador de filas usando o IBM MQ Operator. Esta tarefa deve ser concluída por um administrador do projeto.

Antes de começar

Nota: Se você estiver planejando usar o IBM MQ em um projeto com outros componentes do IBM Cloud Pak for Integration já instalados, poderá não ser necessário seguir estas instruções.

Efetue login em seu cluster usando **cloudctl login** (para o IBM Cloud Pak for Integration) ou **oc login**.

Sobre esta tarefa

As imagens do IBM MQ Operator são extraídas de um registro de contêiner que executa uma verificação de titularidade de licença. Essa verificação requer uma chave de autorização que é armazenada em um segredo de extração de `docker-registry`. Se você ainda não tiver uma chave de autorização, siga estas instruções para obter uma chave de autorização e criar um segredo de extração.

Procedimento

1. Obtenha a chave de autorização designada para o seu ID.
 - a) Efetue login na [MyIBM Container Software Library](#) com o IBMid e a senha que estão associados ao software autorizado.
 - b) Na seção **Chaves de autorização**, selecione **Copiar chave** para copiar a chave de autorização para a área de transferência.
2. Crie um segredo contendo sua chave de autorização no projeto no qual você deseja implementar o seu gerenciador de filas.

Execute o seguinte comando, em que `<entitlement-key>` é a chave recuperada na etapa 1 e `<user-email>` é o ID IBM associado ao software intitulado.

```
oc create secret docker-registry ibm-entitlement-key \
--docker-server=cp.icr.io \
--docker-username=cp \
--docker-password=<entitlement-key> \
--docker-email=<user-email>
```

Como proceder a seguir

[“Implementando um gerenciador de filas usando a CLI do Red Hat OpenShift”](#) na página 84

Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform

Use o recurso customizado QueueManager para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform.

Procedimento

-  [“Implementando um gerenciador de filas usando o IBM Cloud Pak for Integration Platform Navigator”](#) na página 82.
-  [“Implementando um gerenciador de filas usando o console da web do Red Hat OpenShift”](#) na página 83.
-  [“Implementando um gerenciador de filas usando a CLI do Red Hat OpenShift”](#) na página 84.

Tarefas relacionadas

[“Exemplos para configurar um gerenciador de filas”](#) na página 85

Um gerenciador de filas pode ser configurado ajustando o conteúdo do recurso customizado do QueueManager.

Implementando um gerenciador de filas usando o IBM Cloud Pak for Integration Platform Navigator

Use o recurso customizado QueueManager para implementar um gerenciador de filas em um cluster de Plataforma de Contêineres Red Hat OpenShift usando o IBM Cloud Pak for Integration Platform Navigator. Essa tarefa deve ser concluída por um administrador do projeto

Antes de começar

Em um navegador, ative o IBM Cloud Pak for Integration Platform Navigator.

Se esta for a sua primeira implementação de um gerenciador de filas neste projeto do Red Hat OpenShift, siga as etapas para [“Preparando seu projeto Red Hat OpenShift para IBM MQ”](#) na página 79.

Procedimento

1. Implementar um gerenciador de filas.

O exemplo a seguir implementa um gerenciador de filas de "iniciação rápida", que usa armazenamento efêmero (não persistente) e desliga a segurança do MQ. As mensagens não serão persistidas em reinicializações do gerenciador de filas. É possível ajustar a configuração para mudar muitas configurações do gerenciador de filas.

a) No IBM Cloud Pak for Integration Platform Navigator, clique em **Administração**, em seguida, **Tempos de execução de integração**. Em versões mais antigas do IBM Cloud Pak for Integration Platform Navigator, clique em **Tempo de execução e instâncias**.

b) Clique em **Criar instância**.

c) Selecione **Sistema de mensagens** e clique em **Avançar**. Em versões mais antigas do IBM Cloud Pak for Integration Platform Navigator, clique em **Gerenciador de filas** e clique em **Avançar**.

O formulário para criar uma instância de um QueueManager é exibido.

Nota: Também é possível clicar em **Código** para visualizar ou mudar o YAML de configuração do QueueManager.

d) Na seção **Detalhes**, verifique ou atualize o campo **Nome** e especifique o **Namespace** no qual a instância do gerenciador de filas será criada.

e) Se você aceitar o contrato de licença do IBM Cloud Pak for Integration, mude a **Aceitação da licença** para **Ativa**.

Deve-se aceitar a licença para implementar um gerenciador de filas.

f) Na seção **Gerenciador de filas**, verifique ou atualize o **Nome** do gerenciador de filas subjacente. Em versões mais antigas do IBM Cloud Pak for Integration Platform Navigator, use a seção **Configuração do gerenciador de filas**.

Por padrão, o nome do gerenciador de filas usado pelos aplicativos clientes do IBM MQ será o mesmo que o nome do QueueManager, mas com quaisquer caracteres inválidos (como hifens) removidos.

g) Clique em **Criar**.

A lista de gerenciadores de filas no projeto atual (namespace) agora é exibida. O novo QueueManager deve ter um status de Pending

2. Verifique se o gerenciador de filas está em execução

A criação terá sido concluída quando o status de QueueManager for Running.

Tarefas relacionadas

[“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift”](#) na página 108

É necessária uma rota do Red Hat OpenShift para conectar um aplicativo a um gerenciador de filas do IBM MQ de fora de um cluster do Red Hat OpenShift. Deve-se ativar o TLS no gerenciador de filas e no aplicativo cliente do IBM MQ, porque o SNI está disponível apenas no protocolo TLS quando um protocolo TLS 1.2 ou superior é usado. O Red Hat OpenShift Container Platform Router usa o SNI para solicitações de roteamento para o gerenciador de filas IBM MQ.

[“Conectando-se ao IBM MQ Console implementado em um cluster do Red Hat OpenShift” na página 115](#)
Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster Red Hat OpenShift Container Platform.

Implementando um gerenciador de filas usando o console da web do Red Hat OpenShift

Use o recurso customizado do QueueManager para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform usando o console da web do Red Hat OpenShift. Essa tarefa deve ser concluída por um administrador do projeto

Antes de começar

Faça login no seu console da web do cluster Red Hat OpenShift. Será necessário selecionar um Projeto existente (namespace) para usar ou criar um novo.

Se esta for a sua primeira implementação de um gerenciador de filas neste projeto do Red Hat OpenShift, siga as etapas para [“Preparando seu projeto Red Hat OpenShift para IBM MQ” na página 79.](#)

Procedimento

1. Implementar um gerenciador de filas.

O exemplo a seguir implementa um gerenciador de filas de "iniciação rápida", que usa armazenamento efêmero (não persistente) e desliga a segurança do MQ. As mensagens não serão persistidas em reinicializações do gerenciador de filas. É possível ajustar a configuração para mudar muitas configurações do gerenciador de filas.

a) No console da web do Red Hat OpenShift, no painel de navegação, clique em **Operadores > Operadores Instalados**

b) Clique em **IBM MQ**.

c) Clique na guia **Gerenciador de Filas**.

d) Clique no botão **Criar QueueManager**.

Um editor YAML é exibido, contendo o YAML de exemplo para um recurso QueueManager.

Nota: Também é possível clicar em **Editar formulário** para visualizar ou mudar a configuração de QueueManager.

e) Se você aceitar o contrato de licença, mude **Aceitação de licença** para **Ativa**.

O IBM MQ está disponível sob várias licenças diferentes. Para obter mais informações sobre as licenças válidas, consulte [“Referência de licenciamento para mq.ibm.com/v1beta1” na página 126.](#) Deve-se aceitar a licença para implementar um gerenciador de filas.

f) Clique em **Criar**.

A lista de gerenciadores de filas no projeto atual (namespace) agora é exibida. O novo QueueManager deve estar em um estado Pending.

2. Verifique se o gerenciador de filas está em execução

A criação terá sido concluída quando o status de QueueManager for Running.

Tarefas relacionadas

[“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift” na página 108](#)

É necessária uma rota do Red Hat OpenShift para conectar um aplicativo a um gerenciador de filas do IBM MQ de fora de um cluster do Red Hat OpenShift. Deve-se ativar o TLS no gerenciador de filas e no aplicativo cliente do IBM MQ, porque o SNI está disponível apenas no protocolo TLS quando um protocolo TLS 1.2 ou superior é usado. O Red Hat OpenShift Container Platform Router usa o SNI para solicitações de roteamento para o gerenciador de filas IBM MQ.

[“Conectando-se ao IBM MQ Console implementado em um cluster do Red Hat OpenShift” na página 115](#)
Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster Red Hat OpenShift Container Platform.

Hat OpenShift

Use o recurso customizado QueueManager para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform usando a interface da linha de comandos (CLI). Essa tarefa deve ser concluída por um administrador do projeto

Antes de começar

É necessário instalar a [Interface da linha de comandos do Red Hat OpenShift Container Platform](#).

Efetue login em seu cluster usando `cloudctl login` (para o IBM Cloud Pak for Integration) ou `oc login`.

Se esta for a sua primeira implementação de um gerenciador de filas neste projeto do Red Hat OpenShift, siga as etapas para [“Preparando seu projeto Red Hat OpenShift para IBM MQ”](#) na página 79.

Procedimento

1. Implementar um gerenciador de filas.

O exemplo a seguir implementa um gerenciador de filas de "iniciação rápida", que usa armazenamento efêmero (não persistente) e desliga a segurança do MQ. As mensagens não serão persistidas em reinicializações do gerenciador de filas. É possível ajustar o conteúdo do YAML para mudar muitas configurações do gerenciador de filas.

a) Criar um arquivo YAML QueueManager

Por exemplo, para instalar um gerenciador de filas básico no IBM Cloud Pak for Integration, crie o arquivo "mq-quickstart.yaml" com o conteúdo a seguir:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"
```

Importante: Se você aceitar o contrato de licença do IBM Cloud Pak for Integration, altere `accept: false` para `accept: true`. Consulte [“Referência de licenciamento para mq.ibm.com/v1beta1”](#) na página 126 para obter detalhes sobre a licença.

Este exemplo também inclui um servidor da web implementado com o gerenciador de filas, com o console da web ativado com a Conexão única com o IBM Cloud Pak Identity and Access Manager.

Para instalar um gerenciador de filas básico independentemente do IBM Cloud Pak for Integration, crie o arquivo "mq-quickstart.yaml" com o conteúdo a seguir:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart
spec:
```

```

version: 9.2.5.0-r3
license:
  accept: false
  license: L-APIG-BZDDDY
web:
  enabled: true
queueManager:
  name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
template:
  pod:
    containers:
      - name: qmgr
        env:
          - name: MQSNOAUT
            value: "yes"

```

Importante: se você aceitar o contrato de licença MQ, altere `accept: false` para `accept: true`. Consulte [“Referência de licenciamento para mq.ibm.com/v1beta1”](#) na página 126 para obter detalhes sobre a licença.

b) Criar o objeto QueueManager

```
oc apply -f mq-quickstart.yaml
```

2. Verifique se o gerenciador de filas está em execução

É possível validar a implementação executando

```
oc describe queuemanager <QueueManagerResourceName>
```

e, em seguida, verificando o status.

Por exemplo, execute

```
oc describe queuemanager quickstart
```

e confira se o campo `status.Phase` indica `Running`

Tarefas relacionadas

[“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift”](#) na página 108

É necessária uma rota do Red Hat OpenShift para conectar um aplicativo a um gerenciador de filas do IBM MQ de fora de um cluster do Red Hat OpenShift . Deve-se ativar o TLS no gerenciador de filas e no aplicativo cliente do IBM MQ , porque o SNI está disponível apenas no protocolo TLS quando um protocolo TLS 1.2 ou superior é usado O Red Hat OpenShift Container Platform Router usa o SNI para solicitações de roteamento para o gerenciador de filas IBM MQ.

[“Conectando-se ao IBM MQ Console implementado em um cluster do Red Hat OpenShift”](#) na página 115
Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster Red Hat OpenShift Container Platform.

Exemplos para configurar um gerenciador de filas

Um gerenciador de filas pode ser configurado ajustando o conteúdo do recurso customizado do QueueManager.

Sobre esta tarefa

Use os exemplos a seguir para ajudá-lo a configurar um gerenciador de filas usando o arquivo YAML do QueueManager.

Procedimento

- [“Exemplo: fornecendo arquivos MQSC e INI”](#) na página 86
- [“Exemplo: configurando o TLS”](#) na página 87

Este exemplo cria um ConfigMap do Kubernetes que contém dois arquivos MQSC e um arquivo INI. Em seguida, um gerenciador de filas é implementado e processa esses arquivos MQSC e INI.

Sobre esta tarefa

Os arquivos [MQSC](#) e [INI](#) poderão ser fornecidos quando um gerenciador de filas for implementado. Os dados MQSC e INI devem ser definidos em um ou mais Kubernetes [ConfigMaps](#) e [Segredos](#). Eles devem ser criados no namespace (projeto) no qual você implementará o gerenciador de filas.

Nota: Um Segredo do Kubernetes deve ser usado quando o arquivo MQSC ou INI contém dados sensíveis. Fornecer o MQSC e o INI dessa forma requer IBM MQ Operator 1.1 ou mais recente.

Exemplo

O exemplo a seguir cria um ConfigMap do Kubernetes que contém dois arquivos MQSC e um arquivo INI. Em seguida, um gerenciador de filas é implementado e processa esses arquivos MQSC e INI.

Exemplo de ConfigMap: aplique o YAML a seguir no seu cluster:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mqsc-ini-example
data:
  example1.mqsc: |
    DEFINE QLOCAL('DEV.QUEUE.1') REPLACE
    DEFINE QLOCAL('DEV.QUEUE.2') REPLACE
  example2.mqsc: |
    DEFINE QLOCAL('DEV.DEAD.LETTER.QUEUE') REPLACE
  example.ini: |
    Channels:
      MQIBindType=FASTPATH
```

Exemplo de QueueManager: implemente o seu gerenciador de filas com a configuração a seguir, usando a linha de comandos ou o IBM Cloud Pak for Integration Platform Navigator:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-cp4i
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "MQSCINI"
    mqsc:
      - configMap:
          name: mqsc-ini-example
          items:
            - example1.mqsc
            - example2.mqsc
    ini:
      - configMap:
          name: mqsc-ini-example
          items:
            - example.ini
  storage:
    queueManager:
      type: ephemeral
```

Importante: Se você aceitar o contrato de licença do IBM Cloud Pak for Integration, altere `accept: false` para `accept: true`. Consulte [Referência de licenciamento do mq.ibm.com/v1beta1](#) para obter detalhes sobre a licença.

Informações adicionais:

- Um gerenciador de filas pode ser configurado para usar um único ConfigMap ou Segredo do Kubernetes (conforme mostrado neste exemplo) ou diversos ConfigMaps ou Segredos do Kubernetes.
- É possível optar por usar todos os dados MQSC e INI de um ConfigMap ou Segredo do Kubernetes (conforme mostrado neste exemplo) ou configurar cada gerenciador de filas para usar apenas um subconjunto dos arquivos disponíveis.
- Os arquivos MQSC e INI são processados em ordem alfabética com base na chave. Assim, o `example1.mqsc` sempre será processado antes de `example2.mqsc`, independentemente da ordem em que eles aparecem na configuração do gerenciador de filas.
- Se vários arquivos MQSC ou INI tiverem a mesma chave em vários ConfigMaps ou segredos do Kubernetes, este conjunto de arquivos será processado com base na ordem na qual os arquivos estão definidos na configuração do gerenciador de filas.

OpenShift CP4I Linux **Exemplo: configurando o TLS**

Esse exemplo implementa um gerenciador de filas no Red Hat OpenShift Container Platform usando o IBM MQ Operator. A comunicação TLS unidirecional é configurada entre um cliente de amostra e o gerenciador de filas. O exemplo demonstra uma configuração bem-sucedida ao colocar e obter mensagens.

Antes de começar

Para concluir esse exemplo, deve-se primeiro ter concluído os pré-requisitos a seguir:

- Instale o IBM MQ client e inclua `samp/bin` e `bin` no seu `PATH`. Os aplicativos **runmqakm**, **amqspuic** e **amqsgetc** são necessários. Eles podem ser instalados como parte do IBM MQ client da maneira a seguir:
 - **Windows** **Linux** Para Windows e Linux: instale o cliente redistribuível IBM MQ para seu sistema operacional por meio de <https://ibm.biz/mq92redistclients>
 - **mac OS** Para Mac: faça download e configure o IBM MQ MacOS Toolkit: <https://developer.ibm.com/tutorials/mq-macos-dev/>
- Instale a ferramenta OpenSSL no seu sistema operacional.
- Crie um projeto/namespace de Red Hat OpenShift Container Platform (OCP) para este exemplo.
- Na linha de comandos, efetue login no cluster do OCP e alterne para o namespace acima.
- Assegure-se de que o IBM MQ Operator esteja instalado e disponível no namespace acima.

Sobre esta tarefa

Este exemplo fornece um YAML de recurso customizado definindo um gerenciador de filas para ser implementado no Red Hat OpenShift Container Platform. Ele também detalha as etapas adicionais necessárias para implementar o gerenciador de filas com o TLS ativado. Após a conclusão, colocar e obter mensagens valida o gerenciador de filas que está configurado com o TLS.

Crie uma chave privada TLS e certificados para o servidor IBM MQ

Os exemplos de código a seguir mostram como criar um certificado autoassinado para o gerenciador de filas e como incluir o certificado em um banco de dados chave para atuar como o armazenamento confiável do cliente. Se você já tiver uma chave privada e um certificado, será possível usá-los.

Observe que os certificados autoassinados devem ser usados apenas para fins de desenvolvimento.

Crie uma chave privada autoassinada e um certificado público no diretório atual

Execute o seguinte comando:

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

Inclua a chave pública do servidor a um banco de dados de chaves de clientes

O banco de dados de chaves é usado como o armazenamento confiável do aplicativo cliente.

Crie o banco de dados de chaves de clientes:

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

Inclua a chave pública gerada anteriormente no banco de dados de chaves de clientes:

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii  
-stashed
```

Configure a implementação dos Certificados TLS no Gerenciador de Filas

Para que o seu gerenciador de filas possa referenciar e aplicar a chave e o certificado, crie um segredo de TLS de Kubernetes, referenciando os arquivos criados acima. Ao fazer isso, assegure-se de que está no namespace criado antes de iniciar esta tarefa.

```
oc create secret tls example-tls-secret --key="tls.key" --cert="tls.crt"
```

Criar um mapa de configuração contendo comandos MQSC

Crie um mapa de configuração de Kubernetes contendo os comandos MQSC para criar uma nova fila e um Canal SVRCONN e para incluir um registro de autenticação de canal que permita o acesso ao canal bloqueando apenas os usuários chamados de *nobody*.

Observe que essa abordagem deve ser usada apenas para fins de desenvolvimento.

Assegure-se de estar no namespace criado anteriormente (consulte [Antes de começar](#)) e digite o YAML a seguir na IU do OCP ou usando a linha de comandos.

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: example-tls-configmap  
data:  
  tls.mqsc: |  
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE  
    DEFINE CHANNEL(SECUREQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)  
    SSLCIPH('ANY_TLS12_OR_HIGHER')  
    SET CHLAUTH(SECUREQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

Crie a rota do OCP necessária

Assegure-se de estar no namespace criado antes de iniciar esta tarefa e insira o YAML a seguir na IU do OCP ou usando a linha de comandos.

```
apiVersion: route.openshift.io/v1  
kind: Route  
metadata:  
  name: example-tls-route  
spec:  
  host: secureqmchl.chl.mq.ibm.com  
  to:  
    kind: Service  
    name: secureqm-ibm-mq  
  port:  
    targetPort: 1414  
  tls:  
    termination: passthrough
```

Observe que o Red Hat OpenShift Container Platform Router usa o SNI para solicitações de roteamento para o gerenciador de filas IBM MQ. Se você mudar o nome do canal especificado no MQSC no mapa de configuração criado anteriormente, também será necessário mudar o campo de host aqui e no arquivo CCDT criado posteriormente. Para obter informações adicionais, consulte [“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift” na página 108.](#)

Implementar o gerenciador de filas

Importante: neste exemplo usamos a variável `MQSNOAUT` para desativar a autorização no gerenciador de filas, o que nos permite focar nas etapas necessárias para conectar um cliente usando

o TLS. Isso não é recomendado em uma implementação de produção do IBM MQ, pois ela faz com que todos os aplicativos que se conectam tenham poderes administrativos integrais, sem nenhum mecanismo para diminuir as permissões para aplicativos individuais.

Crie um novo gerenciador de filas usando o YAML de recurso customizado a seguir. Observe que ele faz referência ao mapa de configuração e ao segredo criados anteriormente, assim como a variável *MQSNOAUT*.

Assegure-se de estar no namespace criado antes de iniciar esta tarefa e insira o YAML a seguir na IU do OCP, usando a linha de comandos ou o IBM Cloud Pak for Integration Platform Navigator. Verifique se a licença correta está especificada e aceite a licença alterando *false* para *true*.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: secureqm
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    name: SECUREQM
  mqsc:
    - configMap:
        name: example-tls-configmap
        items:
          - tls.mqsc
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
          name: qmgr
  version: 9.2.5.0-r3
  web:
    enabled: true
  pki:
    keys:
      - name: example
        secret:
          secretName: example-tls-secret
          items:
            - tls.key
            - tls.crt
```

Confirmar se o gerenciador de filas está sendo executado

O gerenciador de filas agora está sendo implementado. Confirme se está no estado Running antes de prosseguir. Por exemplo:

```
oc get qmgr secureqm
```

Teste a conexão com o gerenciador de filas

Para confirmar que o gerenciador de filas está configurado para comunicação TLS unidirecional, use os aplicativos de amostra **amqspc** e **amqsgetc**:

Localize o nome do host do gerenciador de filas

Use o comando a seguir para localizar o nome do host completo do gerenciador de filas para a rota `secureqm-ibm-mq-qm`:

```
oc get routes secureqm-ibm-mq-qm
```

Especificar os detalhes do gerenciador de filas

Crie um arquivo CCDT .JSON que especifica os detalhes do gerenciador de filas. Substitua o valor do host pelo nome do host da etapa anterior.

```
{
  "channel":
  [
    {
      "name": "SECUREQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<hostname from previous step>",
            "port": 443
          }
        ],
        "queueManager": "SECUREQM"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

Exportar variáveis de ambiente

Exporte as variáveis de ambiente a seguir, da maneira apropriada para o seu sistema operacional. Essas variáveis serão lidas por **amqsputc** e **amqsgetc**.

Atualize o caminho para os arquivos no seu sistema:

```
export MQCCDTURL='<full path to file>/CCDT.JSON'
export MQSSLKEYR='<full path to file>/clientkey'
```

Colocar mensagens na fila

Execute o seguinte comando:

```
amqsputc EXAMPLE.QUEUE SECUREQM
```

Se a conexão com o gerenciador de filas for bem-sucedida, a resposta a seguir será a saída:

```
target queue is EXAMPLE.QUEUE
```

Coloque várias mensagens na fila, inserindo um texto e pressionando **Enter** cada vez.

Para concluir, pressione **Enter** duas vezes.

Recuperar as mensagens da fila

Execute o seguinte comando:

```
amqsgetc EXAMPLE.QUEUE SECUREQM
```

As mensagens que foram incluídas na etapa anterior foram consumidas e uma saída foi obtida.

Depois de alguns segundos, você verá o resultado do comando.

Parabéns, você implementou com sucesso um gerenciador de filas com TLS ativado e mostrou que é possível colocar e obter mensagens de forma segura no gerenciador de filas de um cliente.

OpenShift

CP4I

Exemplo: Customizando anotações de serviço de licença

O IBM MQ Operator inclui automaticamente as anotações do IBM License Service nos recursos implementados. Eles são monitorados pelo IBM License Service, e são gerados relatórios que correspondem à autorização necessária.

Sobre esta tarefa

As anotações incluídas pelo IBM MQ Operator são aquelas esperadas em situações padrão e são baseadas nos valores de licença selecionados durante a implementação de um gerenciador de filas.

Exemplo

Se o **License** for configurado como L-RJON-BZFQU2 (IBM Cloud Pak for Integration 2021.2.1) e o **Use** for configurado como NonProduction, as seguintes anotações serão aplicadas:

- cloudpakId: c8b82d189e7545f0892db9ef2731b90d
- cloudpakName: IBM Cloud Pak for Integration
- productChargedContainers: qmgr
- productCloudpakRatio: '4:1'
- productID: 21dfe9a0f00f444f888756d835334909
- productName: IBM MQ Advanced for Non-Production
- productMetric: VIRTUAL_PROCESSOR_CORE
- productVersion: 9.2.3.0

Dentro do IBM Cloud Pak for Integration, implementações do IBM App Connect Enterprise incluem uma autorização restrita para o IBM MQ. Nessas situações, essas anotações precisam ser substituídas para garantir que o IBM License Service captura o uso correto. Para fazer isso, use a abordagem descrita em [“Incluindo anotações e rótulos customizados para recursos do gerenciador de filas”](#) na página 114.

Por exemplo, se o IBM MQ for implementado sob autorização do IBM App Connect Enterprise, use a abordagem mostrada no seguinte fragmento de código:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productMetric: FREE
```

Há dois outros motivos comuns pelos quais as anotações de licença podem exigir modificação:

1. IBM MQ Advanced está incluído no direito de outro produto IBM.
 - Nesta situação, use a abordagem descrita anteriormente para o IBM App Connect Enterprise.
2. IBM MQ é implementado em uma licença do IBM Cloud Pak for Integration.
 - Se você tem uma licença do IBM Cloud Pak for Integration, é possível decidir implementar um gerenciador de filas sob a razão IBM MQ ou IBM MQ Advanced. Se você implementar em uma razão IBM MQ, certifique-se de não usar nenhum recurso avançado, como HA nativa ou Advanced Message Security.
 - Nessa situação, use as seguintes anotações para uso de produção:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: c661609261d5471fb4ff8970a36bccea
    productCloudpakRatio: '4:1'
    productName: IBM MQ for Production
    productMetric: VIRTUAL_PROCESSOR_CORE
```

- Use as seguintes anotações para uso não produção:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
```

```
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: 151bec68564a4a47a14e6fa99266deff
    productCloudpakRatio: '8:1'
    productName: IBM MQ for Non-Production
    productMetric: VIRTUAL_PROCESSOR_CORE
```

OpenShift > CP4I Configurando alta disponibilidade para gerenciadores de filas usando o IBM MQ Operator

Sobre esta tarefa

Procedimento

- [V 9.2.3](#)
“HA nativa” na página 92.
- [V 9.2.3](#)
“Exemplo: configurando um gerenciador de filas Native HA” na página 94.
- “Exemplo: configurando um gerenciador de filas de várias instâncias” na página 103.

