

9.3

IBM MQ em contêineres

IBM

Nota

Antes de usar estas informações e o produto que elas suportam, leia as informações em [“Avisos” na página 221](#).

Esta edição se aplica à versão 9 liberação 3 do IBM® MQ e a todas as liberações e modificações subsequentes até que seja indicado de outra forma em novas edições.

Ao enviar informações para a IBM, você concede à IBM um direito não exclusivo de usar ou distribuir as informações da maneira que julgar apropriada, sem incorrer em qualquer obrigação para com você

© **Copyright International Business Machines Corporation 2007, 2024.**

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 em contêineres.....	6
Planejando o licenciamento do IBM MQ em contêineres.....	14
Dependências para o IBM MQ Operator.....	19
Permissões com escopo de cluster necessárias pelo IBM MQ Operator.....	20
Considerações de armazenamento para o IBM MQ Operator.....	21
IBM MQ Advanced for Developers Imagem de contêiner.....	23
Alta disponibilidade para IBM MQ em contêineres.....	25
Recuperação de desastre para o IBM MQ em contêineres.....	27
Planejando proteger o IBM MQ em contêineres.....	28
Planejando a escalabilidade e o desempenho do IBM MQ em contêineres.....	34
Usando o Operador IBM MQ.....	35
Histórico de liberação para o IBM MQ Operator.....	35
Verificando assinaturas de imagem.....	79
Migrando o IBM MQ para o IBM Cloud Pak for Integration.....	79
Instalando o IBM MQ Operator.....	103
Instalando o IBM MQ Operator 2.x em um ambiente de isolamento físico.....	110
Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform..	116
Desinstalando o IBM MQ Operator.....	119
Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas.....	120
Configurando gerenciadores de filas usando o IBM MQ Operator.....	134
Operando o IBM MQ usando o IBM MQ Operator.....	171
Resolução de problemas com o IBM MQ Operator.....	179
Referência da API do IBM MQ Operator.....	182
Construindo seu próprio contêiner e código de implementação do IBM MQ.....	207
Planejando a sua própria imagem do gerenciador de filas do IBM MQ usando um contêiner.....	208
Construindo uma imagem de contêiner de gerenciador de filas do IBM MQ.....	208
Executando aplicativos de ligação local em contêineres separados.....	211
Criando o grupo de HA nativa se estiver criando seus próprios contêineres.....	213
Avisos.....	221
Informações sobre a Interface de Programação.....	222
Marcas comerciais.....	223

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 [OpenShift Container Platform arquitetura](#)).

Procedimento

- [“Escolhendo como você deseja usar o IBM MQ em contêineres”](#) na página 5.
- [“Suporte para IBM MQ em contêineres”](#) na página 6.
- [“Considerações de armazenamento para o IBM MQ Operator”](#) na página 21.
- [“Alta disponibilidade para IBM MQ em contêineres”](#) na página 25.
- [“Recuperação de desastre para o IBM MQ em contêineres”](#) na página 27.
- [“Autenticação e autorização de usuário para o IBM MQ em contêineres”](#) na página 28.

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 estende a API Red Hat OpenShift Container Platform para incluir um novo recurso customizado `QueueManager`. 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 o [“Considerações para executar a sua própria atualização contínua de um gerenciador de filas de alta disponibilidade nativo”](#) na página 215

Alguns recursos do IBM MQ não são suportados ao usar o IBM MQ Operator. Consulte [“Suporte para IBM MQ em contêineres”](#) na página 6 para obter detalhes sobre o que é suportado ao usar o IBM MQ Operator

Observe que o IBM MQ Operator não suporta a instalação em um cluster do OpenShift com máquinas de cálculo de diversas arquiteturas

Construindo suas próprias imagens e código de implementação



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 [“Construindo seu próprio contêiner e código de implementação do IBM MQ”](#) na página 207.

Consulte [“Suporte para IBM MQ em contêineres”](#) na página 6 para obter detalhes do que é suportado ao construir sua própria imagem e código de implementação.

Referências relacionadas

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

Nem todos os recursos do IBM MQ estão disponíveis e são suportados da mesma maneira em contêineres

OpenShift CP4I CD CP4I-LTS Suporte para IBM MQ em contêineres

Nem todos os recursos do IBM MQ estão disponíveis e são suportados da mesma maneira em contêineres

Abaixo está uma tabela que mostra em detalhes como os recursos do IBM MQ são suportados com o IBM MQ Operator ou ao construir seus próprios contêineres e código de implementação.

Nota: As imagens de contêiner do IBM MQ pré-construídas no IBM Container Registry (icr.io e cp.icr.io) são suportadas e elegíveis para correções se usadas com o IBM MQ Operator.

Não é possível "fazer upgrade" da licença da imagem IBM MQ Advanced for Developers pré-construída para uma licença diferente. O IBM MQ Operator implementará diferentes imagens, dependendo de qual licença é selecionada

Nesta tabela, os termos a seguir se aplicam:

"Código de ativação do contêiner"

The executables **runmqserver**, **runmqintegrationserver**, **chkmqhealthy**, **chkmqready** and **chkmqstarted**. Esse código é fornecido como uma de amostra e é suportado apenas como parte dos contêineres pré-construídos quando usado com o IBM MQ Operator

	Usando IBM MQ Operator e uma IBM Cloud Pak for Integration licença	Usando IBM MQ Operator e uma IBM MQ Advanced licença	Usando IBM MQ Operator e uma IBM MQ Advanced for Developers licença	imagem IBM MQ Advanced for Developers pré-construída	Construir seu próprio contêiner
Plataformas suportadas	<p>Suportado apenas no Red Hat OpenShift Container Platform. As liberações do Red Hat OpenShift Container Platform não são mais suportadas pelo IBM MQ quando Red Hat para o suporte.</p> <p>Consulte “Suporte de versão para o IBM MQ Operator” na página 12 para obter mais detalhes.</p>		Disponível apenas no Red Hat OpenShift Container Platform , mas não suportado	Funciona em qualquer plataforma Docker, containerd ou cri-o, mas não é suportado Consulte Requisitos do sistema para IBM MQ para obter detalhes.	Qualquer plataforma Docker, containerd ou cri-o. Consulte Requisitos do sistema para IBM MQ para obter detalhes. A HA nativa é suportada apenas no Kubernetes ou no Red Hat OpenShift Container Platform A imagem de contêiner de amostra usa um Red Hat Universal Base Image (UBI), que inclui Linux® bibliotecas e utilitários usados pelo IBM MQ. A UBI é suportada pelo Red Hat quando executado no Red Hat OpenShift. O <i>código de ativação do contêiner</i> não é suportado.
Arquiteturas de CPU	Suportado no amd64, no s390x z /Linuxe no ppc64le Power Systems		Disponível no amd64, no s390x z /Linuxe no ppc64le Power Systems, mas não suportado..		Conforme o software IBM MQ .

	Usando IBM MQ Operator e uma IBM Cloud Pak for Integration licença	Usando IBM MQ Operator e uma IBM MQ Advanced licença	Usando IBM MQ Operator e uma IBM MQ Advanced for Developers licença	imagem IBM MQ Advanced for Developers pré-construída	Construir seu próprio contêiner
Duração do apoio	<p>IBM Cloud Pak for Integration - Long Term Support ou Continuous Delivery.¹</p> <p>CD operador e gerenciadores de filas são suportados até a próxima liberação IBM Cloud Pak for Integration CD ou CP4I-LTS .</p> <p>O operador e os gerenciadores de filas do CP4I-LTS são suportados até a próxima liberação do IBM Cloud Pak for Integration CP4I-LTS , além de um período de cortesia para permitir o upgrade</p>	<p>Continuous Delivery somente para o IBM MQ Operatore os gerenciadores de fila.</p> <p>Cada versão do IBM MQ Operator e do gerenciador de filas é suportada apenas até a próxima liberação CD ou LTS</p>	Não Suportado		<p>Conforme o software IBM MQ . Consulte IBM MQ FAQ para Long Term Support e Continuous Delivery liberações. O <i>código de ativação do contêiner</i> não é suportado.</p>

¹ O IBM MQ Operator é suportado como uma liberação do IBM MQ CD ou como uma liberação do CP4I-LTS :

- As imagens de contêiner IBM MQ 9.3.0.x implementadas com o IBM MQ Operator 2.0.x, quando usadas como parte do IBM Cloud Pak for Integration 2022.2.1, são elegíveis para o suporte CP4I-LTS . A liberação mais recente do Long Term Support (LTS) do IBM MQ Operator é 2.0.23e a mais recente LTS imagem de contêiner é 9.3.0.17-r3.
- As imagens de contêiner IBM MQ 9.3.5 implementadas com o IBM MQ Operator 3.1.x, quando usadas como parte do IBM Cloud Pak for Integration 2023.4.1, são elegíveis para o suporte CD . A liberação mais recente do Continuous Delivery (CD) do IBM MQ Operator é 3.1.3e a mais recente CD imagem de contêiner é 9.3.5.1-r2.

	Usando IBM MQ Operator e uma IBM Cloud Pak for Integration licença	Usando IBM MQ Operator e uma IBM MQ Advanced licença	Usando IBM MQ Operator e uma IBM MQ Advanced for Developers licença	imagem IBM MQ Advanced for Developers pré-construída	Construir seu próprio contêiner
Disponibilidade e da correção de segurança	Correções periódicas disponíveis como imagens de contêiner no IBM Container Registry				Correções para o software IBM MQ estão disponíveis como software no Fix Central . O <i>código de ativação do contêiner</i> não é suportado.
Disponibilidade e de correção temporária	Correções do gerenciador de filas disponíveis como software e uma construção de imagem customizada é necessária. As correções do IBM MQ Operator não estão disponíveis como correções temporárias		Não há correções temporárias disponíveis		As correções para o software IBM MQ estão disponíveis como software no Fix Central ou por meio do Suporte IBM O <i>código de ativação do contêiner</i> não é suportado.
Recurso: Advanced Message Security	Suportado. Observe que não é fácil usar a criptografia do lado do servidor, pois o IBM MQ Operator não permite diretamente especificar seu próprio armazenamento de chaves para o Advanced Message Security.		Disponível mas não suportado.		Suportado conforme o software IBM MQ , mas nenhuma amostra disponível.
Recurso: Managed File Transfer	Não disponível e não suportado. No entanto, é possível usar o IBM MQ Operator para fornecer um ou mais gerenciadores de filas de Coordenação, Comando ou Agente.			Não disponível e não suportado.	Suportado conforme o software IBM MQ , com <code>sample</code> para o agente.
Recurso: MQTT	Não disponível e não suportado.				Suportado conforme o software IBM MQ , mas nenhuma amostra disponível.

	Usando IBM MQ Operator e uma IBM Cloud Pak for Integration licença	Usando IBM MQ Operator e uma IBM MQ Advanced licença	Usando IBM MQ Operator e uma IBM MQ Advanced for Developers licença	imagem IBM MQ Advanced for Developers pré-construída	Construir seu próprio contêiner
Recurso: AMQP	Não disponível e não suportado.				Suportado conforme o software IBM MQ , mas nenhuma amostra disponível.
Recurso: REST API	Disponível e suportado a partir de IBM MQ Operator 3.0 e IBM MQ 9.3.4 em diante. Antes disso, APIs REST não eram suportadas.	Disponível e suportado.. Fácil de configurar do IBM MQ Operator 3.0 e do IBM MQ 9.3.4 em diante.	Disponível e suportado a partir de IBM MQ Operator 3.0 e IBM MQ 9.3.4 em diante, mas não suportado. Antes disso, APIs REST não estavam disponíveis.	Disponível e suportado do IBM MQ 9.3.4 em diante, mas não suportado. Antes disso, o REST API não estava disponível.	Disponível e suportado conforme o software IBM MQ .
Recurso: Gerenciadores de Filas de Dados Replicados	Não disponível e não suportado. Os gerenciadores de filas de dados replicados são fortemente acoplados ao kernel do Linux e não são suportados em contêineres.				
Recurso: HA nativa	Disponível e suportado..		Disponível, mas não suportado..		Disponível apenas no Kubernetes e no Red Hat OpenShift Container Platform Suportado conforme o software IBM MQ .
Recurso: gerenciadores de filas de várias instâncias	Disponível e suportado..		Disponível, mas não suportado..		Disponível e suportado conforme o software IBM MQ .

	Usando IBM MQ Operator e uma IBM Cloud Pak for Integration licença	Usando IBM MQ Operator e uma IBM MQ Advanced licença	Usando IBM MQ Operator e uma IBM MQ Advanced for Developers licença	imagem IBM MQ Advanced for Developers pré-construída	Construir seu próprio contêiner
Recurso: tipos de logs de recuperação	Criação de log circular ou somente logs replicados A criação de log linear não é suportada				Disponível e suportado conforme o software IBM MQ . É necessário configurar as opções de crtmqm
Recurso: especificando opções da linha de comandos customizada para crtmdir, crtmqm, strmqm e endmqm	Não disponível e não suportado. A maioria das opções pode ser configurada usando um arquivo INI, no entanto, algumas não podem ser configuradas, como o uso de criação de log linear				Opcional, dependendo de como você implementa o seu <i>código de ativação de contêiner</i>
Recurso: usuários do sistema operacional (OS)	Não disponível e não suportado.				Possível e suportado conforme o software IBM MQ , se você instalar o IBM MQ usando RPMs, mas nenhuma amostra estiver disponível. Não é recomendado devido ao risco de segurança
Recurso: IBM MQ Bridge to blockchain	Não disponível e não suportado. Removido do IBM MQ inteiramente do IBM MQ 9.3.2 em diante.				
Recurso: IBM MQ Bridge to Salesforce	Não disponível e não suportado.				Suportado conforme o software IBM MQ , mas descontinuado a partir do IBM MQ 9.3.1 em diante.

Nota: A frase "suportado conforme o software IBM MQ " significa que o IBM Suporte Técnico é limitado ao software principal IBM MQ que está em execução dentro do contêiner.

Conceitos relacionados

[FAQ do IBM MQ para liberações de suporte de longo prazo e de entrega contínua](#)

Referências relacionadas

[IBM Cloud Pak for Integration Anexo de ciclo de vida de suporte de software](#)

OpenShift CP4I CD CP4I-LTS Suporte de versão para o IBM MQ

Operator

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

Nota:

O IBM MQ Operator suporta apenas Extended Update Support (EUS) versões do OpenShift Container Platform. Para obter informações sobre quais liberações isso inclui, consulte [Fases do ciclo de vida](#) na página da web Red Hat OpenShift Container Platform Política de ciclo de vida.

- [“Versões disponíveis do IBM MQ”](#) na página 12
- [“Versões compatíveis do Red Hat OpenShift Container Platform”](#) na página 13
- [“Versões do IBM Cloud Pak for Integration”](#) na página 13
- [“Versões do IBM MQ disponíveis em operadores mais antigos”](#) na página 14
- [“Versões do OpenShift Container Platform compatíveis para operadores mais antigos”](#) na página 14

Versões disponíveis do IBM MQ

Canal do operador	Versão do operador	Versões do IBM MQ									
		9.2.0 EUS	9.2.3	9.2.4	9.2.5	9.3.0	9.3.1	9.3.2	9.3.3	9.3.4	9.3.5
v2.0	2.0	→	⚠	●	●	●□					
v2.1	2.1	→	⚠	⚠	⚠	→	●				
v2.2	2.2	→	⚠	⚠	⚠	→	●				
v2.3	2.3	→	⚠	⚠	⚠	→	⚠	●			
v2.4	2.4	→	⚠	⚠	⚠	→	⚠	⚠	●		
v3.0	3.0					→	⚠	⚠	⚠	●	
v3.1	3.1					→	⚠	⚠	⚠	⚠	●

Chave:



Suporte do Continuous Delivery disponível



IBM Cloud Pak for Integration - Long Term Support disponível



Somente disponível durante a migração do operando IBM Cloud Pak for Integration - Long Term Support para um operando Continuous Delivery.



Deprecated Como as liberações do IBM MQ saem do suporte, elas ainda podem ser configuráveis no operador, mas não são mais elegíveis para suporte e podem ser removidas em liberações futuras.

Consulte [“Histórico de liberação para o IBM MQ Operator”](#) na página 35 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 OpenShift Container Platform ²		
		4.10	4.12	4.14
v2.0	2.0.0-2.0.15	→	□	
	2.0.16		□	
	2.0.17 em diante		□	□
v2.1	2.1	→	●	
v2.2	2.2	→	●	
v2.3	2.3	→	●	
v2.4	2.4.0-2.4.3	→	●	
	2.4.4		●	
	2.4.5 em diante		●	●
v3.0	3.0.0 em diante		●	●
v3.1	3.1.0 em diante		●	●

Chave:

- Suporte do Continuous Delivery disponível
- IBM Cloud Pak for Integration - Long Term Support disponível
- Não é mais suportado. Migre para uma versão mais recente do OpenShift Container Platform

Versões do IBM Cloud Pak for Integration

Suportado para uso como parte da IBM Cloud Pak for Integration versão 2022.2.1, ou independentemente:

- IBM MQ Operator 2.0.x
- IBM MQ Operator 2.1.x

Suportado para uso como parte do IBM Cloud Pak for Integration version 2022.4.1ou independentemente:

- IBM MQ Operator 2.2.x
- IBM MQ Operator 2.3.x

Suportado para uso como parte da IBM Cloud Pak for Integration versão 2023.2.1ou independentemente:

- IBM MQ Operator 2.4.x

Suportado para uso como parte da IBM Cloud Pak for Integration versão 2023.4.1, ou independentemente:

- IBM MQ Operator 3.0.x
- IBM MQ Operator 3.1.x

² As versões do OpenShift Container Platform estão sujeitas às suas próprias datas de suporte. Veja [Política de ciclo de vida do OpenShift Container Platform](#) para obter mais informações.

Versões do IBM MQ disponíveis em operadores mais antigos

Consulte [Versões IBM MQ Disponíveis](#) na documentação do IBM MQ 9.2

Versões do OpenShift Container Platform compatíveis para operadores mais antigos

Consulte [Versões compatíveis OpenShift Container Platform](#) na documentação do IBM MQ 9.2

Planejando o licenciamento do IBM MQ em contêineres

O licenciamento de contêiner permite licenciar apenas a capacidade disponível de seus contêineres individuais do IBM MQ, em vez de requerer que você licencie o servidor inteiro no qual seus contêineres estão em execução. Para tirar vantagem do licenciamento de contêiner, o IBM License Service deve ser usado para controlar o uso sob licença e determinar sua autorização necessária

Informações relacionadas

[Licenças de contêiner do IBM](#)

[Perguntas mais frequentes sobre licenciamento de contêiner](#)

[Instalando o License Service](#)

[Visualizando e rastreando o uso sob licença..](#)

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:

- Licenciar a máquina inteira executando o contêiner.
- Licenciar o 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 em sua autorização e recursos implementados no contêiner, use uma ou mais das anotações a seguir.

Nota: Muitas das anotações contêm uma ou ambas as linhas a seguir:

```
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

Você deve editar essas linhas antes de usar a anotação:

- Para `productChargedContainers`, você deve escolher "All" ou substituir o nome real do contêiner.
- Para `productMetric`, você deve escolher um dos valores oferecidos.

Anotações a serem usadas com uma autorização do produto IBM MQ

Se você tiver uma autorização do produto IBM MQ, selecione a anotação abaixo que corresponde à autorização que você comprou e deseja usar.

- [“IBM MQ” na página 16](#)

- [“IBM MQ Avançado” na página 16](#)
- [“IBM MQ para Ambiente de Não Produção” na página 17](#)
- [“IBM MQ Avançado para Ambiente de Não Produção” na página 17](#)
- [“IBM MQ Avançado para Desenvolvedores” na página 17](#)

As anotações IBM MQ para usar com as configurações de Alta disponibilidade de várias instâncias do IBM MQ são as seguintes. Consulte também [“Selecionando as anotações corretas para configurações de Alta Disponibilidade” na página 15.](#)

- [“IBM MQ Instância múltipla do contêiner” na página 17](#)
- [“IBM MQ Multiinstância de Contêiner Avançado” na página 17](#)
- [“IBM MQ Container Multi Instância para Ambiente de Não Produção” na página 17](#)
- [“IBM MQ Advanced Container Multi Instance para Ambiente de Não Produção” na página 17](#)

Anotações a serem usadas com a autorização do produto CP4I

Se você tiver autorização IBM Cloud Pak for Integration (CP4I), selecione a anotação abaixo que corresponde à autorização que você comprou e deseja usar.

- [“IBM MQ com autorização CP4I” na página 18](#)
- [“Autorização IBM MQ Avançada com CP4I” na página 18](#)
- [“IBM MQ para Ambiente de Não Produção com autorização CP4I” na página 18](#)
- [“Autorização IBM MQ Avançada para Ambiente de Não Produção com CP4I” na página 18](#)

As anotações CP4I para usar com as configurações de Alta disponibilidade de várias instâncias do IBM MQ são as seguintes. Consulte também [“Selecionando as anotações corretas para configurações de Alta Disponibilidade” na página 15.](#)

- [“Autorização do IBM MQ Container Multi Instance com CP4I” na página 18](#)
- [“IBM MQ Autorização Avançada de Contêiner Multi-Instância com CP4I” na página 18](#)
- [“IBM MQ Instância Múltipla de Contêiner para Ambiente de Não Produção com autorização CP4I” na página 19](#)
- [“IBM MQ Autorização do Advanced Container Multi para Ambiente de Não Produção com CP4I” na página 19](#)

Selecionando as anotações corretas para configurações de Alta Disponibilidade

IBM MQ Multi Instância

Ao implementar um par de gerenciadores de filas em uma configuração de alta disponibilidade de várias instâncias do IBM MQ, é necessário usar a mesma anotação em ambas as instâncias. Uma das anotações a seguir deve ser selecionada, dependendo da autorização comprada:

- Autorização independente do IBM MQ ou IBM MQ Advanced
 - [“IBM MQ Instância múltipla do contêiner” na página 17](#)
 - [“IBM MQ Multiinstância de Contêiner Avançado” na página 17](#)
 - [“IBM MQ Container Multi Instância para Ambiente de Não Produção” na página 17](#)
 - [“IBM MQ Advanced Container Multi Instance para Ambiente de Não Produção” na página 17](#)
- IBM Cloud Pak for Integration autorização
 - [“Autorização do IBM MQ Container Multi Instance com CP4I” na página 18](#)
 - [“IBM MQ Autorização Avançada de Contêiner Multi-Instância com CP4I” na página 18](#)
 - [“IBM MQ Instância Múltipla de Contêiner para Ambiente de Não Produção com autorização CP4I” na página 19](#)

- [“IBM MQ Autorização do Advanced Container Multi para Ambiente de Não Produção com CP4I” na página 19](#)

Quando usado com a autorização do IBM Cloud Pak for Integration , as razões de autorização nas anotações asseguram que o consumo de autorização correto seja registrado Quando usado com autorizações IBM MQ ou IBM MQ Advanced independentes, as anotações relatadas no License Service para cada instância precisam ser mapeadas para as partes de autorização do IBM MQ , conforme a seguir:

- IBM MQ Advanced container Várias instâncias
 - 1 x IBM MQ Advanced e 1 x IBM MQ Advanced Réplica de Alta Disponibilidade ou
 - 2 x IBM MQ Advanced³
- IBM MQ Advanced container Multi Instância para Ambiente de Não Produção
 - 1 x IBM MQ Advanced e 1 x IBM MQ Advanced Réplica de Alta Disponibilidade ou
 - 2 x IBM MQ Advanced para Ambiente de Não Produção)³
- IBM MQ Instância múltipla do contêiner
 - 1 x IBM MQ e 1 x IBM MQ Réplica de Alta Disponibilidade ou
 - 2 x IBM MQ³
- IBM MQ Container Multi Instância para Ambiente de Não Produção
 - 1 x IBM MQ e 1 x IBM MQ Réplica de Alta Disponibilidade ou
 - 2 x IBM MQ para Ambiente de Não Produção) ³

IBM MQ HA nativa

Se você estiver implementando três gerenciadores de filas em um quorum de HA nativa, apenas a instância ativa consumirá a autorização. Todas as instâncias devem possuir a mesma anotação. Um dos itens a seguir deve ser selecionado, dependendo da autorização comprada:

- Autorização independente do IBM MQ ou IBM MQ Advanced
 - [“IBM MQ Avançado” na página 16](#)
 - [“IBM MQ Avançado para Ambiente de Não Produção” na página 17](#)
- IBM Cloud Pak for Integration autorização
 - [“Autorização IBM MQ Avançada com CP4I” na página 18](#)
 - [“Autorização IBM MQ Avançada para Ambiente de Não Produção com CP4I” na página 18](#)

Anotações

O restante deste tópico detalha o conteúdo das anotações.

IBM MQ

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

IBM MQ Avançado

```
productID: "208423bb063c43288328b1d788745b0c"  
productName: "IBM MQ Advanced"
```

³ Essa opção de autorização é sub-ideal e deve ser usada apenas se nenhuma autorização da parte relevante da Réplica de Alta Disponibilidade estiver disponível


```
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ para Ambiente de Não Produção

```
productID: "151bec68564a4a47a14e6fa99266deff"  
productName: "IBM MQ for Non-Production Environment"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Avançado para Ambiente de Não Produção

```
productID: "21dfe9a0f00f444f888756d835334909"  
productName: "IBM MQ Advanced for Non-Production Environment"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Avançado para Desenvolvedores

```
productID: "2f886a3eefbe4ccb89b2adb97c78b9cb"  
productName: "IBM MQ Advanced for Developers (Non-Warranted)"  
productMetric: "FREE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Instância múltipla do contêiner

```
productID: "2dea73b866b648b6b4abe2a85eb76964"  
productName: "IBM MQ Container Multi Instance"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Multiinstância de Contêiner Avançado

```
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"  
productName: "IBM MQ Advanced Container Multi Instance"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Container Multi Instância para Ambiente de Não Produção

```
productID: "af11b093f16a4a26806013712b860b60"  
productName: "IBM MQ Container Multi Instance for Non-Production Environment"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Advanced Container Multi Instance para Ambiente de Não Produção

```
productID: "31f844f7a96b49749130cd0708fdbb17"  
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environment"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ com autorização CP4I

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

Autorização IBM MQ Avançada com CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"  
cloudpakName: "IBM Cloud Pak for Integration"  
productID: "208423bb063c43288328b1d788745b0c"  
productName: "IBM MQ Advanced"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productCloudpakRatio: "2:1"
```

IBM MQ para Ambiente de Não Produção com autorização CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"  
cloudpakName: "IBM Cloud Pak for Integration"  
productID: "151bec68564a4a47a14e6fa99266deff"  
productName: "IBM MQ for Non-Production Environment"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productCloudpakRatio: "8:1"
```

Autorização IBM MQ Avançada para Ambiente de Não Produção com CP4I

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

Autorização do IBM MQ Container Multi Instance com CP4I

```
productName: "IBM MQ Container Multi Instance"  
productID: "2dea73b866b648b6b4abe2a85eb76964"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "10:3"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Autorização Avançada de Contêiner Multi-Instância com CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"  
cloudpakName: "IBM Cloud Pak for Integration"  
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"  
productName: "IBM MQ Advanced Container Multi Instance"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productCloudpakRatio: "5:3"
```

IBM MQ Instância Múltipla de Contêiner para Ambiente de Não Produção com autorização CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"  
cloudpakName: "IBM Cloud Pak for Integration"  
productId: "af11b093f16a4a26806013712b860b60"  
productName: "IBM MQ Container Multi Instance for Non-Production Environment"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productCloudpakRatio: "20:3"
```

IBM MQ Autorização do Advanced Container Multi para Ambiente de Não Produção com CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"  
cloudpakName: "IBM Cloud Pak for Integration"  
productId: "31f844f7a96b49749130cd0708fdbb17"  
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environments"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productCloudpakRatio: "10:3"
```

OpenShift

CP4I

Dependências para o IBM MQ Operator

A partir do IBM MQ Operator 3.0 , nenhum outro operador será instalado automaticamente quando você instalar o IBM MQ Operator. Em versões mais antigas do IBM MQ Operator, há uma dependência permanente no IBM Cloud Pak foundational services Operator, que também instala o IBM Operand Deployment Lifecycle Manager (ODLM) Operator.

O IBM Licensing Operator precisa ser instalado separadamente para controlar o uso sob licença Consulte [Implementando License Service](#) na documentação do IBM Cloud Pak for Integration .

IBM MQ Operator 3.0 em diante

V 9.3.4

Ao criar um `QueueManager` usando uma licença do IBM Cloud Pak for Integration , é possível escolher se você deseja ou não usar a conexão única com a instância IBM Cloud Pak for Integration do Keycloak.. O uso de Keycloak é ativado por padrão com uma licença IBM Cloud Pak for Integration , mas se não estiver instalado, o `QueueManager` entrará em um estado "Bloqueado" até que as dependências corretas sejam instaladas. Consulte [“Instalando o IBM MQ Operator”](#) na página 103 para obter mais detalhes sobre as dependências.

Versões mais antigas do IBM MQ Operator

Os operadores IBM Cloud Pak foundational services serão instalados automaticamente quando você instalar versões mais antigas do IBM MQ Operator.. 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-AMRD-XH6P3Q
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-CD3JKX
    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-CD3JKX
    use: "Production"
  web:
    enabled: false
```

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. Consulte [“.spec.queueManager.resources” na página 191](#) para obter mais informações.

Referências relacionadas

[“Referência de licenciamento para mq.ibm.com/v1beta1” na página 182](#)

Permissões com escopo de cluster necessárias pelo IBM 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**
 - verbs: **get, delete**
- 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**
 - verbs: **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**
 - verbs: **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**
 - verbs: **get, list**

Nota: O IBM MQ Operator também requer permissões com escopo definido pelo namespace. Se o IBM MQ Operator for instalado em um escopo de cluster, então as permissões com escopo definido pelo namespace estarão presentes em todos os namespaces.

OpenShift CP4I Kubernetes 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 165](#)). Se isso não for concluído, o IBM MQ precisará ser configurado cada vez que o contêiner for reiniciado.

OpenShift CP4I 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 persistente para assegurar que o gerenciador de filas retenha suas mensagens persistentes e a configuração após uma reinicialização. Esse não é o comportamento padrão. Como há vários provedores de armazenamento, cada um suportando diferentes recursos, isso geralmente significa que a customização da configuração é necessária. O exemplo abaixo descreve os campos comuns que customizam a configuração de armazenamento do IBM MQ na API v1beta1:

- **spec.queueManager.availability** controla o modo de disponibilidade. Se você estiver usando `SingleInstance` ou `NativeHA`, precisará apenas de armazenamento `ReadWriteOnce`. Para `MultiInstance`, você requer uma classe de armazenamento que suporte `ReadWriteMany` com as características de bloqueio de arquivo corretas. 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 25.
- O **spec.queueManager.storage** controla as configurações de armazenamento individuais. Um gerenciador de filas pode ser configurado para ser usado 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: [65534] # Change to 99 for clusters with RHEL7 or earlier worker nodes
```

Para obter informações sobre considerações de armazenamento para gerenciadores de filas de HA nativa, consulte [“HA nativa”](#) na página 144.

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

Capacidade de armazenamento



Ao usar o IBM MQ Operator, o tamanho do armazenamento solicitado é fixo e não pode ser redimensionado após a criação do gerenciador de filas. Você deve assegurar que o volume seja grande o suficiente para suas necessidades.


Criptografia



O IBM MQ não criptografa ativamente os dados em repouso. Portanto, você deve usar o armazenamento criptografado passivamente ou IBM MQ Advanced Message Security, ou ambos, para criptografar suas mensagens. No IBM Cloud, o armazenamento de bloco e de arquivo estão disponíveis com a criptografia passiva em repouso.

OpenShift CP4I Kubernetes IBM MQ Advanced for Developers Imagem de contêiner

Uma imagem de contêiner pré-construída está disponível para o IBM MQ Advanced for Developers. Essa imagem está disponível no IBM Container Registry. Essa imagem é adequada para uso com Docker, Podman, Kubernetes e outros ambientes de contêiner.

Nota:  As imagens do IBM MQ Advanced for Developers estavam anteriormente disponíveis no Docker Hub, mas isso foi descontinuado e nenhuma atualização adicional está disponível no Docker Hub.

Imagens disponíveis

As imagens do IBM MQ são armazenadas no IBM Container Registry:

- IBM MQ Advanced for Developers 9.3.0.17: icr.io/ibm-messaging/mq:9.3.0.17-r3
- IBM MQ Advanced for Developers 9.3.5.1: icr.io/ibm-messaging/mq:9.3.5.1-r2

Referência Rápida

- Licença:
 - [IBM MQ Advanced for Developers e Apache Licença 2.0.](#) Observe que a licença do IBM MQ Advanced for Developers não permite uma distribuição adicional e os termos restringem o uso a uma máquina do desenvolvedor.
- Onde arquivar problemas:
 - [GitHub](#)
- Disponível para as arquiteturas de CPU a seguir:
 - amd64
 - s390x
 - ppc64le

Uso

Execute o [IBM MQ Advanced for Developers](#) em um contêiner.

Consulte a [documentação de uso](#) para obter detalhes sobre como executar um contêiner.

Para poder usar a imagem, deve-se aceitar os termos da licença do IBM MQ configurando a variável de ambiente **LICENSE**.

Variáveis de ambiente suportadas

IDIOMA

Configure o idioma no qual você deseja que a licença seja impressa.

LICENSE

Configure `accept` para concordar com as condições de licença IBM MQ Advanced for Developers.

Configure `view` para visualizar as condições de licenciamento.

Deprecated **log_format**

DESCONTINUADO: Substituído por “[MQ 9.3.2 Feb 2023]MQ_LOGGING_CONSOLE_FORMAT” na página 24.

Altere o formato de logs que são impressos no local do `stdout` do contêiner
Configure `basic` para usar um formato simples legível. Esse é o valor-padrão.
Configure `json` para usar o formato JSON (um objeto JSON em cada linha).

Deprecated **MQ_ADMIN_PASSWORD**

Especifique a senha do usuário administrador.
Deve ter pelo menos 8 caracteres.

V 9.3.4 Não há nenhuma senha padrão para o usuário administrador Para versões IBM MQ Operator anteriores a 3.0.0, o valor padrão é `passwd`.

V 9.3.4 Em IBM MQ 9.3.4, essa variável foi descontinuada. O [YAML de exemplo neste tópico](#) mostra como você mesmo pode criar essa variável e protegê-la com um segredo.

Deprecated **MQ_APP_PASSWORD**

Especifique a senha do usuário do aplicativo

Se configurado, isso faz com que o canal **DEV.APP.SVRCONN** se torne seguro e permita apenas conexões que forneçam um ID do usuário e senha válidos.

Deve ter pelo menos 8 caracteres.

V 9.3.4 Não há nenhuma senha padrão para o usuário do aplicativo Para IBM MQ Operator versões anteriores a 3.0.0, o valor padrão é em branco (nenhuma senha necessária) para clientes IBM MQ e `passwd` para clientes HTTP.

V 9.3.4 Em IBM MQ 9.3.4, essa variável foi descontinuada. O [YAML de exemplo neste tópico](#) mostra como você mesmo pode criar essa variável e protegê-la com um segredo.

MQ_DEV

Configure `false` para parar os objetos padrão que estão sendo criados

MQ_ENABLE_METRICS

Configure `true` para gerar métricas do Prometheus para seu gerenciador de filas.

V 9.3.2 MQ_LOGGING_CONSOLE_SOURCE

Especifique uma lista separada por vírgula de origens para logs que são espelhados para o local do **stdout** do contêiner.

Os valores válidos são `qmgr` e `web`

O valor padrão é `qmgr, web`.

V 9.3.2 MQ_LOGGING_CONSOLE_FORMAT

Substitui o “[Descontinuado]log_format” na página 24

Altere o formato de logs que são impressos no local do **stdout** do contêiner
Configure `basic` para usar um formato simples legível. Esse é o valor-padrão.
Configure `json` para usar o formato JSON (um objeto JSON em cada linha).

V 9.3.2 MQ_LOGGING_CONSOLE_EXCLUDE_ID

Especifique uma lista separada por vírgula de IDs de mensagens para mensagens de log que são excluídas.

As mensagens de log ainda aparecem no arquivo de log no disco, mas não são impressas no local do **stdout** do contêiner

O valor padrão é `AMQ5041I, AMQ5052I, AMQ5051I, AMQ5037I, AMQ5975I`.

mq_qmgr_name

Configure o nome com o qual você deseja que seu gerenciador de fila seja criado

Para obter mais informações sobre a configuração do desenvolvedor padrão suportada pelo IBM MQ Advanced for Developers imagem, consulte a [documentação de configuração do desenvolvedor padrão](#).

V 9.3.4 YAML do gerenciador de filas de exemplo que descreve como especificar senhas para admin e app usuários

No IBM MQ 9.3.4, os IDs do usuário **admin** e **app** não têm mais senhas padrão. Para esses usuários você deve fornecer senhas ao implementar um gerenciador de filas usando a licença do Development . A seguir há um YAML do gerenciador de filas de exemplo que mostra como fazer isso com o IBM MQ Operator

O comando a seguir cria um segredo contendo senhas para usuários **admin** e **app** .

```
oc create secret generic my-mq-dev-passwords --from-literal=dev-admin-password=passwd --from-literal=dev-app-password=passwd
```

O YAML a seguir usa essas senhas ao implementar um gerenciador de filas

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm-dev
spec:
  license:
    accept: false
    license: L-AXAF-JLZ53A
    use: Development
  web:
    enabled: true
  template:
    pod:
      containers:
        - env:
            - name: MQ_DEV
              value: "true"
            - name: MQ_CONNAUTH_USE_HTTP
              value: "true"
            - name: MQ_ADMIN_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-admin-password
            - name: MQ_APP_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-app-password
          name: qmgr
      queueManager:
        storage:
          queueManager:
            type: persistent-claim
            name: QUICKSTART
        version: 9.3.5.1-r2
```

OpenShift CP4I Kubernetes Alta disponibilidade para IBM MQ em contêineres

Há três opções para alta disponibilidade com IBM MQ Operator: **gerenciador de filas de HA nativa** (que possui uma réplica ativa e duas réplicas de espera), **Gerenciador de filas de várias instâncias** (que é um par de espera ativa, usando um sistema de arquivos compartilhado e em rede) ou **Gerenciador de filas resilientes únicos** (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



Os gerenciadores de filas de HA nativa envolvem um pods **ativo** e dois de **réplicas** Kubernetes , que são executados como parte de um Kubernetes StatefulSet com exatamente três réplicas cada uma com seu próprio conjunto de Kubernetes Volumes Persistentes 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, *xf*s 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 144](#).

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

Multi

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, *xf*s 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.

Nota: A mudança dos detalhes do StatefulSet de um gerenciador de filas do IBM MQ, incluindo o número de réplicas, não é suportada quando o gerenciador de filas é criado por meio do IBM MQ Operator

Conceitos relacionados

[Configurações de alta disponibilidade](#)

Tarefas relacionadas

[“Configurando alta disponibilidade para gerenciadores de filas usando o IBM MQ Operator” na página 143](#)

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](#)

OpenShift

CP4I

Planejando proteger o IBM MQ em contêineres

Considerações de segurança ao planejar o seu IBM MQ na configuração de contêineres

Procedimento

- [“Autenticação e autorização de usuário para o IBM MQ em contêineres”](#) na página 28
 - [“Restrições de segurança no uso de usuários do sistema operacional em contêineres”](#) na página 29
- [“Considerações para restringir o tráfego de rede para o IBM MQ em contêineres”](#) na página 29

Autenticação e autorização de usuário para o IBM MQ em contêineres

O IBM MQ em contêineres pode ser configurado para autenticar usuários por meio de LDAP, TLS Mútuo ou um plug-in customizado do MQ .

Observe que o Operador do IBM MQ não permite o uso de usuários e grupos do sistema operacional dentro da imagem de contêiner Para obter informações adicionais, consulte [“Restrições de segurança no uso de usuários do sistema operacional em contêineres”](#) na página 29.

LDAP

Para obter informações sobre como configurar o IBM MQ para usar um repositório do usuário LDAP, consulte [Autenticação de conexão: repositórios do usuário](#) e [autorização LDAP](#).

TLS mútuo

Se você configurar conexões recebidas para um gerenciador de filas para requerer um certificado TLS (TLS mútuo), será possível mapear o nome distinto do certificado para um nome de usuário. Você precisa fazer duas coisas:

- Configure um registro de autenticação de canal para criar o mapeamento para um nome de usuário, usando SSLPEER Para obter mais informações, consulte [Mapeando um Nome Distinto SSL ou TLS para um ID do usuário MCAUSER](#).
- Configure o gerenciador de filas para permitir que você defina registros de autoridade para um nome de usuário que não seja conhecido para o sistema Para obter mais informações, consulte [Sub-rotina de serviço do arquivo qm.ini](#)

Tokens da Web JSON

Para obter informações sobre como configurar o IBM MQ para usar JSON Web Tokens (JWT), consulte [Trabalhando com tokens de autenticação](#).

Plug-in customizado do MQ

Esta é uma técnica avançada, e requer muito mais trabalho. Para obter mais informações, consulte [Usando um serviço de autorização customizado](#)

Tarefas relacionadas

“Exemplo: [Configurando um gerenciador de filas com autenticação TLS mútua](#)” na página 138


Esse exemplo implementa um gerenciador de filas no OpenShift Container Platform usando o IBM MQ Operator. O TLS mútuo é usado para autenticação, para mapear de um certificado TLS para uma identidade no gerenciador de filas.

Restrições de segurança no uso de usuários do sistema operacional em contêineres

O uso de usuários do sistema operacional em contêineres não é recomendado e é proibido com o Operador IBM MQ .

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.** IBM MQ on Linux usa escalação de privilégio para verificar as senhas de 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 definidos nas bibliotecas do sistema operacional dentro de um contêiner. Não há nenhum ID do usuário ou grupo mqm definido no contêiner.

Considerações para restringir o tráfego de rede para o IBM MQ em contêineres

É possível definir políticas de rede para restringir o tráfego para pods em seu cluster no OpenShift Container Platform e Kubernetes. Este tópico descreve algumas considerações sobre como as políticas de rede podem se aplicar ao IBM MQ

Para ingresso de rede para um gerenciador de filas, há várias portas a serem consideradas:

- Porta 1414 para tráfego do gerenciador de filas
- Porta 9414 para HA nativa
- Porta 9157 para métricas
- Porta 9443 para o console da web e APIs REST

A saída de rede é mais complexa. Exemplos de egresso de rede que você pode desejar considerar:

- DNS-se você tiver canais ou outra configuração que use nomes DNS
- Outros gerenciadores de filas

- Online Certificate Status Protocol (OCSP) e Certificate Revocation Lists (CRLs)-determinado por seu provedor de certificado.
- Provedores de autenticação:
 - LDAP
 - Abra o ID Connect ou outro provedor de login configurado para o servidor da web IBM MQ . Isso inclui a IU do IBM Cloud Pak Platform e o IAM de serviços básicos do IBM Cloud Pak.
- Provedores de rastreamento:
 - Instana
 - Painel de operações do Cloud Pak for Integration .⁴

Exemplo de ingresso NetworkPolicy

A seguir há um exemplo de política de rede para controlar o ingresso para um gerenciador de filas chamado "myqm", para uso no Red Hat OpenShift Container Platform.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: myqm
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: myqm
      app.kubernetes.io/name: ibm-mq
  ingress:
    # Allow access to queue manager listener from anywhere
    - ports:
      - protocol: TCP
        port: 1414
    # Allow access to Native HA port from other instances of the same queue manager
    - from:
      - podSelector:
          matchLabels:
            app.kubernetes.io/instance: myqm
            app.kubernetes.io/name: ibm-mq
    ports:
      - protocol: TCP
        port: 9414
    # Allow access to metrics from monitoring project
    - from:
      - namespaceSelector:
          matchLabels:
            network.openshift.io/policy-group: monitoring
    ports:
      - protocol: TCP
        port: 9157
    # Allow access to web server via Route
    - from:
      - namespaceSelector:
          matchLabels:
            network.openshift.io/policy-group: ingress
    ports:
      - protocol: TCP
        port: 9443
```

Conformidade FIPS para IBM MQ em contêineres

Na inicialização, o IBM MQ em contêineres detecta se o sistema operacional no qual o contêiner está sendo iniciado é compatível com FIPS e (se for) configura o suporte FIPS automaticamente. Requisitos e limitações são observados aqui.

⁴ O Painel de operações foi descontinuado do IBM MQ 9.3.0, e removido em IBM MQ 9.3.3 Consulte [“Integrando-se ao IBM Cloud Pak for Integration Operations Dashboard”](#) na página 156.

Federal Information Processing Standards

O governo dos EUA produz aconselhamento técnico sobre sistemas e segurança de TI, inclusive criptografia de dados. O National Institute for Standards and Technology (NIST) é um órgão governamental preocupado com sistemas de TI e segurança. O NIST produz recomendações e normas, inclusive Federal Information Processing Standards (FIPS).

Um padrão FIPS significativo é o FIPS 140-2, que requer o uso de algoritmos criptográficos fortes FIPS 140-2 também especifica requisitos para algoritmos hashing a serem usados para proteger pacotes contra modificação em trânsito.

IBM MQ fornece suporte FIPS 140-2 se ele tiver sido configurado para fazer isso.

Nota: No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o [IBM Crypto for C \(ICC\) certificado](#) e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na [NIST CMVP modules in process list](#).

Requisitos

Para obter os requisitos relacionados à configuração de cluster e outras considerações, consulte [FIPS Wall: Atual IBM abordagem para conformidade FIPS](#).

IBM MQ em contêineres podem ser executados no modo de conformidade FIPS 140-2. Durante a inicialização, o IBM MQ em contêineres (9.3.1.0 e acima) detecta se o sistema operacional do host no qual o contêiner está sendo iniciado é compatível com o FIPS. Se o sistema operacional do host for compatível com FIPS e as chaves privadas e os certificados tiverem sido fornecidos, o contêiner IBM MQ configurará o gerenciador de filas, o servidor da Web do IBM MQ e a transferência de dados entre os nós em uma implementação de Alta Disponibilidade Nativa, para executar no modo de conformidade FIPS

Ao usar IBM MQ Operator para implementar gerenciadores de filas, o operador cria uma rota com um tipo de finalização de **Passthrough**. Isso significa que o tráfego é enviado diretamente ao destino sem que o roteador forneça a finalização TLS. O gerenciador de filas do IBM MQ e o servidor da Web do IBM MQ são os destinos neste caso e eles já fornecem comunicação segura compatível com FIPS.

Principais requisitos:

1. Uma chave privada e certificados, fornecidos em um segredo para o gerenciador de filas e servidor da web, que permitem que clientes externos se conectem com segurança ao gerenciador de filas e servidor da web.
2. Uma chave privada e certificados para transferência de dados entre diferentes nós em uma configuração de Alta Disponibilidade Nativa

Limitações

Para uma implementação compatível com FIPS do IBM MQ em contêineres, considere o seguinte:

- O IBM MQ em contêineres fornece um terminal para coleção de métricas. Atualmente, este terminal é apenas HTTP. É possível desativar o terminal de métricas para tornar o restante do IBM MQ FIPS compatível.
- IBM MQ em contêineres permite substituições de imagem customizada. Ou seja, é possível construir imagens customizadas usando a imagem de contêiner do IBM MQ como a imagem base. A conformidade com o FIPS pode não se aplicar a essas imagens customizadas
- Para rastreamento de mensagens usando IBM Instana, a comunicação entre IBM MQ e IBM Instana é HTTP ou HTTPS, sem conformidade com FIPS.
- O acesso IBM MQ Operator aos serviços de gerenciamento de identidade e acesso (IAM) /Zen do IBM não é compatível com FIPS.

Como a conformidade FIPS é detectada e o suporte FIPS configurado automaticamente

Se o sistema operacional no qual o contêiner está sendo iniciado for compatível com FIPS, o suporte FIPS será configurado automaticamente..

Nota: No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o [IBM Crypto for C \(ICC\) certificado](#) e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na [NIST CMVP modules in process list](#).

Durante a inicialização, IBM MQ em contêineres detecta se o sistema operacional no qual o contêiner está sendo iniciado é compatível com FIPS. Nesse caso, as ações a seguir serão executadas automaticamente:

Gerenciador de filas

Se o sistema operacional do host for compatível com FIPS, e a chave privada e os certificados forem fornecidos, o atributo do gerenciador de filas **SSLFIPS** será configurado como YES Caso contrário, o atributo **SSLFIPS** será configurado como NO .

IBM MQ servidor da web

O servidor da web IBM MQ fornece uma interface HTTP/HTTPS para administrar IBM MQ. Se o sistema operacional do host for compatível com FIPS, as opções da JVM serão atualizadas para que o servidor da web use a criptografia compatível com FIPS. Para poder usar FIPS, a chave privada e os certificados devem ser fornecidos durante o início do contêiner.

HA nativa

A segurança dos dados replicado entre os nós é controlada pela sub-rotina

NativeHALocalInstance do arquivo `qm.ini` Por exemplo:

```
NativeHALocalInstance:
  KeyRepository=/run/runmqserver/ha/tls/key.kdb
  CertificateLabel=NHAQM
  CipherSpec=ECDHE_RSA_AES_256_GCM_SHA384
```

Se FIPS estiver ativado, o atributo **SSLFipsRequired** será incluído na sub-rotina, com o valor configurado como Yes:

```
NativeHALocalInstance:
  KeyRepository=/run/runmqserver/ha/tls/key.kdb
  CertificateLabel=NHAQM
  CipherSpec=ECDHE_RSA_AES_256_GCM_SHA384
  SSLFipsRequired=Yes
```

Se o contêiner estiver em execução em um cluster do OpenShift sem suporte FIPS, o gerenciador de filas, o servidor da Web do IBM MQ e os componentes de HA nativa não terão seu suporte FIPS ativado automaticamente Atualmente, apenas a arquitetura x86-64 é suportada pela plataforma OpenShift para FIPS Para arquiteturas Power e Linux for IBM Z , OpenShift não oferece suporte FIPS. Para ativar explicitamente o suporte FIPS nos componentes IBM MQ para essas arquiteturas, configure a variável de ambiente `MQ_ENABLE_FIPS` como `true` no YAML do gerenciador de filas. O fragmento YAML a seguir descreve o uso da variável de ambiente `MQ_ENABLE_FIPS` :

```
template:
  pod:
    containers:
      - env:
          - name: MQ_ENABLE_FIPS
            value: "true"
        name: qmgr
```

Substituindo o modo FIPS automático para IBM MQ em contêineres

Use a variável de ambiente `MQ_ENABLE_FIPS` para ativar ou desativar explicitamente o modo FIPS para os componentes do IBM MQ no contêiner

Antes de começar

Nota: No AIX, Linux, and Windows, IBM MQ fornece conformidade FIPS 140-2 por meio do módulo criptográfico IBM Crypto for C (ICC) . O certificado deste módulo foi movido para o status Histórico. Os clientes devem visualizar o [IBM Crypto for C \(ICC\) certificado](#) e estar ciente de qualquer aviso fornecido pelo NIST Um módulo FIPS 140-3 de substituição está atualmente em andamento e seu status pode ser visualizado procurando por ele na [NIST CMVP modules in process list](#).

Sobre esta tarefa

`MQ_ENABLE_FIPS` suporta três valores:

automático

Esse é o valor-padrão.

Se o sistema operacional do host for ativado para FIPS, todos os componentes (gerenciador de filas, servidor da Web do IBM MQ e HA nativa) serão executados no modo FIPS.

Se o sistema operacional do host não for ativado por FIPS, então todos os componentes não serão executados no modo FIPS

true

Esse valor ativa o FIPS para componentes selecionados no contêiner.

O atributo do gerenciador de filas **SSLFIPS** é configurado como YES , mesmo se IBM MQ em contêineres estiver em execução em um sistema operacional do host que não seja compatível com FIPS Ou seja, se o gerenciador de filas do IBM MQ , o servidor da web e a HA nativa forem compatíveis com FIPS, mas o sistema operacional do contêiner não for.

false

Esse valor desativa a conformidade FIPS.

O atributo do gerenciador de filas **SSLFIPS** é configurado como NO, mesmo se IBM MQ em contêineres estiver em execução em uma máquina host compatível com FIPS No entanto, o IBM MQ ainda protege as conexões se a chave privada e os certificados forem fornecidos

As opções da JVM não são atualizadas para o servidor da Web IBM MQ No entanto, o servidor da web do IBM MQ ainda executa um terminal HTTPS se a chave privada e os certificados forem fornecidos.

A replicação de dados em HA nativa não usa a criptografia FIPS

Exemplo

Aqui está um YAML do gerenciador de filas de amostra que descreve a ativação de TLS e FIPS para o componente do gerenciador de filas:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  namespace: ibm-mq-fips
  name: ibm-mq-qm-ppcle
spec:
  license:
    accept: true
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: PPCLEQM
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQ_ENABLE_FIPS
              value: "true"
          name: qmgr
  version: 9.3.5.1-r2
  web:
    enabled: false
  pki:
```

```
keys:
- name: ibm-mq-tls-certs
  secret:
    secretName: ibm-mq-tls-secret
    items:
    - tls.key
    - tls.crt
```

Multi 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. Em Red Hat OpenShift Container Platform 4.11, há um limite padrão de 4096 processos por contêiner. Para versões anteriores do OpenShift Container Platform, o limite é 1024 processos. Para obter compatibilidade de versões do IBM MQ Operator com versões do OpenShift Container Platform, consulte [“Versões compatíveis do Red Hat OpenShift Container Platform”](#) na página 13. 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.

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 35.](#)
- [“Migrando o IBM MQ para o IBM Cloud Pak for Integration” na página 79.](#)
- [“Instalando o IBM MQ Operator” na página 103.](#)
- [“Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas” na página 120.](#)
- [“Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform” na página 116.](#)
- [“Operando o IBM MQ usando o IBM MQ Operator” na página 171.](#)
- [“Referência da API do IBM MQ Operator” na página 182.](#)

Notas:

- Para obter informações sobre Operadores IBM MQ anteriores, consulte [Histórico de Liberação para IBM MQ Operator](#) na documentação do IBM MQ 9.2 .
- Para obter informações sobre futuras atualizações do IBM MQ , consulte a página geral [IBM MQ datas de liberação de manutenção planejadas](#) .

IBM MQ Operator 3.1.3



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Canal do operador

v3.1

Valores permitidos para `.spec.version`

[9.3.5.1-r2](#)

Valores permitidos para `.spec.version` durante a migração

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, [9.3.0.17-r2](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2, [9.3.5.1-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versão 4.3 e superior (instalação opcional).

O que mudou

- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 3.1.2

CD

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Canal do operador

v3.1

Valores permitidos para `.spec.version`

9.3.5.1-r1

Valores permitidos para `.spec.version` durante a migração

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2,

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versão 4.3 e superior (instalação opcional).

O que mudou

- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 3.1.1

CD

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Canal do operador

v3.1

Valores permitidos para `.spec.version`

9.3.5.0-r2

Valores permitidos para `.spec.version` durante a migração

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versão 4.3 e superior (instalação opcional).

O que mudou

- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 3.1.0

CD

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Canal do operador

v3.1

Valores permitidos para `.spec.version`

[9.3.5.0-r1](#)

Valores permitidos para `.spec.version` durante a migração

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, [9.3.3.3-r2](#), 9.3.4.0-r1, 9.3.4.1-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versão 4.3 e superior (instalação opcional).

O que mudou

- As vulnerabilidades abordadas são detalhadas nestes Boletins de Segurança:
 - <https://www.ibm.com/support/pages/node/7126571>.
 - <https://www.ibm.com/support/pages/node/7137570>.

IBM MQ Operator 3.0.1



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Canal do operador

v3.0

Valores permitidos para `.spec.version`

[9.3.4.1-r1](#)

Valores permitidos para `.spec.version` durante a migração

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, [9.3.3.3-r1](#), 9.3.4.0-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versão 4.3 e superior (instalação opcional).

O que mudou

- Atualização somente de segurança construída no [“IBM MQ Operator 3.0.0”](#) na página 37
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 3.0.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.4.1

Canal do operador

v3.0

Valores permitidos para `.spec.version`

9.3.4.0-r1

Valores permitidos para `.spec.version` durante a migração

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versão 4.3 e superior (instalação opcional).

O Quê Há de Novo

- É possível configurar o servidor da web IBM MQ incluindo um arquivo `mqwebuser.xml` em um ConfigMap ou Segredo, usando a nova propriedade `manualConfig` YAML (requer IBM MQ 9.3.4 ou superior)
- O `administrative REST API` agora é suportado.. É possível configurar isso por meio de um ConfigMap ou Segredo como acima (requer IBM MQ 9.3.4 ou superior). No entanto, observe que o servidor da web ainda não é considerado um serviço crítico para a análise de atividade, portanto, se ele falhar, o contêiner não será reiniciado automaticamente.
- Você desativa a conexão única ao usar uma licença do IBM Cloud Pak for Integration escolhendo autenticação e autorização "manual" (requer IBM MQ 9.3.4 ou superior)
- É possível ativar um sistema de arquivos raiz somente leitura dentro do contêiner Isso melhora a segurança, evitando gravações na maioria dos arquivos dentro do contêiner no tempo de execução (requer IBM MQ 9.3.4 ou superior). A opção `readOnlyRootFilesystem` é acompanhada por opções adicionais para configurar o tamanho dos volumes "scratch" e "tmp" que são montados para permitir a gravação de arquivos temporários. Consulte ["Executando o contêiner IBM MQ com um sistema de arquivos raiz somente leitura" na página 168](#)

O que mudou

- Liberações removidas (anteriormente descontinuadas): IBM MQ 9.2.0 EUS, 9.2.3, 9.2.4, 9.2.5. Importante: assegure-se de que você não tenha gerenciadores de fila para qualquer uma das versões removidas antes de fazer upgrade do IBM MQ Operator. Após o upgrade, você não poderá mais editar o recurso `QueueManager`, além de fazer upgrade para uma versão em suporte, porque o IBM MQ Operator não reconhece mais as versões mais antigas.
- Instalação e ciclo de vida do operador
 - O IBM MQ Operator agora é suportado no Red Hat OpenShift Container Platform version 4.14
 - O IBM MQ Operator não é mais instalado IBM Cloud Pak foundational services automaticamente. Se você implementar um `QueueManager` que usa uma licença IBM Cloud Pak for Integration e que configura a conexão única (o padrão para gerenciadores de filas com essa licença), o `QueueManager` entrará em um estado "Bloqueado" se as dependências necessárias ainda não estiverem instaladas. Nenhum outro operador será instalado automaticamente..
- Mudanças de Segurança
 - IBM Cloud Pak for Integration 2023.4.1 usa Keycloak para conexão única e autorização, em vez de IBM Cloud Pak Identity and Access Manager.

- O modelo IBM Cloud Pak for Integration "início rápido" não desativa mais a segurança com *MQSNOAUT*. É necessário configurar a autenticação Consulte [“Autenticação e autorização de usuário para o IBM MQ em contêineres”](#) na página 28.
- Usuários padrão desativados no IBM MQ Advanced for Developers da versão 9.3.4. Os usuários padrão ("admin" e "app") e outra configuração fornecida como parte do IBM MQ Advanced for Developers são desativados por padrão.
- Mudanças menores no Pod IBM MQ Operator :
 - O IBM MQ Operator não implementa mais um contêiner de inicialização
 - O nome do contêiner IBM MQ Operator agora é *gerenciador*
 - O prefixo do pod IBM MQ Operator é *ibm-mq-operator* .
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.4.8

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Canal do operador

v2.4

Valores permitidos para `.spec.version`

[9.3.3.3-r2](#)

Valores permitidos para `.spec.version` durante a migração

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente de segurança construída no [“IBM MQ Operator 2.4.0”](#) na página 43
- As vulnerabilidades abordadas são detalhadas nestes Boletins de Segurança:
 - <https://www.ibm.com/support/pages/node/7126571>.
 - <https://www.ibm.com/support/pages/node/7137570>.

IBM MQ Operator 2.4.7

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Canal do operador

v2.4

Valores permitidos para `.spec.version`

[9.3.3.3-r1](#)

Valores permitidos para `.spec.version` durante a migração

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente de segurança construída no [“IBM MQ Operator 2.4.0”](#) na página 43
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.4.6



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Canal do operador

v2.4

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente de segurança construída no [“IBM MQ Operator 2.4.0”](#) na página 43
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.4.5



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Canal do operador

v2.4

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2

r2, 9.3.0.11-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente de segurança construída no [“IBM MQ Operator 2.4.0” na página 43](#)
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.4.4



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Canal do operador

v2.4

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente de segurança construída no [“IBM MQ Operator 2.4.0” na página 43](#)
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).
- O IBM MQ Operator não é mais testado ou suportado no OpenShift Container Platform 4.10

IBM MQ Operator 2.4.3



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Canal do operador

v2.4

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente de segurança construída no [“IBM MQ Operator 2.4.0”](#) na página 43
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.4.2



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Canal do operador

v2.4

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente de segurança construída no [“IBM MQ Operator 2.4.0”](#) na página 43
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.4.1



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Canal do operador

v2.4

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente de segurança construída no [“IBM MQ Operator 2.4.0”](#) na página 43
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.4.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2023.2.1

Canal do operador

v2.4

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, [9.3.3.0-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O Quê Há de Novo

- Integração do Painel de Operações Removidas
- Incluído suporte IBM MQ Operator para **LogFilePages**.

O que mudou

- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.3.3



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Canal do operador

v2.3

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, [9.3.0.5-r2](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, [9.3.2.1-r2](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.3.0](#)” na página 45
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.3.2

CD

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Canal do operador

v2.3

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, [9.3.0.5-r1](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, [9.3.2.1-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.3.0](#)” na página 45
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.3.1

CD

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Canal do operador

v2.3

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, [9.3.0.4-r2](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, [9.3.2.0-r2](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O Quê Há de Novo

- A partir de março de 2023, as imagens de contêiner do gerenciador de filas IBM MQ Operator e do IBM MQ são assinadas digitalmente. As imagens IBM MQ Operator 2.3.1 e IBM MQ 9.3.2.0-r2 foram assinadas com esta liberação. Consulte “[Verificando assinaturas de imagem](#)” na página 79.

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.3.0](#)” na página 45
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.3.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Canal do operador

v2.3

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, [9.3.0.4-r1](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, [9.3.2.0-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O Quê Há de Novo

- No IBM MQ Operator 2.3.0, o suporte FIPS 140-2 pode ser configurado. Consulte “[Conformidade FIPS para IBM MQ em contêineres](#)” na página 30.
- Em IBM MQ Operator 2.3.0, IBM MQ 9.3.1 foi descontinuado.

O que mudou

- Em IBM MQ Operator 2.3.0, “[HA nativa](#)” na página 144 está disponível sob uma licença IBM MQ Advanced ou IBM MQ Advanced for Developers
- O conjunto de permissões requerido pelo IBM MQ Operator é reduzido..
- A partir do IBM MQ Operator 2.3.0, **ibm-automation-core** é removido do OperandRequest feito para IBM Cloud Pak for Integration implementações.
- No IBM MQ Operator 2.3.0, a implementação do IBM MQ Operator especifica um **imagePullPolicy** de IfNotPresent.
- As vulnerabilidades que são abordadas são detalhadas nestes boletins de segurança:
 - [Boletim para CVE-2022-47629 e CVE-2022-35737](#)
 - [Boletim para CVE-2023-26284](#)

IBM MQ Operator 2.2.2



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Canal do operador

v2.2

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, [9.3.0.3-r1](#), 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, [9.3.1.1-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.2.1



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Canal do operador

v2.2

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, [9.3.0.1-r4](#), 9.3.1.0-r1, 9.3.1.0-r2, [9.3.1.0-r3](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O que mudou

- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.2.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.4.1

Canal do operador

v2.2

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, [9.3.0.1-r3](#), 9.3.1.0-r1, [9.3.1.0-r2](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versões 3.19 a 3.24 inclusivo

O Quê Há de Novo

- A partir do IBM MQ Operator 2.2.0 (CD), o rastreamento IBM Instana é suportado nativamente. O suporte está disponível na imagem do gerenciador de filas 9.3.1.0-r2 (CD) IBM MQ . 9.3.1.0-r2 contém

[versão 2.4.0 \(2022.4.0\)](#) do IBM Instana MQ Exit. Para ativar o rastreo do IBM Instana , consulte [“Integrando o IBM MQ ao rastreo do IBM Instana”](#) na página 157

O que mudou

- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).
- No IBM MQ Operator 2.2.0, o Painel de Operações foi descontinuado e não receberá mais atualizações. Nenhum novo uso do Painel de Operações deve ser iniciado.. IBM MQ 2.0.x Os operadores continuam a suportar o Painel de Operações.

IBM MQ Operator 2.1.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.1

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, [9.3.0.1-r2](#), [9.3.1.0-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.X, mas pelo menos 3.19

O Quê Há de Novo

- Inclui IBM MQ 9.3.1..
- Inclui uma nova opção que permite que os usuários [desativem atualizações de valor padrão para a especificação do gerenciador de filas](#).
- Inclui uma nova condição de status que descontinua todas as versões do IBM MQ anteriores a IBM MQ 9.3.1
- Inclui uma nova condição de status que avisa os usuários que estão usando os operandos LTS com esta versão do CD do IBM MQ Operator

O que mudou

- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.23 (LTS)



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

[9.3.0.17-r3](#)

Valores permitidos para `.spec.version` durante a migração

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4,

9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização de segurança construída em “[IBM MQ Operator 2.0.0](#)” na página 58
- IBM MQ A imagem do catálogo foi movida para o formato de catálogo baseado em arquivos do formato do banco de dados SQLite .
- As vulnerabilidades que são abordadas são detalhadas neste [Boletim de segurança](#)

IBM MQ Operator 2.0.22 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

[9.3.0.17-r2](#)

Valores permitidos para `.spec.version` durante a migração

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades que são abordadas são detalhadas neste [Boletim de segurança](#)

IBM MQ Operator 2.0.21 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

[9.3.0.17-r1](#)

Valores permitidos para `.spec.version` durante a migração

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades que são abordadas são detalhadas neste [Boletim de segurança](#)

IBM MQ Operator 2.0.20 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para .spec.version

[9.3.0.16-r2](#)

Valores permitidos para .spec.version durante a migração

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades que são abordadas são detalhadas neste [Boletim de segurança](#)

IBM MQ Operator 2.0.19 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para .spec.version

[9.3.0.16-r1](#)

Valores permitidos para .spec.version durante a migração

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4,

9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas nestes Boletins de Segurança:
 - <https://www.ibm.com/support/pages/node/7126571>.
 - <https://www.ibm.com/support/pages/node/7137570>.

IBM MQ Operator 2.0.18 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para .spec.version

[9.3.0.15-r1](#)

Valores permitidos para .spec.version durante a migração

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.17 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4,

9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, [9.3.0.11-r2](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no [“IBM MQ Operator 2.0.0”](#) na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).
- O IBM MQ Operator não é mais testado ou suportado no OpenShift Container Platform 4.10

IBM MQ Operator 2.0.16 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, [9.3.0.11-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 e superior. **Nota:** Somente as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo 4.14 e 4.16.

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no [“IBM MQ Operator 2.0.0”](#) na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).
- O IBM MQ Operator não é mais testado ou suportado no OpenShift Container Platform 4.10

IBM MQ Operator 2.0.15 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, [9.3.0.10-r2](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no [“IBM MQ Operator 2.0.0”](#) na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.14 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, [9.3.0.10-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no [“IBM MQ Operator 2.0.0”](#) na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.13 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, [9.3.0.6-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.12 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, [9.3.0.5-r3](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.11 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, [9.3.0.5-r2](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.10 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, [9.3.0.5-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no [“IBM MQ Operator 2.0.0”](#) na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.9 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, [9.3.0.4-r2](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O Quê Há de Novo

- A partir de março de 2023, as imagens de contêiner do gerenciador de filas IBM MQ Operator e do IBM MQ são assinadas digitalmente. As imagens IBM MQ Operator 2.0.9 e IBM MQ 9.3.0.4-r2 foram assinadas com esta liberação. Consulte [“Verificando assinaturas de imagem”](#) na página 79)

O que mudou

- Atualização somente segurança construída no [“IBM MQ Operator 2.0.0”](#) na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.8 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades que são abordadas são detalhadas nestes boletins de segurança:
 - [Boletim para CVE-2022-47629 e CVE-2022-35737](#)
 - [Boletim para CVE-2023-26284](#)

IBM MQ Operator 2.0.7 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.6 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, [9.3.0.1-r4](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.5 (LTS)

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, [9.3.0.1-r3](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.4

CP4I-LTS

Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, [9.3.0.1-r2](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.3



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 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, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, [9.3.0.1-r1](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.2



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para .spec.version

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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](#), 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, [9.3.0.0-r3](#)

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no “[IBM MQ Operator 2.0.0](#)” na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.1



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O que mudou

- Atualização somente segurança construída no [“IBM MQ Operator 2.0.0”](#) na página 58
- As vulnerabilidades abordadas são detalhadas neste [Boletim de segurança](#).

IBM MQ Operator 2.0.0



Versão IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2022.2.1

Canal do operador

v2.0

Valores permitidos para `.spec.version`

9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-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, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1

Versões do Red Hat OpenShift Container Platform

OpenShift Container Platform 4.10 e mais recente **Nota:** somente as liberações do OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias de número par, por exemplo, 4.10 e 4.12

Versões do IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.19

O Quê Há de Novo

- Inclui IBM MQ 9.3.0..
- Inclui suporte para POWER (ppc64le).

O que mudou

- Red Hat OpenShift Container Platform 4.10 agora necessário. Consulte [“Suporte de versão para o IBM MQ Operator”](#) na página 12.
- **Deprecated** Versões descontinuadas: IBM MQ 9.2.3 Essas versões podem não ser reconciliadas por versões futuras do IBM MQ Operator.
- **Removed** Removidas (anteriormente descontinuadas) liberações de entrega contínua: IBM MQ 9.1.5, 9.2.0 CD, 9.2.1e 9.2.2
- O gancho da web de validação do IBM MQ Operator agora é instalado pelo Operator Lifecycle Manager (OLM). O OLM agora gerencia o certificado do web hook.
- Foi corrigido um erro que estava anteriormente gerando avisos de preferência do usuário na criação de log do IBM MQ Console
- As vulnerabilidades abordadas são detalhadas nestes Boletins de Segurança:

- <https://www.ibm.com/support/pages/node/6602255>
- <https://www.ibm.com/support/pages/node/6602259>

OpenShift > CP4I > CD > CP4I-LTS **Histórico de liberação para imagens do Contêiner do Gerenciador de filas para uso com o IBM MQ Operator**

Nota: Para obter informações sobre imagens anteriores do Queue Manager Container, consulte [Histórico de Liberação para IBM MQ Operator](#) na documentação do IBM MQ 9.2 .

9.3.5.1-r2



Versão do operador necessária

[3.1.3](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.1-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.1-r2`
- `icr.io/ibm-messaging/mq:9.3.5.1-r2`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.5](#)

O que mudou

- [O que mudou no IBM MQ 9.3.5](#)
- Baseado em [Red Hat Universal Base Image 8.9-1161.1715068733](#)
- A biblioteca [golang.org/x/net](#) foi atualizada para corrigir uma vulnerabilidade relatada

9.3.5.1-r1



Versão do operador necessária

[3.1.2](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.1-r1`
- `icr.io/ibm-messaging/mq:9.3.5.1-r1`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.5](#)

O que mudou

- [O que mudou no IBM MQ 9.3.5](#)
- Baseado em [Red Hat Universal Base Image 8.9-1161](#)
- As vulnerabilidades de segurança relatadas por "dependabot" foram resolvidas

9.3.5.0-r2

CD

Versão do operador necessária

[3.1.1](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.0-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.0-r2](#)
- [icr.io/ibm-messaging/mq:9.3.5.0-r2](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.5](#)

O que mudou

- [O que mudou no IBM MQ 9.3.5](#)
- Com base em [Red Hat Universal Base Image 8.9-1137](#)
- Você só precisará retirar a nova imagem 9.3.5.0-r2 se tiver o Painel de Operações ativado.

9.3.5.0-r1

CD

Versão do operador necessária

[3.1.0](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.0-r1](#)
- [icr.io/ibm-messaging/mq:9.3.5.0-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.5](#)

O que mudou

- [O que mudou no IBM MQ 9.3.5](#)
- Com base em [Red Hat Universal Base Image 8.9-1137](#)
- Um link simbólico é fornecido para o `/var/mam` em que credenciais não criptografadas no `mqwebuser.xml` seriam copiadas.
- A biblioteca [golang.org/x/crypto](#) foi atualizada para corrigir a vulnerabilidade CVE-2023-48795 .
- Algoritmo SHA512 mais seguro usado em vez de SHA256 para criar certificado autoassinado no keystore da web.
- O keystore PKCS#12 para uso com o servidor da web IBM MQ agora é gerado usando a função **Pkcs12.Modern.Encode** , que usa a criptografia SHA-2 (gerada anteriormente usando uma criptografia SHA-1 anterior).
- A vulnerabilidade relatada nos usos do método **PathTraversal** foi corrigida

9.3.4.1-r1

CD

Versão do operador necessária

[3.0.1](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.4.1-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.4.1-r1](#)
- [icr.io/ibm-messaging/mq:9.3.4.1-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.4](#)

O que mudou

- [O que mudou no IBM MQ 9.3.4](#)
- Com base em [Red Hat Universal Base Image 8.9-1108](#)

9.3.4.0-r1

CD

Versão do operador necessária

[3.0.0](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.4.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.4.0-r1](#)
- [icr.io/ibm-messaging/mq:9.3.4.0-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.4](#)

O que mudou

- [O que mudou no IBM MQ 9.3.4](#)
- Com base em [Red Hat Universal Base Image 8.9-1029](#)
- Suporte melhorado para o servidor da web IBM MQ -O log do servidor da web IBM MQ agora aparece por padrão no log do contêiner. O arquivo `messages.log` do servidor da web agora é espelhado automaticamente para a saída do registro de contêiner. Como parte dessa mudança, o arquivo `messages.log` gravado no disco agora está sempre no formato JSON, embora o log do contêiner continue disponível como JSON ou como o formato "básico" legível.
- Foi corrigida a manipulação de sinais dentro da imagem do contêiner do gerenciador de filas, para que ele processe corretamente os sinais de controle se o contêiner for finalizado por Red Hat OpenShift Container Platform antes da inicialização ser concluída

9.3.3.3-r2

Versão do operador necessária

[2.4.8](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.3-r2](#)

- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.3-r2
- icr.io/ibm-messaging/mq:9.3.3.3-r2

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.3](#)

O que mudou

- [O que mudou no IBM MQ 9.3.3](#)
- Com base em [Red Hat Universal Base Image 8.9-1137](#)
- A biblioteca golang.org/x/crypto foi atualizada para corrigir a vulnerabilidade CVE-2023-48795 .
- Algoritmo SHA512 mais seguro usado em vez de SHA256 para criar certificado autoassinado no keystore da web.
- O keystore PKCS#12 para uso com o servidor da web IBM MQ agora é gerado usando a função **Pkcs12.Modern.Encode** , que usa a criptografia SHA-2 (gerada anteriormente usando uma criptografia SHA-1 anterior).
- A vulnerabilidade relatada nos usos do método **PathTraversal1** foi corrigida

9.3.3.3-r1

Versão do operador necessária

[2.4.7](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.3-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.3-r1
- icr.io/ibm-messaging/mq:9.3.3.3-r1

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.3](#)

O que mudou

- [O que mudou no IBM MQ 9.3.3](#)
- Com base em [Red Hat Universal Base Image 8.9-1108](#)

IBM MQ APARs incluídos

- IT44961
- IT44821
- IT44954

9.3.3.2-r3



Versão do operador necessária

[2.4.6](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r3
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.2-r3
- icr.io/ibm-messaging/mq:9.3.3.2-r3

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.3](#)

O que mudou

- [O que mudou no IBM MQ 9.3.3](#)
- Com base em [Red Hat Universal Base Image 8.9-1029](#)

9.3.3.2-r2



Versão do operador necessária

[2.4.5](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.2-r2`
- `icr.io/ibm-messaging/mq:9.3.3.2-r2`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.3](#)

O que mudou

- [O que mudou no IBM MQ 9.3.3](#)
- Com base em [Red Hat Universal Base Image 8.8-1072.1697626218](#)
- IBM MQ A Imagem Contêiner do gerenciador de filas 9.3.3.2-r2 inclui a versão 3.1.7 (2023.4.0) do [Instana MQ Exit](#).

9.3.3.2-r1



Versão do operador necessária

[2.4.4](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.2-r1`
- `icr.io/ibm-messaging/mq:9.3.3.2-r1`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.3](#)

O que mudou

⁵

- [O que mudou no IBM MQ 9.3.3](#)
- Com base em [Red Hat Universal Base Image 8.8-1072.1697626218](#)
- Atualiza o nível de libcurl para 8.4.0

⁵ Uma versão anterior deste tópico declarou incorretamente que a IBM MQ Imagem do contêiner do gerenciador de fila 9.3.3.2-r1 inclui a versão 3.1.7 (2023.4.0) do [Instana MQ Exit](#).

IBM MQ APARs incluídos

- IT41871
- IT44585
- IT44623
- IT44762

9.3.3.1-r2



Versão do operador necessária

[2.4.3](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.1-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.1-r2](#)
- [icr.io/ibm-messaging/mq:9.3.3.1-r2](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.3](#)

O que mudou

- [O que mudou no IBM MQ 9.3.3](#)
- Atualização somente de segurança construída em [IBM MQ 9.3.3.1-r1](#)
- Com base em [Red Hat Universal Base Image 8.8-1037](#)

9.3.3.1-r1



Versão do operador necessária

[2.4.2](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.1-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.1-r1](#)
- [icr.io/ibm-messaging/mq:9.3.3.1-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.3](#)

O que mudou

- [O que mudou no IBM MQ 9.3.3](#)
- Com base em [Red Hat Universal Base Image 8.8-1037](#)

9.3.3.0-r2



Versão do operador necessária

[2.4.1](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.0-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.0-r2
- icr.io/ibm-messaging/mq:9.3.3.0-r2

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.3](#)

O que mudou

- [O que mudou no IBM MQ 9.3.3](#)
- Com base em [Red Hat Universal Base Image 8.8-1014](#)

9.3.3.0-r1



Versão do operador necessária

[2.4.0](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.0-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.0-r1
- icr.io/ibm-messaging/mq:9.3.3.0-r1

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.3](#)

O que mudou

- [O que mudou no IBM MQ 9.3.3](#)
- Com base em [Red Hat Universal Base Image 8.8-860](#)
- IBM MQ Imagem do contêiner do gerenciador de filas 9.3.3.0-r1 inclui a [versão 3.1.2 \(2023.2.0\)](#) do [Instana MQ Exit](#).

9.3.2.1-r2



Versão do operador necessária

[2.3.3](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.1-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.1-r2
- icr.io/ibm-messaging/mq:9.3.2.1-r2

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.2](#)

O que mudou

- [O que mudou no IBM MQ 9.3.2](#)
- Com base na [Red Hat Universal Base Image 8.7-1107](#)

9.3.2.1-r1



Versão do operador necessária

[2.3.2](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.1-r1`
- `icr.io/ibm-messaging/mq:9.3.2.1-r1`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.2](#)

O que mudou

- [O que mudou no IBM MQ 9.3.2](#)
- Com base na [Red Hat Universal Base Image 8.7-1107](#)

9.3.2.0-r2



Versão do operador necessária

[2.3.1](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.0-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.0-r2`
- `icr.io/ibm-messaging/mq:9.3.2.0-r2`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.2](#)

O que mudou

- [O que mudou no IBM MQ 9.3.2](#)
- Baseado em [Red Hat Universal Base Image 8.7-1085](#).

9.3.2.0-r1



Versão do operador necessária

[2.3.0](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.0-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.0-r1`
- `icr.io/ibm-messaging/mq:9.3.2.0-r1`

O Quê Há de Novo

- O que há de novo no [IBM MQ 9.3.2](#)
- A variável de ambiente `MQ_LOGGING_CONSOLE_FORMAT` agora está configurada, que substitui a variável `LOG_FORMAT` descontinuada.

O que mudou

- O que mudou no [IBM MQ 9.3.2](#)
- Certificados do gerenciador de filas com o mesmo Nome Distinto (DN) do emissor (CA) não são suportados. Um certificado deve ter um nome distinto de sujeito exclusivo.
- Com base em [Red Hat Universal Base Image 8.7-1049.1675784874](#)

9.3.1.1-r1



Versão do operador necessária

[2.2.2](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.1-r1`
- `icr.io/ibm-messaging/mq:9.3.1.1-r1`

O Quê Há de Novo

- O que há de novo no [IBM MQ 9.3.1](#)

O que mudou

- O que mudou no [IBM MQ 9.3.1](#)
- Baseado em [Red Hat Universal Base Image 8.7-1031](#).
- IBM MQ A Imagem do contêiner do gerenciador de filas 9.3.1.1-r1 inclui [versão 2.4.3 \(2022.4.3\)](#) do [IBM Instana MQ Exit](#).

9.3.1.0-r3



Versão do operador necessária

[2.2.1](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.0-r3`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.0-r3`
- `icr.io/ibm-messaging/mq:9.3.1.0-r3`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.1](#)

O que mudou

- O que mudou no [IBM MQ 9.3.1](#)
- Com base em [Red Hat Universal Base Image 8.7-923.1669829893](#)
- IBM MQ Imagem Contêiner do gerenciador de filas 9.3.1.0-r3 inclui [versão 2.4.3 \(2022.4.3\) do IBM Instana MQ Exit](#).

9.3.1.0-r2



Versão do operador necessária

[2.2.0](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.0-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.0-r2](#)
- [icr.io/ibm-messaging/mq:9.3.1.0-r2](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.1](#)
- A partir da imagem do Contêiner do Gerenciador de Filas 9.3.1.0-r2 (CD) IBM MQ , IBM Instana o rastreamento é suportado nativamente. O IBM MQ versão 9.3.1.0-r2 inclui [versão 2.4.0 \(2022.4.0\) do IBM Instana MQ Exit](#). Para ativar o rastreamento do IBM Instana , consulte [“Integrando o IBM MQ ao rastreamento do IBM Instana” na página 157](#)

O que mudou

- [O que mudou no IBM MQ 9.3.1](#)
- Baseado em [Red Hat Universal Base Image 8.7-923](#).
- Se uma chave e um certificado não forem fornecidos, o atributo do gerenciador de filas **SSLKEYR** agora será configurado como em branco em vez de ser configurado como `"/run/runmqserver/tls/key"`.

9.3.1.0-r1



Versão do operador necessária

[2.1.0](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.0-r1](#)
- [icr.io/ibm-messaging/mq:9.3.1.0-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.1](#)

O que mudou

- [O que mudou no IBM MQ 9.3.1](#)
- Baseado em [Red Hat Universal Base Image 8.6-941](#).

9.3.0.17-r3

CP4I-LTS

Versão do operador necessária

[2.0.22](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.17-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.17-r3](#)
- [icr.io/ibm-messaging/mq:9.3.0.17-r3](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no IBM MQ 9.3.0.0-r1
- Com base na [Red Hat Universal Base Image 9.4-949.1716471857](#)

9.3.0.17-r2

CP4I-LTS

Versão do operador necessária

[2.0.22](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.17-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.17-r2](#)
- [icr.io/ibm-messaging/mq:9.3.0.17-r2](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização de segurança construída em IBM MQ 9.3.0.0-r1
- Baseado em [Red Hat Universal Base Image 8.9-1161.1715068733](#)
- A biblioteca [golang.org/x/net](#) foi atualizada para corrigir uma vulnerabilidade relatada

9.3.0.17-r1

CP4I-LTS

Versão do operador necessária

[2.0.21](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.17-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.17-r1
- icr.io/ibm-messaging/mq:9.3.0.17-r1

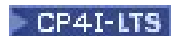
O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização de segurança construída em IBM MQ 9.3.0.0-r1
- Baseado em [Red Hat Universal Base Image 8.9-1161](#)
- As vulnerabilidades de segurança relatadas "dependabot" foram resolvidas.

9.3.0.16-r2



Versão do operador necessária

[2.0.20](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.16-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.16-r2
- icr.io/ibm-messaging/mq:9.3.0.16-r2

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização de segurança construída em IBM MQ 9.3.0.0-r1
- Com base em [Red Hat Universal Base Image 8.9-1137](#)
- Você só precisará retirar a nova imagem 9.3.0.16-r2 se tiver o Painel de Operações ativado.

9.3.0.16-r1



Versão do operador necessária

[2.0.19](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.16-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.16-r1
- icr.io/ibm-messaging/mq:9.3.0.16-r1

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização de segurança construída em IBM MQ 9.3.0.0-r1
- Com base em [Red Hat Universal Base Image 8.9-1137](#)
- A biblioteca [golang.org/x/crypto](#) foi atualizada para corrigir a vulnerabilidade CVE-2023-48795 .
- Algoritmo SHA512 mais seguro usado em vez de SHA256 para criar certificado autoassinado no keystore da web.
- O keystore PKCS#12 para uso com o servidor da web IBM MQ agora é gerado usando a função **Pkcs12.Modern.Encode** , que usa a criptografia SHA-2 (gerada anteriormente usando uma criptografia SHA-1 anterior).
- A vulnerabilidade relatada nos usos do método **PathTraversal** foi corrigida

9.3.0.15-r1

CP4I-LTS

Versão do operador necessária

[2.0.18](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.15-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.15-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.15-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base em [Red Hat Universal Base Image 8.9-1108](#)

9.3.0.11-r2

CP4I-LTS

Versão do operador necessária

[2.0.17](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.11-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.11-r2](#)
- [icr.io/ibm-messaging/mq:9.3.0.11-r2](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Baseado em [Red Hat Universal Base Image 8.9-1029](#).

9.3.0.11-r1

CP4I-LTS

Versão do operador necessária

[2.0.16](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.11-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.11-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.11-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base em [Red Hat Universal Base Image 8.8-1072.1697626218](#)
- Atualiza o nível de libcurl para 8.4.0

9.3.0.10-r2

CP4I-LTS

Versão do operador necessária

[2.0.15](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.10-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.10-r2](#)
- [icr.io/ibm-messaging/mq:9.3.0.10-r2](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base em [Red Hat Universal Base Image 8.8-1037](#)

9.3.0.10-r1

CP4I-LTS

Versão do operador necessária

[2.0.14](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.10-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.10-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.10-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base em [Red Hat Universal Base Image 8.8-1037](#)

9.3.0.6-r1

CP4I-LTS

Versão do operador necessária

[2.0.13](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.6-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.6-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.6-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base em [Red Hat Universal Base Image 8.8-1014](#)

9.3.0.5-r3

CP4I-LTS

Versão do operador necessária

[2.0.12](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.5-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.5-r3](#)
- [icr.io/ibm-messaging/mq:9.3.0.5-r3](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base em [Red Hat Universal Base Image 8.8-860](#)

9.3.0.5-r2

CP4I-LTS

Versão do operador necessária

[2.0.11](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.5-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.5-r2](#)
- [icr.io/ibm-messaging/mq:9.3.0.5-r2](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base na [Red Hat Universal Base Image 8.7-1107](#)

Importante: Para usuários do Painel de Operações no IBM MQ LTS Imagem do Contêiner do Gerenciador de filas 9.3.0.5-r2

Quando o Painel de Operações é ativado, o IBM MQ LTS Queue Manager Container Image 9.3.0.5-r2 implementa as imagens do Operations Dashboard Agent e do Collector que não contêm as correções de segurança mais recentes disponíveis no momento de sua Disponibilidade Geral.

Mitigação: faça upgrade para pelo menos 9.3.0.5-r3 todas as IBM MQ LTS imagens do Queue Manager Container 9.3.0.5-r2 com o Painel de Operações ativado. Consulte [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando Red Hat OpenShift”](#) na página 131.

9.3.0.5-r1

CP4I-LTS

Versão do operador necessária

[2.0.10](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.5-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.5-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.5-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base na [Red Hat Universal Base Image 8.7-1107](#)

Importante: Para usuários do Painel de Operações no IBM MQ LTS Imagem do Contêiner do Gerenciador de filas 9.3.0.5-r1

Quando o Painel de Operações está ativado, o IBM MQ LTS Queue Manager Container Image 9.3.0.5-r1 implementa as imagens do Operations Dashboard Agent e do Collector que não contêm as correções de segurança mais recentes disponíveis no momento de sua Disponibilidade Geral.

Mitigação: faça upgrade para pelo menos 9.3.0.5-r3 todas as IBM MQ LTS imagens do Queue Manager Container 9.3.0.5-r1 com o Painel de Operações ativado. Consulte [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando Red Hat OpenShift” na página 131.](#)

9.3.0.4-r2

CP4I-LTS

Versão do operador necessária

[2.0.9](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.4-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.4-r2`
- `icr.io/ibm-messaging/mq:9.3.0.4-r2`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Baseado em [Red Hat Universal Base Image 8.7-1085](#).

9.3.0.4-r1

CP4I-LTS

Versão do operador necessária

[2.0.8](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.4-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.4-r1`
- `icr.io/ibm-messaging/mq:9.3.0.4-r1`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)

- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base em [Red Hat Universal Base Image 8.7-1049.1675784874](#)

9.3.0.3-r1

CP4I-LTS

Versão do operador necessária

[2.0.7](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.3-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.3-r1](#)
- [icr.io/ibm-messaging/mq:9.3.0.3-r1](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Baseado em [Red Hat Universal Base Image 8.7-1031](#).

9.3.0.1-r4

CP4I-LTS

Versão do operador necessária

[2.0.6](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r4](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r4](#)
- [icr.io/ibm-messaging/mq:9.3.0.1-r4](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Com base em [Red Hat Universal Base Image 8.7-923.1669829893](#)

9.3.0.1-r3

CP4I-LTS

Versão do operador necessária

[2.0.5](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r3`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r3`
- `icr.io/ibm-messaging/mq:9.3.0.1-r3`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Baseado em [Red Hat Universal Base Image 8.7-923](#).

9.3.0.1-r2



Versão do operador necessária

[2.0.4](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r2`
- `icr.io/ibm-messaging/mq:9.3.0.1-r2`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Baseado em [Red Hat Universal Base Image 8.6-941](#).

9.3.0.1-r1



Versão do operador necessária

[2.0.3](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r1`
- `icr.io/ibm-messaging/mq:9.3.0.1-r1`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)

- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Baseado em [Red Hat Universal Base Image 8.6-941](#).

9.3.0.0-r3



Versão do operador necessária

[2.0.2](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.0-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.0-r3](#)
- [icr.io/ibm-messaging/mq:9.3.0.0-r3](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Baseado em [Red Hat Universal Base Image 8.6-902](#).

9.3.0.0-r2



Versão do operador necessária

[2.0.1](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.0-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.0-r2](#)
- [icr.io/ibm-messaging/mq:9.3.0.0-r2](#)

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- Atualização somente de segurança construída no [IBM MQ 9.3.0.0-r1](#)
- Baseado em [Red Hat Universal Base Image 8.6-854](#).

9.3.0.0-r1



Versão do operador necessária

[2.0.0](#) ou superior

Arquiteturas suportadas

amd64, s390x, ppc64le

Imagens

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.0-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.0-r1`
- `icr.io/ibm-messaging/mq:9.3.0.0-r1`

O Quê Há de Novo

- [O que há de novo no IBM MQ 9.3.0](#)

O que mudou

- [O que mudou no IBM MQ 9.3.0](#)
- A configuração do desenvolvedor padrão na imagem MQ Advanced for Developers agora usa `ANY_TLS12_OR_HIGHER`.
- Corrigido um problema com o servidor da web IBM MQ , que causou um erro no log devido a preferências Java ausentes.
- Com base em [Red Hat Universal Base Image 8.6-751.1655117800](#)

OpenShift

CP4I

Verificando assinaturas de imagem

A partir de março de 2023, as imagens de contêiner do gerenciador de filas IBM MQ Operator e do IBM MQ são assinadas digitalmente

Primeiros IBM MQ Operadores a serem assinados:

- 2.3.1 (CD)
- 2.0.9 (LTS)




Primeiras imagens de contêiner do gerenciador de filas do IBM MQ a serem assinadas:

- 9.3.2.0-r2 (CD)
- 9.3.0.4-r2 (LTS)

Sobre esta tarefa

As assinaturas digitais fornecem uma maneira para os consumidores de conteúdo garantirem que o que eles baixam é autêntico (originado da fonte esperada) e tem integridade (é o que esperamos que seja).

Procedimento

- Verifique as assinaturas de imagens de contêiner do gerenciador de filas IBM MQ Operator e IBM MQ :
 -   Para um IBM MQ Operator em 3.0.0 ou mais recente ou uma IBM MQ imagem de contêiner do gerenciador de filas em 9.3.4.0-r1 ou mais recente, consulte [Verificando assinaturas de imagem](#) na documentação do IBM Cloud Pak for Integration (CP4I) 2023.4 .
 -  Para um IBM MQ Operator em 2.4.x ou uma IBM MQ imagem de contêiner do gerenciador de filas em 9.3.3.x, consulte [Verificando assinaturas de imagem](#) na documentação do CP4I 2023.2 .
 - Para um IBM MQ Operator anterior a 2.4.0 ou uma IBM MQ imagem de contêiner do gerenciador de filas anterior a 9.3.3.0-r1, consulte [Verificando assinaturas de imagem](#) na documentação do CP4I 2022.4 .

OpenShift

CD

CP4I-LTS

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 81](#)
2. [“Extraindo a configuração do gerenciador de filas” na página 81](#)
3. Opcional: [“Opcional: extraindo e adquirindo as chaves e os certificados do gerenciador de filas” na página 82](#)
4. Opcional: [“Opcional: configurando o LDAP” na página 84](#)
5. Opcional: [“Opcional: mudando os endereços IP e nomes de host na configuração do IBM MQ” na página 91](#)
6. [“Atualizando a configuração do gerenciador de filas para um ambiente de contêiner” na página 93](#)
7. [“Selecionando a arquitetura de HA de destino para o IBM MQ em execução em contêineres” na página 95](#)
8. [“Criando os recursos para o gerenciador de filas” na página 96](#)
9. [“Criando o novo gerenciador de filas no Red Hat OpenShift” na página 97](#)
10. [“Verificando a nova implementação de contêiner” na página 101](#)

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 79.

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 165.
4. Se o seu sistema IBM MQ incluir Alta Disponibilidade, revise as opções disponíveis.

Consulte [“Alta disponibilidade para IBM MQ em contêineres”](#) na página 25.

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](#).

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.

Nota:  Certificados do gerenciador de filas com o mesmo Nome Distinto (DN) do emissor (CA) não são suportados. Um certificado deve ter um nome distinto de sujeito exclusivo. O produto agora verifica se os DN's não são os mesmos.

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](#).

OpenShift CD CP4I-LTS Opcional: configurando 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 [extrai 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) +
* ALTDATA(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.

LDAP parte 1: atualizando o backup do IBM MQ para o servidor 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 84.

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

- **CONNNAME** 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:

```
Open ▾ [icon]
backup.mqsc
*****
* 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: 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) +
```

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 +
* ALTDATE(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 CD CP4I-LTS **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 84](#).

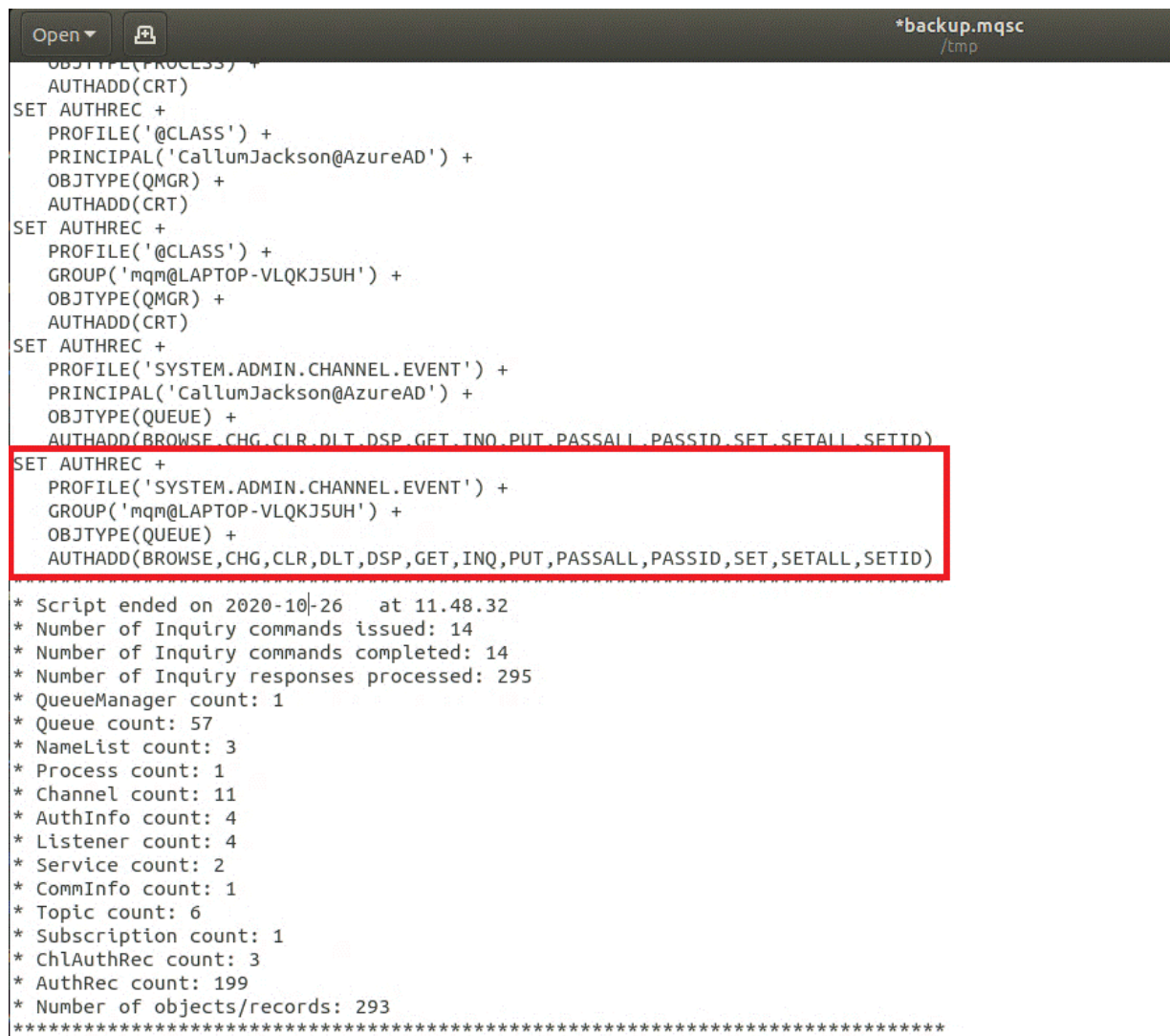
Há duas partes para atualizar as informações de autorização do LDAP:

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 'CONNAME\|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 `CONNAME` 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 `CONNAME`. Por exemplo:

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

Na saída original do comando `grep 'CONNAME\|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 > CD > CP4I-LTS 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 97).

Procedimento

1. Revise as duas opções.

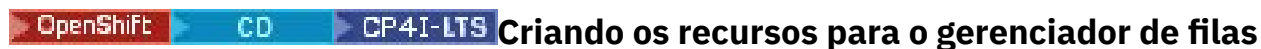
Para obter uma descrição abrangente dessas duas opções, consulte “Alta disponibilidade para IBM MQ em contêineres” na página 25.

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.



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 82, 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 95, 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: 9.3.5.1-r2
  license:
    accept: true
    license: L-VTPK-22YZPK
    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: 9.3.5.1-r2
  license:
    accept: true
    license: L-VTPK-22YZPK
    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”](https://www.ibm.com/docs/en/mq/1beta1) na página 182.

```
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.

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

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-mjsc-migrated` e `backup.mjsc`.

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: 9.3.5.1-r2
  license:
    accept: true
    license: L-VTPK-22YZPK
    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 97](#). 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 79](#).

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 154.

OpenShift

CP4I

Instalando o IBM MQ Operator

O IBM MQ Operator pode ser instalado no Red Hat OpenShift usando o console do OpenShift ou a interface da linha de comandos (CLI)

Antes de começar

Para assegurar que sua instalação seja o mais suave possível, certifique-se de entender todos os pré-requisitos e requisitos antes de iniciar sua instalação. Consulte [“Planejando-se para o IBM MQ em contêineres”](#) na página 5.

Importante: **V 9.3.4** Revise a orientação sobre [estruturação de sua implementação](#) antes de instalar o IBM MQ Operator

Sobre esta tarefa

As etapas a seguir representam o fluxo de tarefas típico para instalar seu IBM MQ Operator:

1. [Instale o Red Hat OpenShift Container Platform.](#)
2. [Configurar o armazenamento.](#)
3. [Espelhar imagens \(somente isolamento físico\).](#)
4. [Inclua o catálogo do operador IBM e prepare seu cluster..](#)
5. [Instale o IBM MQ Operator .](#)
6. [Crie o segredo da chave de autorização \(somente instalações online\)..](#)
7. [Opcional: instale IBM Cloud Pak for Integration \(CP4I\) e suas dependências.](#)
8. [Implementar o License Service](#)
9. [Implementar um Gerenciador de Filas](#)

Procedimento

1. Instale Red Hat OpenShift Container Platform.

Para obter etapas detalhadas para instalar o OpenShift, consulte [Instalando o software Red Hat 4.6 ou posterior](#).

Importante: Assegure-se de instalar uma versão suportada do OpenShift Container Platform. Por exemplo, para usar o IBM MQ Operator 2.0 ou posterior, deve-se instalar o OpenShift Container Platform 4.12 ou posterior. Além disso, observe que apenas as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias numeradas pares, por exemplo, 4.14 e 4.16. Para obter mais informações, consulte [IBM Cloud Pak e Red Hat OpenShift Container Platform compatibilidade](#)..

Para quaisquer etapas que usem a CLI do Red Hat OpenShift Container Platform , deve-se ter efetuado login no cluster do OpenShift com o `oc login`. Para instalar a CLI, consulte [Introdução à OpenShift CLI](#).

Depois de instalar o OpenShift, é possível verificar e obter acesso ao seu software de contêiner usando a chave de autorização do IBM criada em [Criar o segredo da chave de autorização](#).

2. Configurar o armazenamento.

Deve-se definir as classes de armazenamento no Red Hat OpenShift Container Platform e definir sua configuração de armazenamento para satisfazer seus requisitos de dimensionamento..

Importante: Os gerenciadores de filas de instância única e de HA nativa do IBM MQ podem usar o modo de acesso RWO, enquanto os gerenciadores de filas de várias instâncias requerem RWX conforme descrito em “[Considerações de armazenamento para o IBM MQ Operator](#)” na página 21. IBM MQ gerenciadores de filas de várias instâncias requerem características específicas do sistema de arquivos, que podem ser verificadas usando as instruções para [Testando um sistema de arquivo compartilhado para IBM MQ](#).

Uma lista de sistemas de arquivos em conformidade e fora de conformidade conhecidos e notas sobre outros limites ou restrições pode ser localizada na [Instrução de teste para sistemas de arquivos IBM MQ](#).

Os provedores de armazenamento recomendados podem ser localizados na página CP4I [Considerações de armazenamento](#)

3. **9.3.4**

Imagens espelhadas (somente isolamento físico).

Se seu cluster estiver em um ambiente de rede restrito (fisicamente isolado), deve-se espelhar as imagens do IBM MQ . Dependendo de sua configuração, também pode ser necessário espelhar alguns componentes adicionais. Leia as seguintes informações e, em seguida, espelhe as imagens conforme necessário.

- Deve-se espelhar as imagens do IBM MQ Use os valores a seguir:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.1.3
```

- Também deve-se espelhar alguns componentes adicionais necessários se você pretende implementar pelo menos um gerenciador de filas no qual **todas** as instruções a seguir são verdadeiras:

- Você está usando IBM MQ 9.3.4 ou posterior.
- Você está usando uma licença do CP4I
- O IBM MQ Console está ativado
- Você está usando o serviço IBM Cloud Pak for Integration Keycloak para autenticação e autorização de conexão única (SSO) IBM MQ Console (o padrão).

Se as instruções anteriores forem verdadeiras, a SSO será fornecida pelo Keycloak e você deverá espelhar cada um dos componentes a seguir:

- IBM Cloud Pak foundational services
- Certificate Manager. Se você tiver instalado uma versão do operador IBM Cloud Pak foundational services anterior à versão 4.4, deverá espelhar o Certificate Manager.⁶
- IBM Cloud Pak for Integration
- Keycloak (operadorRed Hat OpenShift)

Para criar imagens espelhadas, consulte [Imagens de Espelhamento para um cluster fisicamente isolado](#)

4. Inclua a origem do catálogo IBM MQ Operator .

Inclua a origem do catálogo que torna os operadores disponíveis para o seu cluster Consulte [“Incluindo a origem do catálogo IBM MQ Operator” na página 106](#).

5. Instale o IBM MQ Operator.

Escolha uma das duas opções a seguir (use o console ou use a CLI):

- Opção 1: [Instalar o IBM MQ Operator usando o OpenShift console](#).
- Opção 2: [Instalar o IBM MQ Operator usando a OpenShift CLI](#).

6. Crie o segredo da chave de autorização (somente instalações on-line)

⁶ Na versão 4.4 do IBM Cloud Pak foundational services , esse espelhamento não é mais necessário.

O IBM MQ Operator implementa imagens do gerenciador de filas que são extraídas de um registro de contêiner que executa uma verificação de autorização 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 no namespace no qual instalará gerenciadores de filas, siga estas instruções para obter uma chave de autorização e criar um segredo de extração.

Nota: A chave de autorização não será necessária se somente os gerenciadores de filas IBM MQ Advanced for Developers (Não Garantidos) forem implementados.

É possível criar o segredo da chave de autorização usando o console ou a CLI do OpenShift. O exemplo a seguir usa a CLI:

- a. Obtenha a chave de autorização designada ao seu ID do IBM. 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.
- c. Na CLI do OpenShift, execute o comando a seguir para criar um segredo de extração de imagem chamado `ibm-entitlement-key`.

```
oc create secret docker-registry ibm-entitlement-key \
--docker-server=cp.icr.io \
--docker-username=cp \
--docker-password=<entitlement-key> \
--docker-email=<user-email> \
--namespace=<namespace>
```

Em que `<entitlement-key>` é a chave de autorização que você copiou na etapa b, `<user-email>` é o ID IBM associado ao software autorizado e `<namespace>` é o namespace no qual você instalou o seu IBM MQ Operator.

7. Opcional: Instale o CP4I e suas dependências.

Há alguns componentes necessários adicionais ao implementar pelo menos um gerenciador de filas em que **todas** as instruções a seguir são verdadeiras:

- Você está usando IBM MQ 9.3.4 ou posterior.
- Você está usando uma licença do CP4I
- O IBM MQ Console está ativado
- Você está usando o serviço CP4I Keycloak para autenticação e autorização de conexão única (SSO) IBM MQ Console (o padrão).

Se todas as instruções anteriores forem verdadeiras, a SSO será fornecida pelo Keycloak e você deverá concluir as etapas adicionais a seguir:

- Instale o operador IBM Cloud Pak foundational services no mesmo modo de instalação que o Operador CP4I. Para obter versões suportadas, consulte [Versões do canal do operador para esta liberação](#)
- Se você tiver instalado uma versão do operador IBM Cloud Pak foundational services anterior à versão 4.4, [Instale o operador cert-manager para o Red Hat OpenShift Container Platform](#).⁷
- [Instale o Operador CP4I](#).
- Opcional: implementar a IU da Plataforma.
 - a. Crie o namespace `ibm-common-services`. Quando tiver efetuado login em seu cluster do OpenShift por meio da CLI, execute o comando a seguir:

```
oc new-project ibm-common-services
```

- b. [Implementar a IU da plataforma](#).

8. Implemente o License Service.

⁷ Na versão 4.4 do IBM Cloud Pak foundational services isso não é mais necessário.

Isso é necessário para monitorar o uso sob licença de gerenciadores de fila Siga as instruções em [Implementando License Service](#).

9. Implementar um gerenciador de filas.

Para obter instruções sobre como implementar um gerenciador de filas de "início rápido" de exemplo, consulte [“Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform”](#) na página 116.

Tarefas relacionadas

[“Desinstalando o IBM MQ Operator”](#) na página 119

É possível usar o console ou CLI do Red Hat OpenShift para desinstalar o IBM MQ Operator do Red Hat OpenShift.

Referências relacionadas

[“Instalando o IBM MQ Operator 2.x em um ambiente de isolamento físico”](#) na página 110

Este tutorial o orienta na instalação do IBM MQ Operator 2.x 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.

Incluindo a origem do catálogo IBM MQ Operator

Incluir a origem do catálogo em seu cluster do OpenShift inclui os operadores IBM na lista de operadores que podem ser instalados.

Antes de começar

Esta tarefa supõe que você tenha concluído as primeiras 3 etapas do [“Instalando o IBM MQ Operator”](#) na página 103

Essa tarefa deve ser executada por um administrador de cluster

Sobre esta tarefa

O catálogo do IBM MQ Operator é um índice de operadores disponíveis para estender a API de um cluster Red Hat OpenShift Container Platform para ativar produtos de software IBM .

Conclua **Opção A: Air-gap** ou **Opção B: Internet**, dependendo de seu cluster estar em um ambiente de rede restrito (fisicamente isolado) ou de seu cluster ter acesso à Internet.

Procedimento

Opção A: Isolamento físico Inclua a origem do catálogo em um ambiente de rede fisicamente isolado.

a) Inclua a origem do catálogo IBM MQ Operator .

Siga as instruções em [Incluindo origens de catálogo em um cluster](#).

Nota: Como você já concluiu a etapa de instalação do operador [Mirror images \(air-gap only\)](#), você só precisa concluir a etapa que aplica a origem do catálogo. Por exemplo:

```
oc apply -f ~/.ibm-pak/data/mirror/${OPERATOR_PACKAGE_NAME}/${OPERATOR_VERSION}/catalog-sources.yaml
```

b) Inclua a origem do catálogo para os componentes adicionais necessários

Há alguns componentes necessários adicionais ao implementar pelo menos um gerenciador de filas em que **todas** as instruções a seguir são verdadeiras:

- Você está usando IBM MQ 9.3.4 ou posterior.
- Você está usando uma licença do IBM Cloud Pak for Integration
- O IBM MQ Console está ativado

- Você está usando o serviço IBM Cloud Pak for Integration Keycloak para autenticação e autorização de conexão única (SSO) IBM MQ Console (o padrão).

Se todas as instruções anteriores forem verdadeiras, então a SSO será fornecida pelo Keycloak. Portanto, assim como para a origem do catálogo do IBM MQ Operator, também deve-se seguir as etapas em Incluindo origens do catálogo em um cluster para cada um desses componentes adicionais necessários:

- IBM Cloud Pak foundational services
 - Certificate Manager. Se você tiver instalado uma versão do operador IBM Cloud Pak foundational services anterior à versão 4.4, deverá espelhar o Certificate Manager.⁸
 - IBM Cloud Pak for Integration
- **Opção B: Internet** Inclua a origem do catálogos em um ambiente que tenha acesso à Internet. Crie um CatalogSource usando a CLI do OpenShift

Inclua o catálogo aplicando o arquivo YAML a seguir no cluster do Red Hat OpenShift Container Platform

a) Crie o YAML CatalogSource .

Salve a definição de recurso a seguir como um arquivo chamado `catalog_source.yaml`

```
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 YAML do CatalogSource

Faça isso no console da web do Red Hat OpenShift Container Platform clicando no botão "+" ou usando a linha de comandos.

Por exemplo, aplique o arquivo executando o comando a seguir:

```
oc apply -f catalog_source.yaml -n openshift-marketplace
```

c) Verifique a criação de CatalogSource bem-sucedida

Execute o seguinte comando:

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
```

Você recebe esta saída em caso de êxito:

NAME	DISPLAY	TYPE	PUBLISHER	AGE
ibm-operator-catalog	IBM operator Catalog	grpc	IBM	28s

Resultados

Agora você está pronto para concluir a Etapa 5 de Instalação do IBM MQ Operator

Instalando o IBM MQ Operator usando o console do OpenShift

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

⁸ Na versão 4.4 do IBM Cloud Pak foundational services, esse espelhamento não é mais necessário.

Antes de começar

Esta tarefa supõe que você tenha concluído as Etapas 1 a 4 de [“Instalando o IBM MQ Operator”](#) na página 103

Procedimento

1. Efetue login em seu console de cluster do Red Hat OpenShift.
2. Na área de janela de navegação, clique em **Operadores > OperatorHub**.
A página OperatorHub é exibida.
3. No campo **Todos os itens**, insira "IBM MQ".
A entrada do catálogo do IBM MQ é exibida.
4. Selecione **IBM MQ**.
A janela do IBM MQ é exibida.
5. Clique em **Instalar**.
A página Instalar Operador é exibida.
6. Insira os seguintes valores:
 - a) Configure **Canal** para a sua versão escolhida
Revise [“Suporte de versão para o IBM MQ Operator”](#) na página 12 para determinar qual canal do operador escolher.
 - b) Configure o **Modo de instalação** como "um namespace específico no cluster" (que pode ser criado na próxima etapa), ou o escopo de todo o 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.
 - c) Opcional: Se você escolheu "um namespace específico no cluster", configure o **Namespace** para o valor do projeto (namespace) no qual você deseja instalar o operador..

Nota: Ao usar o console para instalar o operador, é possível usar um namespace existente, o namespace padrão fornecido pelo operador ou criar um novo namespace. Se você deseja criar um novo namespace, poderá criá-lo a partir desse formulário, conforme a seguir: Na área de janela de navegação, clique em **Início > Projetos**, selecione **Criar Projeto**, especifique o **Nome** do projeto (o namespace) que deseja criar e, em seguida, clique em **Criar**.
 - d) Configure a **Estratégia de Aprovação** como Automático
7. Clique em **Instalar** e aguarde o operador instalar.

Você receberá uma confirmação quando a instalação for concluída.

Para verificar a instalação, navegue para **Operadores > Operadores instalados** e selecione seu projeto na lista suspensa **Projetos**. O status do operador é alterado para Bem-sucedido quando a instalação é concluída.

Como proceder a seguir

Agora você está pronto para [Criar o segredo da chave de autorização](#) (etapa 6 de [“Instalando o IBM MQ Operator”](#) na página 103).

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

O IBM MQ Operator pode ser instalado no Red Hat OpenShift usando a interface da linha de comandos (CLI).

Antes de começar

Esta tarefa supõe que você tenha concluído as Etapas 1 a 4 de [“Instalando o IBM MQ Operator”](#) na página 103

Procedimento

1. Faça login na interface da linha de comandos do Red Hat OpenShift (CLI) usando **oc login**.
2. Opcional: Crie um namespace a ser usado para o IBM MQ Operator

O IBM MQ Operator pode ser instalado com escopo em um único namespace ou em todos os namespaces. Essa etapa será necessária apenas se você desejar instalar em um namespace específico que ainda não exista

Para criar um novo namespace na CLI, execute o comando a seguir:

```
oc create namespace <namespace_name>
```

Em que *< namespace_name >* é o nome do namespace que você deseja criar.

3. Visualize a lista de operadores disponíveis para o cluster no OperatorHub:

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

4. Inspecione o IBM MQ Operator para verificar seu **InstallModes** suportado e **Channels** disponível.

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

5. Opcional: Crie um **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 deseja instalar usar o modo AllNamespaces , o namespace openshift-operators já terá um **OperatorGroup** apropriado em vigor e será possível ignorar essa etapa

Se o Operador usar o modo SingleNamespace e você ainda não tiver um **OperatorGroup** apropriado no lugar, crie um executando o comando a seguir:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <operatorgroup_name>
  namespace: <namespace_name>
spec:
  targetNamespaces:
  - <namespace_name>
EOF
```

6. Revise [“Suporte de versão para o IBM MQ Operator”](#) na página 12 para determinar qual canal do operador escolher.
7. Instale o operador..

Use o comando a seguir, mudando *< ibm-mq-operator-channel >* para corresponder ao canal para a versão do Operador do IBM MQ que você deseja instalar e mudando *< namespace_name >* para **openshift-operators** se você estiver usando o modo "AllNamespaces" ou para o namespace no qual deseja implementar o Operador do IBM MQ se estiver usando o modo "SingleNamespace".

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-mq
  namespace: <namespace_name>
spec:
```

```
channel: <ibm-mq-operator-channel>
installPlanApproval: Automatic
name: ibm-mq
source: ibm-operator-catalog
sourceNamespace: openshift-marketplace
EOF
```

8. Após alguns minutos, o operador é instalado. Execute o comando a seguir para verificar se todos os componentes estão no estado Bem-sucedido:

```
oc get csv -n <namespace_name> | grep ibm-mq
```

Em que < namespace_name > será **openshift-operators** se você estiver usando o modo "AllNamespaces" ou o nome do projeto (namespace) se estiver usando o modo "SingleNamespace".

Como proceder a seguir

Agora você está pronto para [Criar o segredo da chave de autorização](#) (etapa 6 de “[Instalando o IBM MQ Operator](#)” na página 103).

Instalando o IBM MQ Operator 2.x em um ambiente de isolamento físico

Este tutorial o orienta na instalação do IBM MQ Operator 2.x 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.

Antes de Começar

Estas instruções são para instalar a versão 2.x do IBM MQ Operator em um ambiente de isolamento físico. Para instalar o IBM MQ Operator 3.0.0 e posterior, consulte “[Instalando o IBM MQ Operator](#)” na página 103, prestando atenção especial às etapas específicas de isolamento físico.

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

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 executar a instalação de 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 12 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 111
2. “[Preparar um registro do Docker](#)” na página 111
3. “[Prepare um host bastion](#)” na página 111
4. “[Crie variáveis de ambiente para o instalador e inventário de imagem](#)” na página 112
5. “[Faça download do instalador do IBM MQ e do inventário de imagens](#)” na página 113

6. [“Efetue login no cluster do OpenShift Container Platform como um administrador de cluster” na página 113](#)
7. [“Crie um namespace do Kubernetes para o IBM MQ Operator” na página 113](#)
8. [“Espelhe as imagens e configure o cluster” na página 113](#)
9. [“Instale o IBM MQ Operator.” na página 115](#)
10. [“Implementar o IBM MQ Queue Manager” na página 115](#)

Pré-requisitos

1. Um cluster do OpenShift Container Platform deve ser instalado. Para as versões do OpenShift Container Platform suportadas, consulte [“Suporte de versão para o IBM MQ Operator” na página 12](#).
2. Um registro do Docker deve estar disponível. Para obter mais informações, consulte [“Preparar um registro do Docker” na página 111](#).
3. Um servidor de bastion deve ser configurado. Para obter informações adicionais, consulte [“Prepare um host bastion” na página 111](#).

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 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:

- Um exemplo de um registro simples é incluído em [Criando um registro de espelho para instalação em uma rede restrita na documentação do Red Hat OpenShift](#)
- Verifique se cada namespace 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 OpenShift Container Platform usa essas credenciais.

Prepare um host bastion

Prepare um host bastion que possa acessar o cluster 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 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 -xvf 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 OpenShift Container Platform da CLI.

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

6. Crie um diretório que sirva de 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 em 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 OpenShift Container Platform como um administrador de cluster

A seguir, está um comando de exemplo para efetuar login no cluster 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 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 12 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 [“Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform”](#) na página 116.

Tarefas relacionadas

[“\[Descontinuado\]Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou gerenciador de filas em um ambiente de isolamento físico”](#) na página 126

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 Operador ou do gerenciador de filas IBM MQ 2.x .

OpenShift CP4I Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform

Este exemplo implementa um gerenciador de filas de "iniciação rápida", que usa armazenamento temporário (não persistente) e desativa a segurança do IBM MQ . As mensagens não são persistidas entre as reinicializações do gerenciador de filas. É possível ajustar a configuração para mudar muitas configurações do gerenciador de filas.

Sobre esta tarefa

Esta tarefa oferece 3 opções para implementar um gerenciador de filas no OpenShift:

1. [Implementar um gerenciador de filas com o OpenShift console.](#)
2. [Implementar um gerenciador de filas com a OpenShift CLI.](#)
3. [Implementar um gerenciador de filas com o IBM Cloud Pak for Integration Platform UI.](#)

Procedimento

• Opção 1: Implementar um gerenciador de filas com o console do OpenShift .

- a) Implementar um gerenciador de filas.
 - a. Efetue login no console do OpenShift com suas credenciais de administrador do cluster do Red Hat OpenShift Container Platform .
 - b. Altere **Projeto** para o namespace no qual o IBM MQ Operator foi instalado. Selecione o namespace na lista suspensa **Projeto** .
 - c. Na área de janela de navegação, clique em **Operadores > Operadores Instalados**
 - d. Na lista no painel Operadores instalados, localize e clique em **IBM MQ**.
 - e. Clique na guia **Gerenciador de Filas**.
 - f. Clique no botão **Criar QueueManager**. O painel de criação da instância é exibido e oferece dois métodos para configurar o recurso; a **Visualização de Formulário** e a **Visualização YAML**. A **Visualização de Formulário** é selecionada por padrão

b) Configure o gerenciador de filas

Etapa 2 Opção 1: Configurar na **visualização Formulário**.

A **Visualização de Formulário** abre um formulário que pode ser usado para visualizar ou modificar a configuração do recurso

- a. Ao lado de **Licença**, clique na seta para expandir a seção de aceitação da licença
- b. Configure **Licença aceita** como **true** se você aceitar o contrato de licença.
- c. Clique na seta para abrir a lista suspensa e selecione uma licença. 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 182. Deve-se aceitar a licença para implementar um gerenciador de filas.
- d. 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.

Etapa 2 Opção 2: configure na **visualização YAML**.

A **visualização YAML** abre um editor contendo um arquivo YAML de exemplo para um QueueManager. Atualize os valores no arquivo seguindo as etapas abaixo.

- a. Altere metadata . namespace para o nome do projeto (namespace).

- b. Altere o valor de `spec.license.license` para a sequência de licença que corresponda aos seus requisitos.. Consulte [“Referência de licenciamento para mq.ibm.com/v1beta1”](#) na página 182 para obter os detalhes da licença
 - c. Mude `spec.license.accept` para `true` se você aceitar o contrato de licença
 - d. 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`.
- c) Verifique a criação do gerenciador de filas
- É possível verificar se você criou um Gerenciador de Filas concluindo as seguintes etapas:
- a. Assegure-se de que você esteja no namespace no qual criou seu IBM MQ Operator .
 - b. Na tela **Início** , clique em **Operadores > Operadores instalados**, em seguida, selecione o IBM MQ Operator instalado para o qual você criou o gerenciador de filas.
 - c. Clique na guia **Gerenciador de Filas**. A criação terá sido concluída quando o status de `QueueManager` for `Running`.
- **Opção 2: implementar um gerenciador de filas com a CLI do OpenShift .**
- 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: 9.3.5.1-r2
  license:
    accept: false
    license: L-VTPK-22YZPK
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

Importante: Se aceitar o contrato de licença, altere `accept: false` para `accept: true`. Consulte [“Referência de licenciamento para mq.ibm.com/v1beta1”](#) na página 182 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 Conexão Única no IBM Cloud Pak for Integration **V 9.3.4**. A partir do IBM Cloud Pak for Integration 2023.4.1, para a Conexão Única funcionar, primeiro será necessário [instalar outros IBM Cloud Pak for Integration componentes](#).

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: 9.3.5.1-r2
  license:
    accept: false
    license: L-AMRD-XH6P3Q
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

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 182 para obter detalhes sobre a licença.

- b) Crie o objeto `QueueManager` ..

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

- c) Verifique a criação do gerenciador de filas

Verifique se você criou um gerenciador de filas concluindo as etapas a seguir:

- a. Valide a implementação:

```
oc describe queuemanager <QueueManagerResourceName>
```

- b. Verifique o status:

```
oc describe queuemanager quickstart
```

• **Opção 3: Implementar um gerenciador de filas com o IBM Cloud Pak for Integration Platform UI.**

- a) Em um navegador, ative o IBM Cloud Pak for Integration Platform UI.

- b) No IBM Cloud Pak for Integration Platform UI, clique em **Criar instância**

- c) Selecione **Sistema de Mensagense**, em seguida, clique em **Avançar**

O painel de criação da instância é exibido e oferece dois métodos para configurar o recurso; a **Visualização de Formulário** e a **Visualização YAML**. A **Visualização de Formulário** é selecionada por padrão

- 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**.

Consulte [“Referência de licenciamento para mq.ibm.com/v1beta1”](#) na página 182 para obter detalhes sobre a licença. 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 UI, use a seção **Configuração do gerenciador de filas**.

Por padrão, o nome do gerenciador de filas usado pelos aplicativos clientes do IBM MQ é igual ao 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`

- h) Verifique a criação do gerenciador de filas

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 154

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 171
Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster do Red Hat OpenShift Container Platform

[“Exemplos para configurar um gerenciador de filas”](#) na página 134

Um gerenciador de filas pode ser configurado ajustando o conteúdo do recurso customizado do QueueManager.

OpenShift

CP4I

Desinstalando o IBM MQ Operator

É possível usar o console ou CLI do Red Hat OpenShift para desinstalar o IBM MQ Operator do Red Hat OpenShift.

Procedimento

- Opção 1: Desinstalar o IBM MQ Operator com o console do OpenShift

Nota: Se o IBM MQ Operator for instalado em todos os projetos / namespaces no cluster, repita as etapas 2-6 do procedimento a seguir para cada projeto no qual você deseja excluir gerenciadores de filas.

- a) Efetue login no console da web do Red Hat OpenShift Container Platform com as suas credenciais de administrador do cluster do Red Hat OpenShift Container Platform
- b) Altere **Projeto** para o namespace do qual você deseja desinstalar o IBM MQ Operator. Selecione o namespace na lista suspensa **Projeto**.
- c) Na área de janela de navegação, clique em **Operadores > Operadores instalados**.
- d) Clique no operador **IBM MQ**.
- e) Clique na guia **Gerenciadores de filas** para visualizar os gerenciadores de filas gerenciados por este IBM MQ Operator.
- f) 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.

- g) Opcional: Se apropriado, repita as etapas 2-6 para cada projeto no qual você deseja excluir gerenciadores de filas.
- h) Retorne para **Operadores > Operadores instalados**.
 - i) Ao lado do operador **IBM MQ**, clique no menu de três pontos e selecione **Desinstalar operador**.

- Opção 2: Desinstalar o IBM MQ Operator com a CLI do OpenShift

- a) Efetue login no cluster do Red Hat OpenShift usando `oc login`.
- b) Se o IBM MQ Operator for instalado em um único namespace, conclua as subetapas a seguir:
 - a. Assegure-se de que você esteja no projeto que contém o IBM MQ Operator a ser desinstalado:

```
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. Se nada mais estiver usando serviços comuns, talvez você deseje desinstalar o operador de serviços comuns e excluir o grupo do operador:

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

ii) Visualize o grupo do operador:

```
oc get operatorgroup
```

iii) Exclua o grupo do operador:

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

c) 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. Opcional: se nada mais estiver usando serviços comuns, talvez você queira desinstalar o operador de serviços comuns... Para isso, siga as instruções em [Desinstalando serviços básicos](#) na documentação do produto IBM Cloud Pak foundational services .

Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas

Há diferentes processos de upgrade para as versões Continuous Delivery (CD) e Long Term Support (LTS) do IBM MQ Operator. Conclua a etapa de upgrade para o seu tipo de implementação

Sobre esta tarefa

Para fazer upgrade do IBM MQ Operator e dos gerenciadores de filas, conclua uma das etapas a seguir:

Procedimento

- Opção 1: **Fazer upgrade de implementações para a versão mais recente em seu canal do Operador atual.**

Para fazer upgrade de implementações do IBM MQ Operator para a versão mais recente no canal do Operador atual, consulte [“Fazendo upgrade para uma liberação de segurança mais recente do canal do IBM MQ Operator”](#) na página 121.

- Opção 2: **Atualizar CD implementações.**

Para fazer upgrade de implementações anteriores do CD do IBM MQ Operator para a versão CD mais recente do IBM MQ Operator (versão 3.1.3), consulte [“Migrando para o canal atual do CD do IBM MQ Operator”](#) na página 123.

Nota:

A versão 2.0.x foi liberada como uma liberação CD e LTS , portanto, é possível usar o procedimento em [“Migrando para o canal atual do CD do IBM MQ Operator”](#) na página 123 para fazer upgrade de qualquer versão 2.0.x IBM MQ Operator para a versão CD mais recente do IBM MQ Operator.

Fazendo upgrade para uma liberação de segurança mais recente do canal do IBM MQ Operator

Fazer upgrade do IBM MQ Operator permite fazer upgrade dos gerenciadores de filas.

Antes de começar

Importante: Este tópico é para fazer upgrade de implementações do IBM MQ Operator para a Liberação de Segurança mais recente no Canal da implementação. Se isso não se aplicar à sua implementação, consulte os caminhos de upgrade alternativos descritos em [“Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas”](#) na página 120

Para implementações do IBM MQ Operator em um cluster Red Hat OpenShift que não tem conectividade com a Internet, siga o procedimento em [“\[Descontinuado\]Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operator ou gerenciador de filas em um ambiente de isolamento físico”](#) na página 126

Procedimento

1. Faça o upgrade do IBM MQ Operator para uma versão mais nova



Se você tiver upgrades automáticos configurados, então, após a liberação de uma nova Liberação de Segurança, seu IBM MQ Operator concluirá um upgrade.

Se você não tiver upgrades automáticos configurados, então aprove manualmente seu upgrade do IBM MQ Operator :

- Se houver um upgrade disponível, o **Upgrade Status** poderá ser "Upgrade disponível".
- Nesse caso, pode haver um controle disponível que você pode usar para aprovar o **InstallPlan** que faz upgrade do IBM MQ Operator

2. Faça upgrade de um gerenciador de filas do IBM MQ para uma versão mais nova

A tabela a seguir descreve a versão mais recente do gerenciador de filas do IBM MQ para cada canal ativo do Operador. Usando a versão relevante, siga o procedimento em [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando Red Hat OpenShift”](#) na página 131

Canal do operador	Gerenciador de filas IBM MQ mais recente
 v2.0 (LTS)	9.3.0.17-r3
 v3.1 (CD)	9.3.5.1-r2

Fazer upgrade do IBM MQ Operator permite fazer upgrade de seus gerenciadores de filas.

Antes de começar

Importante: Este tópico é para fazer upgrade de implementações do 1.3.x Long Term Support (LTS) IBM MQ Operator para o fluxo LTS de IBM MQ Operator 2.0.x **apenas**. Se isso não se aplicar à sua implementação, consulte os caminhos de upgrade alternativos descritos em [Atualizando o IBM MQ Operator e os gerenciadores de filas](#).

Para implementações do IBM MQ Operator em um cluster Red Hat OpenShift que não tem conectividade de Internet, siga o procedimento em “[Descontinuado]Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou gerenciador de filas em um ambiente de isolamento físico” na página 126.

Importante: IBM MQ Operator 2.0.x requer:

- Red Hat OpenShift Container Platform 4.12.

Para fazer upgrade, siga o procedimento em [Fazendo upgrade Red Hat OpenShift.](#)

- IBM Cloud Pak foundational services 3.19.x

Ao fazer upgrade do IBM MQ Operator 1.3.x (2020.4), *ambas* as instâncias de um gerenciador de filas de várias instâncias são reiniciadas simultaneamente. Isso ocorre ao mudar a versão do IBM MQ para 9.2.0.5-r3-eus. Há uma atualização contínua do gerenciador de filas do IBM MQ ao fazer upgrade de IBM MQ Operator 1.3.x para 2.0.x. Se você tiver o IBM Cloud Pak for Integration Platform UI instalado, haverá reinicializações adicionais de IBM MQ ao mudar a versão do IBM Cloud Pak for Integration Platform UI para 2020.4.1-8-euse para 2022.2.1-0.

Procedimento

1. Antes de seguir o link na etapa 2, deve-se ler as informações essenciais a seguir para seu upgrade:

- Você deve omitir todas as subetapas para componentes que não estão instalados. Isso inclui o IBM Cloud Pak for Integration Platform UI se você não tiver instalado.
- A Etapa 2 leva você para a documentação do IBM Cloud Pak for Integration . Durante o processo de upgrade você é trazido de volta para o tópico IBM MQ a seguir, para fazer upgrade do IBM MQ Operand: [Fazendo upgrade de um IBM MQ gerenciador de filas](#).

- Todos os usuários do IBM MQ são recomendados para concluir pelo menos as tarefas a seguir, usando as instruções do link na etapa 2, bem como quaisquer outras que se aplicam ao seu ambiente:

– Correção IBM MQ Operator e operando (Correção 2020.4):

- Faça upgrade do IBM MQ Operator para pelo menos a versão 1.3.5 no Canal do operador v1.3-eus .
- Faça upgrade de seu IBM MQ Operand (Queue Manager Container image) para pelo menos a versão 9.2.0.5-r3-eus.

Nota: É recomendado que você atualize seu Operando do IBM MQ para pelo menos essa versão, mas isso não é obrigatório

– Dependências de upgrade:

- Faça upgrade do IBM Cloud Pak foundational services.
- Faça upgrade do OpenShift Container Platform.

– Atualize os operadores:

- Faça upgrade do IBM MQ Operator para 2.0.23.

– Atualize os recursos:

- Faça upgrade do IBM MQ Operand (Queue Manager Container image) para a versão 9.3.0 mais recente (9.3.0.17-r3) para receber as correções de segurança mais recentes.
2. Faça upgrade do IBM MQ Operator e dos gerenciadores de filas, concluindo [Upgrade de IBM MQ Operator 1.3-eus](#) (IBM Cloud Pak for Integration 2020.4).

Tarefas relacionadas

“[\[Descontinuado\]Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou gerenciador de filas em um ambiente de isolamento físico](#)” na página 126

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 Operador ou do gerenciador de filas IBM MQ 2.x .

“[Fazendo upgrade do IBM MQ Operator usando Red Hat OpenShift](#)” na página 129

É possível fazer upgrade do IBM MQ Operator usando o console da web ou a CLI do Red Hat OpenShift .

“[Fazendo upgrade de um gerenciador de filas do IBM MQ usando Red Hat OpenShift](#)” na página 131

Migrando para o canal atual do CD do IBM MQ Operator

Faça upgrade de um IBM MQ Operator anterior para a versão 3.1.3. Fazer upgrade do operador permite fazer upgrade de seus gerenciadores de filas.

Antes de começar

Este tópico é para fazer upgrade de implementações do Continuous Delivery (CD) do IBM MQ Operator anterior à versão 3.1.0, para a versão 3.1.3 **apenas**. Se isso não se aplicar à sua implementação, consulte os caminhos de upgrade alternativos descritos em [Atualizando o IBM MQ Operator e os gerenciadores de filas](#)

Para fazer upgrade para o IBM MQ Operator 3.1.3 , deve-se estar executando Red Hat OpenShift Container Platform 4.12 ou posterior. Para atualizar a plataforma, consulte [Fazendo upgrade Red Hat OpenShift](#).

Nota: Apenas as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias pares, por exemplo, 4.14 e 4.16.

Procedimento

1. Opcional: **Faça upgrade de um IBM MQ Operator que esteja atualmente em uma versão CD anterior a 2.0.0.**

Se o seu IBM MQ Operator estiver atualmente em uma versão 1.x CD , primeiro siga o procedimento em [“Migrando um 1.x CD IBM MQ Operator para a versão 2.0.x”](#) na página 124, em seguida, retorne aqui para fazer upgrade para a versão CD mais recente.

2. Opcional: **Atualize um IBM MQ Operator que esteja atualmente na versão 2.2.x ou 2.3.x para 2.4.x.**

Se seu IBM MQ Operator estiver atualmente em uma versão 2.2.x ou 2.3.x , siga as etapas relevantes em [“Migrando para o canal v2.4 do IBM MQ Operator”](#) na página 125, em seguida, retorne aqui para fazer upgrade para a versão mais recente do CD . Observe que esta é uma etapa de pré-requisito obrigatória antes de fazer upgrade para a versão 3.1.3

3. **Atualizar componentes.**

Selecione uma das opções a seguir:

- **Opção 1:** se você for um usuário CP4I ou tiver implementado pelo menos um de seus gerenciadores de filas usando uma licença CP4I , siga as etapas relevantes para **fazer upgrade de todos os seus componentes**, incluindo o IBM MQ Operator e os gerenciadores de filas, por meio do plano de upgrade gerado:
 - Para fazer upgrade da versão 2023.2, consulte [Fazendo upgrade do 2023.2 gerando um plano de upgrade](#).

- Para fazer upgrade da versão 2022.2, consulte [Fazendo upgrade do 2022.2 gerando um plano de upgrade](#).
 - **Opção 2:** para todos os outros usuários:
 - a. **Espelhar imagens (apenas isolamento físico).**
Deve-se espelhar as imagens do IBM MQ Conclua as etapas no link a seguir, usando apenas estes valores:


```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.1.3
```

Você deve omitir a seção 3.5 "Configurar o cluster", porque a conexão com o registro de imagem deve ter sido configurada durante instalações ou upgrades anteriores.

Link: [imagens de espelhamento para um cluster fisicamente isolado](#).
 - b. **Faça upgrade do IBM MQ Operator para 3.1.3.**
Consulte [“Fazendo upgrade do IBM MQ Operator usando Red Hat OpenShift”](#) na página 129.
 - c. **Atualizar as instâncias.**
Para receber os recursos mais recentes e correções de segurança, faça upgrade do IBM MQ Operand (Queue Manager Container image) para a versão CD mais recente (9.3.5.1-r2). Consulte [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando Red Hat OpenShift”](#) na página 131.
4. Opcional: **Atualize Red Hat OpenShift Container Platform 4.12 para 4.14.**

Em IBM MQ Operator 3.0.0, Red Hat OpenShift Container Platform 4.12 é necessário. Observe que, opcionalmente, é possível optar por fazer upgrade ainda mais para Red Hat OpenShift 4.14. Para verificar as versões compatíveis para cada canal IBM MQ Operator, consulte [“Versões compatíveis do Red Hat OpenShift Container Platform”](#) na página 13. Para fazer upgrade, consulte [Fazendo upgrade Red Hat OpenShift..](#)


5. Opcional: **Fixar a origem específica do catálogo para o IBM MQ Operator.**

Se a instalação da qual você está fazendo upgrade usar o Catálogo do IBM MQ Operator, será necessário fixar a origem do catálogo específica para o IBM MQ Operator Consulte [Movendo para origens de catálogos específicas para cada operador](#)

Tarefas relacionadas

[“\[Descontinuado\]Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou gerenciador de filas em um ambiente de isolamento físico”](#) na página 126

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 Operador ou do gerenciador de filas IBM MQ 2.x .

 **Migrando um 1.x CD IBM MQ Operator para a versão 2.0.x**

Fazer upgrade do IBM MQ Operator permite fazer upgrade de seus gerenciadores de filas.

Antes de começar

Importante: Este tópico é para fazer upgrade de implementações do Continuous Delivery (CD) do IBM MQ Operator anterior à versão 2.0.x para a versão 2.0.x **apenas**. Se isso não se aplicar à sua implementação, consulte os caminhos de upgrade alternativos descritos em [Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas](#)

Para implementações do IBM MQ Operator em um cluster Red Hat OpenShift que não tem conectividade de Internet, siga o procedimento em [“\[Descontinuado\]Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou gerenciador de filas em um ambiente de isolamento físico”](#) na página 126.

Para concluir esse upgrade, os requisitos a seguir para IBM MQ Operator 2.0.0 devem ser atendidos:

- Red Hat OpenShift Container Platform 4.12.

Para fazer upgrade, siga o procedimento em [Fazendo upgrade Red Hat OpenShift.](#)

- IBM Cloud Pak foundational services 3.19

Procedimento

1. Antes de seguir o link na etapa 2, deve-se ler as informações essenciais a seguir para seu upgrade:

- Omita todas as subetapas para os componentes que você não possui instalados Isso inclui o IBM Cloud Pak for Integration Platform UI se você não tiver instalado.
- A Etapa 2 leva você para a documentação do IBM Cloud Pak for Integration . Durante o processo de upgrade você é trazido de volta para o tópico IBM MQ a seguir, para fazer upgrade do IBM MQ Operand: [Fazendo upgrade de um IBM MQ gerenciador de filas.](#)
- Todos os usuários do IBM MQ são recomendados para concluir pelo menos as tarefas a seguir, usando as instruções do link na etapa 2, bem como quaisquer outras que se aplicam ao seu ambiente:
 - Correção IBM MQ Operator e operando (Correção 2021.4):
 - Faça upgrade do seu IBM MQ Operator para pelo menos a versão 1.8.0 no Canal do Operador v1.8
 - Faça upgrade de seu IBM MQ Operand (Queue Manager Container image) para pelo menos a versão 9.2.5.0-r3.
Nota: É recomendado que você atualize seu IBM MQ Operand para a versão atual (9.3.0.17-r3), mas isso não é obrigatório.
 - Dependências de upgrade:
 - Faça upgrade do IBM Cloud Pak foundational services.
 - Faça upgrade do OpenShift Container Platform.
 - Atualize os operadores:
 - Faça upgrade do IBM MQ Operator para 2.0.23.
 - Atualize os recursos:
 - Faça upgrade do IBM MQ Operand (Queue Manager Container image) para a versão 9.3.0 mais recente (9.3.0.17-r3) para receber as correções de segurança mais recentes.

2. Faça upgrade do IBM MQ Operator e dos gerenciadores de fila, concluindo [Upgrade do IBM MQ Operator 1.8 \(IBM Cloud Pak for Integration 2021.4\)](#) ou versão anterior CD IBM MQ Operator .

Como proceder a seguir

Agora você está pronto para fazer upgrade do IBM MQ Operator e dos gerenciadores de filas para a versão CD mais recente (3.1.3). Consulte [“Migrando para o canal atual do CD do IBM MQ Operator”](#) na página 123.

Migrando para o canal v2.4 do IBM MQ Operator

Fazer upgrade do IBM MQ Operator permite fazer upgrade dos gerenciadores de filas.

Antes de começar

Importante: Este tópico é para fazer upgrade de implementações Continuous Delivery (CD) do IBM MQ Operator anterior à versão 2.4.0 para a versão 2.4.8 **apenas**. Esta é uma etapa intermediária para fazer upgrade para a versão CD mais recente do IBM MQ Operator; o canal v2.4 não recebe atualizações de segurança. Se isso não se aplicar à sua implementação, consulte os caminhos de upgrade alternativos descritos em [Atualizando o IBM MQ Operator e os gerenciadores de filas](#)

Para implementações do IBM MQ Operator em um cluster Red Hat OpenShift que não tem conectividade com a Internet, siga o procedimento em “[Descontinuado]Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou gerenciador de filas em um ambiente de isolamento físico” na página 126

Para concluir esse upgrade, os seguintes requisitos para o IBM MQ Operator 2.4.8 devem ser atendidos:

- Red Hat OpenShift Container Platform 4.12.

Para fazer upgrade, siga o procedimento em [Fazendo upgrade Red Hat OpenShift.](#)

Nota: Apenas as liberações OpenShift Container Platform Extended Update Support (EUS) são suportadas, que são as liberações secundárias pares, por exemplo, 4.14 e 4.16.

- IBM Cloud Pak foundational services 3.19 para 3.24 inclusivo

Procedimento

1. Opcional: **Fazer upgrade de um IBM MQ Operator que está atualmente em uma versão CD anterior a 2.0.0**

Se o seu IBM MQ Operator estiver atualmente em uma versão 1.x CD , primeiro siga o procedimento em [“Migrando um 1.x CD IBM MQ Operator para a versão 2.0.x”](#) na página 124, em seguida, retorne aqui para fazer upgrade para a versão 2.4 mais recente.

2. **Faça upgrade de um IBM MQ Operator que esteja na CD versão 2.x.x para a versão 2.4 mais recente (2.4.8).**

Siga o procedimento em [“Fazendo upgrade do IBM MQ Operator usando Red Hat OpenShift”](#) na página 129.

3. Opcional: **Faça upgrade de outros componentes do IBM Cloud Pak for Integration.**

Se você for um usuário do IBM Cloud Pak for Integration , poderá ter outros componentes que deseja atualizar. Para obter as etapas para fazer upgrade de outros componentes, consulte as etapas relevantes abaixo com base em sua implementação:

- Opção 1: [Upgrade do IBM MQ Operator 2.0.x/2.1.x](#) (IBM Cloud Pak for Integration 2022.2).
- Opção 2: [Upgrade do IBM MQ Operator 2.2.x/2.3.x](#) (IBM Cloud Pak for Integration 2022.4).

4. Opcional: **Faça upgrade do IBM Cloud Pak foundational services.**

Se você for um usuário do IBM Cloud Pak for Integration , poderá desejar fazer upgrade do IBM Cloud Pak foundational services da versão 3.19.x para a versão 3.24.x. Para obter as etapas para concluir esse upgrade, consulte [Fazendo upgrade IBM Cloud Pak foundational services](#)

Tarefas relacionadas

[“\[Descontinuado\]Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou gerenciador de filas em um ambiente de isolamento físico”](#) na página 126

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 Operador ou do gerenciador de filas IBM MQ 2.x .

[“Fazendo upgrade do IBM MQ Operator usando Red Hat OpenShift”](#) na página 129

É possível fazer upgrade do IBM MQ Operator usando o console da web ou a CLI do Red Hat OpenShift .

[“Fazendo upgrade de um gerenciador de filas do IBM MQ usando Red Hat OpenShift”](#) na página 131

Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou 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 Operador ou do gerenciador de filas IBM MQ 2.x .

Antes de começar

Nota: Essas instruções são para upgrade para a versão 2.x do IBM MQ Operator em um ambiente de isolamento físico. Para fazer upgrade para IBM MQ Operator 3.0.0 e mais recente, consulte [“Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas”](#) na página 120, prestando atenção particular às etapas específicas de isolamento físico.

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 poder fazer upgrade do IBM MQ Operator ou do gerenciador de fila em um ambiente de isolamento físico, deve-se espelhar as IBM Cloud Pak for Integration imagens mais recentes.

Observe que as quatro primeiras etapas nesta tarefa são as mesmas que as etapas executadas quando [“Instalando o IBM MQ Operator 2.x em um ambiente de isolamento físico”](#) na página 110.

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 OpenShift Container Platform como um administrador de cluster.

A seguir, está um comando de exemplo para efetuar login no cluster 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`.

- 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
```

- 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`.

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

i) Verifique se um reinício de nó é necessário.

- Na 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 mosquem 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

Para concluir um upgrade do IBM Cloud Pak for Integration , pode ser necessário retornar para a documentação do IBM Cloud Pak for Integration

Caso contrário, agora você estará 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 Red Hat OpenShift” na página 129](#)
- [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando Red Hat OpenShift” na página 131](#)

Fazendo upgrade do IBM MQ Operator usando Red Hat OpenShift

É possível fazer upgrade do IBM MQ Operator usando o console da web ou a CLI do Red Hat OpenShift .

Procedimento

Para fazer upgrade do IBM MQ Operator usando Red Hat OpenShift, conclua uma das tarefas a seguir:

- [“Fazendo upgrade do IBM MQ Operator usando o console da web do Red Hat OpenShift”](#) na página 130
- [“Fazendo upgrade do IBM MQ Operator usando a CLI do Red Hat OpenShift”](#) na página 130

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

Nota: A versão mais recente do CD do IBM MQ Operator é 3.1.3 A versão mais recente do LTS do IBM MQ Operator é 2.0.23 Para obter as notas sobre a liberação mais recentes do IBM MQ Operator , consulte [“Histórico de liberação para o IBM MQ Operator”](#) na página 35

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

Procedimento

1. Revise [“Suporte de versão para o IBM MQ Operator”](#) na página 12 para determinar para qual canal do operador fazer upgrade.
2. Aplique a origem do Catálogo mais recente

Se você estiver usando a origem do catálogo específica do IBM MQ (todas as instalações de isolamento físico), em vez do `ibm-operator-catalog`, deverá aplicar a origem do catálogo para sua versão do IBM MQ .

Siga as instruções em [Incluindo origens de catálogo em um cluster](#).

Nota: Se você já tiver concluído a etapa de instalação do operador para [Imagens do espelho \(somente isolamento físico\)](#) de isolamento físico, será necessário apenas concluir a etapa que se aplica à origem do catálogo. Por exemplo:

```
oc apply -f ~/.ibm-pak/data/mirror/${OPERATOR_PACKAGE_NAME}/${OPERATOR_VERSION}/catalog-sources.yaml
```

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) 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 [“Suporte de versão para o IBM MQ Operator”](#) na página 12.

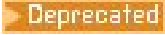
Fazendo upgrade do IBM MQ Operator usando a CLI do Red Hat OpenShift

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

Antes de começar

Nota: A versão mais recente do CD do IBM MQ Operator é 3.1.3 A versão mais recente do LTS do IBM MQ Operator é 2.0.23 Para obter as notas sobre a liberação mais recentes do IBM MQ Operator , consulte [“Histórico de liberação para o IBM MQ Operator”](#) na página 35

Efetue login em seu cluster usando **oc login**.

Antes de poder fazer upgrade do IBM MQ Operator em um ambiente de isolamento físico, deve-se espelhar as imagens IBM Cloud Pak for Integration mais recentes. Para fazer upgrade para o IBM MQ Operator 3.0 ou superior, [Migrando para o canal CD atual do IBM MQ Operator](#) inclui as etapas específicas de isolamento físico. Para fazer upgrade para versões anteriores do operador IBM MQ , consulte  [Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou gerenciador de filas em um ambiente de isolamento físico.](#)

Procedimento

1. Revise [“Suporte de versão para o IBM MQ Operator”](#) na página 12 para determinar para qual canal do operador fazer upgrade.
2. Aplique a origem do Catálogo mais recente

Se você estiver usando a origem do catálogo específica do IBM MQ (todas as instalações de isolamento físico), em vez do `ibm-operator-catalog`, deverá aplicar a origem do catálogo para sua versão do IBM MQ .

Siga as instruções em [Incluindo origens de catálogo em um cluster](#).

Nota: Se você já tiver concluído a etapa de instalação do operador para [Imagens do espelho \(somente isolamento físico\)](#) de isolamento físico, será necessário apenas concluir a etapa que se aplica à origem do catálogo. Por exemplo:

```
oc apply -f ~/.ibm-pak/data/mirror/${OPERATOR_PACKAGE_NAME}/${OPERATOR_VERSION}/catalog-sources.yaml
```

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 ou secundária, será necessário atualizar o canal selecionado em 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
```

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

Antes de começar

Como parte do processo para atualizar os gerenciadores de filas do IBM MQ , você pode ter sido enviado para este tópico da documentação do IBM Cloud Pak for Integration .

Procedimento

Para fazer upgrade do gerenciador de filas do IBM MQ usando Red Hat OpenShift, conclua uma das tarefas a seguir:

- [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando o console da web do Red Hat OpenShift” na página 132](#)
- [“Fazendo upgrade de um gerenciador de filas do IBM MQ usando a CLI do Red Hat OpenShift” na página 133](#)
- [“Fazendo upgrade de um gerenciador de filas do IBM MQ no Red Hat OpenShift usando a UI da Plataforma” na página 133](#)

Como proceder a seguir

Para concluir um upgrade do IBM Cloud Pak for Integration , pode ser necessário retornar para a documentação do IBM Cloud Pak for Integration


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

Nota: A versão mais recente do CD do gerenciador de filas do IBM MQ é 9.3.5.1-r2 A versão mais recente do LTS do gerenciador de filas do IBM MQ é 9.3.0.17-r3 Para obter as notas sobre a liberação mais recentes do IBM MQ , consulte [“Histórico de liberação para imagens do Contêiner do Gerenciador de filas para uso com o IBM MQ Operator” na página 59.](#)

- 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 usando Red Hat OpenShift” na página 129.](#)

Antes de fazer upgrade do gerenciador de filas em um ambiente de isolamento físico, deve-se espelhar as imagens mais recentes do IBM Cloud Pak for Integration . Para fazer upgrade para o IBM MQ Operator 3.0 ou superior, [Migrando para o canal CD atual do IBM MQ Operator](#) inclui as etapas específicas de isolamento físico. Para fazer upgrade para versões anteriores do operador IBM MQ , consulte  [Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operator ou gerenciador de filas em um ambiente de isolamento físico.](#)

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 [“Histórico de liberação para imagens do Contêiner do Gerenciador de filas para uso com o IBM MQ Operator” na página 59](#) para obter um mapeamento de versões do IBM MQ Operator e IBM MQ imagens de contêiner do gerenciador de filas.

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

Nota: A versão mais recente do CD do gerenciador de filas do IBM MQ é 9.3.5.1-r2 A versão mais recente do LTS do gerenciador de filas do IBM MQ é 9.3.0.17-r3 Para obter as notas sobre a liberação mais recentes do IBM MQ , consulte [“Histórico de liberação para imagens do Contêiner do Gerenciador de filas para uso com o IBM MQ Operator”](#) na página 59.

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 120.

Antes de fazer upgrade do gerenciador de filas em um ambiente de isolamento físico, deve-se espelhar as imagens mais recentes do IBM Cloud Pak for Integration . Para fazer upgrade para o IBM MQ Operator 3.0 ou superior, [Migrando para o canal CD atual do IBM MQ Operator](#) inclui as etapas específicas de isolamento físico. Para fazer upgrade para versões anteriores do operador IBM MQ , consulte  [Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operator ou gerenciador de filas em um ambiente de isolamento físico.](#)

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 12 para obter um mapeamento de canais para versões do IBM MQ Operator e versões do gerenciador de filas IBM MQ.

Utilize o seguinte comando:

```
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 a UI da Plataforma

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 UI (previously the Platform Navigator).

Antes de começar

Nota: A versão mais recente do CD do gerenciador de filas do IBM MQ é 9.3.5.1-r2 A versão mais recente do LTS do gerenciador de filas do IBM MQ é 9.3.0.17-r3 Para obter as notas sobre a liberação mais recentes do IBM MQ , consulte [“Histórico de liberação para imagens do Contêiner do Gerenciador de filas para uso com o IBM MQ Operator”](#) na página 59.

- Efetue login no IBM Cloud Pak for Integration Platform UI 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 o [“Fazendo upgrade do IBM MQ Operator e dos gerenciadores de filas”](#) na página 120.

Antes de fazer upgrade do gerenciador de filas em um ambiente de isolamento físico, deve-se espelhar as imagens mais recentes do IBM Cloud Pak for Integration . Para fazer upgrade para o IBM MQ Operator 3.0 ou superior, [Migrando para o canal CD atual do IBM MQ Operator](#) inclui as etapas específicas de isolamento físico. Para fazer upgrade para versões anteriores do operador IBM MQ , consulte  [Preparando-se para fazer upgrade para o mais recente IBM MQ 2.x Operador ou gerenciador de filas em um ambiente de isolamento físico.](#)

Procedimento

1. Na página inicial do IBM Cloud Pak for Integration Platform UI (previously the 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.

Configurando gerenciadores de filas usando o IBM MQ Operator

Exemplos de configuração; configuração de HA; conexão de fora de um cluster OpenShift ; integração com o painel CP4i ; integração com o rastreamento do Instana; construção de uma imagem com arquivos MQSC e INI customizados; inclusão de anotações e rótulos customizados.

Sobre esta tarefa

Procedimento

- [“Exemplos para configurar um gerenciador de filas” na página 134.](#)
- [“Configurando alta disponibilidade para gerenciadores de filas usando o IBM MQ Operator” na página 143.](#)
- [“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift” na página 154.](#)
- [“Integrando-se ao IBM Cloud Pak for Integration Operations Dashboard” na página 156.](#)
- [“Integrando o IBM MQ ao rastreamento do IBM Instana” na página 157.](#)
- [“Construindo uma imagem com arquivos customizados de MQSC e INI, usando a CLI do Red Hat OpenShift” na página 165.](#)
- [“Incluindo anotações e rótulos customizados para recursos do gerenciador de filas” na página 166.](#)
- [“Desativando as verificações de webhook de tempo de execução” na página 167.](#)
- [“Desativando atualizações de valor padrão para a especificação do gerenciador de filas” na página 168.](#)

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 135](#)
- [“Exemplo: Configurando um gerenciador de filas com autenticação TLS mútua” na página 138](#)

Exemplo: fornecendo arquivos MQSC e INI

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.

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 QueueManager -implemente seu gerenciador de filas com a seguinte configuração, usando a linha de comandos ou usando o console da web do Red Hat OpenShift Container Platform :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-qm
spec:
  version: 9.3.5.1-r2
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  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 MQ Advanced, altere `accept: false` para `accept: true`. Consulte [Referência de licenciamento do mq.ibm.com/v1beta1](https://www.ibm.com/support/ctodoc/p0628161) para obter detalhes sobre a licença.

Informações adicionais:

- Um gerenciador de filas pode ser configurado para usar um único Kubernetes ConfigMap ou Secret (conforme mostrado neste exemplo) ou vários ConfigMaps e Secrets.
- 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.
- Quando um pod do gerenciador de filas estiver em execução, quaisquer mudanças no Kubernetes ConfigMap não serão selecionadas porque o IBM MQ Operator não está ciente da mudança.. Se você fizer mudanças no ConfigMap, por exemplo, mudanças nos comandos MQSC ou nos arquivos INI, deverá reiniciar manualmente os gerenciadores de filas para selecionar essas mudanças. Para os gerenciadores de fila de instância única, exclua o pod para acionar a reinicialização necessária. Para implementações de HA nativa, reinicie os pods em espera primeiro excluindo-os. Quando estiverem novamente em um estado de execução, exclua o pod ativo para reiniciá-lo. Essa ordem de reinicializações assegura um tempo de inatividade mínimo para o gerenciador de filas




Criando uma PKI autoassinada usando OpenSSL

O IBM MQ permite usar o TLS mútuo para autenticação, em que ambas as extremidades de uma conexão fornecem um certificado e os detalhes no certificado são usados para estabelecer uma identidade com o gerenciador de filas. Este tópico apresenta como criar um exemplo de Public Key Infrastructure (PKI) usando a ferramenta de linha de comandos OpenSSL , criando dois certificados que podem ser usados em outros exemplos.

Antes de começar

Assegure-se de que a ferramenta de linha de comandos do OpenSSL esteja instalada

Instale o IBM MQ client e inclua `samp/bin` e `bin` no seu `PATH`. É necessário o comando `runmqicred`, que pode ser instalado como parte do IBM MQ client da seguinte forma:

-   Para Windows e Linux: Instale o cliente redistribuível IBM MQ para seu sistema operacional a partir de <https://ibm.biz/mq93redistclients>
-  Para Mac: faça download e configure o IBM MQ MacOS Toolkit: <https://developer.ibm.com/tutorials/mq-macos-dev/>

Sobre esta tarefa

Importante: Os exemplos descritos aqui não são adequados para um ambiente de produção e são exclusivamente destinados como exemplos para começar rapidamente. O gerenciamento de certificado é um assunto complexo para usuários avançados. Para produção, você deve considerar coisas como rotação, revogação, comprimento da chave, recuperação de desastre e muito mais.

Essas etapas foram testadas usando o OpenSSL 3.1.4

Procedimento

1. Crie uma chave privada para usar para sua autoridade de certificação interna

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out ca.key
```

Uma chave privada da autoridade de certificação interna é criada em um arquivo chamado *ca.key*. Este arquivo deve ser mantido seguro e secreto-ele será usado para assinar certificados para sua autoridade de certificação interna.

2. Emitir um certificado autoassinado para sua autoridade de certificação interna

```
openssl req -x509 -new -nodes -key ca.key -sha512 -days 30 -subj "/CN=example-selfsigned-ca" -out ca.crt
```

-days especifica o número de dias que o certificado de autoridade de certificação raiz será válido.

Um certificado é criado em um arquivo chamado *ca.crt*. Este certificado contém as informações públicas sobre a autoridade de certificação interna e é livremente compartilhável

3. Criar uma chave privada e um certificado para um gerenciador de fila

- a) Criar uma chave privada e solicitação de assinatura de certificado para um gerenciador de filas

```
openssl req -new -nodes -out example-qm.csr -newkey rsa:4096 -keyout example-qm.key -subj '/CN=example-qm'
```

Uma chave privada é criada em um arquivo chamado *example-qm.key* e uma solicitação de assinatura de certificado é criada em um arquivo chamado *example-qm.csr*

- b) Assinar a chave do gerenciador de filas com sua autoridade de certificação interna

```
openssl x509 -req -in example-qm.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out example-qm.crt -days 7 -sha512
```

-days especifica o número de dias que o certificado será válido.

Um certificado assinado é criado em um arquivo chamado *example-qm.crt*

- c) Crie um segredo do Kubernetes com a chave e o certificado do gerenciador de fila...

```
oc create secret generic example-qm-tls --type="kubernetes.io/tls" --from-file=tls.key=example-qm.key --from-file=tls.crt=example-qm.crt --from-file=ca.crt
```

Um Kubernetes segredo chamado *example-qm-tls* é criado. Esse segredo contém a chave privada para o gerenciador de fila, o certificado público, e o certificado de autoridade de certificação.

4. Criar uma chave privada e certificado para um aplicativo

- a) Criar uma solicitação de assinatura de chave privada e certificado para um aplicativo

```
openssl req -new -nodes -out example-app1.csr -newkey rsa:4096 -keyout example-app1.key -subj '/CN=example-app1'
```

Uma chave privada é criada em um arquivo chamado *example-app1.key* e uma solicitação de assinatura de certificado é criada em um arquivo chamado *example-app1.csr*

- b) Assinar a chave do gerenciador de filas com sua autoridade de certificação interna

```
openssl x509 -req -in example-app1.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out example-app1.crt -days 7 -sha512
```

-days especifica o número de dias que o certificado será válido.

Um certificado assinado é criado em um arquivo chamado *example-app1.crt*

- c) Crie um keystore PKCS#12 com a chave e o certificado do aplicativo.

O IBM MQ usa um banco de dados de chave, e não arquivos de chave individuais O gerenciador de filas containerizado criará o banco de dados de chave para o gerenciador de filas a partir de um Segredo, mas para aplicativos clientes, é necessário criar manualmente o banco de dados de chave

```
openssl pkcs12 -export -in "example-app1.crt" -name "example-app1" -certfile "ca.crt"
-inkey "example-app1.key" -out "example-app1.p12" -passout pass:<PASSWORD>
```

Substitua < *PASSWORD* > por uma senha de sua própria escolha

Um keystore é criado em um arquivo chamado *example-app1.p12*. A chave e o certificado do aplicativo são armazenados dentro, com um "rótulo" ou "nome fácil" de "example-app1", bem como o certificado de CA.

- d) Se você estiver usando um Apple Mac arm64 , será necessário configurar um arquivo adicional combinando o aplicativo e os certificados de autoridade de certificação.
Por exemplo:

```
cat example-app1.crt ca.crt > example-app1-chain.crt
```

Tarefas relacionadas

[“Exemplo: Configurando um gerenciador de filas com autenticação TLS mútua” na página 138](#)

Esse exemplo implementa um gerenciador de filas no OpenShift Container Platform usando o IBM MQ Operator. O TLS mútuo é usado para autenticação, para mapear de um certificado TLS para uma identidade no gerenciador de filas.

[“Exemplo: configurando HA nativa usando o IBM MQ Operator” na página 146](#)

Este exemplo implementa um gerenciador de filas usando o recurso de alta disponibilidade nativo no OpenShift Container Platform usando o IBM MQ Operator O TLS mútuo é usado para autenticação, para mapear de um certificado TLS para uma identidade no gerenciador de filas.

[“Configurando um gerenciador de filas de várias instâncias usando o IBM MQ Operator” na página 152](#)

Este exemplo implementa um gerenciador de filas de várias instâncias usando no OpenShift Container Platform usando o IBM MQ Operator O TLS mútuo é usado para autenticação, para mapear de um certificado TLS para uma identidade no gerenciador de filas.

Exemplo: Configurando um gerenciador de filas com autenticação TLS mútua

Esse exemplo implementa um gerenciador de filas no OpenShift Container Platform usando o IBM MQ Operator. O TLS mútuo é usado para autenticação, para mapear de um certificado TLS para uma identidade no gerenciador de filas.

Antes de começar

Para concluir esse exemplo, deve-se primeiro ter concluído os pré-requisitos a seguir:

- Crie um projeto/namespace de 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 OpenShift Container Platform. Ele também detalha as etapas adicionais necessárias para implementar o gerenciador de filas com o TLS ativado.

Procedimento

1. Crie um par de certificados, conforme descrito em [“Criando uma PKI autoassinada usando OpenSSL” na página 136](#)
2. Crie um mapa de configuração contendo comandos MQSC e um arquivo INI

Crie um Kubernetes ConfigMap contendo os comandos MQSC para criar uma nova fila e um canal SVRCONN e para incluir um registro de autenticação de canal que permita acesso ao canal.

Assegure-se de que você esteja no namespace criado anteriormente (consulte [Antes de iniciar](#)), em seguida, insira o YAML a seguir no console da web do OCP ou usando a linha de comandos.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-tls-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
    ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
    MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
    AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
```

O MQSC define um canal chamado *MTLS.SVRCONN* e uma fila chamada *EXAMPLE.QUEUE*. O canal é configurado para permitir acesso apenas a clientes que apresentam um certificado com um "nome comum" de *example-app1*. Esse é o nome comum usado em um dos certificados criados na [Etapa "1" na página 138](#) Conexões nesse canal com esse nome comum são mapeadas para um ID do usuário de *app1*, que está autorizado a se conectar ao gerenciador de filas e a acessar a fila de exemplo. O arquivo INI ativa uma política de segurança que significa que o ID do usuário *app1* não precisa existir em um registro do usuário externo-ele existe apenas como um nome nesta configuração.

3. Implementar o gerenciador de filas

Crie um novo gerenciador de filas usando o YAML de recurso customizado a seguir. Assegure-se de que você esteja no namespace criado antes de iniciar essa tarefa, em seguida, insira o YAML a seguir no console da web do OCP ou usando a linha de comandos. Verifique se a licença correta foi especificada e aceite a licença mudando `false` para `true`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: EXAMPLEQM
    mqsc:
      - configMap:
          name: example-tls-configmap
          items:
            - example-tls.mqsc
    ini:
      - configMap:
          name: example-tls-configmap
          items:
            - example-tls.ini
  storage:
    queueManager:
      type: ephemeral
  version: 9.3.5.1-r2
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
        items:
          - tls.key
```

```
- tls.crt
- ca.crt
```

Observe que o Segredo *example-qm-tls* foi criado na Etapa “1” na página 138 e o ConfigMap *example-tls-configmap* foi criado na Etapa “2” na página 138

4. 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 exampleqm
```

5. Teste a conexão com o gerenciador de filas

Para confirmar que o gerenciador de filas está configurado para a comunicação TLS mútua, siga as etapas em “Testando uma conexão TLS mútua com um gerenciador de filas de seu laptop” na página 140

Resultados




Parabéns, você implementou com sucesso um gerenciador de filas com o TLS ativado e que usa detalhes fornecidos no certificado TLS para autenticar com o gerenciador de fila e fornecer uma identidade

Testando uma conexão TLS mútua com um gerenciador de filas de seu laptop

Depois de ter criado um gerenciador de fila usando o IBM MQ Operator, é possível testar se ele está funcionando conectando a ele e colocando e obtendo uma mensagem. Esta tarefa orientará você sobre como se conectar usando os programas de amostra do IBM MQ, executando-os em uma máquina fora do cluster do Kubernetes, como seu laptop, por exemplo,

Antes de começar

Para concluir esse exemplo, deve-se primeiro ter concluído os pré-requisitos a seguir:

- Instale o IBM MQ client. Você precisa dos comandos **amqsputc** e **amqsgetc**, que podem ser instalados como parte do IBM MQ client da seguinte forma:
 -   Para Windows e Linux: Instale o cliente redistribuível IBM MQ para seu sistema operacional a partir de <https://ibm.biz/mq93redistclients>
 -  Para Mac: faça download e configure o IBM MQ MacOS Toolkit: <https://developer.ibm.com/tutorials/mq-macos-dev/>
- Assegure-se de ter os arquivos de chave e de certificado necessários transferidos por download para um diretório em sua máquina e que você saiba a senha do armazenamento de chaves. Por exemplo, esses arquivos são criados em “Criando uma PKI autoassinada usando OpenSSL” na página 136:
 - `example-app1.p12`
 - `example-app1-chain.crt` (somente se você estiver usando um arm64 Apple Mac)
- Implemente um gerenciador de filas configurado com TLS para o cluster OCP, por exemplo, seguindo as etapas em “Exemplo: Configurando um gerenciador de filas com autenticação TLS mútua” na página 138

Sobre esta tarefa

Esse exemplo usa os programas de amostra IBM MQ em execução em uma máquina fora do cluster Kubernetes, como seu laptop, para se conectar a um QueueManager configurado com TLS e para colocar e obter mensagens.

Procedimento

1. 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 exampleqm
```

2. Localize o nome do host do gerenciador de filas

Use o comando a seguir para localizar o nome completo do host do gerenciador de filas para o gerenciador de filas de fora do cluster OCP, usando a rota que é criada automaticamente: `exampleqm-ibm-mq-qm`:

```
oc get route exampleqm-ibm-mq-qm --template="{{.spec.host}}"
```

3. Criar uma Tabela de Definição de Canal do Cliente (CCDT) do IBM MQ

Crie um arquivo chamado `ccdt.json` com os seguintes conteúdos:

```
{
  "channel": [
    {
      "name": "MTLS.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "<hostname from previous step>",
            "port": 443
          }
        ],
        "queueManager": "EXAMPLEQM"
      },
      "transmissionSecurity": {
        "cipherSpecification": "ANY_TLS13",
        "certificateLabel": "example-app1"
      },
      "type": "clientConnection"
    }
  ]
}
```

A conexão usa a porta 443, porque essa é a porta na qual o roteador Red Hat OpenShift Container Platform está atendendo. O tráfego será encaminhado ao gerenciador de filas na porta 1414.

Se você tiver usado um nome de canal diferente, então você também precisará ajustar isso. Os exemplos de TLS mútuo usam um canal denominado `MTLS.SVRCONN`

Para obter mais detalhes, consulte [Configurando um formato JSON CCDT](#)

4. Crie um arquivo INI do cliente para configurar os detalhes da conexão

Crie um arquivo chamado `mqclient.ini`, no diretório atual. Esse arquivo será lido por `amqspu` e `amqsgetc`..

```
Channels:
  ChannelDefinitionDirectory=.
  ChannelDefinitionFile=ccdt.json
SSL:
  OutboundSNI=HOSTNAME
  SSLKeyRepository=example-app1.p12
  SSLKeyRepositoryPassword=<password you used when creating the p12 file>
```

Certifique-se de atualizar a `SSLKeyRepositorySenha` para a senha escolhida ao criar o arquivo PKCS#12. Há outras maneiras de configurar a senha do keystore, incluindo o uso de uma senha criptografada. Para obter mais informações, consulte [Fornecendo a senha de repositório de chaves para um IBM MQ MQI client em AIX, Linux, and Windows ..](#)

Observe que o Red Hat OpenShift Container Platform Router usa o SNI para solicitações de roteamento para o gerenciador de filas IBM MQ. O atributo `OutboundSNI= HOSTNAME` assegura que o cliente IBM MQ inclua as informações necessárias para o roteador trabalhar com a rota padrão configurada pelo IBM MQ Operator. 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 154.

5. Se você estiver usando um arm64 Apple Mac, será necessário configurar uma variável de ambiente adicional.

```
export MQSSLTRUSTSTORE=example-app1-chain.crt
```

Esse arquivo contém a cadeia de certificados completa, incluindo os certificados de aplicativo e de CA.

6. Colocar mensagens na fila

Execute o seguinte comando:

```
/opt/mqm/samp/bin/amqsputc EXAMPLE.QUEUE EXAMPLEQM
```

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.

7. Recuperar as mensagens da fila

Execute o seguinte comando:

```
/opt/mqm/samp/bin/amqsgetc EXAMPLE.QUEUE EXAMPLEQM
```

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.

Resultados

Parabéns, você testou com sucesso a conexão de um gerenciador de filas com TLS ativado e mostrou que é possível colocar e obter mensagens com segurança no gerenciador de filas de um cliente.

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-RJ0N-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 166.

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
```

Configurando alta disponibilidade para gerenciadores de filas usando o IBM MQ Operator

Sobre esta tarefa

Procedimento

- [“HA nativa” na página 144.](#)
- [“Exemplo: configurando HA nativa usando o IBM MQ Operator” na página 146.](#)
- [“Configurando um gerenciador de filas de várias instâncias usando o IBM MQ Operator” na página 152.](#)

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.

Para desempenho e confiabilidade, o armazenamento persistente RWO (ReadWriteOnce) é recomendado para uso com uma configuração de HA nativa. Os volumes RWO de qualquer provedor de armazenamento serão suportados se eles atenderem às condições a seguir:

- Obtido de um provedor de armazenamento de bloco.
- Formatado como ext4 ou XFS (que assegura a conformidade POSIX).
- Suporta fornecimento de volume dinâmico e "volumeBindingModo: WaitForFirstConsumer".

Os seguintes provedores são explicitamente proibidos:

- NFS
- GlusterFS
- Outros provedores não de bloco.

A figura a seguir mostra uma implementação típica com três instâncias de um gerenciador de filas implementadas em três contêineres.

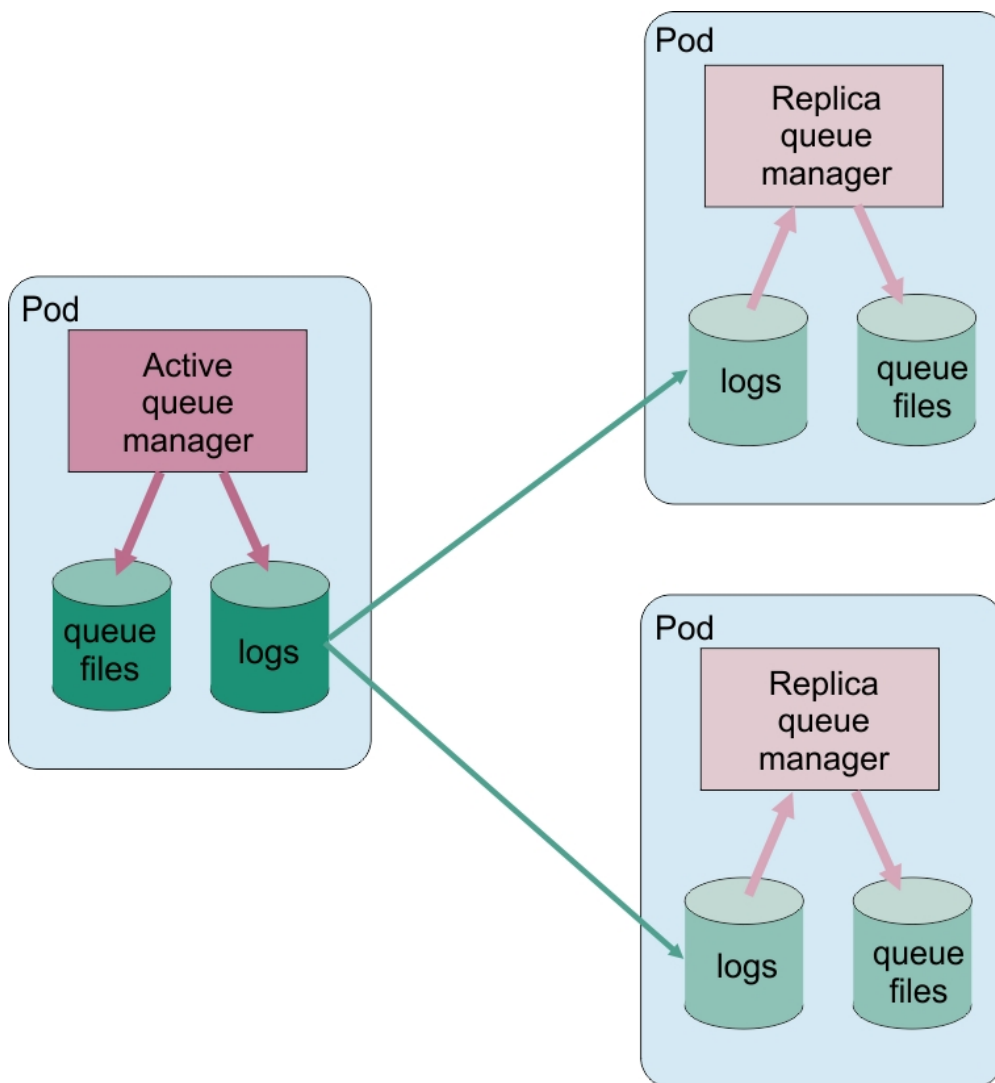


Figura 1. Exemplo de configuração de HA nativa

OpenShift **MQ Adv.** 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-AMRD-XH6P3Q
    use: Production
  queueManager:
    availability:
      type: NativeHA
    version: 9.3.5.1-r2
```

O campo `.spec.queueManager.availability.type` deve ser configurado como `NativeHA`.

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 HA nativa usando o IBM MQ Operator”](#) na página 146.

Tarefas relacionadas

[“Exemplo: configurando HA nativa usando o IBM MQ Operator”](#) na página 146

Este exemplo implementa um gerenciador de filas usando o recurso de alta disponibilidade nativo no OpenShift Container Platform usando o IBM MQ Operator. O TLS mútuo é usado para autenticação, para mapear de um certificado TLS para uma identidade no gerenciador de filas.

 [Exemplo: configurando HA nativa usando o IBM MQ Operator](#)

Este exemplo implementa um gerenciador de filas usando o recurso de alta disponibilidade nativo no OpenShift Container Platform usando o IBM MQ Operator. O TLS mútuo é usado para autenticação, para mapear de um certificado TLS para uma identidade no gerenciador de filas.

Antes de começar

Para concluir esse exemplo, deve-se primeiro ter concluído os pré-requisitos a seguir:

- Crie um projeto/namespace de 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 OpenShift Container Platform. Ele também detalha as etapas adicionais necessárias para implementar o gerenciador de filas com o TLS ativado.

Procedimento

1. Crie um par de certificados, conforme descrito em [“Criando uma PKI autoassinada usando OpenSSL”](#) na página 136
2. Crie um mapa de configuração contendo comandos MQSC e um arquivo INI
Crie um Kubernetes ConfigMap contendo os comandos MQSC para criar uma nova fila e um canal SVRCONN e para incluir um registro de autenticação de canal que permita acesso ao canal.
Assegure-se de que você esteja no namespace criado anteriormente (consulte [Antes de iniciar](#)), em seguida, insira o YAML a seguir no console da web do OCP ou usando a linha de comandos.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-nativeha-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
```

```

SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*) USERSRC(NOACCESS)
ACTION(REPLACE)
SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
MCAUSER('app1') ACTION(REPLACE)
SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
AUTHADD(BROWSE,PUT,GET,INQ)
example-tls.ini: |
Service:
  Name=AuthorizationService
  EntryPoints=14
  SecurityPolicy=UserExternal

```

O MQSC define um canal chamado *MTLS.SVRCONN* e uma fila chamada *EXAMPLE.QUEUE*. O canal é configurado para permitir acesso apenas a clientes que apresentam um certificado com um "nome comum" de *example-app1*. Esse é o nome comum usado em um dos certificados criados na Etapa “1” na página 146 Conexões nesse canal com esse nome comum são mapeadas para um ID do usuário de *app1*, que está autorizado a se conectar ao gerenciador de filas e a acessar a fila de exemplo. O arquivo INI ativa uma política de segurança que significa que o ID do usuário *app1* não precisa existir em um registro do usuário externo-ele existe apenas como um nome nesta configuração.

3. Implementar o gerenciador de filas

Crie um novo gerenciador de filas usando o YAML de recurso customizado a seguir. Assegure-se de que você esteja no namespace criado antes de iniciar essa tarefa, em seguida, insira o YAML a seguir no console da web do OCP ou usando a linha de comandos. Verifique se a licença correta foi especificada e aceite a licença mudando `false` para `true`.

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: NativeHA
    tls:
      secretName: example-qm-tls
  mqsc:
    - configMap:
        name: example-nativeha-configmap
        items:
          - example-tls.mqsc
  ini:
    - configMap:
        name: example-nativeha-configmap
        items:
          - example-tls.ini
  storage:
    queueManager:
      type: persistent-claim
  version: 9.3.5.1-r2
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt

```

Observe que o Segredo *example-qm-tls* foi criado na Etapa “1” na página 146e o ConfigMap *example-nativeha-configmap* foi criado na Etapa “2” na página 146

O tipo de disponibilidade é configurado para *NativeHA*, e o armazenamento persistente é selecionado. A classe de armazenamento padrão configurada no cluster Kubernetes será usada. 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`.

Os três pods em um gerenciador de filas de HA nativa replicam dados por meio da rede. Esse link não é criptografado por padrão, mas esse exemplo usa o certificado do gerenciador de filas para criptografar o tráfego. É possível especificar um certificado diferente para a segurança adicional. O segredo do TLS de HA nativa deve ser um segredo do TLS do Kubernetes, que tenha uma estrutura específica (por exemplo, a chave privada deve ser chamada `tls.key`).

4. 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 exampleqm
```

5. Teste a conexão com o gerenciador de filas

Para confirmar que o gerenciador de filas está configurado e disponível, siga as etapas em [“Testando uma conexão TLS mútua com um gerenciador de filas de seu laptop”](#) na página 140

6. Forçar o pod ativo a falhar

Para validar a recuperação automática do gerenciador de filas, simule uma falha de pod:

a) Visualizar os pods ativos e em espera

Execute o seguinte comando:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

Observe que, no campo **READY**, o pod ativo retorna o valor `1/1`, enquanto os pods de réplica retornam o valor `0/1`.

b) Excluir o pod ativo

Execute o comando a seguir, especificando o nome completo do pod ativo:

```
oc delete pod exampleqm-ibm-mq-<value>
```

c) Visualizar o status do pod novamente

Execute o seguinte comando:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

d) 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 EXAMPLEQM
```

O status deve mostrar que a instância ativa foi mudada, por exemplo:

```
QMNAME(EXAMPLEQM) 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)
```

e) Testar a conexão com o gerenciador de filas novamente..

Para confirmar que o gerenciador de filas se recuperou, siga as etapas em [“Testando uma conexão TLS mútua com um gerenciador de filas de seu laptop”](#) na página 140

Resultados

Parabéns, você implementou com sucesso um gerenciador de filas com alta disponibilidade nativa e autenticação TLS mútua e verificou que ele é recuperado automaticamente quando o Pod ativo falha.

IBM MQ

Para contêineres IBM MQ, é possível visualizar o status das instâncias de HA nativa executando o comando **dspmq** dentro de um dos Pods em execução.

Sobre esta tarefa

É possível usar o comando **dspmq** 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) ALTDATE() 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)
```

Tarefas relacionadas

“Exemplo: configurando HA nativa usando o IBM MQ Operator” na página 146

Este exemplo implementa um gerenciador de filas usando o recurso de alta disponibilidade nativo no OpenShift Container Platform usando o IBM MQ Operator. O TLS mútuo é usado para autenticação, para mapear de um certificado TLS para uma identidade no gerenciador de filas.

Referências relacionadas

[Comando dspmq \(exibir gerenciadores de filas\)](#)

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 135 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.

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

várias instâncias usando o IBM MQ Operator

Este exemplo implementa um gerenciador de filas de várias instâncias usando no OpenShift Container Platform usando o IBM MQ Operator. O TLS mútuo é usado para autenticação, para mapear de um certificado TLS para uma identidade no gerenciador de filas.

Antes de começar

Para concluir esse exemplo, deve-se primeiro ter concluído os pré-requisitos a seguir:

- Crie um projeto/namespace de 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 OpenShift Container Platform. Ele também detalha as etapas adicionais necessárias para implementar o gerenciador de filas com o TLS ativado.

Procedimento

1. Determinar uma classe de armazenamento adequada

O armazenamento em um cluster do Kubernetes pode ser acessado usando diversos [Modos de acesso do volume persistente](#). Um gerenciador de filas de várias instâncias cria vários volumes persistentes: um para cada gerenciador de filas e pelo menos um volume compartilhado... O volume compartilhado para um gerenciador de filas de várias instâncias deve usar uma classe de armazenamento ReadWriteMany. A classe de armazenamento padrão em um cluster Kubernetes geralmente é para uma classe de armazenamento ReadWriteOnce (armazenamento de bloco). Por exemplo, se você estiver usando Red Hat OpenShift Data Foundation, a classe de armazenamento `ocs-storagecluster-cephfs` fornece um sistema de arquivo compartilhado adequado. A escolha do sistema de arquivos é muito importante, porque nem todos os sistemas de arquivos compartilhados lidam com o bloqueio de arquivos da mesma maneira. Consulte [Planejando o suporte do sistema de arquivos em multiplataformas](#) e [Instrução de teste para sistemas de arquivos do gerenciador de filas de várias instâncias do IBM MQ](#).

2. Crie um par de certificados, conforme descrito em [“Criando uma PKI autoassinada usando OpenSSL”](#) na página 136

3. Crie um mapa de configuração contendo comandos MQSC e um arquivo INI

Crie um Kubernetes ConfigMap contendo os comandos MQSC para criar uma nova fila e um canal SVRCONN e para incluir um registro de autenticação de canal que permita acesso ao canal.

Assegure-se de que você esteja no namespace criado anteriormente (consulte [Antes de iniciar](#)), em seguida, insira o YAML a seguir no console da web do OCP ou usando a linha de comandos.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-miqm-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('Mtls.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('Mtls.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*)' USERSRC(NOACCESS)
    ACTION(REPLACE)
    SET CHLAUTH('Mtls.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
    MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
    AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
```



```
Name=AuthorizationService
EntryPoints=14
SecurityPolicy=UserExternal
```

O MQSC define um canal chamado *MTLS.SVRCONN* e uma fila chamada *EXAMPLE.QUEUE*. O canal é configurado para permitir acesso apenas a clientes que apresentam um certificado com um "nome comum" de *example-app1*. Esse é o nome comum usado em um dos certificados criados na [Etapa "2" na página 152](#) Conexões nesse canal com esse nome comum são mapeadas para um ID do usuário de *app1*, que está autorizado a se conectar ao gerenciador de filas e a acessar a fila de exemplo. O arquivo INI ativa uma política de segurança que significa que o ID do usuário *app1* não precisa existir em um registro do usuário externo-ele existe apenas como um nome nesta configuração.

4. Implementar o gerenciador de filas

Crie um novo gerenciador de filas usando o YAML de recurso customizado a seguir. Assegure-se de que você esteja no namespace criado antes de iniciar essa tarefa, em seguida, insira o YAML a seguir no console da web do OCP ou usando a linha de comandos. Verifique se a licença correta foi especificada e aceite a licença mudando `false` para `true`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: MultiInstance
  mqsc:
    - configMap:
        name: example-miqm-configmap
        items:
          - example-tls.mqsc
  ini:
    - configMap:
        name: example-miqm-configmap
        items:
          - example-tls.ini
    storage:
      defaultClass: <STORAGE CLASS>
  version: 9.3.5.1-r2
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

Altere `< STORAGE CLASS>` para a classe de armazenamento identificada na [Etapa "1" na página 152](#).

Observe que o Segredo *example-qm-tls* foi criado na [Etapa "2" na página 152](#) e o ConfigMap *example-miqm-configmap* foi criado na [Etapa "3" na página 152](#)

O tipo de disponibilidade é configurado como *MultiInstance* faz com que o armazenamento persistente seja selecionado automaticamente.

5. 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 exampleqm
```

6. Teste a conexão com o gerenciador de filas

Para confirmar que o gerenciador de filas está configurado e disponível, siga as etapas em ["Testando uma conexão TLS mútua com um gerenciador de filas de seu laptop" na página 140](#)

7. Forçar o pod ativo a falhar

Para validar a recuperação automática do gerenciador de filas, simule uma falha de pod:

a) Visualizar os pods ativos e em espera

Execute o seguinte comando:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

Observe que, no campo **READY**, o pod ativo retorna o valor 1/1, enquanto que o pod de espera retorna o valor 0/1.

b) Excluir o pod ativo

Execute o comando a seguir, especificando o nome completo do pod ativo:

```
oc delete pod exampleqm-ibm-mq-<value>
```

c) Visualizar o status do pod novamente

Execute o seguinte comando:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

d) Visualizar o status do gerenciador de filas

Execute o seguinte comando, especificando o nome completo do outro pod:

```
oc exec -t Pod -- dspmq -x
```

O status deve mostrar que a instância ativa foi mudada, por exemplo:

```
QMNAME(EXAMPLEQM)                                STATUS(Running as standby)
  INSTANCE(exampleqm-ibm-mq-1) MODE(Active)
  INSTANCE(exampleqm-ibm-mq-0) MODE(Standby)
```

e) Testar a conexão com o gerenciador de filas novamente..

Para confirmar que o gerenciador de filas se recuperou, siga as etapas em [“Testando uma conexão TLS mútua com um gerenciador de filas de seu laptop”](#) na página 140

Resultados

Parabéns, você implementou com sucesso um gerenciador de filas de várias instâncias com a autenticação TLS mútua e verificou que ele se recupera automaticamente quando o Pod ativo falha

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

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](#).

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.

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 de IBM MQ 9.3.0)

A partir do IBM MQ 9.3.0, o cliente IBM MQ gerenciado .NET foi atualizado para configurar SERVERNAME para o respectivo nome do host se a propriedade **OutboundSNI** estiver configurada como HOSTNAME, o que permite que um cliente IBM MQ gerenciado .NET se conecte a um gerenciador de filas usando rotas do Red Hat OpenShift.

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 CERTLABEL, 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 CERTLABEL 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 canal do MQ não terminem em uma letra minúscula devido à maneira como o IBM MQ mapeia os nomes de canais para cabeçalhos 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 171
Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster do Red Hat OpenShift Container Platform




Integrando-se ao IBM Cloud Pak for Integration Operations Dashboard



A capacidade de rastrear transações por meio do IBM Cloud Pak for Integration é fornecida pelo Painel de operações.

Antes de começar



Atenção:

   No IBM MQ Operator 2.0.0, o Painel de Operações foi descontinuado e não receberá mais atualizações. Nenhum novo uso do Painel de Operações deve ser criado

  No IBM MQ Operator 2.4.0, o Painel de Operações é removido. Observe que o Painel de Operações ainda pode ser usado para gerenciadores de filas existentes

que são mais antigos que 9.3.3.0-r1 se em um IBM MQ Operator que suporta essa imagem de contêiner do gerenciador de filas. Para obter o suporte de versão para o IBM MQ Operator, consulte [“Versões disponíveis do IBM MQ”](#) na página 12

O suporte para o Painel de Operações termina em 30 de junho de 2024.. Para obter mais informações, consulte [Retirada de software e / ou descontinuação de suporte](#)

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 rastreamento 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.

Procedimento

1. Implemente um gerenciador de filas com o rastreamento ativado

Por padrão, o recurso de rastreamento é desativado.

Ao realizar uma implementação usando o IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator), é possível ativar o rastreamento durante ela configurando **Ativar rastreamento** como **Ativo** e **Namespace do rastreamento** como o namespace no qual o Operations Dashboard está instalado. Para obter mais informações sobre como implementar um gerenciador de filas, consulte [Implementar um gerenciador de fila com o IBM Cloud Pak for Integration Platform UI](#)

Se estiver implementando usando a [Red Hat OpenShift CLI](#) ou [Red Hat OpenShift console da web](#), será possível ativar o rastreamento com o fragmento YAML a seguir:

```
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.

Integrando o IBM MQ ao rastreamento do IBM

Instana

IBM Instana pode ser usado para rastrear transações em IBM Cloud Pak for Integration.

Antes de começar

Este documento cobre o rastreamento do IBM Instana, que é o processo de rastreamento de mensagens por meio de um sistema. Ele não cobre o monitoramento do IBM Instana, no qual os detalhes são recuperados sobre o estado de um gerenciador de fila do IBM MQ. Para obter informações sobre monitoramento do IBM MQ por IBM Instana consulte [Monitoramento IBM MQ](#). Para obter instruções detalhadas sobre monitoramento autenticado, consulte [“Configurando o monitoramento autenticado do IBM Instana com TLS”](#) na página 159.

Nota:

- Esse recurso só pode ser usado com o IBM MQ Operator versão 2.2.0 e mais recente. Esse recurso é suportado somente em Operandos do IBM MQ versão 9.3.1.0-r2 ou posterior.
- É possível executar o rastreamento do IBM Instana nas versões anteriores do Operador e do gerenciador de filas do IBM MQ, mas não nativamente. Consulte [Configurando o IBM MQ Rastreamento](#) na documentação do IBM Instana.

Antes de poder executar o rastreamento do IBM Instana com o Operador do IBM MQ, deve-se implementar um agente de IBM Instana backend e IBM Instana. Por padrão, um gerenciador de filas do IBM MQ se comunica com um agente IBM Instana implementado no mesmo nó que o pod do gerenciador de filas.

Sobre esta tarefa

Ativar a integração com o IBM Instana faz com que uma saída de API IBM MQ seja instalada em seu gerenciador de filas. A saída de API envia dados de rastreamento para agentes IBM Instana sobre mensagens que estão fluindo através do gerenciador de filas.

A saída de API inclui cabeçalhos RFH2 em cada mensagem. Esses cabeçalhos contêm informações de rastreamento..

Os agentes IBM Instana são responsáveis por enviar os dados de rastreamento para o backend IBM Instana.

Para obter informações sobre como implementar um IBM Instana backend e IBM Instana agentes, consulte [Ativando o IBM Instana Monitoring na CP4I Platform UI](#) na documentação do IBM Instana.

Procedimento

Implementação padrão

- Implementar um gerenciador de filas com o rastreamento do IBM Instana ativado.

Por padrão, o rastreamento do IBM Instana é desativado.

Se você estiver usando o IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) ou o console da web OpenShift:

1. Clique em **Telemetria > Rastreamento > Instana**
2. Configure a alternância **Ativar rastreamento da Instana** para `true`

Se você estiver implementando por meio do YAML, use o fragmento a seguir:

```
spec:
  telemetry:
    tracing:
      instana:
        enabled: true
```

Implementação Avançada

- Comunique-se com o agente do IBM Instana sobre https

Por padrão, a saída IBM Instana para IBM MQ comunica-se com o agente IBM Instana sobre http. O endereço do host do agente é configurado para o endereço IP do nó no qual o gerenciador de filas está em execução. Isso corresponde à configuração descrita em [Ativando IBM Instana monitoramento](#) na documentação do IBM Instana, em que os agentes IBM Instana são implementados pelo IBM Instana Agent Operator como um daemonset.

Atualmente, a comunicação entre a saída IBM Instana para IBM MQ e o agente IBM Instana suporta protocolos http ou https. Para usar https, o agente do IBM Instana deve primeiro ser configurado para usar a criptografia TLS. Consulte [Configurando a criptografia TLS para o terminal do agente](#) na documentação do IBM Instana. O protocolo pode, então, ser configurado como https da seguinte forma:

Se você estiver usando o console da web do OpenShift:

1. Clique em **Telemetria > Instana**

2. Expanda a lista suspensa **Configuração avançada** .
3. Configure o **protocolo de comunicação do agente do Instana** como https

Se você estiver implementando por meio do YAML, use o fragmento a seguir:

```
spec:
  telemetry:
    instana:
      enabled: true
      protocol: https
```

- Configure o **agentHost**

Se os agentes do IBM Instana não tiverem sido implementados como um daemonset no cluster OpenShift no qual o gerenciador de filas está em execução, deve-se configurar o valor **agentHost** para o nome do host ou endereço IP no qual o agente IBM Instana está em execução. O valor **agentHost** não deve incluir um protocolo ou porta.

Se você estiver usando o console da web do OpenShift :

1. Clique em **Telemetria > Instana**
2. Expanda a lista suspensa **Configuração avançada** .
3. Digite seu nome do host na caixa de texto **Host do agente do Instana**

Se você estiver implementando por meio do YAML, use o fragmento a seguir:

```
spec:
  telemetry:
    instana:
      enabled: true
      agentHost: 9.9.9.9
```

Como proceder a seguir

Consulte também “[Implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform](#)” na página 116.

Configurando o monitoramento autenticado do IBM Instana com TLS

Para poder monitorar um gerenciador de fila por meio de um agente IBM Instana , deve-se configurar o agente e o gerenciador de fila.

Antes de começar

A seção "[Configuração](#)" de "[Monitoramento IBM MQ](#)" na documentação do IBM Instana fornece informações gerais sobre a configuração de monitoramento do IBM Instana . No entanto, ele não inclui detalhes sobre a configuração do gerenciador de fila

Antes de poder executar o rastreamento do IBM Instana com o Operador do IBM MQ , deve-se implementar um agente de IBM Instana backend e IBM Instana . Para fazer isso, consulte [Ativando o monitoramento do IBM Instana na IU do CP4I Platform](#) na documentação do IBM Instana .

Procedimento

1. [Gerar certificados](#)
2. [Configure os IBM Instana agentes..](#)
3. [Configure o Gerenciador de Filas](#)
4. [Verifique e depure.](#)

Tarefas relacionadas

[“Integrando o IBM MQ ao rastreamento do IBM Instana”](#) na página 157

IBM Instana pode ser usado para rastrear transações em IBM Cloud Pak for Integration.

OpenShift Operator 2.2.0 CP4I Gerar um certificado e uma chave para o agente do IBM Instana e o gerenciador de filas

Para a comunicação TLS entre o agente do IBM Instana e o gerenciador de fila, ambos devem ter um certificado e a chave privada correspondente

Antes de começar

Esta é a primeira de quatro tarefas para [configurar o monitoramento autenticado do IBM Instana com TLS](#)

Nota: Os valores usados na geração desses certificados são para propósitos de demonstração. Ao implementar em um ambiente de produção, certifique-se de que o assunto e a expiração do certificado sejam apropriados

Procedimento

IBM MQ Gerenciador de filas

Para se comunicar com o agente do IBM Instana por meio de TLS, o gerenciador de filas deve ter um certificado e a chave privada correspondente. Se você já tiver isso, ignore esta seção.

1. Gerar um certificado e uma chave privada para o gerenciador de fila.

Execute o seguinte comando:

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

IBM Instana agente

Para que o agente execute a comunicação TLS com o gerenciador de fila do IBM MQ, o agente deve ter um certificado e uma chave privada correspondente. Se você já tiver uma chave privada e um certificado em um keystore JKS que gostaria de usar, ignore esta seção.

2. Gere um certificado e uma chave privada para o agente do IBM Instana

Execute o seguinte comando:

```
openssl req \
  -newkey rsa:2048 -nodes -keyout application.key \
  -subj "/CN=instana-agent/OU=app team1" \
  -x509 -days 3650 -out application.crt
```

3. Armazene o certificado e a chave privada em um keystore PKCS12.

Execute o comando a seguir, substituindo *your_password* pela senha que você deseja usar para proteger o keystore. Execute esta substituição em todas as etapas subsequentes

```
openssl pkcs12 -export -out application.p12 -inkey application.key -in application.crt
-passout pass:your_password
```

4. Converta o keystore PKCS12 em um keystore JKS.

Execute o seguinte comando:

```
keytool -importkeystore \
  -srckeystore application.p12 \
  -srcstoretype pkcs12 \
  -destkeystore application.jks \
  -deststoretype JKS \
  -srcstorepass your_password \
  -deststorepass your_password \
  -noprompt
```

5. Rotular o certificado.

Execute o seguinte comando:

```
keytool -changealias -alias "1" -destalias "instana" -keypass your_password -keystore application.jks -storepass your_password -noprompt
```

6. Importe o certificado do gerenciador de filas para o keystore.

Execute o seguinte comando:

```
keytool -importcert -file server.crt -keystore application.jks -storepass your_password -alias myca -noprompt
```

Como proceder a seguir

Agora você está pronto para [configurar os agentes para IBM Instana monitoramento](#).

OpenShift Operator 2.2.0 CP4I **Monitoramento do Instana: Configurando agentes.**

Monte o keystore para os agentes IBM Instana , em seguida, configure o monitoramento para um gerenciador de filas específico.

Antes de começar

Esta tarefa supõe que você [tenha gerado um certificado e uma chave para os agentes IBM Instana e o gerenciador de fila](#)

Procedimento

Montando o keystore para os agentes IBM Instana

1. Crie um segredo de seu keystore JKS no namespace do agente IBM Instana .

Execute o comando a seguir, substituindo *keystore_secret_name* pelo nome que deseja usar. Execute esta substituição em todas as etapas subsequentes

```
oc create secret generic keystore_secret_name --from-file=./application.jks -n instana-agent
```

2. No namespace instana-agent, use o comando `oc edit daemonset instana-agent` para editar o `daemonset instana-agent` para incluir o `volumeMount` e o `volume` adicionais a seguir:

```
volumeMounts:
- name: mq-key-jks-name
  subPath: application.jks
  mountPath: /opt/instana/agent/etc/application.jks
volumes:
- name: mq-key-jks-name
  secret:
    secretName: keystore_secret_name
```

Configurando o monitoramento para um gerenciador de filas específico

3. No namespace instana-agent, use o comando `oc edit configmap instana-agent` para editar o `configmap instana-agent`.

4. Inclua a seção a seguir em `configuration.yaml`: | Se você já tiver definido essa seção, apenas inclua o novo gerenciador de filas na lista

```
com.ibmqa.plugin.ibmmq:
  enabled: true
  poll_rate: 60
  queueManagers:
    QUEUE_MANAGER_NAME:
      channel: 'INSTANA.A.SVRCONN'
      keystorePassword: 'your_password'
      keystore: '/opt/instana/agent/etc/application.jks'
      cipherSuite: 'TLS_RSA_WITH_AES_256_CBC_SHA256'
```

em que

- *your_password* é a senha para seu keystore JKS

- `QUEUE_MANAGER_NAME` é o nome do gerenciador de filas do IBM MQ subjacente a ser implementado, em vez do nome do Operando do Gerenciador de Filas

Nota: Se `QUEUE_MANAGER_NAME` não estiver configurado para o nome do gerenciador de filas subjacente e, em vez disso, estiver configurado para o Operando, o monitoramento não funcionará. O nome subjacente é definido em `spec.queuemanager.name` para o Operando do Gerenciador de Filas

5. Exclua os pods `instana-agent` no namespace `instana-agent`. Isso os faz reiniciar e iniciar o monitoramento com as novas configurações.

Como proceder a seguir

Agora você está pronto para [configurar o gerenciador de filas para IBM Instana monitoramento](#).

OpenShift Operator 2.2.0 CP4I Monitoramento do Instana: Configurando o gerenciador de filas

Configure um gerenciador de filas que use TLS para se comunicar com o agente IBM Instana. A autenticação para essa conexão é feita usando um [SSLPEERMAP](#)

Antes de começar

Esta tarefa assume que você [configurou os agentes para IBM Instana monitoramento](#)

Procedimento

1. Configure o gerenciador de filas através de MQSC e INI.

O MQSC é usado para configurar um novo canal ativado pelo TLS e, em seguida, configurar esse canal para autenticar o agente IBM Instana de conexão se ele tiver um certificado com os campos necessários. Nesse caso, mapeamos qualquer cliente de conexão com um certificado contendo os campos `CN=instana-agent,OU=app_team1` para usuário `app1`. O MQSC então concede permissão para o usuário `app1` para executar as operações necessárias para o monitoramento do IBM Instana.

O arquivo INI é usado para conceder permissões para nosso usuário externo `app1`

O configmap a seguir contém as configurações de MQSC e INI necessárias. Implemente-o no namespace do gerenciador de filas.

```
apiVersion: v1
data:
  channel.mqsc: |-
    DEFINE CHANNEL('INSTANA.A.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    ALTER QMGR CONNAUTH(' ')
    REFRESH SECURITY
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
  action.replace: |-
    ACTION(REPLACE)
  set.chlauth.wildcard: |-
    SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
  set.chlauth.instana: |-
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=instana-agent,OU=app
  team1') USERSRC(MAP) MCAUSER('app1')
  set.authrec.principal.app1: |-
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(ALL)
  set.authrec.profile.system.admin.command.queue: |-
    SET AUTHREC PROFILE('SYSTEM.ADMIN.COMMAND.QUEUE') PRINCIPAL('app1') OBJTYPE(Queue)
  auth.add.put.inq.dsp.chg: |-
    AUTHADD(PUT,INQ,DSP,CHG)
  set.authrec.profile.system.wildcard: |-
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(Topic) AUTHADD(DSP)
  set.authrec.profile.system.wildcard.dsp: |-
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(Topic) AUTHADD(DSP)
  set.authrec.profile.system.wildcard.queue: |-
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(Queue) AUTHADD(DSP,CHG,GET)
  set.authrec.profile.system.wildcard.listener: |-
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(Listener) AUTHADD(DSP)
  set.authrec.profile.amq.wildcard: |-
    SET AUTHREC PROFILE('AMQ.*') PRINCIPAL('app1') OBJTYPE(Queue) AUTHADD(DSP,CHG)
  refresh.security.type.connauth: |-
    REFRESH SECURITY TYPE(CONNAUTH)
  auth.ini: |-
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
kind: ConfigMap
metadata:
  namespace: your-queue-manager-namespace
  name: qmgr-monitoring-config
```

em que *your-queue-manager-namespace* é o namespace no qual seu gerenciador de filas será implementado.

Nota: Se você estiver monitorando filas definidas pelo usuário, deverá incluir linhas adicionais no configmap MQSC, concedendo permissões DSP, CHG e GET para essas filas. Por exemplo:

```
SET AUTHREC PROFILE('MYQUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG, GET).
```

Este exemplo usa um configmap para os dados MQSC e INI, mas é possível usar um segredo se quaisquer adições feitas forem confidenciais. Para obter informações gerais sobre a implementação com MQSC e INI, consulte [“Exemplo: fornecendo arquivos MQSC e INI”](#) na página 135.

2. Para uma conexão TLS ser feita, o gerenciador de fila deve confiar no certificado do agente IBM Instana . Para fazer isso, crie um segredo contendo apenas o certificado do agente IBM Instana :

```
oc create secret generic instana-certificate-secret --from-file=./application.crt -n your-queue-manager-namespace
```

3. O gerenciador de filas deve apresentar seu próprio certificado para o handshake TLS e requer acesso à chave privada associada Implemente um segredo contendo a chave e o certificado que você criou anteriormente ou já possui:

```
oc create secret tls qm-tls-secret --cert server.crt --key server.key -n your-queue-manager-namespace
```

Com o configmap e o segredo criados, você está pronto para criar o próprio gerenciador de filas

4. Assegure-se de que seu gerenciador de filas YAML não configure a variável de ambiente **MQSNOAUT** no contêiner do gerenciador de filas.

Caso contrário, após ser ativado, o mecanismo de autenticação não funcionará. A remoção da variável após a implementação não faz com que o mecanismo seja reativado e o gerenciador de filas precisa ser recriado.

5. Inclua as seguintes seções em sua definição de gerenciador de filas, em que *MYQM* é o nome de seu gerenciador de filas:

```
spec:
  queueManager:
    name: MYQM #(a)
    ini: #(b)
    - configMap:
      items:
        - auth.ini
      name: qmgr-monitoring-config
    mqsc: #(c)
    - configMap:
      items:
        - channel.mqsc
      name: qmgr-monitoring-config
  pki:
    keys: #(d)
    - name: default
      secret:
        items:
          - tls.key
          - tls.crt
        secretName: qm-tls-secret
    trust: #(e)
    - name: app
      secret:
        items:
          - application.crt
        secretName: instana-certificate-secret
```

As seções sinalizadas da especificação são descritas a seguir:

- a. Assegure-se de ter fornecido ao seu gerenciador de filas subjacente um nome exclusivo. Se o gerenciador de filas subjacente não tiver um nome exclusivo, o monitoramento poderá não funcionar conforme desejado. Esse nome deve corresponder ao nome no configmap do agente IBM Instana que foi editado anteriormente.

- b. As informações INI gravadas no configmap são incluídas no gerenciador de filas.
 - c. As informações MQSC que foram gravadas no configmap são incluídas no gerenciador de fila.
 - d. O certificado do gerenciador de filas e a chave privada são adicionados ao keystore do gerenciador de filas
 - e. O certificado do agente do IBM Instana é incluído no armazenamento confiável do gerenciador de filas
6. Opcional: Ative o Rastreamento do IBM Instana em seu gerenciador de filas monitorado
- Para fazer isso, consulte [“Integrando o IBM MQ ao rastreamento do IBM Instana” na página 157.](#)
7. Implementar o gerenciador de filas.

Como proceder a seguir

Agora você está pronto para [verificar e depurar o IBM Instana monitoramento.](#)

Monitoramento do Instana: Verificando e Depurando

Para poder monitorar um gerenciador de fila por meio de um agente IBM Instana , deve-se configurar o agente e o gerenciador de fila.

Antes de começar

Essa tarefa supõe que você [configurou o gerenciador de fila para IBM Instana monitoramento.](#)

Procedimento

Verificando

1. Para verificar se você foi bem-sucedido em sua implementação, visualize seu gerenciador de filas no painel do IBM Instana
- O gerenciador de filas deve estar visível na seção de serviços da página do aplicativo e também na visualização Infraestrutura.

Depurando

Nota: Essas etapas de depuração assumem uma implementação do Openshift do agente IBM Instana em execução como um daemonset.

Se não for possível ver seu gerenciador de filas no painel do IBM Instana , você poderá ter configurado incorretamente seu gerenciador de filas.. Use as etapas a seguir para investigar.

2. Identifique o nó no qual o seu pod do gerenciador de filas ativo está em execução

Execute o comando a seguir no namespace do gerenciador de filas:

```
oc get pods -o wide -n your-queue-manager-namespace
```

3. Para determinar qual pod do agente do IBM Instana está em execução no mesmo nó que seu gerenciador de filas, execute o mesmo comando no namespace instana-agent:

```
oc get pods -o wide -n instana-agent-namespace
```

4. Para ajudar a entender quaisquer problemas do lado do agente IBM Instana , obtenha os logs do pod do agente IBM Instana e procure entradas relacionadas a 'mq' ou ao nome de seu gerenciador de filas.

Execute o seguinte comando:

```
oc logs instana-agent-pod -c instana-agent -n instana-agent
```

5. Verifique os logs do gerenciador de filas..

Se o agente tiver feito uma tentativa de se conectar ao gerenciador de fila, os logs do gerenciador de filas deverão indicar por que a conexão não foi bem-sucedida. Execute o seguinte comando:

```
oc logs your-queue-manager-name -n your-queue-manager-namespace
```

Resultados

Você concluiu todas as quatro tarefas para [configurar o monitoramento autenticado IBM Instana com o TLS](#)

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 tiver um Red Hat OpenShift Secret para o IBM Entitled Registry em seu projeto Red Hat OpenShift, siga as etapas para [Criar o segredo da chave de autorização](#).

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:9.3.5.1-r2
      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:9.3.5.1-r2"
      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 35 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 116, 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/namespave 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 116

Este exemplo implementa um gerenciador de filas de "iniciação rápida", que usa armazenamento temporário (não persistente) e desativa a segurança do IBM MQ. As mensagens não são persistidas entre as reinicializações do gerenciador de filas. É possível ajustar a configuração para mudar muitas configurações do gerenciador de filas.

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.

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"
```

Desativando atualizações de valor padrão para a especificação do gerenciador de filas

O IBM MQ Operator atualiza quaisquer valores não especificados na especificação do gerenciador de filas com os seus valores padrão. É possível desativar esse comportamento se você deseja evitar qualquer modificação na especificação do gerenciador de filas. Os campos de status do gerenciador de filas ainda estão atualizados.

Procedimento

- Desativar atualizações de valor padrão do gerenciador de filas.

Inclua a anotação a seguir em metadata. Por exemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.mq/write-defaults-spec" : "false"
```

Nota: Esse recurso só pode ser usado com o Operador do MQ 2.1.0 e versões mais recentes. Em IBM MQ Operator 2.1.0, os exemplos de iniciação rápida têm essa anotação aplicada por padrão.

V 9.3.4

Executando o contêiner IBM MQ com um sistema de arquivos raiz somente leitura

No IBM MQ Operator 3.0.0 e no IBM MQ contêiner 9.3.4.0, é possível configurar o contêiner IBM MQ para executar com um sistema de arquivos raiz somente leitura. Isso impede que invasores copiem e executem código malicioso no contêiner.

Sobre esta tarefa

A ativação do sistema de arquivos raiz somente leitura torna os arquivos de contêiner imutáveis. Ou seja, no sistema de arquivos de contêiner, os arquivos podem ser visualizados, mas não modificados e nenhum novo arquivo pode ser criado. Os arquivos só podem ser modificados ou criados em um sistema de arquivos montado.

Quando um sistema de arquivos raiz somente leitura é ativado, dois volumes efêmeros `Rascunho` e `Tmp` são criados e montados nos diretórios `/run` e `/tmp` respectivamente no contêiner.

- O volume `Rascunho` contém os arquivos, keystores e outros arquivos usados para configurar o gerenciador de filas.
- O volume `Tmp` contém arquivos de diagnóstico, por exemplo, arquivos `RAS` do gerenciador de filas.

Como esses volumes são efêmeros, os arquivos nesses volumes são perdidos na reinicialização do pod.

O tipo do volume criado para os dados do gerenciador de filas depende do tipo de armazenamento. Por padrão, um volume persistente é montado. Ou, se o tipo de armazenamento for efêmero, um volume efêmero será montado. Se o tamanho dos dados no volume exceder o valor especificado para a propriedade `sizeLimit`, o Kubernetes poderá ejetar o contêiner e criar um novo. Antes do IBM MQ Operator 3.0.0, o limite de tamanho não era aplicado ao usar o armazenamento efêmero para dados do gerenciador de filas.

Um sistema de arquivos raiz somente leitura não é ativado por padrão. Para ativá-la, conclua as seguintes etapas:

Procedimento

- Use a API spec `.securityContext` para ativar o sistema de arquivos raiz somente leitura.

Para seu gerenciador de filas, configure a propriedade **readOnlyRootFilesystem** em [“.spec.securityContext”](#) na página 196 como true..

O IBM MQ Operator cria dois volumes temporários, Scratch e Tmp.

2. Opcional: Configure ou altere o tipo de armazenamento de dados do gerenciador de filas

Por padrão, uma solicitação de volume persistente é montada em /mnt/mqm. Ou, se a propriedade **type** for configurada como efêmero em [“.spec.queueManager.storage.queueManager”](#) na página 194, um volume efêmero será criado e montado.

3. Para cada volume efêmero, considere cuidadosamente quanto os dados podem crescer. Configure o valor da propriedade **sizeLimit** adequadamente, incluindo unidades SI.

- Para o volume efêmero Rascunho , configure a propriedade **sizeLimit** em [“.spec.queueManager.storage.scratch”](#) na página 195 O valor-padrão é "100M".
- Para o volume efêmero Tmp , configure a propriedade **sizeLimit** em [“.spec.queueManager.storage.tmp”](#) na página 196.. O valor padrão é "2Gi". "
- Se o **type** do volume do gerenciador de filas estiver configurado como ephemeral, configure a propriedade **sizeLimit** em [“.spec.queueManager.storage.queueManager”](#) na página 194 O valor padrão é "2Gi". "

Configurando o IBM MQ Console com um registro básico usando o IBM MQ Operator

Para efetuar login no IBM MQ Console, é possível fornecer sua própria configuração ao gerenciador de filas.

Antes de começar

Se você estiver implementando um gerenciador de filas com uma licença do IBM MQ Advanced for Developers , haverá uma configuração simples integrada... Consulte [“\[MQ 9.3.4 Dez 2023\]YAML do gerenciador de filas de exemplo que descreve como especificar senhas para admin e app usuários”](#) na página 25.

Se você estiver implementando um gerenciador de filas de licença do IBM Cloud Pak for Integration , será possível ativar a integração com o IBM Cloud Pak for Integration Keycloak para efetuar login no IBM MQ Console usando Conexão única. Consulte [“Conectando-se ao IBM MQ Console implementado em um cluster do Red Hat OpenShift”](#) na página 171.

Procedimento

1. Crie uma senha e criptografe-a usando **securityUtility**.

Um ConfigMap é usado para armazenar as credenciais usadas para acessar seu gerenciador de filas. Para segurança melhorada, você codifica essas credenciais com o comando **securityUtility**.

Como alternativa, é possível usar um Segredo, que protege credenciais na camada Kubernetes . No entanto, ferramentas de monitoramento ou resolução de problemas podem expor o arquivo subjacente de forma insegura.

2. Opcional: **Efetue login na Red Hat OpenShift interface da linha de comandos (CLI)**.

Se estiver usando a CLI do OpenShift , efetue login usando o `oc login`

Como alternativa, é possível usar o console do OpenShift

3. **Crie um ConfigMap com sua configuração.**

Para obter ajuda com a criação da configuração XML, consulte [IBM MQ Console e REST API segurança](#)

O exemplo a seguir cria um usuário dentro do grupo MQWebAdminGroup Membros do MQWebAdminGroup são designados à função MQWebAdmin . Neste exemplo:

- Você **deve** substituir *USERNAME* e *PASSWORD* por seus próprios valores. Observe que *USERNAME* é usado duas vezes no exemplo..

Você **deve** especificar o *NAMESPACE* como aquele no qual seu IBM MQ Operator está implementado e onde seu gerenciador de filas estará, ou já está, implementado.

a) Use o console do OpenShift ou a linha de comandos para criar o seguinte ConfigMap:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: mqwebuserconfigmap
  namespace: NAMESPACE
data:
  mqwebuser.xml: |
    <?xml version="1.0" encoding="UTF-8"?>
    <server>
      <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>basicAuthenticationMQ-1.0</feature>
      </featureManager>
      <enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
          <security-role name="MQWebAdmin">
            <group name="MQWebAdminGroup" realm="defaultRealm"/>
          </security-role>
        </application-bnd>
      </enterpriseApplication>
      <basicRegistry id="basic" realm="defaultRealm">
        <user name="USERNAME" password="PASSWORD"/>
        <group name="MQWebAdminGroup">
          <member name="USERNAME"/>
        </group>
      </basicRegistry>
      <sslDefault sslRef="mqDefaultSSLConfig"/>
    </server>
```

b) Opcional: Se estiver usando a linha de comandos, aplique o ConfigMap:

```
oc apply -f mqwebuserconfigmap.yaml
```

Para as etapas restantes, escolha uma das opções a seguir:

- Implementar um novo gerenciador de filas com a configuração para acessar o IBM MQ Console.
 - Aplique a configuração que fornece ao IBM MQ Console o acesso a um gerenciador de filas existente
4. Opcional: **Implementar um novo gerenciador de filas com a configuração para acessar o IBM MQ Console.**

a) Crie o seu gerenciador de filas.

Configure os provedores de autenticação e autorização como manual e forneça o ConfigMap mqwebuserconfigmap recém-criado por meio de uma das opções a seguir:

- Opção 1: Através do YAML do gerenciador de filas

Inclua o código a seguir na seção web do gerenciador de filas YAML:

```
...
web:
  enabled: true
  console:
    authentication:
      provider: manual
    authorization:
      provider: manual
  manualConfig:
    configMap:
      name: mqwebuserconfigmap
```

- Opção 2: Por meio da visualização Formulário do console do OpenShift :
 - i) No OpenShift console, selecione **Operadores > Operadores instalados**.
 - ii) Selecione a sua implementação do IBM MQ Operator

- iii) Selecione **Queue Manager** e clique em **Criar QueueManager**.
- iv) Selecione as opções relevantes para o seu gerenciador de fila.
- v) Selecione **Web** e configure **Ativar servidor da web** como `true`.
- vi) Abra a caixa de listagem **Configuração Avançada**
- vii) Na caixa de listagem **Console**, configure o **provedor** para **Autenticação** e **Autorização** como `manual`.
- viii) Abra a caixa de lista **Configuração**.
- ix) Abra a caixa de listagem **ConfigMap** e selecione o ConfigMap `mqwebuserconfigmap` que foi criado na etapa “3” na página 169
- x) Clique em **Criar**.

Agora é possível acessar o IBM MQ Console de seu novo gerenciador de filas por meio das credenciais especificadas no ConfigMap criado na etapa “3” na página 169.

5. Opcional: **Aplicar configuração que ativa o IBM MQ Console para um gerenciador de filas existente.**

Edite o YAML do gerenciador de filas para o qual você está ativando o IBM MQ Console:

- a. No console OpenShift, selecione **Operadores > Operadores instalados**.
- b. Selecione a sua implementação do IBM MQ Operator
- c. Selecione **Gerenciador de filas** e selecione o nome do gerenciador de filas.
- d. Selecione **YAML**.
- e. Substitua a seção `web` existente do YAML do gerenciador de filas pelo código a seguir:

```
...
web:
  enabled: true
  console:
    authentication:
      provider: manual
    authorization:
      provider: manual
  manualConfig:
    configMap:
      name: mqwebuserconfigmap
```

- f. Clique em **Salvar**.

Agora é possível acessar o IBM MQ Console de seu gerenciador de filas existente por meio das credenciais especificadas no ConfigMap criado na etapa “3” na página 169.

Operando o IBM MQ usando o IBM MQ Operator

Procedimento

- [“Conectando-se ao IBM MQ Console implementado em um cluster do Red Hat OpenShift” na página 171.](#)
- [“Monitorando ao usar o IBM MQ Operator” na página 173.](#)
- [“Fazer backup e restaurar a configuração do gerenciador de filas usando a CLI do Red Hat OpenShift” na página 178.](#)

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 do 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 UI (previously the 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 estiver usando uma licença do IBM Cloud Pak for Integration :

- Para IBM MQ Operator 3.0.0 e posterior, o IBM MQ Console usa Keycloak para gerenciamento de identidade e acesso. Consulte [Gerenciamento de Identidade e Acesso](#) na documentação do IBM Cloud Pak for Integration
- Para implementações do IBM MQ Operator anteriores à versão 3.0.0, o IBM MQ Console usa o IBM Cloud Pak Identity and Access Manager (IAM). O componente IAM pode já ter sido configurado pelo administrador de cluster. No entanto, se essa for a primeira vez que o IAM foi usado no cluster do Red Hat OpenShift , será necessário recuperar a senha do administrador inicial. Consulte [Obtendo a Senha do Administrador Inicial](#)

Se você estiver usando uma licença do IBM MQ , o IBM MQ Console não será pré-configurado e será necessário configurá-lo sozinho. Para obter mais informações, veja [Configurando usuários e funções](#). Para obter um exemplo, consulte [“Configurando o IBM MQ Console com um registro básico usando o IBM MQ Operator”](#) na página 169.

Tarefas relacionadas

[“Configurando uma Rota para se conectar a um gerenciador de filas de fora de um cluster do Red Hat OpenShift”](#) na página 154

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 UI (previously the 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.
- Selecione a equipe para editá-la,
 - Selecione **Recursos > Gerenciar Recursos**..
 - Selecione os namespaces que você deseja que essa equipe administre Eles podem ser quaisquer namespaces com um gerenciador de filas

OpenShift CP4I 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 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 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 Operator 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.

OpenShift CP4I 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
<code>ibmmq_qmgr_commit_total</code>	counter	Contagem de confirmações
<code>ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage</code>	gauge	Carregamento da CPU - média de quinze minutos
<code>ibmmq_qmgr_cpu_load_five_minute_average_percentage</code>	gauge	Carregamento da CPU - média de cinco minutos
<code>ibmmq_qmgr_cpu_load_one_minute_average_percentage</code>	gauge	Carregamento da CPU - média de um minuto
<code>ibmmq_qmgr_destructive_get_bytes_total</code>	counter	Obtenção destrutiva de total de intervalo - contagem de bytes
<code>ibmmq_qmgr_destructive_get_total</code>	counter	Obtenção destrutiva de total de intervalo - contagem

Métrica	tipo	Descrição
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
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/MQCONN 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

Métrica	tipo	Descrição
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
ibmmq_qmgr_log_max_bytes	gauge	Log - máximo de bytes
ibmmq_qmgr_log_occupied_by_reusable_extents_bytes	gauge	Log - bytes ocupados por extensões reutilizáveis
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_required_for_media_recovery_bytes	gauge	Log - bytes necessários para a recuperação de mídia
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

Métrica	tipo	Descrição
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
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

Métrica	tipo	Descrição
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_browser_bytes_total	counter	Procura de mensagem persistente - contagem de bytes
ibmmq_qmgr_persistent_message_browser_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
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

Métrica	tipo	Descrição
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
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

Informações relacionadas

[Métricas publicadas sobre os tópicos do sistema](#)

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.

- 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 **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

- [“Coletando informações de resolução de problemas para os gerenciadores de filas implementados com o IBM MQ Operator”](#) na página 179
- [“Resolução de Problemas: Obtendo Acesso aos Dados do Gerenciador de Filas”](#) na página 181

OpenShift

CP4I

Coletando informações de resolução de problemas para os gerenciadores de filas implementados com o IBM MQ Operator

Coletando informações de resolução de problemas que devem ser fornecidas ao Suporte IBM ao levantar um novo caso de suporte.

Procedimento

1. Coletar informações do provedor em nuvem.

Esse é o provedor em nuvem que hospeda seu cluster do Red Hat OpenShift (por exemplo, IBM Cloud).

2. Coletar informações de arquitetura.

A arquitetura do cluster do Red Hat OpenShift é uma das seguintes:

- Linux for x86-64
 - Linux on Power Systems (ppc64le)
 - Linux for IBM Z
3. Colete as informações de implementação do IBM MQ
- a) Efetue logon no cluster do Red Hat OpenShift usando um shell bash/zsh .
 - b) Configure as variáveis de ambiente a seguir:

```
export QM=QueueManager_name
export QM_NAMESPACE=QueueManager_namespace
export MQ_OPERATOR_NAMESPACE=mq_operator_namespace
```

Em que *QueueManager_name* é o nome de seu recurso QueueManager , *QueueManager_namespace* é o namespace no qual ele é implementado e *mq_operator_namespace* é o namespace no qual o IBM MQ Operator é implementado Isso pode ser igual ao namespace QueueManager .

- c) Execute os comandos a seguir e forneça todos os arquivos de saída resultantes para o Suporte do IBM

```
# OCP / Kubernetes: Version
oc version -o yaml > ocversion.yaml

# QueueManager: YAML
oc get qmgr $QM -n $QM_NAMESPACE -o yaml > "queue-manager-$QM.yaml"

# MQ Queue Manager: Pods
oc get pods -n $QM_NAMESPACE -o wide --selector "app.kubernetes.io/instance=$QM" > "qm-pods-$QM.txt"

# MQ Queue Manager: Pod YAML
oc get pods -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-pods-$QM.yaml"

# MQ Queue Manager: Pod Logs
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc logs -n $QM_NAMESPACE --previous "$p" > "qm-logs-previous-$p.txt"; oc logs -n $QM_NAMESPACE $p > "qm-logs-$p.txt";done

# MQ Web UI: Console Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/installations/Installation1/servers/mqweb/logs/console.log" "web-$p-console.log"; done

# MQ Web UI: Messages Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/installations/Installation1/servers/mqweb/logs/messages.log" "web-$p-messages.log"; done

# MQ Queue Manager: routes defined by operator
oc get routes -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-routes-$QM.yaml"

# MQ Queue Manager: routes to QM
oc get routes -n $QM_NAMESPACE -o yaml --field-selector "spec.to.name=$QM-ibm-mq" > "qm-routes2-$QM.yaml"

# MQ Queue Manager: stateful set
oc get statefulset -n $QM_NAMESPACE -o yaml ${QM}-ibm-mq > "qm-statefulset-$QM.yaml"

# MQ Queue Manager: services
oc get services -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-services-$QM.yaml"

# MQ Queue Manager: PVCs
oc get pvc -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-pvcs-$QM.yaml"

# MQ Operator: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-mq\|NAME" > mq-operator-csv.txt

# Cloud Pak Foundational Services: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-common-service-operator\|NAME" > common-services-csv.txt
```

```
# Cloud Pak for Integration: Version (if applicable)
oc get csv -n $QM_NAMESPACE | grep "^ibm-integration-platform-navigator\|NAME" > cp4i-
csv.txt

# Output from runmqras (this may take a while to execute)
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do timestamp=$(TZ=UTC date +"%Y%m%d_%H%M%S");
oc exec -n $QM_NAMESPACE $p -- runmqras -workdirectory "/tmp/runmqras_${timestamp}"
-section logger,mqweb,nativeha,trace; oc cp -n $QM_NAMESPACE --retries=10 "$p:tmp/
runmqras_${timestamp}/" .; done

# MQ Operator: Pod Log
oc logs -n $MQ_OPERATOR_NAMESPACE $(oc get pods -n $MQ_OPERATOR_NAMESPACE --no-headers
--selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/managed-by=olm | cut -d ' ' -f
1) > mq-operator-log.txt
```

Nota:

A maioria desses comandos requer acesso ao namespace no qual o gerenciador de filas é implementado.. No entanto, a coleta do log IBM MQ Operator pode requerer adicionalmente o acesso de **administrador de cluster** se o IBM MQ Operator estiver instalado **com escopo definido no cluster**

Tarefas relacionadas

[Coletando informações de resolução de problemas para o Suporte IBM](#)

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 ish pvc-inspector-pod-name
```

Você é colocado no diretório contendo os diretórios PVC montados.

- b) Liste os diretórios PVC executando o comando a seguir:

```
ls
```

- c) Consulte uma lista de PVCs executando 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 CP4I-LTS 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-