CP4I > CD > V 9.2.3 **HA nativa**

A HA nativa é uma solução de alta disponibilidade nativa (integrada) para o IBM MQ que é adequada para uso com armazenamento de bloco de nuvem.

Uma configuração de HA nativa fornece um gerenciador de filas altamente disponível no qual os dados do MQ recuperáveis (por exemplo, as mensagens) são replicados em vários conjuntos de armazenamento, impedindo a perda de falhas de armazenamento. O gerenciador de filas consiste em várias instâncias em execução, uma é a líder, as outras estão prontas para assumir o controle rapidamente no caso de uma falha, maximizando o acesso ao gerenciador de filas e suas mensagens.

Uma configuração de HA nativa consiste em três pods do Kubernetes, cada um com uma instância do gerenciador de filas. Uma instância é o gerenciador de filas ativas, processando mensagens e gravando em seu log de recuperação. Sempre que o log de recuperação é gravado, o gerenciador de filas ativas envia os dados para as outras duas instâncias, conhecidas como réplicas. Cada réplica grava em seu próprio log de recuperação, reconhece os dados e, em seguida, atualiza seus próprios dados de fila no log de recuperação replicado. Se o pod que executa o gerenciador de filas ativas falhar, uma das instâncias de réplica do gerenciador de filas assumirá a função ativa e terá dados atuais com os quais operar.

O tipo de log é conhecido como um 'log replicado'. Um log replicado é essencialmente um log linear, com gerenciamento de log automático e imagens de mídia automáticas ativadas. Consulte [Tipos de criação de log](#). Use as mesmas técnicas para gerenciar o log replicado usado para gerenciar um log linear.

Um Serviço do Kubernetes é usado para rotear conexões do cliente TCP/IP para a instância ativa atual, que é identificada como sendo o único pod que está pronto para o tráfego de rede. Isso acontece sem a necessidade de que o aplicativo cliente esteja ciente das diferentes instâncias.

Três pods são usados para reduzir consideravelmente a possibilidade do surgimento de uma situação de split-brain. Em um sistema de alta disponibilidade de dois pods, o split-brain pode ocorrer quando a conectividade entre os dois pods se quebra. Sem conectividade, ambos os pods poderiam executar o gerenciador de filas ao mesmo tempo, acumulando dados diferentes. Quando a conexão for restaurada, haverá duas versões diferentes dos dados (um 'split-brain') e a intervenção manual será necessária para decidir qual conjunto de dados manter e qual descartar.

A HA nativa usa um sistema de três pods com quorum para evitar a situação de split-brain. Os pods que podem se comunicar com pelo menos um dos outros pods formam um quorum. Um gerenciador de filas pode se tornar a instância ativa apenas em um pod que tem quorum. O gerenciador de filas não pode

se tornar ativo em um pod que não está conectado a pelo menos um outro pod, portanto, nunca poderá haver duas instâncias ativas ao mesmo tempo:

- Se um único pod falhar, o gerenciador de filas em um dos outros dois pods poderá assumir o controle. Se dois dos três pods falharem, o gerenciador de filas não poderá se tornar a instância ativa no restante porque ele não tem quorum (ele não pode informar se os outros falharam ou se ainda estão em execução e ele perdeu a conectividade).
- Se um único pod perder a conectividade, o gerenciador de filas não poderá se tornar ativo nele porque ele não terá quorum. O gerenciador de filas em um dos dois pods restantes pode assumir o controle, os quais têm quorum. Se todos os pods perderem a conectividade, o gerenciador de filas não poderá se tornar ativo em nenhum deles, pois nenhum terá quorum.

Se um pod ativo falhar e, posteriormente, se recuperar, ele poderá se unir novamente ao grupo em uma função de réplica.

A figura a seguir mostra uma implementação típica com três instâncias de um gerenciador de filas implementadas em três contêineres.

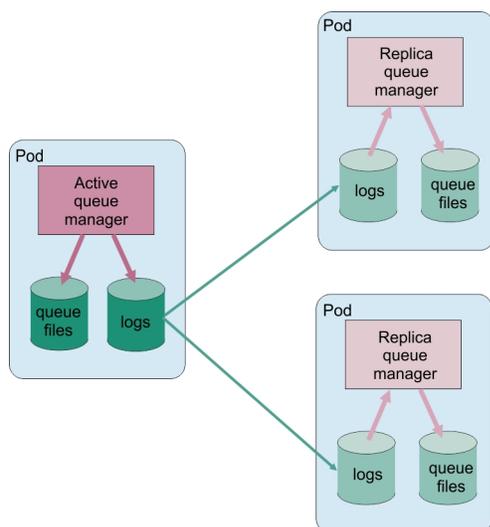


Figura 1. Exemplo de configuração de HA nativa

CP4I > CD > V9.2.3 Configurando o Native HA usando o IBM MQ Operator

O Native HA é configurado usando a API QueueManager, e opções avançadas estão disponíveis usando um arquivo INI.

O Native HA é configurado usando o `.spec.queueManager.availability` da API QueueManager, por exemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: nativeha-example
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    availability:
      type: NativeHA
    version: 9.2.5.0-r3
```

O campo `.spec.queueManager.availability.type` deve ser configurado como NativeHA.

O Native HA está disponível no IBM MQ 9.2.3 ou superior.

Em `.spec.queueManager.availability`, também é possível configurar um segredo de TLS e cifras para usar entre instâncias do gerenciador de filas ao replicar. Isso é fortemente recomendado e um guia

passo a passo está disponível em [“Exemplo: configurando um gerenciador de filas Native HA”](#) na página 94.

Referências relacionadas

[“Exemplo: configurando um gerenciador de filas Native HA”](#) na página 94

Este exemplo mostra como você implementa um gerenciador de filas usando o recurso de alta disponibilidade nativa no Red Hat OpenShift Container Platform (OCP) usando o IBM MQ Operator.

 *Exemplo: configurando um gerenciador de filas Native HA*

Este exemplo mostra como você implementa um gerenciador de filas usando o recurso de alta disponibilidade nativa no Red Hat OpenShift Container Platform (OCP) usando o IBM MQ Operator.

Antes de Começar

Para concluir esse exemplo, deve-se primeiro ter concluído os pré-requisitos a seguir:

- Instale o IBM MQ client e inclua os diretórios `samp/bin` e `bin` instalados em seu `PATH`. O cliente fornece os aplicativos `runmqakm`, `amqspuic` e `amqsgetc` que são necessários para este exemplo. Instale o IBM MQ client como a seguir:
 -  Para Windows e Linux: instale o cliente redistribuível IBM MQ para seu sistema operacional por meio de <https://ibm.biz/mq92redistclients>
 -  Para Mac: faça download e configure o IBM MQ MacOS Toolkit. Consulte <https://ibm.biz/mqdevmacclient>.
- Instale a ferramenta OpenSSL no seu sistema operacional. Ela é necessária para gerar um certificado autoassinado para o gerenciador de filas quando você ainda não tem uma chave privada e um certificado.
- Crie um projeto/namespace do Red Hat OpenShift Container Platform (OCP) para este exemplo e siga as etapas na tarefa [“Preparando seu projeto Red Hat OpenShift para IBM MQ”](#) na página 79
- Na linha de comandos, efetue login no cluster do OCP e alterne para o namespace que você acabou de criar.
- Assegure-se de que o IBM MQ Operator esteja instalado e disponível no namespace.
- Configure uma classe de armazenamento padrão no OCP, para ser usada por seu gerenciador de filas. Para concluir este tutorial sem configurar uma classe de armazenamento padrão, consulte [Nota 2: usando uma classe de armazenamento não padrão](#).

Sobre esta tarefa

Os gerenciadores de filas de HA nativa envolvem um Pod ativo e dois Pods de réplica do Kubernetes. Eles são executados como parte de um Conjunto Stateful do Kubernetes com exatamente três réplicas e um conjunto de Volumes Persistentes do Kubernetes. Para obter mais informações sobre os gerenciadores de filas de HA nativa, consulte [“Alta disponibilidade para IBM MQ em contêineres”](#) na página 16.

O exemplo fornece um YAML de recurso customizado que define um gerenciador de filas Native HA que usa armazenamento persistente e é configurado com TLS. Depois de implementar o gerenciador de filas no OCP, você simula a falha do pod do gerenciador de filas ativas. Você vê a recuperação automática ocorrer e prova que ela foi bem-sucedida colocando e recebendo mensagens após a falha.

exemplo

Criar uma chave privada TLS e certificados para servidor MQ

É possível criar um certificado autoassinado para o gerenciador de filas e incluir o certificado em um banco de dados de chaves para agir como o armazenamento confiável para o cliente. Se você já tiver uma chave privada e um certificado, será possível usá-los. Observe que o uso de certificados autoassinados é necessário apenas para propósitos de desenvolvimento.

Para criar uma chave privada autoassinada e um certificado público no diretório atual, execute o comando a seguir:

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

Criar uma chave privada TLS e certificados para uso interno por HA nativa

Os três pods em um gerenciador de filas de HA nativa replicam dados por meio da rede. É possível criar um certificado autoassinado para uso ao replicar internamente. Observe que o uso de certificados autoassinados é necessário apenas para propósitos de desenvolvimento.

Para criar uma chave privada autoassinada e um certificado público no diretório atual, execute o comando a seguir:

```
openssl req -newkey rsa:2048 -nodes -keyout nativeha.key -subj "/CN=localhost" -x509 -days 3650 -out nativeha.crt
```

Incluir a chave pública do gerenciador de filas em um banco de dados de chaves do cliente

Um banco de dados de chaves do cliente é usado como o armazenamento confiável para o aplicativo cliente.

Crie o banco de dados de chaves de clientes:

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

Inclua a chave pública gerada anteriormente no banco de dados de chaves de clientes:

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

Criar um segredo contendo certificados TLS para implementação do gerenciador de filas

Para que o seu gerenciador de filas possa referenciar e aplicar a chave e o certificado, crie um segredo de TLS do Kubernetes, referenciando os arquivos criados acima. Ao fazer isso, assegure-se de que está no namespace criado antes de iniciar esta tarefa.

```
oc create secret tls example-ha-secret --key="tls.key" --cert="tls.crt"
```

Criar um segredo contendo o certificado TLS de HA nativa interno e a chave

Para que o seu gerenciador de filas possa referenciar e aplicar a chave e o certificado, crie um segredo de TLS do Kubernetes, referenciando os arquivos criados acima. Ao fazer isso, assegure-se de que está no namespace criado antes de iniciar esta tarefa.

```
oc create secret tls example-ha-secret-internal --key="nativeha.key" --cert="nativeha.crt"
```

Criar um mapa de configuração contendo comandos MQSC

Crie um mapa de configuração do Kubernetes contendo os comandos MQSC para criar uma nova fila e um Canal SVRCONN e para incluir um registro de autenticação de canal que permita o acesso ao canal, bloqueando somente os usuários chamados *nobody*.

Observe que essa abordagem deve ser usada apenas para fins de desenvolvimento.

Assegure-se de estar no namespace criado anteriormente (consulte [“Antes de Começar”](#) na página 94), em seguida, insira o YAML a seguir na IU do OCP ou usando a linha de comandos:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-mi-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') DEFPSIST(YES) REPLACE
    DEFINE CHANNEL(HAQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    SET CHLAUTH(HAQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

Configurar roteamento

Se você estiver usando um IBM MQ client ou kit de ferramentas no IBM MQ 9.2.1 ou mais recente, será possível configurar o roteamento para o gerenciador de filas usando um arquivo de configuração do gerenciador de filas (um arquivo INI). No arquivo, você configura a variável *OutboundSNI* para rotear com base no nome do host, em vez de no nome do canal.

Crie um arquivo no diretório em que você está executando comandos, denominado `mqclient.ini`, contendo exatamente o seguinte texto:

```
SSL:
  OutboundSNI=HOSTNAME
```

Não mude nenhum valor neste arquivo INI. Por exemplo, a sequência `HOSTNAME` não deve ser alterada.

Para obter detalhes adicionais, consulte [Sub-rotina SSL do arquivo de configuração do cliente](#).

Se você estiver usando um IBM MQ client ou kit de ferramentas anterior a IBM MQ 9.2.1, será necessário criar uma rota OCP em vez do arquivo de configuração anterior. Siga as etapas em [Nota 1: criando uma rota](#).

Implementar o gerenciador de filas

Importante: neste exemplo usamos a variável *MQSNOAUT* para desativar a autorização no gerenciador de filas, o que nos permite focar nas etapas necessárias para conectar um cliente usando o TLS. Isso não é recomendado em uma implementação de produção do IBM MQ, pois ela faz com que todos os aplicativos que se conectem tenham poderes administrativos integrais, sem nenhum mecanismo para diminuir as permissões para aplicativos individuais.

Copie e atualize o YAML a seguir.

- Assegure-se de que a licença correta esteja especificada. Consulte [Referência de licenciamento para mq.ibm.com/v1beta1](#). No IBM Cloud Pak for Integration 2021.1.1, a licença deve ser a licença de avaliação L - RJON - BYRMYW
- Aceite a licença alterando `false` para `true`.

YAML do recurso customizado do gerenciador de filas:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: nativeha-example
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    name: HAEXAMPLE
    availability:
      type: NativeHA
    tls:
      secretName: example-ha-secret-internal
  mqsc:
    - configMap:
        name: example-mi-configmap
        items:
          - tls.mqsc
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
          name: qmgr
  version: 9.2.5.0-r3
  pki:
    keys:
      - name: example
        secret:
          secretName: example-ha-secret
```

```
items:
- tls.key
- tls.crt
```

Assegurando que você esteja no namespace criado anteriormente, implemente o YAML atualizado, usando o console da web do Red Hat OpenShift Container Platform, a linha de comandos ou usando o IBM Cloud Pak for Integration Platform Navigator.

Há um breve atraso enquanto o sistema configura o gerenciador de filas Native HA, após o qual o gerenciador de filas deve estar disponível para uso.

Validação

Nesta seção, nós validamos que o gerenciador de filas se comporta conforme o esperado.

Confirmar se o gerenciador de filas está sendo executado

O gerenciador de filas agora está sendo implementado. Confirme se está no estado Running antes de prosseguir. Por exemplo:

```
oc get qmgr nativeha-example
```

Teste a conexão com o gerenciador de filas

Para confirmar que o gerenciador de filas está configurado para comunicação TLS unidirecional, use os aplicativos de amostra **amqsputc** e **amqsgetc**:

Localize o nome do host do gerenciador de filas

Para encontrar o nome do host do gerenciador de filas para a rota nativeha-example-ibm-mq-qm, execute o seguinte comando. O nome do host é retornado no campo HOST.

```
oc get routes nativeha-example-ibm-mq-qm
```

Especificar os detalhes do gerenciador de filas

Crie um arquivo CCDT .JSON que especifica os detalhes do gerenciador de filas. Substitua o valor do host pelo nome do host retornado pela etapa anterior.

```
{
  "channel":
  [
    {
      "name": "HAQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<host from previous step>",
            "port": 443
          }
        ],
        "queueManager": "HAEXAMPLE"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

Exportar variáveis de ambiente

Exporte as variáveis de ambiente a seguir, da maneira apropriada para o seu sistema operacional. Essas variáveis serão lidas por **amqsputc** e **amqsgetc**.

Atualize o caminho para os arquivos no seu sistema:

```
export MQCCDTURL='<full_path_to_file>/CCDT.JSON'
export MQSSLKEYR='<full_path_to_file>/clientkey'
```

Colocar mensagens na fila

Execute o seguinte comando:

```
amqsputc EXAMPLE.QUEUE HAEXAMPLE
```

Se a conexão com o gerenciador de filas for bem-sucedida, a resposta a seguir será a saída:

```
target queue is EXAMPLE.QUEUE
```

Coloque várias mensagens na fila, inserindo um texto e pressionando **Enter** cada vez.

Para concluir, pressione **Enter** duas vezes.

Recuperar as mensagens da fila

Execute o seguinte comando:

```
amqsgetc EXAMPLE.QUEUE HAEXAMPLE
```

As mensagens que foram incluídas na etapa anterior foram consumidas e uma saída foi obtida.

Depois de alguns segundos, você verá o resultado do comando.

Forçar o pod ativo a falhar

Para validar a recuperação automática do gerenciador de filas, simule uma falha de pod:

Visualizar os pods ativos e em espera

Execute o seguinte comando:

```
oc get pods --selector app.kubernetes.io/instance=nativeha-example
```

Observe que, no campo **READY**, o pod ativo retorna o valor 1/1, enquanto os pods de réplica retornam o valor 0/1.

Excluir o pod ativo

Execute o comando a seguir, especificando o nome completo do pod ativo:

```
oc delete pod nativeha-example-ibm-mq-<value>
```

Visualizar o status do pod novamente

Execute o seguinte comando:

```
oc get pods --selector app.kubernetes.io/instance=nativeha-example
```

Visualizar o status do gerenciador de filas

Execute o comando a seguir, especificando o nome completo de um dos outros pods:

```
oc exec -t Pod -- dspmq -o nativeha -x -m HAEXAMPLE
```

O status deve mostrar que a instância ativa foi mudada, por exemplo:

```
QMNAME(HAEXAMPLE) ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Colocar e receber mensagens novamente

Depois que o pod de espera se tornar o pod ativo (ou seja, depois que o valor do campo READY se tornar 1/1), use os comandos a seguir novamente, conforme descrito anteriormente, para colocar mensagens no gerenciador de filas e, em seguida, recuperar as mensagens do gerenciador de filas:

```
amqsputc EXAMPLE.QUEUE HAEXAMPLE
```

```
amqsgetc EXAMPLE.QUEUE HAEXAMPLE
```

Parabéns, você implementou com sucesso um gerenciador de filas de HA nativa e mostrou que ele pode se recuperar automaticamente de uma falha de pod.

Informações Adicionais

Nota 1: criando uma rota

Se você estiver usando um IBM MQ client ou kit de ferramentas anterior a IBM MQ 9.2.1, será necessário criar uma Rota.

Para criar a rota, assegure-se de estar no namespace criado anteriormente (consulte [“Antes de Começar”](#) na página 94) e, em seguida, insira o YAML a seguir no console da web do Red Hat OpenShift Container Platform ou por meio da linha de comandos:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: example-mi-route
spec:
  host: hamqchl.chl.mq.ibm.com
  to:
    kind: Service
    name: nativeha-example-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Observe que o Red Hat OpenShift Container Platform Router usa o SNI para solicitações de roteamento para o gerenciador de filas IBM MQ. Se você mudar o nome do canal especificado no mapa de configuração contendo comandos MQSC, também deverá mudar o campo host aqui e no arquivo CCDT . JSON que especifica os detalhes do gerenciador de filas. Para obter informações adicionais, consulte [“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift”](#) na página 108.

Nota 2: usando uma classe de armazenamento não padrão

Este exemplo espera que uma classe de armazenamento padrão tenha sido configurada no Red Hat OpenShift Container Platform, portanto, nenhuma informação de armazenamento é necessária no YAML do recurso customizado do gerenciador de filas. Se você não tiver uma classe de armazenamento configurada como padrão ou deseja usar uma classe de armazenamento diferente, inclua `defaultClass: <storage_class_name>` em `spec.queueManager.storage`.

O nome da classe de armazenamento deve corresponder exatamente ao nome de uma classe de armazenamento que já existe. Ou seja, deve corresponder ao nome retornado pelo comando `oc get storageclass`. Ele também deve oferecer suporte a `ReadWriteMany`. Para obter informações adicionais, consulte [“Considerações de armazenamento para o IBM MQ Operator”](#) na página 11.

Tarefas relacionadas

[“Visualizando o status de gerenciadores de filas de HA nativa para contêineres certificados do IBM MQ ..”](#) na página 99

Para contêineres certificados pelo IBM MQ , é possível visualizar o status das instâncias de HA nativa executando o comando **dspmqs** dentro de um desses Pods em execução

 *Visualizando o status de gerenciadores de filas de HA nativa para contêineres certificados do IBM MQ ..*

Para contêineres certificados pelo IBM MQ , é possível visualizar o status das instâncias de HA nativa executando o comando **dspmqs** dentro de um desses Pods em execução

Sobre esta tarefa

Importante:

É possível usar o comando **dspmqs** em um dos Pods em execução para visualizar o status operacional de uma instância do gerenciador de filas. As informações retornadas estarão de acordo com o estado ativo

ou de réplica da instância. As informações fornecidas pela instância ativa são definitivas; as informações de nós de réplica podem estar desatualizadas.

Você pode executar as seguintes ações:

- Visualizar se a instância do gerenciador de filas no nó atual está ativa ou é uma réplica.
- Visualizar o status operacional de HA nativa da instância no nó atual.
- Visualizar o status operacional de todas as três instâncias em uma configuração de HA nativa.

Os campos de status a seguir são usados para relatar o status de configuração da HA nativa:

ROLE

Especifica a função atual da instância e é Active, Replica ou Unknown.

INSTÂNCIA

O nome fornecido para esta instância do gerenciador de filas quando ela foi criada usando a opção **-lr** do comando **crtmqm**.

INSYNC

Indica se a instância está apta a assumir o controle como a instância ativa, se necessário.

QUORUM

Relata o status de quorum no formato *number_of_instances_in-sync/number_of_instances_configured*.

REPLADDR

O endereço de replicação da instância do gerenciador de filas.

CONNECTV

Indica se o nó está conectado à instância ativa.

BACKLOG

Indica o número de KB que a instância está atrasada.

CONNINST

Indica se a instância nomeada está conectada a esta instância.

ALTDATE

Indica a data da última atualização dessas informações (em branco se elas nunca foram atualizadas).

ALTTIME

Indica o horário da última atualização dessas informações (em branco se elas nunca foram atualizadas).

Procedimento

- Localize os pods que fazem parte de seu gerenciador de filas

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- Executar o dspmq em um dos pods

```
oc exec -t Pod dspmq
```

```
oc rsh Pod
```

para um shell interativo, em que é possível executar o dspmq diretamente.

- Para determinar se uma instância do gerenciador de filas está em execução como a instância ativa ou como uma réplica:

```
oc exec -t Pod dspmq -o status -m QMgrName
```

Uma instância ativa de um gerenciador de filas denominado BOB relataria o status a seguir:

```
QMNAME(BOB)          STATUS(Running)
```

Uma instância de réplica de um gerenciador de filas denominado BOB relataria o status a seguir:

```
QMNAME(BOB)                STATUS(Replica)
```

Uma instância inativa relataria o status a seguir:

```
QMNAME(BOB)                STATUS(Ended Immediately)
```

- Para determinar o status operacional de HA nativa da instância no pod especificado:

```
oc exec -t Pod dspmq -o nativeha -m QMgrName
```

A instância ativa de um gerenciador de filas denominado BOB pode relatar o status a seguir:

```
QMNAME(BOB)                ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Uma instância de réplica de um gerenciador de filas denominado BOB pode relatar o status a seguir:

```
QMNAME(BOB)                ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Uma instância inativa de um gerenciador de filas denominado BOB pode relatar o status a seguir:

```
QMNAME(BOB)                ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Para determinar o status operacional da HA nativa de todas as instâncias na configuração de HA nativa:

```
oc exec -t Pod dspmq -o nativeha -x -m QMgrName
```

Se você emitir este comando no nó que executa a instância ativa do gerenciador de filas BOB, o status a seguir poderá ser recebido:

```
QMNAME(BOB)                ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Ao emitir este comando em um nó que executa uma instância de réplica do gerenciador de filas BOB, você recebe o status a seguir, que indica que uma das réplicas está ficando atrasada:

```
QMNAME(BOB)                ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Se você emitir este comando em um nó que executa uma instância inativa do gerenciador de filas BOB, o status a seguir poderá ser recebido:

```
QMNAME(BOB)                ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

Se você emitir o comando quando as instâncias ainda estiverem negociando quais estão ativas e quais são réplicas, receberá o status a seguir:

```
QMNAME(BOB)                STATUS(Negotiating)
```

Referências relacionadas

[Comando dspmq \(exibir gerenciadores de filas\)](#)

“Exemplo: configurando um gerenciador de filas Native HA” na página 94

Este exemplo mostra como você implementa um gerenciador de filas usando o recurso de alta disponibilidade nativa no Red Hat OpenShift Container Platform (OCP) usando o IBM MQ Operator.

CP4I > CD > V 9.2.3 *Ajuste avançado para Native HA*

Configurações avançadas para sincronizações e intervalos de ajuste. Não deve haver necessidade de usar essas configurações, a menos que seja conhecido que os padrões não correspondem aos requisitos de seu sistema.

As opções básicas para configurar o Native HA são manipuladas usando a API `QueueManager`, que o IBM MQ Operator usa para configurar os arquivos INI do gerenciador de filas subjacentes para você. Há algumas opções mais avançadas que são configuráveis apenas usando um arquivo INI, na sub-rotina `NativeHALocalInstance`. Consulte também “Exemplo: fornecendo arquivos MQSC e INI” na página 86 para obter mais informações sobre como configurar um arquivo INI..

HeartbeatInterval

O intervalo de pulsação define com que frequência em milissegundos uma instância ativa de um gerenciador de filas de HA nativa envia uma pulsação de rede. O intervalo válido do valor de intervalo de pulsação é de 500 (0,5 segundos) a 60.000 (1 minuto). Um valor fora desse intervalo causa uma falha na inicialização do gerenciador de filas. Se esse atributo for omitido, um valor padrão de 5.000 (5 segundos) será usado. Cada instância deve usar o mesmo intervalo de pulsação.

HeartbeatTimeout

O tempo limite de pulsação define quanto tempo uma instância de réplica de um gerenciador de filas de HA nativa aguarda antes de decidir que a instância ativa está não responsiva. O intervalo válido do valor de tempo limite do intervalo de pulsação é de 500 (0,5 segundos) a 120.000 (2 minutos). O valor do tempo limite de pulsação deve ser maior ou igual ao intervalo de pulsação.

Um valor inválido gera uma falha na inicialização do gerenciador de filas. Se este atributo for omitido uma réplica aguardará 2 x `HeartbeatInterval` antes de iniciar o processo para eleger uma nova instância ativa. Cada instância deve usar o mesmo tempo limite de pulsação.

RetryInterval

O intervalo de nova tentativa define com que frequência em milissegundos um gerenciador de filas de HA nativa deve tentar novamente um link de replicação com falha. O intervalo válido do intervalo de nova tentativa é de 500 (0,5 segundos) a 120.000 (2 minutos). Se este atributo for omitido uma réplica aguardará por 2 x `HeartbeatInterval` antes de tentar novamente um link de replicação com falha.

CP4I *Finalizando gerenciadores de filas de alta disponibilidade nativa*

É possível usar o comando `endmqm` para terminar um gerenciador de filas ativo ou de réplica que faz parte de um grupo de HA nativa

Procedimento

- Para terminar a instância ativa de um gerenciador de filas, consulte [Terminando Gerenciadores de Filas de HA Nativa](#) na seção Configuração desta documentação

CP4I > V 9.2.2 > CD *Avaliando o recurso de HA Nativa no IBM Cloud Pak for Integration 2021.1.1*

O período de avaliação do IBM Cloud Pak for Integration 2021.1.1 Native HA terminou. Use o recurso Native HA atualizado que está disponível por meio do IBM Cloud Pak for Integration 2021.2.1, usando o IBM MQ Operator 1.6 ou superior com o IBM MQ 9.2.3 ou superior.

Tarefas relacionadas

“Visualizando o status de gerenciadores de filas de HA nativa para contêineres certificados do IBM MQ ..” na página 99

Para contêineres certificados pelo IBM MQ , é possível visualizar o status das instâncias de HA nativa executando o comando `dspmqs` dentro de um desses Pods em execução

Referências relacionadas

“Exemplo: configurando um gerenciador de filas Native HA” na página 94

Este exemplo mostra como você implementa um gerenciador de filas usando o recurso de alta disponibilidade nativa no Red Hat OpenShift Container Platform (OCP) usando o IBM MQ Operator.

Exemplo: configurando um gerenciador de filas de várias instâncias

Este exemplo mostra como você implementa um gerenciador de filas de várias instâncias no Red Hat OpenShift Container Platform (OCP) usando o IBM MQ Operator. Neste exemplo, você também configura a comunicação TLS unidirecional entre um cliente de amostra e o gerenciador de filas. O exemplo demonstra uma configuração bem-sucedida, colocando e obtendo mensagens antes e depois de uma falha de pod simulada.

Antes de Começar

Para concluir esse exemplo, deve-se primeiro ter concluído os pré-requisitos a seguir:

- Instale o IBM MQ client e inclua os diretórios `samp/bin` e `bin` instalados em seu `PATH`. O cliente fornece os aplicativos `runmqakm`, `amqsputc` e `amqsgetc` que são necessários para este exemplo. Instale o IBM MQ client como a seguir:
 -   Para Windows e Linux: instale o cliente redistribuível IBM MQ para seu sistema operacional por meio de <https://ibm.biz/mq92redistclients>
 -  Para Mac: faça download e configure o IBM MQ MacOS Toolkit. Consulte <https://developer.ibm.com/tutorials/mq-macos-dev/>.
- Instale a ferramenta OpenSSL no seu sistema operacional. Ela é necessária para gerar um certificado autoassinado para o gerenciador de filas quando você ainda não tem uma chave privada e um certificado.
- Crie um projeto/namespace de Red Hat OpenShift Container Platform (OCP) para este exemplo.
- Na linha de comandos, efetue login no cluster do OCP e alterne para o namespace acima.
- Assegure-se de que o IBM MQ Operator esteja instalado e disponível no namespace acima.
- Configure uma classe de armazenamento padrão no OCP, para ser usada por seu gerenciador de filas. Para concluir este tutorial sem configurar uma classe de armazenamento padrão, consulte [Nota 2: usando uma classe de armazenamento não padrão](#).

Sobre esta tarefa

Os gerenciadores de filas de várias instâncias envolvem um Pod ativo e um Pod de espera do Kubernetes. Eles são executados como parte de um Conjunto Stateful do Kubernetes com exatamente duas réplicas e um conjunto de Volumes Persistentes do Kubernetes. Para obter mais informações sobre os gerenciadores de filas de várias instâncias, consulte [“Alta disponibilidade para IBM MQ em contêineres”](#) na página 16.

O exemplo fornece um YAML de recurso customizado definindo um gerenciador de filas de várias instâncias com armazenamento persistente e configurado com TLS. Depois de implementar o gerenciador de filas no OCP, você simula a falha do pod do gerenciador de filas ativas. Você vê a recuperação automática ocorrer e prova que ela foi bem-sucedida colocando e recebendo mensagens após a falha.

exemplo

Criar uma chave privada TLS e certificados para servidor MQ

Esta seção documenta como criar um certificado autoassinado para o gerenciador de filas e como incluir o certificado em um banco de dados de chaves para agir como o armazenamento confiável para o cliente. Se você já tiver uma chave privada e um certificado, será possível usá-los. Observe que o uso de certificados autoassinados é necessário apenas para propósitos de desenvolvimento.

Para criar uma chave privada autoassinada e um certificado público no diretório atual, execute o comando a seguir:

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650
-out tls.crt
```

Incluir a chave pública do gerenciador de filas em um banco de dados de chaves do cliente

Um banco de dados de chaves do cliente é usado como o armazenamento confiável para o aplicativo cliente.

Crie o banco de dados de chaves de clientes:

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

Inclua a chave pública gerada anteriormente no banco de dados de chaves de clientes:

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii
-stashed
```

Criar um segredo contendo certificados TLS para implementação do gerenciador de filas

Para que o seu gerenciador de filas possa referenciar e aplicar a chave e o certificado, crie um segredo de TLS do Kubernetes, referenciando os arquivos criados acima. Ao fazer isso, assegure-se de que está no namespace criado antes de iniciar esta tarefa.

```
oc create secret tls example-mi-secret --key="tls.key" --cert="tls.crt"
```

Criar um mapa de configuração contendo comandos MQSC

Crie um mapa de configuração do Kubernetes contendo os comandos MQSC para criar uma nova fila e um Canal SVRCONN e para incluir um registro de autenticação de canal que permita o acesso ao canal, bloqueando somente os usuários chamados *nobody*.

Observe que essa abordagem deve ser usada apenas para fins de desenvolvimento.

Assegure-se de estar no namespace criado anteriormente (consulte [“Antes de Começar”](#) na página 103), em seguida, insira o YAML a seguir na IU do OCP ou usando a linha de comandos:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-mi-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') DEFPSIST(YES) REPLACE
    DEFINE CHANNEL(MIQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    SET CHLAUTH(MIQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

Configurar roteamento

Se você estiver usando um IBM MQ client ou kit de ferramentas no IBM MQ 9.2.1 ou mais recente, será possível configurar o roteamento para o gerenciador de filas usando um arquivo de configuração do gerenciador de filas (um arquivo INI). No arquivo, você configura a variável *OutboundSNI* para rotear com base no nome do host, em vez de no nome do canal.

Crie um arquivo no diretório no qual você está executando comandos, chamado `mqclient.ini`, contendo o texto a seguir:

```
##* Module Name: mqclient.ini ##*
##* Type : IBM MQ MQI client configuration file ##*
##* Function : Define the configuration of a client ##*
##* ##*
##***** ##*
##* Notes : ##*
##* 1) This file defines the configuration of a client ##*
##* ##*
##***** ##*
SSL:
  OutboundSNI=HOSTNAME
```

Nota: Não mude nenhum valor nesta página. Por exemplo a sequência HOSTNAME deve ser deixada como está.

Para obter detalhes adicionais, consulte [Sub-rotina SSL do arquivo de configuração do cliente](#).

Se você estiver usando um IBM MQ client ou kit de ferramentas anterior a IBM MQ 9.2.1, será necessário criar uma rota OCP em vez do arquivo de configuração anterior. Siga as etapas em [Nota 1: criando uma rota](#).

Implementar o gerenciador de filas

Importante: neste exemplo usamos a variável `MQSNOAUT` para desativar a autorização no gerenciador de filas, o que nos permite focar nas etapas necessárias para conectar um cliente usando o TLS. Isso não é recomendado em uma implementação de produção do IBM MQ, pois ela faz com que todos os aplicativos que se conectam tenham poderes administrativos integrais, sem nenhum mecanismo para diminuir as permissões para aplicativos individuais.

Copie e atualize o YAML a seguir.

- Assegure-se de que a licença correta esteja especificada. Consulte [Referência de licenciamento para mq.ibm.com/v1beta1](#).
- Aceite a licença alterando `false` para `true`.
- Se você estiver usando o IBM Cloud File Storage, consulte [Nota 3: usando o IBM Cloud File Storage](#)

YAML do recurso customizado do gerenciador de filas:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: miexample
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  queueManager:
    name: MIEXAMPLE
    availability:
      type: MultiInstance
  mqsc:
    - configMap:
        name: example-mi-configmap
        items:
          - tls.mqsc
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
          name: qmgr
  version: 9.2.5.0-r3
  web:
    enabled: true
  pki:
    keys:
      - name: example
        secret:
          secretName: example-mi-secret
          items:
            - tls.key
            - tls.crt
```

Assegurando que você esteja no namespace criado anteriormente, implemente o YAML atualizado na IU do OCP, usando a linha de comandos ou usando o IBM Cloud Pak for Integration Platform Navigator.

Validação

Após um breve atraso, o gerenciador de filas de várias instâncias deve ser configurado e disponibilizado para uso. Nesta seção, nós validamos que o gerenciador de filas se comporta conforme o esperado.

Confirmar se o gerenciador de filas está sendo executado

O gerenciador de filas agora está sendo implementado. Confirme se está no estado Running antes de prosseguir. Por exemplo:

```
oc get qmgr miexample
```

Teste a conexão com o gerenciador de filas

Para confirmar que o gerenciador de filas está configurado para comunicação TLS unidirecional, use os aplicativos de amostra **amqspu**tc e **amqsget**tc:

Localize o nome do host do gerenciador de filas

Para encontrar o nome do host do gerenciador de filas para a rota `miexample-ibm-mq-qm`, execute o seguinte comando. O nome do host é retornado no campo HOST.

```
oc get routes miexample-ibm-mq-qm
```

Especificar os detalhes do gerenciador de filas

Crie um arquivo CCDT .JSON que especifica os detalhes do gerenciador de filas. Substitua o valor do host pelo nome do host retornado pela etapa anterior.

```
{
  "channel":
  [
    {
      "name": "MIQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<host from previous step>",
            "port": 443
          }
        ],
        "queueManager": "MIEXAMPLE"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

Exportar variáveis de ambiente

Exporte as variáveis de ambiente a seguir, da maneira apropriada para o seu sistema operacional. Essas variáveis serão lidas por **amqspu**tc e **amqsget**tc.

Atualize o caminho para os arquivos no seu sistema:

```
export MQCCDTURL='<full_path_to_file>/CCDT.JSON'
export MQSSLKEYR='<full_path_to_file>/clientkey'
```

Colocar mensagens na fila

Execute o seguinte comando:

```
amqspu tc EXAMPLE.QUEUE MIEXAMPLE
```

Se a conexão com o gerenciador de filas for bem-sucedida, a resposta a seguir será a saída:
`target queue is EXAMPLE.QUEUE`

Coloque várias mensagens na fila, inserindo um texto e pressionando **Enter** cada vez.

Para concluir, pressione **Enter** duas vezes.

Recuperar as mensagens da fila

Execute o seguinte comando:

```
amqsgetc EXAMPLE.QUEUE MIEXAMPLE
```

As mensagens que foram incluídas na etapa anterior foram consumidas e uma saída foi obtida.

Depois de alguns segundos, você verá o resultado do comando.

Forçar o pod ativo a falhar

Para validar a recuperação automática do gerenciador de filas de várias instâncias, simule uma falha de pod:

Visualizar os pods ativos e em espera

Execute o seguinte comando:

```
oc get pods
```

Observe que, no campo **READY**, o pod ativo retorna o valor 1/1, enquanto que o pod de espera retorna o valor 0/1.

Excluir o pod ativo

Execute o comando a seguir, especificando o nome completo do pod ativo:

```
oc delete pod miexample-ibm-mq-<value>
```

Visualizar o status do pod novamente

Execute o seguinte comando:

```
oc get pods
```

Visualizar o log do pod em espera

Execute o comando a seguir, especificando o nome completo do pod em espera:

```
oc logs miexample-ibm-mq-<value>
```

Você deverá ver a seguinte mensagem:

```
IBM MQ queue manager 'MIEXAMPLE' becoming the active instance.
```

Colocar e receber mensagens novamente

Depois que o pod de espera se tornar o pod ativo (ou seja, depois que o valor do campo **READY** se tornar 1/1), use os comandos a seguir novamente, conforme descrito anteriormente, para colocar mensagens no gerenciador de filas e, em seguida, recuperar as mensagens do gerenciador de filas:

```
amqsputc EXAMPLE.QUEUE MIEXAMPLE
```

```
amqsgetc EXAMPLE.QUEUE MIEXAMPLE
```

Parabéns, você implementou com sucesso um gerenciador de filas de várias instâncias e mostrou que ele pode se recuperar automaticamente de uma falha de pod.

Informações Adicionais

Nota 1: criando uma rota

Se você estiver usando um IBM MQ client ou kit de ferramentas anterior ao IBM MQ 9.2.1, será necessário criar uma rota de OCP.

Para criar a rota, assegure-se de estar no namespace criado anteriormente (consulte [“Antes de Começar”](#) na página 103), em seguida, insira o YAML a seguir na IU do OCP, ou usando a linha de comandos:

```
apiVersion: route.openshift.io/v1
kind: Route
```

```
metadata:
  name: example-mi-route
spec:
  host: miqmchl.chl.mq.ibm.com
  to:
    kind: Service
    name: miexample-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Observe que o Red Hat OpenShift Container Platform Router usa o SNI para solicitações de roteamento para o gerenciador de filas IBM MQ. Se você mudar o nome do canal especificado no mapa de configuração contendo comandos MQSC, também deverá mudar o campo `host` aqui e no arquivo CCDT .JSON que especifica os detalhes do gerenciador de filas. Para obter informações adicionais, consulte [“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift”](#) na página 108.

Nota 2: usando uma classe de armazenamento não padrão

Este exemplo espera que uma classe de armazenamento padrão tenha sido configurada no OCP, portanto, nenhuma informação de armazenamento é necessária no YAML do recurso customizado do gerenciador de filas. Se você não tiver uma classe de armazenamento configurada como padrão ou deseja usar uma classe de armazenamento diferente, inclua `defaultClass: <storage_class_name>` em `spec.queueManager.storage`.

O nome da classe de armazenamento deve corresponder exatamente ao nome de uma classe de armazenamento que existe em seu sistema OCP. Ou seja, deve corresponder ao nome retornado pelo comando `oc get storageclass`. Ele também deve oferecer suporte a `ReadWriteMany`. Para obter informações adicionais, consulte [“Considerações de armazenamento para o IBM MQ Operator”](#) na página 11.

Nota 3: usando o IBM Cloud File Storage

Em algumas situações, por exemplo ao usar IBM Cloud File Storage, também será necessário especificar o campo `securityGroups` no YAML do recurso customizado do gerenciador de filas. Por exemplo, incluindo o campo filho a seguir diretamente em `spec`:

```
securityContext:
  supplementalGroups: [99]
```

Para obter informações adicionais, consulte [“Considerações de armazenamento para o IBM MQ Operator”](#) na página 11.

OpenShift CP4I CD Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift

É necessária uma rota do Red Hat OpenShift para conectar um aplicativo a um gerenciador de filas do IBM MQ de fora de um cluster do Red Hat OpenShift . Deve-se ativar o TLS no gerenciador de filas e no aplicativo cliente do IBM MQ , porque o SNI está disponível apenas no protocolo TLS quando um protocolo TLS 1.2 ou superior é usado O Red Hat OpenShift Container Platform Router usa o SNI para solicitações de roteamento para o gerenciador de filas IBM MQ.

Sobre esta tarefa



Atenção: Esse documento se aplica às versões 9.2.1 Continuous Delivery e mais recentes dos clientes do IBM MQ. Se o seu cliente estiver usando a versão 9.2.0 Long Term Support ou anterior, consulte a página de documentação do IBM MQ [9.1 Conectando-se a um gerenciador de fila implementado em um cluster do Red Hat OpenShift](#).

V 9.2.1 A configuração necessária do Red Hat OpenShift Route depende do comportamento `Server Name Indication` (SNI) do aplicativo cliente. O IBM MQ suporta duas configurações de cabeçalho SNI diferentes dependendo da configuração e do tipo de cliente. Um cabeçalho SNI é configurado para o nome do host do destino do cliente ou, como alternativa, configurado para o nome do canal IBM MQ .

Para obter informações sobre como o IBM MQ mapeia um nome de canal para um nome do host, consulte [Como o IBM MQ fornece capacidade de vários recursos](#).

V 9.2.1 Se um cabeçalho SNI é configurado para um nome do canal IBM MQ ou um nome do host é controlado usando o atributo **OutboundSNI**. Os valores possíveis são OutboundSNI=CHANNEL (o valor padrão) ou OutboundSNI=HOSTNAME. Para obter mais informações, consulte a [sub-rotina SSL do arquivo de configuração do cliente](#). Observe que CHANNEL e HOSTNAME são os valores exatos que você usa; eles não são nomes de variáveis que você substitui por um nome de canal ou nome de host real.

V 9.2.1

Comportamentos do cliente com configurações diferentes do OutboundSNI

Se **OutboundSNI** for configurado como HOSTNAME, os clientes a seguir configurarão um SNI de nome do host desde que um nome do host seja fornecido no nome da conexão:

- Clientes C
- Os clientes do .NET em modo não gerenciado
- Clientes Java/JMS

Se **OutboundSNI** for configurado como HOSTNAME e um endereço IP for usado no nome de conexão, os clientes a seguir enviarão um cabeçalho SNI em branco:

- Clientes C
- Os clientes do .NET em modo não gerenciado
- Java/JMS Clientes (que não pode fazer uma pesquisa DNS reversa do nome do host)

Se **OutboundSNI** for configurado como CHANNEL ou não configurado, um nome de canal do IBM MQ será usado em vez disso e sempre será enviado, independentemente se um nome do host ou nome de conexão de endereço IP tenha sido usado.

Os tipos de clientes a seguir não suportam a configuração de um cabeçalho SNI para um nome de canal do IBM MQ. Por isso, sempre tente configurar o cabeçalho SNI para um nome do host independentemente da configuração **OutboundSNI**:

- Clientes do AMQP
- Clientes do XR
- Clientes .NET no modo gerenciado (Antes do IBM MQ 9.2.0 Fix Pack 4 para Long Term Support e antes do IBM MQ 9.2.3 para o Continuous Delivery.)

V 9.2.0.4 V 9.2.3

No IBM MQ 9.2.0 Fix Pack 4 para o Long Term Support e IBM MQ 9.2.3 para o Continuous Delivery, o IBM MQ cliente .NET gerenciado foi atualizado para configurar SERVERNAME para o respectivo nome do host se a propriedade **OutboundSNI** é configurada como HOSTNAME, o que permite que um IBM MQ cliente .NET gerenciado se conecte a um gerenciador de filas usando rotas do Red Hat OpenShift. Observe que, em IBM MQ 9.2.0 Fix Pack 4, a propriedade **OutboundSNI** é incluída e suportada apenas no arquivo `mqclient.ini`; não é possível configurar a propriedade a partir do aplicativo .NET.

V 9.2.5

Se um aplicativo cliente se conectar a um gerenciador de filas implementado em um cluster do Red Hat OpenShift por meio de IBM MQ Internet Pass-Thru (MQIPT), MQIPT pode ser configurado para configurar o SNI para o nome do host usando a propriedade [SSLClientOutboundSNI](#) na definição de rota.

OutboundSNI, diversos certificados e rotas do Red Hat OpenShift

O IBM MQ usa o cabeçalho SNI para fornecer a funcionalidade de vários certificados. Se um aplicativo estiver se conectando a um canal do IBM MQ configurado para usar um certificado diferente por

meio do campo CERTLABL, o aplicativo deverá se conectar a uma configuração **OutboundSNI** de CHANNEL.

Se a sua configuração de rota do Red Hat OpenShift requerer um HOSTNAME SNI, não será possível usar a funcionalidade de vários certificados do IBM MQ e não será possível definir uma configuração CERTLABL em qualquer objeto do canal IBM MQ.

Se um aplicativo com uma configuração **OutboundSNI** diferente de CHANNEL se conectar a um canal com um rótulo certificado configurado, o aplicativo será rejeitado com um MQRC_SSL_INITIALIZATION_ERROR e uma mensagem AMQ9673 será impressa nos logs de erro do gerenciador de filas.

Para obter mais informações sobre como o IBM MQ fornece a funcionalidade de diversos certificados, consulte [Como IBM MQ fornece a capacidade de diversos certificados](#)

Exemplo

Aplicativos clientes que configuram o SNI para o canal MQ requerem que uma nova Rota Red Hat OpenShift seja criada para cada canal ao qual você deseja se conectar. Também é necessário usar nomes de canais exclusivos em todo o cluster Red Hat OpenShift Container Platform para permitir o roteamento para o gerenciador de filas correto.

É importante que os nomes de canais MQ não terminem em uma letra minúscula devido à maneira como o IBM MQ mapeia nomes de canais para cabeçalhos do SNI.

Para determinar o nome do host necessário para cada uma das suas novas Rotas Red Hat OpenShift, é necessário mapear cada nome do canal para um endereço do SNI. Consulte [Como o IBM MQ fornece capacidade de vários certificados](#) para obter mais informações.

Deve-se então criar uma nova Rota Red Hat OpenShift para cada canal, aplicando o seguinte yaml em seu cluster:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: <provide a unique name for the Route>
  namespace: <the namespace of your MQ deployment>
spec:
  host: <SNI address mapping for the channel>
  to:
    kind: Service
    name: <the name of the Kubernetes Service for your MQ deployment (for example "<Queue Manager Name>-ibm-mq")>
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Configurando os detalhes de conexão do aplicativo cliente

É possível determinar o nome do host a ser usado para a conexão do cliente executando o comando a seguir:

```
oc get route <Name of hostname based Route (for example "<Queue Manager Name>-ibm-mq-qm")>
-n <namespace of your MQ deployment> -o jsonpath="{.spec.host}"
```

A porta para a conexão do seu cliente deve ser configurada para a porta usada pelo Red Hat OpenShift Container Platform Router - normalmente 443.

Tarefas relacionadas

[“Conectando-se ao IBM MQ Console implementado em um cluster do Red Hat OpenShift” na página 115](#)
[Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster Red Hat OpenShift Container Platform.](#)

Dashboard

A capacidade de rastrear transações por meio do IBM Cloud Pak for Integration é fornecida pelo Painel de operações.

Sobre esta tarefa

A ativação da integração com o Painel de operações instala uma saída de API do MQ em seu gerenciador de filas. A saída da API enviará ao armazenamento de dados do Painel de operações os dados de rastreo sobre as mensagens que estão fluindo por meio do gerenciador de filas.

Observe que apenas as mensagens que são enviadas usando as ligações do cliente MQ são rastreadas.

Além disso, observe que, para as versões do IBM MQ Operator anteriores à 1.5, quando o rastreo é ativado, as imagens de rastreo do agente e do coletor implementadas ao lado do gerenciador de filas são sempre as versões mais recentes disponíveis, o que pode introduzir uma incompatibilidade ao não usar a versão mais recente do IBM Cloud Pak for Integration.

Procedimento

1. Implemente um gerenciador de filas com o rastreo ativado

Por padrão, o recurso de rastreo é desativado.

Ao realizar uma implementação usando o IBM Cloud Pak for Integration Platform Navigator, é possível ativar o rastreo durante ela configurando **Ativar rastreo** como **Ativo** e **Namespace do rastreo** como o namespace no qual o Operations Dashboard está instalado. Para obter mais informações sobre a implementação de um gerenciador de filas, consulte [“Implementando um gerenciador de filas usando o IBM Cloud Pak for Integration Platform Navigator”](#) na página 82

Se você estiver implementando usando a CLI do Red Hat OpenShift ou console da web do Red Hat OpenShift, então você pode ativar o rastreo com o seguinte trecho YAML:

```
spec:
  tracing:
    enabled: true
    namespace: <Operations_Dashboard_Namespace
```

Importante: o gerenciador de filas não será iniciado até que o MQ tenha sido registrado no Painel de operações (consulte a próxima etapa).

Note que quando esse recurso estiver ativado, ele executará dois contêineres de sidecar ("Agente" e "Coletor"), além do contêiner do gerenciador de filas. As imagens para esses contêineres de sidecar estarão disponíveis no mesmo registro da imagem principal do MQ e usarão a mesma política de extração e o mesmo segredo de extração. Há configurações adicionais disponíveis para definir limites de CPU e de memória.

2. Se esta for a primeira vez que um gerenciador de filas com integração do Operations Dashboard foi implementado neste namespace, então você precisa [Registrar-se](#) com o Operations Dashboard.

O registro cria um objeto Segredo que o pod do gerenciador de filas precisa para ser iniciado com sucesso.

Implementando ou fazendo upgrade da integração do IBM MQ 9.2.2 ou 9.2.3 com Operations Dashboard no IBM Cloud Pak for Integration 2021.4

Cada versão do IBM MQ está associada a uma versão específica do agente do Painel de operações e dos componentes do coletor, que são implementados junto com um gerenciador de filas. O IBM Cloud Pak for Integration 2021.4.1 apresenta uma alteração que faz com que os componentes mais antigos do agente e do coletor não funcionem com o Painel de operações. Para corrigir isso, deve-se substituir a versão do agente do Operations Dashboard e as imagens do coletor que você usa, ao usar o IBM MQ 9.2.2 ou 9.2.3.

Implementando um novo gerenciador IBM MQ 9.2.2 ou 9.2.3

Ao usar o IBM Cloud Pak for Integration 2021.4.1 com o IBM MQ 9.2.2 ou 9.2.3, é necessário substituir o agente do PAInel de operações e as imagens do coletor para as versões 2.4 em seu YAML QueueManager. Por exemplo:

```
spec:
  tracing:
    agent:
      image: cp.icr.io/cp/icp4i/od/icp4i-od-
agent@sha256:27a211f0f78eff765d1f9520e0f9841f902600bb556827477b206e209cb44d20
    collector:
      image: cp.icr.io/cp/icp4i/od/icp4i-od-
collector@sha256:dc70b1341b23dc72642ce68809811f9db0e8a0c46bda2508e8eb3d4035e04f4b
```

Se você não fizer isso, seu Pod QueueManager ficará travado no estado Pending. Ao fazer upgrade para o IBM MQ 9.2.4, é possível remover essas substituições.

Fazendo upgrade para IBM Cloud Pak for Integration 2021.4.1

Nota: Se você estiver mantendo o seu gerenciador de filas IBM MQ 9.2.2 ou 9.2.3, não conclua a etapa 3.

1. Atualize seu QueueManager para substituir as imagens do agente e do coletor, conforme descrito anteriormente.
2. Faça upgrade de seus operadores do IBM Cloud Pak for Integration, incluindo o Operations Dashboard e o IBM MQ Operator, conforme descrito em [“Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas”](#) na página 71.
3. (Opcional) Para fazer o upgrade para IBM MQ 9.2.4 ou posterior, atualize o seu QueueManager para usar o `.spec.version` para a sua versão do IBM MQ, depois remova a substituição das imagens do agente e do coletor.

Construindo uma imagem com arquivos customizados de MQSC e INI, usando a CLI do Red Hat OpenShift

Use um Red Hat OpenShift Container Platform Pipeline para criar uma nova imagem de contêiner do IBM MQ, com os arquivos MQSC e INI que você deseja que sejam aplicados a gerenciadores de filas usando essa imagem. Essa tarefa deve ser concluída por um administrador do projeto

Antes de começar

É necessário instalar a [Interface da linha de comandos do Red Hat OpenShift Container Platform](#).

Efetue login em seu cluster usando `cloudctl login` (para o IBM Cloud Pak for Integration) ou `oc login`.

Se você não tem um Red Hat OpenShift Secret para o IBM Entitled Registry em seu projeto Red Hat OpenShift, então siga as etapas para o [“Preparando seu projeto Red Hat OpenShift para IBM MQ”](#) na página 79.

Procedimento

1. Crie um ImageStream

Um fluxo de imagem e suas tags associadas fornecem uma abstração para referenciar imagens de contêiner de dentro do Red Hat OpenShift Container Platform. O fluxo de imagem e suas tags permitem que você veja quais imagens estão disponíveis e que você se certifique de estar usando a imagem específica necessária, mesmo que uma mudança ocorra na imagem no repositório.

```
oc create imagestream mymq
```

2. Crie um BuildConfig para sua nova imagem

Um BuildConfig permitirá construções para sua nova imagem, que será baseada nas imagens oficiais da IBM, mas incluirá quaisquer arquivos MQSC ou INI que você deseja que sejam executados na inicialização do contêiner.

a) Crie um arquivo YAML definindo o recurso BuildConfig

Por exemplo, crie um arquivo chamado "mq-build-config.yaml" com o conteúdo a seguir:

```
apiVersion: build.openshift.io/v1
kind: BuildConfig
metadata:
  name: mymq
spec:
  source:
    dockerfile: |-
      FROM cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3
      RUN printf "DEFINE QLOCAL(foo) REPLACE\n" > /etc/mqm/my.mqsc \
        && printf "Channels:\n\tMQIBindType=FASTPATH\n" > /etc/mqm/my.ini
      LABEL summary "My custom MQ image"
  strategy:
    type: Docker
    dockerStrategy:
      from:
        kind: "DockerImage"
        name: "cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

Será necessário substituir os dois locais em que o IBM MQ base é mencionado, para apontar para a imagem base correta da versão e da correção que você deseja usar (consulte [“Histórico de liberação para o IBM MQ Operator”](#) na página 20 para obter detalhes). Conforme as correções forem aplicadas, será necessário repetir essas etapas para reconstruir sua imagem.

Este exemplo cria uma nova imagem com base na imagem oficial da IBM e inclui arquivos chamados "my.mqsc" e "my.ini" no diretório /etc/mqm. Quaisquer arquivos MQSC ou INI localizados nesse diretório serão aplicados pelo contêiner na inicialização. Os arquivos INI são aplicados usando a opção **crtmqm -ii** e mesclados com os arquivos INI existentes. Os arquivos MQSC são aplicados em ordem alfabética.

É importante que seus comandos MQSC sejam repetidos, já que eles serão executados *a cada vez* que o gerenciador de filas for inicializado. Isso geralmente significa incluir o parâmetro REPLACE em quaisquer comandos DEFINE e incluir o parâmetro IGNSTATE (YES) em quaisquer comandos START ou STOP.

b) Aplique o BuildConfig no servidor.

```
oc apply -f mq-build-config.yaml
```

3. Execute uma construção para criar sua imagem

a) Inicie a construção

```
oc start-build mymq
```

Você deve ver uma saída semelhante à seguinte:

```
build.build.openshift.io/mymq-1 started
```

b) Verifique o status da construção

Por exemplo, é possível executar o comando a seguir, usando o identificador de construção retornado na etapa anterior:

```
oc describe build mymq-1
```

4. Implemente um gerenciador de filas, usando sua nova imagem

Siga as etapas descritas em [“Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform”](#) na página 81, incluindo sua nova imagem personalizada no YAML. É possível incluir o seguinte fragmento de YAML em seu YAML QueueManager normal, em que *my-namespace* é o projeto/namespace Red Hat OpenShift que você está usando e *image* é o nome da imagem que você criou anteriormente (por exemplo, "mymq:latest-amd64"):

```
spec:
  queueManager:
    image: image-registry.openshift-image-registry.svc:5000/my-namespace/my-image
```

Tarefas relacionadas

[“Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform”](#) na página 81

Use o recurso customizado QueueManager para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform.

Incluindo anotações e rótulos customizados para recursos do gerenciador de filas

Inclua anotações e rótulos customizados aos metadados do QueueManager.

Sobre esta tarefa

As anotações e rótulos customizados são incluídos em todos os recursos, exceto PVCs. Se uma anotação ou rótulo customizado corresponde a uma chave existente, o valor configurado pelo IBM MQ Operator é usado.

Procedimento

- Inclua anotações customizadas.

Para incluir anotações customizadas aos recursos do gerenciador de filas, incluindo o pod, inclua as anotações em metadata. Por exemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    annotationKey: "value"
```

- Inclua rótulos customizados.

Para incluir rótulos customizados aos recursos do gerenciador de filas, incluindo o pod, inclua os rótulos em metadata. Por exemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  labels:
    labelKey: "value"
```

Desativando as verificações de webhook de tempo de execução

As verificações de webhook de tempo de execução asseguram que as classes de armazenamento sejam viáveis para o seu gerenciador de filas. Elas são desativadas para melhorar o desempenho ou porque não são válidas no seu ambiente.

Sobre esta tarefa

As verificações de webhook de tempo de execução são feitas na configuração do gerenciador de filas. Eles verificam se as classes de armazenamento são adequadas para o seu tipo de gerenciador de filas selecionado.

Você pode optar por desativar essas verificações para diminuir o tempo necessário para a criação do gerenciador de filas ou porque as verificações não são válidas para o seu ambiente específico.

Nota: Depois de desativar as verificações do webhook de tempo de execução, todos os valores de classe de armazenamento são permitidos. Isso pode resultar em um gerenciador de filas quebrado.

O suporte para verificações de tempo de execução foi introduzido no IBM MQ Operator 1.2.

Procedimento

- Desative verificações de webhook de tempo de execução.

Inclua a anotação a seguir em metadata. Por exemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.cp4i/disable-webhook-runtime-checks" : "true"
```

OpenShift

CP4I

Operando o IBM MQ usando o IBM MQ Operator

Sobre esta tarefa

Procedimento

- [“Preparando seu projeto Red Hat OpenShift para IBM MQ”](#) na página 79.
- [“Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform”](#) na página 81.

OpenShift

CP4I

Conectando-se ao IBM MQ Console implementado em um cluster do Red Hat OpenShift

Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster Red Hat OpenShift Container Platform.

Sobre esta tarefa

A URL do IBM MQ Console pode ser localizada na página de detalhes do QueueManager no console da web do Red Hat OpenShift ou no IBM Cloud Pak for Integration Platform Navigator. Alternativamente, ele pode ser encontrado por meio da CLI do Red Hat OpenShift executando o seguinte comando:

```
oc get queuemanager <QueueManager Name> -n <namespace of your MQ deployment> --output jsonpath='{.status.adminUiUrl}'
```

Se você estiver usando uma licença IBM Cloud Pak for Integration, então o console da web IBM MQ está configurado para usar o IBM Cloud Pak Identity and Access Manager (IAM). O componente IAM pode já ter sido configurado por seu administrador de cluster. No entanto, se esta for a primeira vez que o IAM foi usado em seu cluster do Red Hat OpenShift, então você precisará recuperar a senha inicial admin. Para obter mais informações, consulte [Obtendo a senha inicial do administrador](#).

Se você estiver usando uma licença do IBM MQ, o console da web do MQ não será pré-configurado e será necessário configurá-lo por conta própria. Para obter mais informações, consulte [Configurando usuários e funções](#).

Tarefas relacionadas

“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift” na página 108

É necessária uma rota do Red Hat OpenShift para conectar um aplicativo a um gerenciador de filas do IBM MQ de fora de um cluster do Red Hat OpenShift . Deve-se ativar o TLS no gerenciador de filas e no aplicativo cliente do IBM MQ , porque o SNI está disponível apenas no protocolo TLS quando um protocolo TLS 1.2 ou superior é usado O Red Hat OpenShift Container Platform Router usa o SNI para solicitações de roteamento para o gerenciador de filas IBM MQ.

Fornecendo permissões para o IBM MQ Console usando o IBM Cloud Pak IAM

As permissões para o IBM MQ Console são gerenciadas por meio do IBM Cloud Pak Administration Hub e não do IBM Cloud Pak for Integration Platform Navigator. O IBM MQ não usa as permissões de "Automação" fornecidas pelo IBM Cloud Pak for Integration, mas, em vez disso, usa as permissões básicas ativadas pelo IBM Cloud Pak Identity and Access Manager (IAM).

Procedimento

1. Abra o console de administração do IBM Cloud Pak.
No IBM Cloud Pak for Integration Platform UI, clique no alternador do Cloud Pak (ícone de 9 pontos) no canto superior direito da barra de ferramentas, em seguida, clique no painel **IBM Cloud Pak** .
2. No menu de navegação no canto superior esquerdo, selecione **Identidade e acesso**, em seguida, selecione **IDs de equipes e serviços**.
3. Crie uma equipe e, em seguida, inclua usuários nela.
 - a) Selecione **Criar equipe**..
 - b) Insira um nome de equipe e, em seguida, selecione o domínio de segurança para os usuários que você deseja gerenciar
 - c) Procurar usuários.
Esses usuários já devem existir no provedor de identidade.
 - d) Quando você localizar cada usuário, dê a ele uma função Deve ser "Administrador" ou "Administrador de Cluster", para administrar IBM MQ usando o IBM MQ Console.
4. Inclua cada usuário em um namespace.
 - a) Selecione a equipe para editá-la,
 - b) Selecione **Recursos > Gerenciar Recursos**..
 - c) Selecione os namespaces que você deseja que essa equipe administre Eles podem ser quaisquer namespaces com um gerenciador de filas

Monitorando ao usar o IBM MQ Operator

Os gerenciadores de filas gerenciados pelo IBM MQ Operator podem produzir métricas compatíveis com o Prometheus.

É possível visualizar essas métricas usando a Red Hat OpenShift Container Platform (OCP) [monitoring stack](#). Abra a guia **Métricas** em OCP, em seguida, clique em **Observar > Métricas**.

As métricas do gerenciador de filas são ativadas por padrão, mas podem ser desativadas configurando **.spec.metrics.enabled** como `false`.

Prometheus é um banco de dados de séries temporais e um mecanismo de avaliação de regras para métricas. Os contêineres do IBM MQ expõem um terminal de métricas que pode ser consultado pelo Prometheus. As métricas são geradas por meio dos tópicos do sistema MQ para monitoramento e rastreamento de atividade.

O Red Hat OpenShift Container Platform inclui uma pilha de monitoramento pré-configurada, pré-instalada e de autoatualização que usa um servidor Prometheus. A pilha de monitoramento do Red Hat OpenShift Container Platform precisa ser configurada para monitorar projetos definidos pelo usuário. Para

obter mais informações, consulte [Ativando o monitoramento para projetos definidos pelo usuário](#). O IBM MQ Operator cria um ServiceMonitor quando você cria um QueueManager com métricas ativadas, o que o Operador do Prometheus pode descobrir.

Em versões mais antigas do IBM Cloud Pak for Integration, também era possível usar o serviço [IBM Cloud Platform Monitoring](#) para fornecer um servidor Prometheus como alternativa.

Métricas publicadas ao usar o IBM MQ Operator

Os contêineres gerenciadores de filas podem publicar métricas compatíveis com o Red Hat OpenShift Monitoring.

Métrica	Tipo	Descrição
ibmmq_qmgr_commit_total	counter	Contagem de confirmações
ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage	gauge	Carregamento da CPU - média de quinze minutos
ibmmq_qmgr_cpu_load_five_minute_average_percentage	gauge	Carregamento da CPU - média de cinco minutos
ibmmq_qmgr_cpu_load_one_minute_average_percentage	gauge	Carregamento da CPU - média de um minuto
ibmmq_qmgr_destructive_get_bytes_total	counter	Obtenção destrutiva de total de intervalo - contagem de bytes
ibmmq_qmgr_destructive_get_total	counter	Obtenção destrutiva de total de intervalo - contagem
ibmmq_qmgr_durable_subscription_alter_total	counter	Contagem de assinaturas duráveis de alteração
ibmmq_qmgr_durable_subscription_create_total	counter	Contagem de assinaturas duráveis de criação
ibmmq_qmgr_durable_subscription_delete_total	counter	Contagem de assinaturas duráveis de exclusão
ibmmq_qmgr_durable_subscription_resume_total	counter	Contagem de assinaturas duráveis de continuação
ibmmq_qmgr_errors_file_system_free_space_percentage	gauge	Sistema de arquivos de erros do MQ - espaço livre
ibmmq_qmgr_errors_file_system_in_use_bytes	gauge	Sistema de arquivos de erros do MQ - bytes em uso
ibmmq_qmgr_expired_message_total	counter	Contagem de mensagens expiradas

Métrica	Tipo	Descrição
ibmmq_qmgr_failed_browse_total	counter	Contagem de procuras com falha
ibmmq_qmgr_failed_mqcb_total	counter	Contagem de MQCB com falha
ibmmq_qmgr_failed_mqclose_total	counter	Contagem de MQCLOSE com falha
ibmmq_qmgr_failed_mqconn_mqconnx_total	counter	Contagem de MQCONN/MQCONNX com falha
ibmmq_qmgr_failed_mqget_total	counter	MQGET com falha - contagem
ibmmq_qmgr_failed_mqinq_total	counter	Contagem de MQINQ com falha
ibmmq_qmgr_failed_mqopen_total	counter	Contagem de MQOPEN com falha
ibmmq_qmgr_failed_mqput1_total	counter	Contagem de MQPUT1 com falha
ibmmq_qmgr_failed_mqput_total	counter	Contagem de MQPUT com falha
ibmmq_qmgr_failed_mqset_total	counter	Contagem de MQSET com falha
ibmmq_qmgr_failed_mqsubrq_total	counter	Contagem de MQSUBRQ com falha
ibmmq_qmgr_failed_subscription_create_alter_resume_total	counter	Contagem de assinaturas de criação/alteração/continuação com falha
ibmmq_qmgr_failed_subscription_delete_total	counter	Contagem de falhas de exclusão de assinatura
ibmmq_qmgr_failed_topic_mqput_mqput1_total	counter	Contagem de MQPUT/MQPUT1 de tópico com falha
ibmmq_qmgr_fdc_files	gauge	Contagem de arquivos FDC do MQ
ibmmq_qmgr_log_file_system_in_use_bytes	gauge	Sistema de arquivos de log - bytes em uso
ibmmq_qmgr_log_file_system_max_bytes	gauge	Sistema de arquivos de log - máximo de bytes
ibmmq_qmgr_log_in_use_bytes	gauge	Log - bytes em uso
ibmmq_qmgr_log_logical_written_bytes_total	counter	Log - bytes lógicos gravados

Métrica	Tipo	Descrição
ibmmq_qmgr_log_max_bytes	gauge	Log - máximo de bytes
ibmmq_qmgr_log_physical_written_bytes_total	counter	Log - bytes físicos gravados
ibmmq_qmgr_log_primary_space_in_use_percentage	gauge	Log - espaço primário atual em uso
ibmmq_qmgr_log_workload_primary_space_utilization_percentage	gauge	Log - utilização de espaço primário de carga de trabalho
ibmmq_qmgr_log_write_latency_seconds	gauge	Log - latência de gravação
ibmmq_qmgr_log_write_size_bytes	gauge	Log - tamanho de gravação
ibmmq_qmgr_mqcb_total	counter	Contagem de MQCB
ibmmq_qmgr_mqclose_total	counter	Contagem de MQCLOSE
ibmmq_qmgr_mqconn_mqconnx_total	counter	Contagem de MQCONN/MQCONNX
ibmmq_qmgr_mqctl_total	counter	Contagem de MQCTL
ibmmq_qmgr_mqdisc_total	counter	Contagem de MQDISC
ibmmq_qmgr_mqinq_total	counter	Contagem de MQINQ
ibmmq_qmgr_mqopen_total	counter	Contagem de MQOPEN
ibmmq_qmgr_mqput_mqput1_bytes_total	counter	Contagem total de bytes de MQPUT/MQPUT1 de intervalo
ibmmq_qmgr_mqput_mqput1_total	counter	Contagem total de MQPUT/MQPUT1 de intervalo
ibmmq_qmgr_mqset_total	counter	Contagem de MQSET
ibmmq_qmgr_mqstat_total	counter	Contagem de MQSTAT
ibmmq_qmgr_mqsubrq_total	counter	Contagem de MQSUBRQ
ibmmq_qmgr_non_durable_subscription_create_total	counter	Contagem de assinaturas não duráveis de criação

Métrica	Tipo	Descrição
ibmmq_qmgr_non_durable_subscription_delete_total	counter	Contagem de assinaturas não duráveis de exclusão
ibmmq_qmgr_non_persistent_message_browse_bytes_total	counter	Procura de mensagem não persistente - contagem de bytes
ibmmq_qmgr_non_persistent_message_browse_total	counter	Procura de mensagem não persistente - contagem
ibmmq_qmgr_non_persistent_message_destructive_get_total	counter	Obtenção destrutiva de mensagem não persistente - contagem
ibmmq_qmgr_non_persistent_message_get_bytes_total	counter	Mensagens não persistentes obtidas - contagem de bytes
ibmmq_qmgr_non_persistent_message_mqput1_total	counter	Contagem de MQPUT1 de mensagem não persistente
ibmmq_qmgr_non_persistent_message_mqput_total	counter	Contagem de MQPUT de mensagem não persistente
ibmmq_qmgr_non_persistent_message_put_bytes_total	counter	Mensagens não persistentes colocadas - contagem de bytes
ibmmq_qmgr_non_persistent_topic_mqput_mqput1_total	counter	Não persistente - contagem de MQPUT/MQPUT1 de tópico
ibmmq_qmgr_persistent_message_browse_bytes_total	counter	Procura de mensagem persistente - contagem de bytes
ibmmq_qmgr_persistent_message_browse_total	counter	Procura de mensagem persistente - contagem
ibmmq_qmgr_persistent_message_destructive_get_total	counter	Obtenção destrutiva de mensagem persistente - contagem
ibmmq_qmgr_persistent_message_get_bytes_total	counter	Mensagens persistentes obtidas - contagem de bytes
ibmmq_qmgr_persistent_message_mqput1_total	counter	Contagem de MQPUT1 de mensagem persistente
ibmmq_qmgr_persistent_message_mqput_total	counter	Contagem de MQPUT de mensagem persistente

Métrica	Tipo	Descrição
ibmmq_qmgr_persistent_message_put_bytes_total	counter	Mensagens persistentes colocadas - contagem de bytes
ibmmq_qmgr_persistent_topic_mqput_mqput1_total	counter	Persistente - contagem de MQPUT/MQPUT1 de tópico
ibmmq_qmgr_published_to_subscribers_bytes_total	counter	Publicado para assinantes - contagem de bytes
ibmmq_qmgr_published_to_subscribers_message_total	counter	Publicado para assinantes - contagem de mensagens
ibmmq_qmgr_purged_queue_total	counter	Contagem de filas limpas
ibmmq_qmgr_queue_manager_file_system_free_space_percentage	gauge	Sistema de arquivos do Gerenciador de Filas - espaço livre
ibmmq_qmgr_queue_manager_file_system_in_use_bytes	gauge	Sistema de arquivos do Gerenciador de Filas - bytes em uso
ibmmq_qmgr_ram_free_percentage	gauge	Porcentagem de RAM livre
ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes	gauge	Total de bytes de RAM - estimativa para o gerenciador de filas
ibmmq_qmgr_rollback_total	counter	Contagem de retrocessos
ibmmq_qmgr_system_cpu_time_estimate_for_queue_manager_percentage	gauge	Tempo de CPU do sistema - estimativa de porcentagem para o gerenciador de filas
ibmmq_qmgr_system_cpu_time_percentage	gauge	Porcentagem de tempo de CPU do sistema
ibmmq_qmgr_topic_mqput_mqput1_total	counter	Total do intervalo de MQPUT/MQPUT1 de tópico
ibmmq_qmgr_topic_put_bytes_total	counter	Total de bytes de tópico do intervalo colocados
ibmmq_qmgr_trace_file_system_free_space_percentage	gauge	Sistema de arquivos de rastreamento do MQ - espaço livre
ibmmq_qmgr_trace_file_system_in_use_bytes	gauge	Sistema de arquivos de rastreamento do MQ - bytes em uso

Métrica	Tipo	Descrição
ibmmq_qmgr_user_cpu_time_estimate_for_queue_manager_percentage	gauge	Tempo de CPU do usuário - estimativa de porcentagem para o gerenciador de filas
ibmmq_qmgr_user_cpu_time_percentage	gauge	Porcentagem de tempo de CPU do usuário

CP4I V9.2.2 CD Visualizando o status de gerenciadores de filas de HA nativa para contêineres certificados do IBM MQ ..

Para contêineres certificados pelo IBM MQ , é possível visualizar o status das instâncias de HA nativa executando o comando **dspm** dentro de um desses Pods em execução

Sobre esta tarefa

Importante:

É possível usar o comando **dspm** em um dos Pods em execução para visualizar o status operacional de uma instância do gerenciador de filas. As informações retornadas estarão de acordo com o estado ativo ou de réplica da instância. As informações fornecidas pela instância ativa são definitivas; as informações de nós de réplica podem estar desatualizadas.

Você pode executar as seguintes ações:

- Visualizar se a instância do gerenciador de filas no nó atual está ativa ou é uma réplica.
- Visualizar o status operacional de HA nativa da instância no nó atual.
- Visualizar o status operacional de todas as três instâncias em uma configuração de HA nativa.

Os campos de status a seguir são usados para relatar o status de configuração da HA nativa:

ROLE

Especifica a função atual da instância e é Active, Replica ou Unknown.

INSTÂNCIA

O nome fornecido para esta instância do gerenciador de filas quando ela foi criada usando a opção **-lr** do comando **crtmqm**.

INSYNC

Indica se a instância está apta a assumir o controle como a instância ativa, se necessário.

QUORUM

Relata o status de quorum no formato *number_of_instances_in-sync/number_of_instances_configured*.

REPLADDR

O endereço de replicação da instância do gerenciador de filas.

CONNECTV

Indica se o nó está conectado à instância ativa.

BACKLOG

Indica o número de KB que a instância está atrasada.

CONNINST

Indica se a instância nomeada está conectada a esta instância.

ALTDAT

Indica a data da última atualização dessas informações (em branco se elas nunca foram atualizadas).

ALTTIME

Indica o horário da última atualização dessas informações (em branco se elas nunca foram atualizadas).

Procedimento

- Localize os pods que fazem parte de seu gerenciador de filas

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- Executar o `dspm` em um dos pods

```
oc exec -t Pod dspm
```

```
oc ish Pod
```

para um shell interativo, em que é possível executar o `dspm` diretamente.

- Para determinar se uma instância do gerenciador de filas está em execução como a instância ativa ou como uma réplica:

```
oc exec -t Pod dspm -o status -m QMgrName
```

Uma instância ativa de um gerenciador de filas denominado BOB relataria o status a seguir:

```
QMNAME(BOB)          STATUS(Running)
```

Uma instância de réplica de um gerenciador de filas denominado BOB relataria o status a seguir:

```
QMNAME(BOB)          STATUS(Replica)
```

Uma instância inativa relataria o status a seguir:

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- Para determinar o status operacional de HA nativa da instância no pod especificado:

```
oc exec -t Pod dspm -o nativeha -m QMgrName
```

A instância ativa de um gerenciador de filas denominado BOB pode relatar o status a seguir:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Uma instância de réplica de um gerenciador de filas denominado BOB pode relatar o status a seguir:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Uma instância inativa de um gerenciador de filas denominado BOB pode relatar o status a seguir:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Para determinar o status operacional da HA nativa de todas as instâncias na configuração de HA nativa:

```
oc exec -t Pod dspm -o nativeha -x -m QMgrName
```

Se você emitir este comando no nó que executa a instância ativa do gerenciador de filas BOB, o status a seguir poderá ser recebido:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Ao emitir este comando em um nó que executa uma instância de réplica do gerenciador de filas BOB, você recebe o status a seguir, que indica que uma das réplicas está ficando atrasada:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Se você emitir este comando em um nó que executa uma instância inativa do gerenciador de filas BOB, o status a seguir poderá ser recebido:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

Se você emitir o comando quando as instâncias ainda estiverem negociando quais estão ativas e quais são réplicas, receberá o status a seguir:

```
QMNAME(BOB)          STATUS(Negotiating)
```

Referências relacionadas

[Comando dspmq \(exibir gerenciadores de filas\)](#)

“Exemplo: configurando um gerenciador de filas Native HA” na página 94

Este exemplo mostra como você implementa um gerenciador de filas usando o recurso de alta disponibilidade nativa no Red Hat OpenShift Container Platform (OCP) usando o IBM MQ Operator.

Fazer backup e restaurar a configuração do gerenciador de filas usando a CLI do Red Hat OpenShift

Fazer backup da configuração do gerenciador de filas poderá ajudar a reconstruir um gerenciador de filas por meio de suas definições, se a configuração do gerenciador de filas for perdida. Esse procedimento não faz backup dos dados de log do gerenciador de filas. Devido à natureza transitória das mensagens, os dados históricos do log provavelmente serão irrelevantes no momento da restauração.

Antes de começar

Efetue login em seu cluster usando **cloudctl login** (para o IBM Cloud Pak for Integration) ou **oc login**.

Procedimento

- Faça backup da configuração do gerenciador de filas.

É possível usar o comando **dmpmqcfig** para fazer dump da configuração de um gerenciador de filas do IBM MQ.

- Obtenha o nome do pod para o seu gerenciador de filas.
Por exemplo, seria possível executar o comando a seguir, em que *queue_manager_name* é o nome de seu recurso QueueManager:

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

- Execute o comando **dmpmqcfig** no pod, direcionando a saída para um arquivo na máquina local.
dmpmqcfig gera a configuração do MQSC do gerenciador de filas.

```
oc exec -it pod_name -- dmpmqcfig > backup.mqsc
```

- Restaure a configuração do gerenciador de filas.

Tendo seguido o procedimento de backup descrito na etapa anterior, deve-se ter um arquivo `backup.mqsc` que contém a configuração do gerenciador de filas. É possível restaurar a configuração aplicando esse arquivo a um novo gerenciador de filas.

- a) Obtenha o nome do pod para o seu gerenciador de filas.

Por exemplo, seria possível executar o comando a seguir, em que `queue_manager_name` é o nome de seu recurso `QueueManager`:

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

- b) Execute o comando `runmqsc` no pod, direcionando no conteúdo do arquivo `backup.mqsc`.

```
oc exec -i pod_name -- runmqsc < backup.mqsc
```

OpenShift

CP4I

Resolução de problemas com o IBM MQ Operator

Se estiver com problemas com o IBM MQ Operator, use as técnicas descritas para ajudá-lo a diagnosticar e resolver esses problemas.

Procedimento

- [“Resolução de Problemas: Obtendo Acesso aos Dados do Gerenciador de Filas” na página 125](#)

OpenShift

CP4I

Resolução de Problemas: Obtendo Acesso aos Dados do Gerenciador de Filas

Use a ferramenta de inspetor do PVC para obter acesso aos arquivos em um PVC do gerenciador de filas no qual um shell remoto não pode ser estabelecido para o pod do gerenciador de filas. Isso pode ser porque o pod está em um estado **Error** ou **CrashLoopBackOff**. Essa ferramenta foi projetada para uso com os gerenciadores de filas implementados pelo IBM MQ Operator.

Antes de começar

Para usar a ferramenta do inspetor do PVC você deve ter acesso ao namespace do gerenciador de filas.

Sobre esta tarefa

Para ajudar na resolução de problemas, é possível acessar os dados armazenados nas Solicitações de Volume Persistente (PVCs) associadas a um determinado gerenciador de filas. Para fazer isso, você usa uma ferramenta para montar os PVCs em um conjunto de pods do inspetor. É possível então obter um shell remoto em qualquer um dos pods do inspetor para ler os arquivos.

Dependendo do tipo de implementação, entre um e três pods do inspetor são criados. Os volumes específicos para um determinado pod de um gerenciador de filas de HA nativa ou de várias instâncias estão disponíveis no pod do inspetor de PVC associado. Volumes compartilhados estão disponíveis em todos os inspetores. O nome do pod do inspetor contém o nome do pod do gerenciador de filas associado.

Procedimento

1. Faça download da ferramenta do inspetor de PVC MQ .
A ferramenta está disponível aqui <https://github.com/ibm-messaging/mq-pvc-tool>.
2. Certifique-se de ter efetuado login em seu cluster.
3. Descubra o nome do gerenciador de filas e o namespace no qual o gerenciador de filas está em execução.
4. Execute a ferramenta do inspetor com relação ao seu gerenciador de filas

- a) Execute o comando a seguir, especificando o nome do gerenciador de filas e o nome do namespace..

```
./pvc-tool.sh queue_manager_name queue_manager_namespace_name
```

- b) Após a conclusão da ferramenta, execute o comando a seguir para visualizar os pods do inspetor que estão sendo criados.

```
oc get pods
```

5. Visualize os arquivos montadas para o pod do inspetor

- a) Cada pod do inspetor de PVC é associado a um pod do gerenciador de filas, portanto, pode haver vários pods do inspetor. Acesse um desses pods, executando o seguinte comando:

```
oc rsh pvc-inspector-pod-name
```

Você é colocado no diretório contendo os diretórios PVC montados.

- b) Abra um shell remoto no pod, executando o comando a seguir:

```
ls
```

- c) É possível ver diretórios com o mesmo nome que os PVCs que foram montados. Acesse os arquivos nos PVCs do gerenciador de filas procurando esses diretórios. Para ver uma lista de PVCs, execute o comando a seguir fora da sessão de shell remoto:

```
oc get pvc
```

- d) Limpe os pods criados pela ferramenta, executando o comando a seguir:

```
'oc delete pods -l tool=mq-pvc-inspector
```

OpenShift

CP4I

Referência da API do IBM MQ Operator

IBM MQ fornece um Kubernetes Operator, que fornece integração nativa com Red Hat OpenShift Container Platform.

OpenShift

CP4I

Referência da API para mq.ibm.com/v1beta1

A API v1beta1 pode ser usada para criar e gerenciar recursos do QueueManager.

OpenShift

CP4I

CD

EUS

Referência de licenciamento para mq.ibm.com/v1beta1

Versões atuais de licença

O campo `spec.license.license` deve conter o identificador da licença que você está aceitando. Os valores válidos são:

Valor de <code>spec.license.license</code>	Valor de <code>spec.license.use</code>	Informações sobre licença	Versões aplicáveis do IBM MQ
L-RJON-C7QG3S	Production ou NonProduction	IBM Cloud Pak for Integration 2021.4.1	9.2.4 ou 9.2.5
L-RJON-C7QFZX	Production ou NonProduction	IBM Cloud Pak for Integration Edição Ltda 2021.4.1	9.2.4 ou 9.2.5
L-RJON-C5CSNH	Production ou NonProduction	IBM Cloud Pak for Integration 2021.3.1	9.2.3 ou 9.2.4

Valor de spec.license.license	Valor de spec.license.use	Informações sobre licença	Versões aplicáveis do IBM MQ
L-RJON-C5CSM2	Production ou NonProduction	IBM Cloud Pak for Integration Edição Ltda 2021.3.1	9.2.3 ou 9.2.4
L-RJON-BZFQU2	Production ou NonProduction	IBM Cloud Pak for Integration 2021.2.1	9.2.3
L-RJON-BZFQSB	Production ou NonProduction	IBM Cloud Pak for Integration Edição Limitada 2021.2.1 ..	9.2.3
L-RJON-BUVMQX	Production ou NonProduction	IBM Cloud Pak for Integration 2020.4.1	9.2.0 EUS ou 9.2.1
L-RJON-BUVMYB	Production ou NonProduction	IBM Cloud Pak for Integration Limited Edition 2020.4.1	9.2.0 EUS ou 9.2.1
L-APIG-BZDDDY	Production	IBM MQ Advanced e IBM MQ Advanced para Ambiente de Não Produção 9.2 -07/2021	9.2.3, 9.2.4 ou 9.2.5
L-APIG-BYHCL7	Development	IBM MQ Advanced for Developers (Não Warranted) V9.2 -07/2021	9.2.3, 9.2.4 ou 9.2.5
L-APIG-BVJJB3	Production	IBM MQ Advanced e IBM MQ Advanced for Non-Production Environment 9.2 -03/2021	9.2.2
L-APIG-BMJJBM	Production	IBM MQ Advanced V9.2	9.2.0 CD ou 9.2.1
L-APIG-BMKG5H	Development	IBM MQ Advanced for Developers (Sem garantia) V9.2	9.2.0 CD, 9.2.1 ou 9.2.2

Observe que a *versão* da licença é especificada, que nem sempre é a mesma da versão do IBM MQ.

Versões de licença mais antigas

O campo spec.license.license deve conter o identificador da licença que você está aceitando. Os valores válidos são:

Valor de spec.license.license	Valor de spec.license.use	Informações sobre licença	Versões aplicáveis do IBM MQ
L-RJON-BXUPZ2	Production ou NonProduction	IBM Cloud Pak for Integration 2021.1.1	9.2.2
L-RJON-BXUQ34	Production ou NonProduction	IBM Cloud Pak for Integration Edição Limitada 2021.1.1 ..	9.2.2
L-RJON-BYRMYW	NonProduction	IBM Cloud Pak for Integration Avaliação-demo 2021.1.1. Liberação antecipada para uso somente com o Native HA com o IBM MQ Operator 1.5.	9.2.2
L-RJON-BQPGWD	Production ou NonProduction	IBM Cloud Pak for Integration 2020.3.1	9.2.0 CD
L-RJON-BN7PN3	Production ou NonProduction	IBM Cloud Pak for Integration 2020.2.1	9.1.5 ou 9.2.0 CD

Valor de <code>spec.license.license</code>	Valor de <code>spec.license.use</code>	Informações sobre licença	Versões aplicáveis do IBM MQ
L-RJON-BPHL2Y	Production ou NonProduction	IBM Cloud Pak for Integration Edição Limitada 2020.2.1 ..	9.1.5
L-APIG-BJAKBF	Production	IBM MQ Advanced V9.1 -04/2020	9.1.5
L-APIG-BM7GDH	Development	IBM MQ Advanced for Developers (Sem garantia) V9.1 - 04/2020	9.1.5

Observe que a *versão* da licença é especificada, que nem sempre é a mesma da versão do IBM MQ.

  **Referência de API para QueueManager (mq.ibm.com/v1beta1)**

QueueManager

Um QueueManager é um Servidor do IBM MQ que fornece serviços de enfileiramento e de publicar/ assinar para aplicativos.

Campo	Descrição
Sequência de <code>apiVersion</code>	<code>APIVersion</code> define o esquema com versão dessa representação de um objeto. Os servidores devem converter os esquemas reconhecidos para o valor interno mais recente e podem rejeitar valores não reconhecidos. Mais informações: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources .
Sequência de <code>kind</code>	<code>Kind</code> é um valor de sequência que representa o recurso REST que este objeto representa. Os servidores podem inferir isso do terminal para o qual o cliente envia solicitações. Não é possível ser atualizado. In CamelCase. Mais informações: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds .
<code>metadata</code>	
<code>spec QueueManagerSpec</code>	O estado desejado do QueueManager.
<code>status QueueManagerStatus</code>	O estado observado do QueueManager.

.spec

O estado desejado do QueueManager.

Aparece em:

- [“QueueManager”](#) na página 128

Campo	Descrição
<code>affinity</code>	Regras de afinidade do Kubernetes padrão. Para obter mais informações, consulte https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core .
<code>annotations</code> <u>Anotações</u>	O campo de anotações serve como uma passagem para anotações do Pod. Os usuários podem incluir qualquer anotação nesse campo e aplicá-las no Pod. As anotações aqui sobrescrevem as anotações padrão, se fornecidas. Requer o MQ Operator 1.3.0 ou mais recente.

Campo	Descrição
<code>imagePullSecrets</code> Matriz LocalObjectReference	Uma lista opcional de referências a segredos no mesmo namespace a ser usado para extrair qualquer uma das imagens usadas por este QueueManager. Se especificado, esses segredos serão transmitidos a implementações de extratores individuais para o uso. Por exemplo, no caso do docker, apenas os segredos do tipo DockerConfig são honrados. Para obter mais informações, consulte https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod .
<code>labels</code> Rótulos	O campo labels serve como uma passagem para rótulos do Pod. Os usuários podem incluir qualquer rótulo nesse campo e aplicá-lo no Pod. Os rótulos aqui sobrescrevem os rótulos padrão, se fornecidos. Requer o MQ Operator 1.3.0 ou mais recente.
<code>license</code> Licença	Configurações que controlam sua aceitação da licença e quais as métricas de licença a serem usadas.
<code>pki</code> PKI	Configurações de Infraestrutura de chave pública, para definição de chaves e certificados para uso com a Segurança da Camada de Transporte (TLS) ou o MQ Advanced Message Security (AMS).
<code>queueManager</code> QueueManagerConfig	Configurações para o contêiner do Gerenciador de Filas e o Gerenciador de Filas subjacente.
<code>securityContext</code> SecurityContext	Configurações de segurança a serem adicionadas ao securityContext do pod do gerenciador de filas.
<code>template</code> Modelo	Modelagem avançada para recursos do Kubernetes. O modelo permite que os usuários substituam como o IBM MQ gera os recursos do Kubernetes subjacentes, como StatefulSet, Pods e Serviços. Destina-se somente a usuários avançados, pois terá o potencial de interromper a operação normal do MQ se for usado incorretamente. Quaisquer valores especificados em qualquer outro lugar no recurso QueueManager serão substituídos por configurações no modelo.
Número inteiro <code>terminationGracePeriod</code> Seconds	Duração opcional em segundos que o Pod precisa finalizar normalmente. O valor deve ser um número inteiro não negativo. O valor zero indica exclusão imediatamente. O horário alvo no qual o término do gerenciador de filas é tentado, escalando as fases de desconexão do aplicativo. Se necessário, tarefas de manutenção essenciais do gerenciador de filas serão interrompidas. O padrão é 30 segundos.
<code>tracing</code> TracingConfig	Configurações para integração de rastreamento com o Cloud Pak for Integration Operations Dashboard.
Sequência de version	Configuração que controla a versão do MQ que será usada (obrigatório). Por exemplo: 9.1.5.0-r2 especificaria o MQ versão 9.1.5.0, usando a segunda revisão da imagem do contêiner. Correções específicas do contêiner são frequentemente aplicadas em revisões, como correções para a imagem base.
<code>web</code> WebServerConfig	Configurações para o servidor da web do MQ.

.spec.annotations

O campo de anotações serve como uma passagem para anotações do Pod. Os usuários podem incluir qualquer anotação nesse campo e aplicá-las no Pod. As anotações aqui sobrescrevem as anotações padrão, se fornecidas. Requer o MQ Operator 1.3.0 ou mais recente.

Aparece em:

- [“.spec” na página 128](#)

.spec.imagePullSecrets

LocalObjectReference contém informações suficientes para deixar você localizar o objeto referenciado dentro do mesmo namespace.

Aparece em:

- [“.spec” na página 128](#)

Campo	Descrição
Sequência de name	Nome do referente. Mais informações: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Adicione outros campos úteis. apiVersão, tipo, uid?

.spec.labels

O campo labels serve como uma passagem para rótulos do Pod. Os usuários podem incluir qualquer rótulo nesse campo e aplicá-lo no Pod. Os rótulos aqui sobrescrevem os rótulos padrão, se fornecidos. Requer o MQ Operator 1.3.0 ou mais recente.

Aparece em:

- [“.spec” na página 128](#)

.spec.license

Configurações que controlam sua aceitação da licença e quais as métricas de licença a serem usadas.

Aparece em:

- [“.spec” na página 128](#)

Campo	Descrição
Booleano accept	Indica se você aceita ou não a licença associada a este software (obrigatório).
Sequência de license	O identificador da licença que você está aceitando. Esse deve ser o identificador de licença correto para a versão do MQ que você está usando. Consulte http://ibm.biz/BdqvCF para obter valores válidos.
Sequência de metric	Configuração que especifica qual métrica de licença usar. Por exemplo, ProcessorValueUnit, VirtualProcessorCore ou ManagedVirtualServer. O padrão é ProcessorValueUnit ao usar uma licença do MQ e VirtualProcessorCore ao usar uma licença do Cloud Pak for Integration.
Sequência de use	Configuração que controla como o software será usado, na qual a licença suporta diversos usos. Consulte http://ibm.biz/BdqvCF para obter valores válidos.

.spec.pki

Configurações de Infraestrutura de chave pública, para definição de chaves e certificados para uso com a Segurança da Camada de Transporte (TLS) ou o MQ Advanced Message Security (AMS).

Aparece em:

- [“.spec” na página 128](#)

Campo	Descrição
keys Matriz do <u>PKISource</u>	Chaves privadas a serem adicionadas ao repositório de chaves do Queue Manager.

Campo	Descrição
trust matriz de PKISource	Certificados a serem adicionados ao repositório de chaves do Queue Manager.

.spec.pki.keys

O PKISource define uma origem de informações de Infraestrutura de chave pública, como chaves ou certificados.

Aparece em:

- [“.spec.pki”](#) na página 130

Campo	Descrição
Sequência de name	O nome é usado como o rótulo para a chave ou o certificado. Deve ser uma sequência alfanumérica minúscula.
secret Segredo	Forneça uma chave usando um Segredo do Kubernetes.

.spec.pki.keys.secret

Forneça uma chave usando um Segredo do Kubernetes.

Aparece em:

- [“.spec.pki.keys”](#) na página 131

Campo	Descrição
Matriz de items	Chaves dentro do segredo do Kubernetes que devem ser incluídas no contêiner do Gerenciador de Filas.
Sequência de secretName	O nome do segredo do Kubernetes.

.spec.pki.trust

O PKISource define uma origem de informações de Infraestrutura de chave pública, como chaves ou certificados.

Aparece em:

- [“.spec.pki”](#) na página 130

Campo	Descrição
Sequência de name	O nome é usado como o rótulo para a chave ou o certificado. Deve ser uma sequência alfanumérica minúscula.
secret Segredo	Forneça uma chave usando um Segredo do Kubernetes.

.spec.pki.trust.secret

Forneça uma chave usando um Segredo do Kubernetes.

Aparece em:

- [“.spec.pki.trust”](#) na página 131

Campo	Descrição
Matriz de items	Chaves dentro do segredo do Kubernetes que devem ser incluídas no contêiner do Gerenciador de Filas.
Sequência de secretName	O nome do segredo do Kubernetes.

.spec.queueManager

Configurações para o contêiner do Gerenciador de Filas e o Gerenciador de Filas subjacente.

Aparece em:

- [“.spec” na página 128](#)

Campo	Descrição
<code>availability</code> Disponibilidade	Configurações de disponibilidade para o Gerenciador de Filas, como o uso ou não de um par ativo-em espera ou da alta disponibilidade nativa.
Booleano <code>debug</code>	Indica se mensagens de depuração do código específico do contêiner devem ou não ser registradas no log do contêiner. Usa <code>false</code> como padrão.
Sequência de <code>image</code>	A imagem de contêiner que será usada.
Sequência de <code>imagePullPolicy</code>	Configuração que controla quando o kubelet tenta extrair a imagem especificada. O padrão é <code>IfNotPresent</code> .
<code>ini</code> Matriz de INISource	Configurações para fornecer INI para o Gerenciador de Filas. Requer o MQ Operator 1.1.0 ou mais recente.
<code>livenessProbe</code> QueueManagerLivenessProbe	Configurações que controlam a análise de vivacidade.
Sequência de <code>logFormat</code>	Qual o formato de log a usar para este contêiner. Use <code>JSON</code> para logs formatados por JSON do contêiner. Use <code>Basic</code> para mensagens formatadas por texto. O padrão é <code>Basic</code> .
<code>metrics</code> QueueManagerMetrics	Configurações para métricas de estilo Prometheus.
<code>mqsc</code> Matriz MQSCSource	Configurações para fornecer MQSC para o Gerenciador de Filas. Requer o MQ Operator 1.1.0 ou mais recente.
Sequência de <code>name</code>	Nome do Gerenciador de Filas do MQ subjacente, se diferente de <code>metadata.name</code> . Use este campo se você quiser um nome do Gerenciador de Filas que não esteja em conformidade com as regras do Kubernetes para nomes (por exemplo, um nome que inclua letras maiúsculas).
<code>readinessProbe</code> QueueManagerReadinessProbe	Configurações que controlam a análise de prontidão.
<code>resources</code> Recursos	Configurações que controlam os requisitos de recursos.
<code>route</code> Rota	Configurações da rota do Gerenciador de Filas. Requer MQ Operator 1.4.0 ou mais recente.
<code>startupProbe</code> StartupProbe	Configurações que controlam a análise de inicialização. Aplica-se apenas às implementações de <code>MultiInstance</code> e <code>NativeHA</code> . Requer o MQ Operator 1.5.0 ou mais recente.
<code>storage</code> QueueManagerStorage	Configurações de armazenamento para controlar o uso de volumes persistentes e classes de armazenamento pelo Gerenciador de Filas.

.spec.queueManager.availability

Configurações de disponibilidade para o Gerenciador de Filas, como o uso ou não de um par ativo-em espera ou da alta disponibilidade nativa.

Aparece em:

- [“.spec.queueManager” na página 132](#)

Campo	Descrição
<code>tls Tls</code>	Configurações opcionais de TLS para configurar a comunicação segura entre as réplicas de NativeHA. Requer o MQ Operator 1.5.0 ou mais recente.
Sequência de <code>type</code>	O tipo de disponibilidade para usar. Use <code>SingleInstance</code> para um único Pod, que será reiniciado automaticamente (em alguns casos) por Kubernetes. Use <code>MultiInstance</code> para um par de Pods, um dos quais é o Gerenciador de Filas ativo e o outro do qual é uma espera. Use <code>NativeHA</code> para replicação de alta disponibilidade nativa (requer o MQ Operator 1.5.0 ou mais recente). O padrão é <code>SingleInstance</code> . Consulte http://ibm.biz/BdqAQa para obter mais detalhes.
<code>updateStrategy string</code>	A estratégia de atualização a ser usada para gerenciadores de filas de várias instâncias e Native HA. Use <code>RollingUpdate</code> para ativar as atualizações contínuas automáticas sempre que a configuração do gerenciador de filas mudar. Use <code>OnDelete</code> para desativar as atualizações contínuas automáticas. As mudanças do gerenciador de filas serão aplicadas somente quando os Pods forem excluídos (inclusive as exclusões de Pods acionadas por fatores externos). Padronizado para <code>RollingUpdate</code> . Requer o MQ Operator 1.6.0 ou superior.

.spec.queueManager.availability.tls

Configurações opcionais de TLS para configurar a comunicação segura entre as réplicas de NativeHA. Requer o MQ Operator 1.5.0 ou mais recente.

Aparece em:

- [“.spec.queueManager.availability” na página 132](#)

Campo	Descrição
Sequência de <code>cipherSpec</code>	O nome do CipherSpec para TLS NativeHA.
Sequência de <code>secretName</code>	O nome do segredo do Kubernetes.

.spec.queueManager.ini

Origem de arquivos de configuração INI.

Aparece em:

- [“.spec.queueManager” na página 132](#)

Campo	Descrição
<code>configMap ConfigMapINISource</code>	O ConfigMap representa um ConfigMap do Kubernetes que contém informações de INI.
<code>secret SecretINISource</code>	O Segredo representa um Segredo do Kubernetes que contém informações de INI.

.spec.queueManager.ini.configMap

O ConfigMap representa um ConfigMap do Kubernetes que contém informações de INI.

Aparece em:

- [“.spec.queueManager.ini” na página 133](#)

Campo	Descrição
Matriz de itens	Chaves dentro da origem do Kubernetes que devem ser aplicadas.
Sequência de name	O nome da origem do Kubernetes.

.spec.queueManager.ini.secret

O Segredo representa um Segredo do Kubernetes que contém informações de INI.

Aparece em:

- [“.spec.queueManager.ini”](#) na página 133

Campo	Descrição
Matriz de itens	Chaves dentro da origem do Kubernetes que devem ser aplicadas.
Sequência de name	O nome da origem do Kubernetes.

.spec.queueManager.livenessProbe

Configurações que controlam a análise de vivacidade.

Aparece em:

- [“.spec.queueManager”](#) na página 132

Campo	Descrição
Número inteiro <code>failureThreshold</code>	As falhas mínimas consecutivas para que a análise seja considerada com falha após ter sido bem-sucedida. O padrão é 1.
Número inteiro <code>initialDelaySeconds</code>	Número de segundos após o contêiner ter sido iniciado antes que a análise seja iniciada. Padronizado para 90 segundos para <code>SingleInstance</code> . Padronizado para 0 segundos para implementações <code>MultiInstance</code> e <code>NativeHA</code> . Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro <code>periodSeconds</code>	Com que frequência (em segundos) realizar a análise. O padrão é 10 segundos.
Número inteiro <code>successThreshold</code>	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida depois de ter falhado. O padrão é 1.
Número inteiro <code>timeoutSeconds</code>	Número de segundos após os quais a análise atingirá o tempo limite. O padrão é 5 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.metrics

Configurações para métricas de estilo Prometheus.

Aparece em:

- [“.spec.queueManager”](#) na página 132

Campo	Descrição
Booleano <code>enabled</code>	Se um terminal de métricas compatíveis com Prometheus deve ou não ser ativado. O padrão é verdadeiro.

.spec.queueManager.mqsc

Origem de arquivos de configuração do MQSC.

Aparece em:

- [“.spec.queueManager” na página 132](#)

Campo	Descrição
configMap ConfigMapMQSCSource	O ConfigMap representa um ConfigMap do Kubernetes que contém informações do MQSC.
secret SecretMQSCSource	O Segredo representa um Segredo do Kubernetes que contém informações do MQSC.

.spec.queueManager.mqsc.configMap

O ConfigMap representa um ConfigMap do Kubernetes que contém informações do MQSC.

Aparece em:

- [“.spec.queueManager.mqsc” na página 135](#)

Campo	Descrição
Matriz de itens	Chaves dentro da origem do Kubernetes que devem ser aplicadas.
Sequência de name	O nome da origem do Kubernetes.

.spec.queueManager.mqsc.secret

O Segredo representa um Segredo do Kubernetes que contém informações do MQSC.

Aparece em:

- [“.spec.queueManager.mqsc” na página 135](#)

Campo	Descrição
Matriz de itens	Chaves dentro da origem do Kubernetes que devem ser aplicadas.
Sequência de name	O nome da origem do Kubernetes.

.spec.queueManager.readinessProbe

Configurações que controlam a análise de prontidão.

Aparece em:

- [“.spec.queueManager” na página 132](#)

Campo	Descrição
Número inteiro <code>failureThreshold</code>	As falhas mínimas consecutivas para que a análise seja considerada com falha após ter sido bem-sucedida. O padrão é 1.
Número inteiro <code>initialDelaySeconds</code>	Número de segundos após o contêiner ter sido iniciado antes que a análise seja iniciada. Padronizado para 10 segundos para SingleInstance. Padronizado para 0 para implementações MultiInstance e NativeHA. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro <code>periodSeconds</code>	Com que frequência (em segundos) realizar a análise. O padrão é 5 segundos.

Campo	Descrição
Número inteiro successThreshold	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida depois de ter falhado. O padrão é 1.
Número inteiro timeoutSeconds	Número de segundos após os quais a análise atingirá o tempo limite. O padrão é 3 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.resources

Configurações que controlam os requisitos de recursos.

Aparece em:

- [“.spec.queueManager” na página 132](#)

Campo	Descrição
limits <u>Limites</u>	Configurações de CPU e memória.
requests <u>Solicitações</u>	Configurações de CPU e memória.

.spec.queueManager.resources.limits

Configurações de CPU e memória.

Aparece em:

- [“.spec.queueManager.resources” na página 136](#)

Campo	Descrição
cpu	
memory	

.spec.queueManager.resources.requests

Configurações de CPU e memória.

Aparece em:

- [“.spec.queueManager.resources” na página 136](#)

Campo	Descrição
cpu	
memory	

.spec.queueManager.route

Configurações da rota do Gerenciador de Filas. Requer MQ Operator 1.4.0 ou mais recente.

Aparece em:

- [“.spec.queueManager” na página 132](#)

Campo	Descrição
Booleano enabled	Se a rota será ativada ou não. O padrão é verdadeiro.

.spec.queueManager.startupProbe

Configurações que controlam a análise de inicialização. Aplica-se apenas às implementações de MultiInstance e NativeHA. Requer o MQ Operator 1.5.0 ou mais recente.

Aparece em:

- [“.spec.queueManager” na página 132](#)

Campo	Descrição
Número inteiro <code>failureThreshold</code>	Falhas mínimas consecutivas para que a análise seja considerada com falha. Padronizado como 60.
Número inteiro <code>initialDelaySeconds</code>	Número de segundos após o contêiner ter sido iniciado antes que a análise seja iniciada. Padronizado para 0 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro <code>periodSeconds</code>	Com que frequência (em segundos) realizar a análise. O padrão é 5 segundos.
Número inteiro <code>successThreshold</code>	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida. O padrão é 1.
Número inteiro <code>timeoutSeconds</code>	Número de segundos após os quais a análise atingirá o tempo limite. O padrão é 5 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.storage

Configurações de armazenamento para controlar o uso de volumes persistentes e classes de armazenamento pelo Gerenciador de Filas.

Aparece em:

- [“.spec.queueManager” na página 132](#)

Campo	Descrição
Sequência de caracteres <code>defaultClass</code>	A classe de armazenamento para aplicar a todos os volumes persistentes desse Gerenciador de Filas por padrão. Os volumes persistentes específicos podem definir a própria classe de armazenamento que substituirá essa configuração de classe de armazenamento padrão. Se <code>type of availability for SingleInstance</code> ou <code>NativeHA</code> , a classe de armazenamento poderá ser do tipo <code>ReadWriteOnce</code> ou <code>ReadWriteMany</code> . Se <code>type of availability for MultiInstance</code> , a classe de armazenamento deverá ser do tipo <code>ReadWriteMany</code> .
Booleano <code>defaultDeleteClaim</code>	Se todos os volumes devem ou não ser excluídos quando o gerenciador de filas é excluído. Os volumes persistentes específicos podem definir o próprio valor para <code>deleteClaim</code> que substituirá essa configuração <code>defaultDeleteClaim</code> . Usa <code>false</code> como padrão.
<code>persistedData</code> QueueManagerOptionalVolume	Detalhes de <code>persistentVolume</code> para dados persistidos do MQ, incluindo configuração, filas e mensagens. Necessário ao usar o Gerenciador de Filas de várias instâncias.
<code>queueManager</code> QueueManagerVolume	<code>PersistentVolume</code> padrão para quaisquer dados normalmente em <code>/var/mqm</code> . Conterá todos os dados persistidos e logs de recuperação, se nenhum outro volume for especificado.

Campo	Descrição
recoveryLogs QueueManagerOptionalVolume	Detalhes de volume persistente para logs de recuperação do MQ. Necessário ao usar o Gerenciador de Filas de várias instâncias.

.spec.queueManager.storage.persistedData

Detalhes de persistentVolume para dados persistidos do MQ, incluindo configuração, filas e mensagens. Necessário ao usar o Gerenciador de Filas de várias instâncias.

Aparece em:

- [“.spec.queueManager.storage”](#) na página 137

Campo	Descrição
Sequência de class	Classe de armazenamento a usar para este volume. Apenas válido se type for persistent-claim. Se type of availability for SingleInstance ou NativeHA, a classe de armazenamento poderá ser do tipo ReadWriteOnce ou ReadWriteMany. Se type of availability for MultiInstance, a classe de armazenamento deverá ser do tipo ReadWriteMany.
Booleano deleteClaim	Se esse volume deve ou não ser excluído quando o gerenciador de filas é excluído.
Booleano enabled	Indica se esse volume deve ou não ser ativado como um volume separado ou ser colocado no volume padrão queueManager. Usa false como padrão.
Sequência de size	Tamanho do PersistentVolume para transmitir ao Kubernetes, incluindo unidades do SI. Apenas válido se type for persistent-claim. Por exemplo, 2Gi. O padrão é 2Gi.
Sequência de sizeLimit	Limite de tamanho ao usar um volume efêmero. Os arquivos ainda são gravados em um diretório temporário, portanto, é possível usar essa opção para limitar o tamanho. Apenas válido se type for efêmero.
Sequência de type	Tipo de volume a ser usado. Escolha ephemeral para usar o armazenamento não persistente ou persistent-claim para usar um volume persistente. O padrão é persistent-claim.

.spec.queueManager.storage.queueManager

PersistentVolume padrão para quaisquer dados normalmente em /var/mqm. Conterá todos os dados persistidos e logs de recuperação, se nenhum outro volume for especificado.

Aparece em:

- [“.spec.queueManager.storage”](#) na página 137

Campo	Descrição
Sequência de class	Classe de armazenamento a usar para este volume. Apenas válido se type for persistent-claim. Se type of availability for SingleInstance ou NativeHA, a classe de armazenamento poderá ser do tipo ReadWriteOnce ou ReadWriteMany. Se type of availability for MultiInstance, a classe de armazenamento deverá ser do tipo ReadWriteMany.
Booleano deleteClaim	Se esse volume deve ou não ser excluído quando o gerenciador de filas é excluído.

Campo	Descrição
Sequência de <code>size</code>	Tamanho do PersistentVolume para transmitir ao Kubernetes, incluindo unidades do SI. Apenas válido se <code>type for persistent-claim</code> . Por exemplo, 2Gi. O padrão é 2Gi.
Sequência de <code>sizeLimit</code>	Limite de tamanho ao usar um volume efêmero. Os arquivos ainda são gravados em um diretório temporário, portanto, é possível usar essa opção para limitar o tamanho. Apenas válido se <code>type for efêmero</code> .
Sequência de <code>type</code>	Tipo de volume a ser usado. Escolha <code>ephemeral</code> para usar o armazenamento não persistente ou <code>persistent-claim</code> para usar um volume persistente. O padrão é <code>persistent-claim</code> .

.spec.queueManager.storage.recoveryLogs

Detalhes de volume persistente para logs de recuperação do MQ. Necessário ao usar o Gerenciador de Filas de várias instâncias.

Aparece em:

- [“.spec.queueManager.storage” na página 137](#)

Campo	Descrição
Sequência de <code>class</code>	Classe de armazenamento a usar para este volume. Apenas válido se <code>type for persistent-claim</code> . Se <code>type of availability for SingleInstance</code> ou <code>NativeHA</code> , a classe de armazenamento poderá ser do tipo <code>ReadWriteOnce</code> ou <code>ReadWriteMany</code> . Se <code>type of availability for MultiInstance</code> , a classe de armazenamento deverá ser do tipo <code>ReadWriteMany</code> .
Booleano <code>deleteClaim</code>	Se esse volume deve ou não ser excluído quando o gerenciador de filas é excluído.
Booleano <code>enabled</code>	Indica se esse volume deve ou não ser ativado como um volume separado ou ser colocado no volume padrão <code>queueManager</code> . Usa <code>false</code> como padrão.
Sequência de <code>size</code>	Tamanho do PersistentVolume para transmitir ao Kubernetes, incluindo unidades do SI. Apenas válido se <code>type for persistent-claim</code> . Por exemplo, 2Gi. O padrão é 2Gi.
Sequência de <code>sizeLimit</code>	Limite de tamanho ao usar um volume efêmero. Os arquivos ainda são gravados em um diretório temporário, portanto, é possível usar essa opção para limitar o tamanho. Apenas válido se <code>type for efêmero</code> .
Sequência de <code>type</code>	Tipo de volume a ser usado. Escolha <code>ephemeral</code> para usar o armazenamento não persistente ou <code>persistent-claim</code> para usar um volume persistente. O padrão é <code>persistent-claim</code> .

.spec.securityContext

Configurações de segurança a serem adicionadas ao `securityContext` do pod do gerenciador de filas.

Aparece em:

- [“.spec” na página 128](#)

Campo	Descrição
Número inteiro fsGroup	Um grupo suplementar especial que se aplica a todos os contêineres em um pod. Alguns tipos de volume permitem que o Kubelet altere a propriedade desse volume para que seja de propriedade do pod: 1. O GOD proprietário será o FSGroup 2. O bit setgid está definido (novos arquivos criados no volume serão de propriedade do FSGroup) 3. Os bits de permissão são OR'd com rw-rw---- Se não estiver definido, o Kubelet não modificará a propriedade e as permissões de nenhum volume.
Booleano initVolumeAsRoot	Afeta o securityContext usado pelo contêiner que inicializa o PersistentVolume. Configure como true se você estiver usando um provedor de armazenamento que requeira que você seja o usuário raiz para acessar volumes recém-provisionados. A configuração como true determina qual objeto de Security Context Constraints (SCC) é possível usar; além disso, poderá ocorrer uma falha na inicialização do Gerenciador de Filas se você não estiver autorizado a usar uma SCC que permita o usuário raiz. Usa false como padrão. Para obter mais informações, consulte https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html .
Matriz de supplementalGroups	Uma lista de grupos aplicados ao primeiro processo executado em cada contêiner, além do GID principal do contêiner. Se não especificado, nenhum grupo será incluído em nenhum contêiner.

.spec.template

Modelagem avançada para recursos do Kubernetes. O modelo permite que os usuários substituam como o IBM MQ gera os recursos do Kubernetes subjacentes, como StatefulSet, Pods e Serviços. Destina-se somente a usuários avançados, pois terá o potencial de interromper a operação normal do MQ se for usado incorretamente. Quaisquer valores especificados em qualquer outro lugar no recurso QueueManager serão substituídos por configurações no modelo.

Aparece em:

- [“.spec” na página 128](#)

Campo	Descrição
pod	Substituições para o modelo usado para o Pod. Consulte https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core .

.spec.tracing

Configurações para integração de rastreamento com o Cloud Pak for Integration Operations Dashboard.

Aparece em:

- [“.spec” na página 128](#)

Campo	Descrição
agent TracingAgent	No Cloud Pak for Integration apenas, é possível definir configurações para o Agente de rastreamento opcional.
collector TracingCollector	No Cloud Pak for Integration apenas, é possível definir configurações para o Coletor de rastreamento opcional.
Booleano enabled	Indica se a integração com o Cloud Pak for Integration Operations Dashboard deve ou não ser ativada por meio do rastreamento. Usa false como padrão.

Campo	Descrição
Sequência de namespace	Namespace no qual o Cloud Pak for Integration Operations Dashboard está instalado.

.spec.tracing.agent

No Cloud Pak for Integration apenas, é possível definir configurações para o Agente de rastreo opcional.

Aparece em:

- [“.spec.tracing”](#) na página 140

Campo	Descrição
Sequência de imagem	A imagem de contêiner que será usada.
Sequência de <code>imagePullPolicy</code>	Configuração que controla quando o kubelet tenta extrair a imagem especificada. O padrão é <code>IfNotPresent</code> .
<code>livenessProbe</code> <code>TracingProbe</code>	Configurações que controlam a análise de vivacidade.
<code>readinessProbe</code> <code>TracingProbe</code>	Configurações que controlam a análise de prontidão.

.spec.tracing.agent.livenessProbe

Configurações que controlam a análise de vivacidade.

Aparece em:

- [“.spec.tracing.agent”](#) na página 141

Campo	Descrição
Número inteiro <code>failureThreshold</code>	As falhas mínimas consecutivas para que a análise seja considerada com falha após ter sido bem-sucedida. O padrão é 1.
Número inteiro <code>initialDelaySeconds</code>	Número de segundos após o contêiner ter iniciado antes que as análises de vivacidade sejam iniciadas. O padrão é 10 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro <code>periodSeconds</code>	Com que frequência (em segundos) realizar a análise. O padrão é 10 segundos.
Número inteiro <code>successThreshold</code>	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida depois de ter falhado. O padrão é 1.
Número inteiro <code>timeoutSeconds</code>	Número de segundos após os quais a análise atingirá o tempo limite. O padrão é 2 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.agent.readinessProbe

Configurações que controlam a análise de prontidão.

Aparece em:

- [“.spec.tracing.agent”](#) na página 141

Campo	Descrição
Número inteiro <code>failureThreshold</code>	As falhas mínimas consecutivas para que a análise seja considerada com falha após ter sido bem-sucedida. O padrão é 1.
Número inteiro <code>initialDelaySeconds</code>	Número de segundos após o contêiner ter iniciado antes que as análises de vivacidade sejam iniciadas. O padrão é 10 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro <code>periodSeconds</code>	Com que frequência (em segundos) realizar a análise. O padrão é 10 segundos.
Número inteiro <code>successThreshold</code>	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida depois de ter falhado. O padrão é 1.
Número inteiro <code>timeoutSeconds</code>	Número de segundos após os quais a análise atingirá o tempo limite. O padrão é 2 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.collector

No Cloud Pak for Integration apenas, é possível definir configurações para o Coletor de rastreamento opcional.

Aparece em:

- [“.spec.tracing”](#) na página 140

Campo	Descrição
Sequência de imagem	A imagem de contêiner que será usada.
Sequência de <code>imagePullPolicy</code>	Configuração que controla quando o kubelet tenta extrair a imagem especificada. O padrão é <code>IfNotPresent</code> .
<code>livenessProbe</code> <code>TracingProbe</code>	Configurações que controlam a análise de vivacidade.
<code>readinessProbe</code> <code>TracingProbe</code>	Configurações que controlam a análise de prontidão.

.spec.tracing.collector.livenessProbe

Configurações que controlam a análise de vivacidade.

Aparece em:

- [“.spec.tracing.collector”](#) na página 142

Campo	Descrição
Número inteiro <code>failureThreshold</code>	As falhas mínimas consecutivas para que a análise seja considerada com falha após ter sido bem-sucedida. O padrão é 1.
Número inteiro <code>initialDelaySeconds</code>	Número de segundos após o contêiner ter iniciado antes que as análises de vivacidade sejam iniciadas. O padrão é 10 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro <code>periodSeconds</code>	Com que frequência (em segundos) realizar a análise. O padrão é 10 segundos.
Número inteiro <code>successThreshold</code>	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida depois de ter falhado. O padrão é 1.

Campo	Descrição
Número inteiro timeoutSeconds	Número de segundos após os quais a análise atingirá o tempo limite. O padrão é 2 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.collector.readinessProbe

Configurações que controlam a análise de prontidão.

Aparece em:

- “[.spec.tracing.collector](#)” na página 142

Campo	Descrição
Número inteiro failureThreshold	As falhas mínimas consecutivas para que a análise seja considerada com falha após ter sido bem-sucedida. O padrão é 1.
Número inteiro initialDelaySeconds	Número de segundos após o contêiner ter iniciado antes que as análises de vivacidade sejam iniciadas. O padrão é 10 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro periodSeconds	Com que frequência (em segundos) realizar a análise. O padrão é 10 segundos.
Número inteiro successThreshold	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida depois de ter falhado. O padrão é 1.
Número inteiro timeoutSeconds	Número de segundos após os quais a análise atingirá o tempo limite. O padrão é 2 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.web

Configurações para o servidor da web do MQ.

Aparece em:

- “[.spec](#)” na página 128

Campo	Descrição
Booleano enabled	Indica se o servidor da web deve ou não ser ativado. Usa false como padrão.

.status

O estado observado do QueueManager.

Aparece em:

- “[QueueManager](#)” na página 128

Campo	Descrição
Sequência de adminUiUrl	URL para a IU do Administrador.
availability Disponibilidade	Status de disponibilidade para o Gerenciador de Filas.
conditions Matriz QueueManagerStatusConditio n	As condições representam as últimas observações disponíveis do estado do Gerenciador de Filas.

Campo	Descrição
endpoints Matriz QueueManagerStatusEndpoint	Informações sobre os terminais que este Gerenciador de Filas está expondo, como terminais de API ou IU.
Sequência de name	O nome do Gerenciador de Filas.
Sequência de phase	Fase do estado do Gerenciador de Filas.
versions QueueManagerStatusVersion	Versão do MQ que está sendo usada e outras versões disponíveis no IBM Entitled Registry.

.status.availability

Status de disponibilidade para o Gerenciador de Filas.

Aparece em:

- [“.status” na página 143](#)

Campo	Descrição
Booleano initialQuorumEstablished	Se um quorum inicial foi ou não estabelecido para NativeHA.

.status.conditions

QueueManagerStatusCondition define as condições do Gerenciador de Filas.

Aparece em:

- [“.status” na página 143](#)

Campo	Descrição
Sequência de lastTransitionTime	Última vez que a condição transitava de um status para outro.
Sequência de message	Mensagem legível por humanos indicando detalhes sobre a última transição.
Sequência de reason	Razão para a última transição deste status.
Sequência de status	Status da condição.
Sequência de type	Tipo de condição.

.status.endpoints

QueueManagerStatusEndpoint define os terminais para o QueueManager.

Aparece em:

- [“.status” na página 143](#)

Campo	Descrição
Sequência de name	Nome do terminal.
Sequência de type	O tipo de endpoint, por exemplo, 'UI' para um endpoint de UI, 'API' para um endpoint de API, 'OpenAPI' para documentação de API.
Sequência de uri	URI para o terminal.

.status.versions

Versão do MQ que está sendo usada e outras versões disponíveis no IBM Entitled Registry.

Aparece em:

- [“.status”](#) na página 143

Campo	Descrição
<code>available</code> <code>QueueManagerStatusVersionA</code> <code>available</code>	Outras versões do MQ disponíveis no IBM Entitled Registry.
Sequência de <code>reconciled</code>	A versão específica do IBM MQ que está sendo usada. Se uma imagem customizada for especificada, isso poderá não corresponder à versão do MQ realmente sendo usada.

.status.versions.available

Outras versões do MQ disponíveis no IBM Entitled Registry.

Aparece em:

- [“.status.versions”](#) na página 145

Campo	Descrição
<code>channels</code>	Canais que estão disponíveis para atualizar automaticamente a versão do MQ.
<code>versions</code> Matriz de Versões	Versões específicas do MQ que estão disponíveis.

.status.versions.available.versions

`QueueManagerStatusVersion` define uma versão do MQ.

Aparece em:

- [“.status.versions.available”](#) na página 145

Campo	Descrição
Sequência de <code>name</code>	<code>name</code> da versão para esta versão de <code>QueueManager</code> . Estes são valores válidos para o campo <code>spec.version</code> .

Condições de status para o QueueManager (mq.ibm.com/v1beta1)

Os campos **status.conditions** são atualizados para refletir a condição do recurso `QueueManager`. Em geral, as condições descrevem situações anormais. Um gerenciador de filas em um estado pronto e em funcionamento não tem as condições **Error** ou **Pending**. Ele pode ter algumas condições de aviso **Warning**.

O suporte às condições foi introduzido em IBM MQ Operator 1.2.

As condições a seguir são definidas para um recurso `QueueManager`:

Tabela 1. Condições de status do gerenciador de filas

Componente	Tipo de condição	Código de razão	Aviso de mensagem
QueueManager ⁷	Pendente	Criando	O gerenciador de filas MQ está sendo implementado
	Pendente	OidcPending	O gerenciador de filas MQ está aguardando o registro de cliente do OIDC
	Erro	Falhou	O gerenciador de filas MQ teve uma falha ao implementar
	Aviso	UnsupportedVersion	⁸ Um operando foi instalado por um operador que não é suportado na versão OCP <ocp_version>. Este operando não é suportado.
	Aviso	EUSSupport	⁹ Um operando do EUS <mq_version> foi instalado mas está sendo gerenciado por um operador que não se qualifica para a duração de suporte estendido. Esse operando não se qualifica para a duração do suporte estendido.
	Aviso	EUSSupport	¹⁰ Um operando do EUS <mq_version> foi instalado, mas a versão OCP 4<ocp_version> não se qualifica para a duração de suporte estendido. Esse operando não se qualifica para a duração do suporte estendido.
	Aviso	EUSSupport	¹¹ Um operando do EUS <mq_version> foi instalado mas a versão OCP <ocp_version> não se qualifica para a duração de suporte estendido. Esse operando é suportado conforme uma liberação de CD regular.

⁷ As condições Creating e Failed monitoram o progresso geral do gerenciador de filas. Se você estiver usando uma licença do IBM Cloud Pak for Integration e o console da web MQ estiver ativado, então a condição OidcPending registra o status do gerenciador de filas enquanto aguarda o registro do cliente OIDC para ser concluído com o IAM.

Tabela 1. Condições de status do gerenciador de filas (continuação)

Componente	Tipo de condição	Código de razão	Aviso de mensagem
Pod ¹²	Pendente	PodPending	O pod para o gerenciador de filas MQ está sendo implementado
	Erro	PodFailed	O pod para o gerenciador de filas MQ está sendo implementado
Armazenamento ¹³	Pendente	StoragePending	O armazenamento para o gerenciador de filas MQ está sendo provisionado
	Aviso	StorageEphemeral	Uso do armazenamento efêmero para um gerenciador de filas MQ de produção
	Erro	StorageFailed	Falha de provisionamento do armazenamento do gerenciador de filas MQ

Multi Construindo seu próprio contêiner e código de implementação do IBM MQ

Desenvolva um contêiner autoconstruído. Essa é a solução de contêiner mais flexível, mas requer que você tenha um bom conhecimento técnico sobre configuração de contêineres e que você seja o "proprietário" do contêiner resultante.

Antes de começar

Antes de desenvolver o seu próprio contêiner, considere se é possível, em vez disso, usar um dos contêineres predefinidos fornecidos pela IBM. Consulte [IBM MQ em contêineres](#)

Sobre esta tarefa

Ao empacotar o IBM MQ como uma imagem de contêiner, as mudanças em seu aplicativo podem ser implementadas para testar e preparar sistemas de forma rápida e fácil. Isso pode ser um grande benefício para a entrega contínua em sua empresa.

⁸ Operador 1.4.0 e mais recente

⁹ Operador 1.4.0 e mais recente

¹⁰ Operador 1.4.0 e mais recente

¹¹ Apenas Operador 1.3.0

¹² As condições do Pod monitoram o status dos pods durante a implementação de um gerenciador de filas. Se você vir qualquer condição PodFailed, então a condição geral do gerenciador de fila também será configurada como Failed

¹³ As condições de armazenamento monitoram o progresso (condição StoragePending) de solicitações para criar volumes para armazenamento persistente, e relatar erros de ligação e outras falhas. Se ocorrer algum erro durante o fornecimento de armazenamento, a condição StorageFailed será incluída na lista de condições e a condição geral do gerenciador de filas também será configurada como Failed

Procedimento

- [“Planejando a sua própria imagem do gerenciador de filas do IBM MQ usando um contêiner”](#) na página 148
- [“Construindo uma imagem de contêiner de gerenciador de filas do IBM MQ”](#) na página 148
- [“Executando aplicativos de ligação local em contêineres separados”](#) na página 151

Conceitos relacionados

[IBM MQ em contêineres](#)

Multi

Planejando a sua própria imagem do gerenciador de filas do IBM MQ usando um contêiner

Há vários requisitos a serem considerados ao executar um gerenciador de filas do IBM MQ em um contêiner. A imagem de contêiner de amostra fornece uma maneira de manipular esses requisitos, mas se você quiser usar a sua própria imagem, será necessário considerar como esses requisitos são manipulados.

Supervisão do processo

Ao executar um contêiner, você está essencialmente executando um único processo (PID 1 dentro do contêiner) que poderá, posteriormente, fazer spawn de processos-filhos.

Se o processo principal terminar, o tempo de execução do contêiner parará o contêiner. Um gerenciador de filas do IBM MQ requer vários processos para estar em execução em segundo plano.

Por essa razão, é necessário assegurar-se de que seu principal processo permanece ativo enquanto o gerenciador de filas estiver em execução. É necessário verificar se o gerenciador de filas está ativo a partir deste processo, por exemplo, executando consultas administrativas.

Preenchendo /var/mqm

Os contêineres devem ser configurados com `/var/mqm` como um volume.

Ao fazer isso, o diretório do volume estará vazio quando o contêiner for iniciado pela primeira vez. Esse diretório geralmente é preenchido no momento da instalação, mas a instalação e o tempo de execução são ambientes separados ao usar um contêiner.

Para resolver isso, quando o seu contêiner for iniciado, será possível usar o comando `crtmqdir` para preencher `/var/mqm` quando ele for executado pela primeira vez

Segurança do contêiner

A fim de minimizar os requisitos de segurança de tempo de execução, as imagens de contêiner de amostra são instaladas usando a instalação que não pode ser descompactada do IBM MQ. Isso garante que nenhum bit `setuid` esteja configurado e que o contêiner não precise usar a escalada de privilégio. Alguns sistemas de contêiner definem quais IDs de usuário você é capaz de usar, e a instalação não descompactável não faz nenhuma suposição sobre os usuários do sistema operacional disponíveis.

Multi

Construindo uma imagem de contêiner de gerenciador de filas do IBM MQ

Use essas informações para construir uma imagem de contêiner de amostra para a execução de um gerenciador de filas do IBM MQ em um contêiner.

Sobre esta tarefa

Em primeiro lugar, você constrói uma imagem base contendo um sistema de arquivos do Red Hat Universal Base Image e uma instalação limpa do IBM MQ.

Em segundo lugar, você constrói na base outra camada de imagem de contêiner, que inclui algumas configurações do IBM MQ para permitir o ID do usuário básico e a segurança de senha.

Por fim, você executa um contêiner usando essa imagem como o sistema de arquivos dele, com o conteúdo de `/var/mqm` fornecido por um volume específico de contêiner no sistema de arquivos do host.

Procedimento

- Para obter informações sobre como construir uma imagem de contêiner de amostra para a execução de um gerenciador de filas do IBM MQ em um contêiner, consulte os subtópicos a seguir:
 - [“Construindo uma imagem do gerenciador de filas do IBM MQ de base de amostra”](#) na página 149
 - [“Construindo uma imagem do gerenciador de filas do IBM MQ de amostra configurada”](#) na página 149

Multi Construindo uma imagem do gerenciador de filas do IBM MQ de base de amostra

Para usar o IBM MQ em sua própria imagem de contêiner, é necessário inicialmente construir uma imagem base com uma instalação limpa do IBM MQ. As etapas a seguir mostram como construir uma imagem base de amostra usando o código de amostra hospedado no GitHub.

Procedimento

- Use os makefiles fornecidos no [repositório mq-container do GitHub](#) para construir a sua imagem de contêiner de produção.

Siga as instruções em [Construindo uma imagem de container](#) no GitHub. Se você planeja configurar o acesso seguro usando o Red Hat OpenShift Container Platform "restricted" Security Context Constraint (SCC), deverá usar o pacote 'Não Instalar' IBM MQ .

Resultados

Agora você tem uma imagem de contêiner base com o IBM MQ instalada.

Agora você está pronto para [construir uma imagem de gerenciador de filas IBM MQ configurada por amostra](#).

Multi Construindo uma imagem do gerenciador de filas do IBM MQ de amostra configurada

Depois de ter construído sua imagem de contêiner base genérica do IBM MQ, é necessário aplicar a própria configuração para permitir o acesso seguro. Para isso, você cria sua própria camada de imagem de contêiner, usando a imagem genérica como pai.

Antes de começar

V 9.2.0 Essa tarefa supõe que, quando você construiu sua imagem de gerenciador de filas IBM MQ da base de amostra, você usou o pacote "No-Install" do IBM MQ. Caso contrário, não é possível configurar o acesso seguro usando o Security Context Constraint (SCC) do Red Hat OpenShift Container Platform "restrito". O SCC "restrito", que é usado por padrão, usa IDs aleatórios de usuários e previne a escalada de privilégios, mudando para um usuário diferente. O instalador baseado em RPM tradicional do IBM MQ conta com um usuário e grupo `mqm` e também usa `bits setuid` em programas executáveis. Em IBM MQ 9.2, ao usar o pacote "No-Install" do IBM MQ, não há mais um usuário `mqm`, nem um grupo `mqm`.

Procedimento

1. Crie um novo diretório e inclua um arquivo denominado `config.mqsc`, com o seguinte conteúdo:

```
DEFINE QLOCAL(EXAMPLE.QUEUE.1) REPLACE
```

Observe que o exemplo anterior usa ID do usuário e autenticação de senha simples. No entanto, é possível aplicar qualquer configuração de segurança que sua empresa necessite.

2. Crie um arquivo denominado `Dockerfile`, com o seguinte conteúdo:

```
FROM mq
COPY config.mqsc /etc/mqm/
```

3. Construa a sua imagem de contêiner customizada usando o comando a seguir:

```
docker build -t mymq .
```

em que "." é o diretório contendo os dois arquivos que você acabou de criar.

O Docker cria então um contêiner provisório usando essa imagem e executa os comandos restantes.

Nota: No Red Hat Enterprise Linux (RHEL), use o comando **docker** (RHEL V7) ou **podman** (RHEL V7 ou RHEL V8). No Linux, será necessário executar os comandos **docker**, com **sudo** no início do comando, para obter privilégios adicionais.

4. Execute sua nova imagem customizada para criar um novo contêiner, com a imagem do disco recém-criada.

A nova camada de imagem não especificou nenhum comando específico a ser executado, de forma que ele foi herdado da imagem pai. O ponto de entrada do pai (o código está disponível no GitHub):

- Cria um gerenciador de filas
- Inicia o gerenciador de filas
- Cria um listener padrão
- Em seguida, executa qualquer comando MQSC de `/etc/mqm/config.mqsc`.

Emita os seguintes comandos para executar a nova imagem customizada:

```
docker run \
  --env LICENSE=accept \
  --env MQ_QMGR_NAME=QM1 \
  --volume /var/example:/var/mqm \
  --publish 1414:1414 \
  --detach \
  mymq
```

em que o:

Primeiro parâmetro env

Transmite uma variável de ambiente no contêiner, que reconhece sua aceitação da licença para IBM WebSphere MQ. Também é possível configurar a variável `LICENSE` para visualizar a licença.

Consulte [IBM MQ informações de licença](#) para obter detalhes adicionais sobre as licenças do IBM MQ.

Segundo parâmetro env

Configura o nome do gerenciador de filas que você está usando.

Parâmetro Volume

Diz ao contêiner que tudo o que MQ escreve para `/var/mqm` deve realmente ser gravado em `/var/example` no host.

Esta opção significa que é possível facilmente excluir o contêiner posteriormente e ainda manter quaisquer dados persistentes. Esta opção também torna mais fácil para visualizar arquivos de log.

Parâmetro Publish

Mapeia as portas no sistema host para portas no contêiner. O contêiner é executado por padrão com seu próprio endereço IP interno, o que significa que é necessário mapear especificamente quaisquer portas que você deseja expor.

Neste exemplo, isso significa a porta de mapeamento 1414 no host para a porta 1414 no contêiner.

Parâmetro Detach

Executa o contêiner em segundo plano.

Resultados

Você construiu uma imagem de contêiner configurada e pode visualizar contêineres em execução usando o comando **docker ps**. É possível visualizar os processos do IBM MQ em execução em seu contêiner usando o comando **docker top**.



Atenção:

É possível visualizar os logs de um contêiner usando o comando **docker logs \$ {CONTAINER_ID}**.

Como proceder a seguir

- Seu contêiner poderá ter falhado se não for mostrado ao usar o comando **docker ps**. É possível ver contêineres com falha usando o comando **docker ps -a ..**
- Ao usar o comando **docker ps -a**, o ID do contêiner é exibido. Esse ID também foi impresso quando você emitiu o comando **docker run**.
- É possível visualizar os logs de um contêiner usando o comando **docker logs \${CONTAINER_ID}**.

Multi

Executando aplicativos de ligação local em contêineres separados

Com o compartilhamento do namespace do processo entre contêineres no Docker, é possível executar aplicativos que requeiram uma conexão de ligação local com o IBM MQ em contêineres separados do gerenciador de filas do IBM MQ.

Sobre esta tarefa

Essa funcionalidade é suportada no IBM MQ 9.0.3 e gerenciadores de filas mais recentes.

Deve-se aderir às restrições a seguir:

- Deve-se compartilhar o namespace de PID de contêineres usando o argumento `--pid`.
- Deve-se compartilhar o namespace de IPC de contêineres usando o argumento `--ipc`.
- Para isso:
 1. Compartilhe o namespace de UTS de contêineres com o host usando o argumento `--uts` ou
 2. Assegure que os contêineres tenham o mesmo nome do host usando o argumento `-h` ou `--hostname`.
- Deve-se montar o diretório de dados do IBM MQ em um volume que esteja disponível para todos os contêineres no diretório do `/var/mqm`.

É possível experimentar essa funcionalidade concluindo as etapas a seguir em um sistema Linux que já tenha o Docker instalado.

O exemplo a seguir usa a imagem de contêiner do IBM MQ de amostra. É possível localizar detalhes dessa imagem no [Github](#).

Procedimento

1. Crie um diretório temporário para agir como seu volume, emitindo o comando a seguir:

```
mkdir /tmp/dockerVolume
```

2. Crie um gerenciador de filas (QM1) em um contêiner, com o nome `sharedNamespace`, emitindo o comando a seguir:

```
docker run -d -e LICENSE=accept -e MQ_QMGR_NAME=QM1 --volume /tmp/dockerVol:/mnt/mqm
--uts host --name sharedNamespace ibmcom/mq
```

3. Inicie um segundo contêiner chamado `secondaryContainer`, com base em `ibmcom/mq`, mas não crie um gerenciador de filas, emitindo o comando a seguir:

```
docker run --entrypoint /bin/bash --volumes-from sharedNamespace --pid
container:sharedNamespace --ipc container:sharedNamespace --uts host --name
secondaryContainer -it --detach ibmcom/mq
```

4. Execute o comando **dspm** no segundo contêiner, para ver o status para ambos os gerenciadores de filas, emitindo o comando a seguir:

```
docker exec secondaryContainer dspmq
```

5. Execute o comando a seguir para processar comandos MQSC com relação ao gerenciador de filas em execução no outro contêiner:

```
docker exec -it secondaryContainer runmqsc QM1
```

Resultados

Agora você tem aplicativos locais em execução em contêineres separados e pode executar com sucesso comandos como **dspm**, **amqsput**, **amqsget** e **runmqsc** como ligações locais com o gerenciador de filas QM1 do contêiner secundário.

Se você não vir o resultado esperado, veja [“Resolução de problemas de seus aplicativos de namespace”](#) na página 152 para obter mais informações.

Resolução de problemas de seus aplicativos de namespace

Ao usar namespaces compartilhados, deve-se garantir o compartilhamento de todos os namespaces (IPC, PID e UTS/nome de host) e volumes montados, caso contrário, os aplicativos não funcionarão.

Veja [“Executando aplicativos de ligação local em contêineres separados”](#) na página 151 para obter uma lista de restrições que se deve seguir.

Se seu aplicativo não atender todas as restrições listadas, você poderá encontrar problemas em que o contêiner inicia, mas a funcionalidade esperada não funciona.

A lista a seguir descreve algumas causas comuns e o comportamento que você provavelmente verá se tiver esquecido de atender uma das restrições.

- Se você se esquecer de compartilhar o namespace (UTS/PID/IPC) ou o nome do host dos contêineres e montar o volume, seu contêiner poderá ver o gerenciador de filas, mas não interagirá com ele.

- Para comandos **dspm**, você vê o seguinte:

```
docker exec container dspmq
QMNAME(QM1)                STATUS(Status not available)
```

- Para comandos **runmqsc** ou outros comandos que tentam se conectar ao gerenciador de filas, você provavelmente receberá uma mensagem de erro AMQ8146:

```
docker exec -it container runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2025.
Starting MQSC for queue manager QM1.
AMQ8146: IBM MQ queue manager not available
```

- Se você compartilha todos os namespaces necessários mas você não monta um volume compartilhado para o diretório `/var/mqm` e você tem um caminho de dados IBM MQ válido, então seus comandos também recebem a mensagens de erro AMQ8146.

No entanto, **dspmq** não é capaz de ver seu gerenciador de filas e em seu lugar retorna uma resposta em branco:

```
docker exec container dspmq
```

- Se você compartilha todos os namespaces necessários mas você não monta um volume compartilhado para o diretório `/var/mqm` e você não tem um caminho de dados IBM MQ válido (ou nenhum caminho de dados IBM MQ), então você verá vários erros, pois o caminho de dados é um componente chave de uma instalação IBM MQ Sem o caminho de dados, o IBM MQ não pode operar.

Se você executar qualquer um dos comandos a seguir e vir respostas semelhantes às mostradas nesses exemplos, será necessário verificar se você montou o diretório ou criou um diretório de dados do IBM MQ:

```
docker exec container dspmq
'No such file or directory' from /var/mqm/mqs.ini
AMQ6090: IBM MQ was unable to display an error message FFFFFFFF.
AMQffff

docker exec container dspmqver
AMQ7047: An unexpected error was encountered by a command. Reason code is 0.

docker exec container mqrc
<file path>/mqrc.c[1152]
lpiObtainQMDetails --> 545261715

docker exec container crtmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container strmqm QM1
AMQ6239: Permission denied attempting to access filesystem location '/var/mqm'.
AMQ7002: An error occurred manipulating a file.

docker exec container endmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container dltmqm QM1
AMQ7002: An error occurred manipulating a file.

docker exec container strmqweb
<file path>/mqrc.c[1152]
lpiObtainQMDetails --> 545261715
```

CP4I Criando o grupo de HA nativa se estiver criando seus próprios contêineres

Deve-se criar, configurar e iniciar três gerenciadores de fila para criar o grupo de HA nativa

Sobre esta tarefa

O método recomendado para criar uma solução de HA nativa é usar o operador IBM MQ (consulte [HA nativa](#)). Como alternativa, se você criar seus próprios contêineres, será possível seguir essas instruções

Para criar um grupo de HA nativa, você cria três gerenciadores de fila em três nós com seus tipos de log configurados como `log replication`. Em seguida, você edita o arquivo `qm.ini` para cada gerenciador de filas para incluir os detalhes de conexão para cada um dos três nós para que eles possam replicar dados do log entre si.

Em seguida, deve-se iniciar os três gerenciadores de filas para que eles possam verificar se as três instâncias podem se comunicar entre si e determinar qual delas será a instância ativa e quais serão as réplicas.

Procedimento

1. Em cada um dos três nós, crie um gerenciador de filas, especificando um tipo de registro de réplica de log e fornecendo um nome exclusivo para cada instância de registro. Cada gerenciador de filas tem o mesmo nome:

```
crtmqm -lr instance_name qmname
```

Por exemplo:

```
node 1> crtmqm -lr qm1_inst1 qm1
node 2> crtmqm -lr qm1_inst2 qm1
node 3> crtmqm -lr qm1_inst3 qm1
```

2. Na criação bem-sucedida de cada gerenciador de filas, uma sub-rotina adicional chamada `NativeHALocalInstance` é incluída no arquivo de configuração do gerenciador de filas, `qm.ini`. Um atributo `Name` é incluído na sub-rotina, especificando o nome da instância fornecido

É possível incluir opcionalmente os atributos a seguir na sub-rotina `NativeHALocalInstance` no arquivo `qm.ini`:

KeyRepository

O local do repositório de chaves que contém o certificado digital a ser usado para proteção do tráfego de replicação do log. O local é fornecido no formato de raiz, ou seja, ele inclui o caminho completo e o nome do arquivo sem uma extensão. Se o atributo de sub-rotina `KeyRepository` for omitido, os dados de replicação de log serão trocados entre as instâncias em texto simples.

CertificateLabel

O rótulo de certificado que identifica o certificado digital a ser usado para proteção do tráfego de replicação do log. Se `KeyRepository` for fornecido, mas `CertificateLabel` for omitido, um valor padrão `ibmwebsphermqueue_manager` será usado.

CipherSpec

O MQ CipherSpec a ser usado para proteger o tráfego de replicação do log. Se esse atributo de sub-rotina for fornecido, `KeyRepository` também deverá ser fornecido. Se `KeyRepository` for fornecido, mas `CipherSpec` for omitido, um valor padrão `ANY` será usado.

LocalAddress

O endereço de interface de rede local que aceita o tráfego de replicação do log. Se esse atributo de sub-rotina for fornecido, ele identificará a interface de rede local e / ou porta usando o formato "[addr] [(port)]". O endereço de rede pode ser especificado como um nome do host, IPv4 decimal pontuado ou IPv6 formato hexadecimal. Se esse atributo for omitido, o gerenciador de filas tentará ligar a todas as interfaces de rede, ele usará a porta especificada na `ReplicationAddress` na sub-rotina `NativeHAInstances` correspondente ao nome da instância local.

HeartbeatInterval

O intervalo de pulsação define com que frequência em milissegundos uma instância ativa de um gerenciador de filas de HA nativa envia uma pulsação de rede. O intervalo válido do valor de intervalo de pulsação é de 500 (0,5 segundos) a 60.000 (1 minuto). Um valor fora desse intervalo causa uma falha na inicialização do gerenciador de filas. Se esse atributo for omitido, um valor padrão de 5.000 (5 segundos) será usado. Cada instância deve usar o mesmo intervalo de pulsação.

HeartbeatTimeout

O tempo limite de pulsação define quanto tempo uma instância de réplica de um gerenciador de filas de HA nativa aguarda antes de decidir que a instância ativa está não responsiva. O intervalo válido do valor de tempo limite do intervalo de pulsação é de 500 (0,5 segundos) a 120.000 (2 minutos). O valor do tempo limite de pulsação deve ser maior ou igual ao intervalo de pulsação.

Um valor inválido gera uma falha na inicialização do gerenciador de filas. Se este atributo for omitido uma réplica aguardará 2 x `HeartbeatInterval` antes de iniciar o processo para eleger uma nova instância ativa. Cada instância deve usar o mesmo tempo limite de pulsação.

RetryInterval

O intervalo de nova tentativa define com que frequência em milissegundos um gerenciador de filas de HA nativa deve tentar novamente um link de replicação com falha. O intervalo válido do intervalo de nova tentativa é de 500 (0,5 segundos) a 120.000 (2 minutos). Se este atributo for omitido uma réplica aguardará por 2 x HeartbeatInterval antes de tentar novamente um link de replicação com falha.

3. Edite o arquivo `qm.ini` para cada gerenciador de filas e inclua detalhes da conexão.. Você inclui três sub-rotinas `NativeHAInstance`, uma para cada instância do gerenciador de filas no grupo de HA nativa (incluindo a instância local). Inclua os seguintes atributos:

Nome

Especifique o nome da instância usado ao criar a instância do gerenciador de filas.

ReplicationAddress

Especifique o nome do host, IPv4 decimal pontilhado ou IPv6 endereço do formato hexadecimal da instância. É possível especificar o endereço como um nome do host, decimal pontilhado IPv4 ou endereço de formato hexadecimal IPv6. O endereço de replicação deve ser resolvível e roteável de cada instância no grupo. O número da porta a ser usado para a replicação de log deve ser especificado entre colchetes, por exemplo:

```
ReplicationAddress=host1.example.com(4444)
```

Nota: As sub-rotinas `NativeHAInstance` são idênticas em cada instância e podem ser fornecidas usando a configuração automática (`crtmqm -ii`).

4. Inicie cada uma das três instâncias:

```
strmqm QMgrName
```

Quando as instâncias são iniciadas, elas se comunicam para verificar se todas as três instâncias estão em execução, em seguida, decidir qual das três é a instância ativa, enquanto as outras duas instâncias continuam a ser executadas como réplicas.

Exemplo

O exemplo a seguir mostra a seção de um arquivo `qm.ini` especificando os detalhes de HA nativa necessários para uma das três instâncias:

```
NativeHALocalInstance:  
  LocalName=node-1  
  
NativeHAInstance:  
  Name=node-1  
  ReplicationAddress=host1.example.com(4444)  
NativeHAInstance:  
  Name=node-2  
  ReplicationAddress=host2.example.com(4444)  
NativeHAInstance:  
  Name=node-3  
  ReplicationAddress=host3.example.com(4444)
```

Kubernetes Considerações para executar a sua própria atualização contínua de um gerenciador de filas de alta disponibilidade nativo

Qualquer atualização na versão IBM MQ ou especificação de Pod para um gerenciador de filas de alta disponibilidade nativo exigirá seja executada uma atualização contínua das instâncias do gerenciador de filas. O IBM MQ Operator manipula isso para você automaticamente, mas se você está criando seu próprio código de implementação, há algumas considerações importantes.

Nota: O gráfico Helm de amostra inclui um script de shell para executar uma atualização contínua, mas o script **não** é adequado para uso em produção, pois não aborda as considerações neste tópico.

No Kubernetes, os recursos `StatefulSet` são usados para gerenciar a inicialização ordenada e as atualizações contínuas. Parte do procedimento de inicialização é iniciar cada Pod individualmente,

esperar que fique pronto e depois passar para o próximo Pod. Isso não funcionará para a HA nativa, pois todos os Pods precisam ser iniciados para que possam realizar uma eleição de líder. Portanto, o campo `.spec.podManagementPolicy` no `StatefulSet` precisa ser configurado como `Parallel`. Isso também significa que todos os Pods também serão atualizados em paralelo, o que é particularmente indesejável. Por esta razão, o `StatefulSet` também deve usar a estratégia de atualização `OnDelete`.

A incapacidade de usar o código de atualização contínua `StatefulSet` gera a necessidade de um código de atualização contínua customizado, que deve considerar o seguinte:

- Procedimento geral de atualização contínua
- Minimizando o tempo de inatividade atualizando os pods na melhor ordem
- Manipulando mudanças no estado do cluster
- Manipulando Erros
- Manipulando problemas de sincronização

Procedimento geral de atualização contínua

O código de atualização contínua deve aguardar que cada instância mostre um status de `REPLICA` de `dspm`. Isso significa que a instância executou algum nível de inicialização (por exemplo, o contêiner foi iniciado e os processos do MQ estão em execução), mas ainda não conseguiu necessariamente se comunicar com as outras instâncias. Por exemplo: o Pod A é reiniciado e assim que está no estado `REPLICA`, o Pod B é reiniciado. Depois que o Pod B iniciar com a nova configuração, ele poderá se comunicar com o Pod A e formar o quorum, e A ou B se tornará a nova instância ativa.

Como parte disso, é útil ter um atraso depois que cada Pod atingir o estado `REPLICA`, para permitir que ele se conecte aos seus pares e estabeleça o quorum.

Minimizando o tempo de inatividade atualizando os pods na melhor ordem

O código de atualização contínua deve excluir os pods um por vez, começando com os pods que estão em um estado de erro conhecido, seguidos por todos os pods que não foram iniciados com êxito. O Pod de gerenciador de filas ativo geralmente deve ser atualizado por último.

Também é importante pausar a exclusão de Pods se a última atualização resultou em um Pod entrando em um estado de erro conhecido. Isso impede o lançamento de uma atualização interrompida em todos os pods. Por exemplo, isso pode acontecer se o pod for atualizado para usar uma nova imagem de contêiner que não está acessível (ou contém um erro de digitação).

Manipulando mudanças no estado do cluster

O código de atualização contínua precisa reagir adequadamente às alterações em tempo real no estado do cluster. Por exemplo, um dos Pods do gerenciador de filas pode ser removido devido a uma reinicialização do Nó ou devido à pressão do Nó. É possível que um pod removido não seja reprogramado imediatamente se o cluster estiver ocupado. Nesse caso, o código de atualização contínua precisaria aguardar o momento adequado antes de reiniciar qualquer outro pod.

Manipulando Erros

O código de atualização contínua precisa ser robusto a falhas ao chamar a API do Kubernetes e com relação a outros comportamentos inesperados de cluster.

Além disso, o próprio código de atualização contínua precisa ser tolerante a ser reiniciado. Uma atualização contínua pode ser de longa duração e o código pode precisar ser reiniciado.

Manipulando problemas de sincronização

O código de atualização contínua precisa verificar as revisões de atualização do pod para garantir que o pod foi reiniciado. Isso evita problemas de sincronização em que um pod pode indicar que está "Iniciado", mas na verdade ainda não foi encerrado.

Conceitos relacionados

“Escolhendo como você deseja usar o IBM MQ em contêineres” na página 5

Há várias opções para usar o IBM MQ em contêineres: é possível optar por usar o IBM MQ Operator, que usa imagens de contêiner pré-empacotadas ou construir suas próprias imagens e código de implementação.

CP4I Visualizando o status de gerenciadores de filas de HA nativa para contêineres customizados

Para contêineres customizados, é possível visualizar o status das instâncias de HA nativa usando o comando **dspmqr**.

Sobre esta tarefa

É possível usar o comando **dspmqr** para visualizar o status operacional de uma instância do gerenciador de filas em um nó. As informações retornadas estarão de acordo com o estado ativo ou de réplica da instância. As informações fornecidas pela instância ativa são definitivas; as informações de nós de réplica podem estar desatualizadas.

Você pode executar as seguintes ações:

- Visualizar se a instância do gerenciador de filas no nó atual está ativa ou é uma réplica.
- Visualizar o status operacional de HA nativa da instância no nó atual.
- Visualizar o status operacional de todas as três instâncias em uma configuração de HA nativa.

Os campos de status a seguir são usados para relatar o status de configuração da HA nativa:

ROLE

Especifica a função atual da instância e é Active, Replica ou Unknown.

INSTÂNCIA

O nome fornecido para esta instância do gerenciador de filas quando ela foi criada usando a opção **-lr** do comando **crtmqm**.

INSYNC

Indica se a instância está apta a assumir o controle como a instância ativa, se necessário.

QUORUM

Relata o status de quorum no formato *number_of_instances_in-sync/number_of_instances_configured*.

REPLADDR

O endereço de replicação da instância do gerenciador de filas.

CONNECTV

Indica se o nó está conectado à instância ativa.

BACKLOG

Indica o número de KB que a instância está atrasada.

CONNINST

Indica se a instância nomeada está conectada a esta instância.

ALTDAT

Indica a data da última atualização dessas informações (em branco se elas nunca foram atualizadas).

ALTTIME

Indica o horário da última atualização dessas informações (em branco se elas nunca foram atualizadas).

Procedimento

- Para determinar se uma instância do gerenciador de filas está em execução como a instância ativa ou como uma réplica:

```
dspmqr -o status -m QMgrName
```

Uma instância ativa de um gerenciador de filas denominado BOB relataria o status a seguir:

```
QMNAME(BOB)          STATUS(Running)
```

Uma instância de réplica de um gerenciador de filas denominado BOB relataria o status a seguir:

```
QMNAME(BOB)          STATUS(Replica)
```

Uma instância inativa relataria o status a seguir:

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- Para determinar o status operacional de HA nativa da instância no nó atual:

```
dspmqr -o nativeha -m QMgrName
```

A instância ativa de um gerenciador de filas denominado BOB pode relatar o status a seguir:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Uma instância de réplica de um gerenciador de filas denominado BOB pode relatar o status a seguir:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Uma instância inativa de um gerenciador de filas denominado BOB pode relatar o status a seguir:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Para determinar o status operacional da HA nativa de todas as instâncias na configuração de HA nativa:

```
dspmqr -o nativeha -x -m QMgrName
```

Se você emitir este comando no nó que executa a instância ativa do gerenciador de filas BOB, o status a seguir poderá ser recebido:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Ao emitir este comando em um nó que executa uma instância de réplica do gerenciador de filas BOB, você recebe o status a seguir, que indica que uma das réplicas está ficando atrasada:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

Se você emitir este comando em um nó que executa uma instância inativa do gerenciador de filas BOB, o status a seguir poderá ser recebido:

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

Se você emitir o comando quando as instâncias ainda estiverem negociando quais estão ativas e quais são réplicas, receberá o status a seguir:

```
QMNAME(BOB)          STATUS(Negotiating)
```

Referências relacionadas

dspm

CP4I Finalizando gerenciadores de filas de alta disponibilidade nativa

É possível usar o comando **endmqm** para terminar um gerenciador de filas ativo ou de réplica que faz parte de um grupo de HA nativa

Procedimento

- Para terminar a instância ativa de um gerenciador de filas, consulte [Terminando Gerenciadores de Filas de HA Nativa](#) na seção Configuração desta documentação

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte seu representante local do IBM para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a um IBM produto, programa ou serviço não se destina a estado ou significa que apenas esse produto IBM, programas ou serviços possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado em substituição. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou aplicativos de patentes pendentes relativas aos assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum sobre tais patentes. É possível enviar pedidos de licença, por escrito, para:

Relações Comerciais e Industriais da IBM
Av. Pasteur, 138-146
Botafogo
Rio, RJ 10504-1785
U.S.A.

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

licença de propriedade intelectual
IBM World Trade Asia Corporation Licensing
IBM Japan, Ltd.
Minato-ku
Tóquio 103-8510, Japão

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. Periodicamente, são feitas nas informações aqui contidas; essas alterações serão incorporadas em futuras edições desta publicação. IBM pode aperfeiçoar e/ou alterar no produto(s) e/ou programa(s) descritos nesta publicação a qualquer momento sem aviso prévio.

Referências nestas informações a websites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais desse produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Av. Pasteur, 138-146
Av. Pasteur, 138-146

Botafogo
Rio de Janeiro, RJ
U.S.A.

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível para ele são fornecidos pela IBM sob os termos do IBM Customer Agreement, IBM Contrato de Licença do Programa Internacional ou qualquer contrato equivalente entre as partes.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disto, algumas medidas podem ter sido estimadas através de extrapolação. Os resultados reais podem variar. usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam somente metas e objetivos.

Essas informações contêm exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com os nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT :

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, uso, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de amostra são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas.

Se estiver visualizando estas informações em formato eletrônico, as fotografias e ilustrações coloridas poderão não aparecer.

Informações sobre a Interface de Programação

As informações da interface de programação, se fornecidas, têm como objetivo ajudá-lo a criar software de aplicação para uso com este programa.

Este manual contém informações sobre interfaces de programação desejadas que permitem que o cliente grave programas para obter os serviços do WebSphere MQ.

No entanto, estas informações também podem conter informações sobre diagnósticos, modificações e ajustes. As informações sobre diagnósticos, modificações e ajustes são fornecidas para ajudá-lo a depurar seu software aplicativo.

Importante: Não use este diagnóstico, modificação e ajuste de informações como uma interface de programação porque está sujeito a alterações.

Marcas comerciais

IBM, o logotipo IBM , ibm.com, são marcas registradas da IBM Corporation, registradas em várias jurisdições no mundo todo Uma lista atual de marcas registradas da IBM está disponível na Web em "Informações de copyright e marca registrada" www.ibm.com/legal/copytrade.shtml. Outros nomes de produtos e serviços podem ser marcas comerciais da IBM ou de outras empresas.

Microsoft e Windows são marcas comerciais da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Este produto inclui software desenvolvido pelo Projeto Eclipse (<https://www.eclipse.org/>).

Java e todas as marcas comerciais e logotipos baseados em Java são marcas comerciais ou marcas registradas da Oracle e/ou de suas afiliadas.



Part Number:

(1P) P/N: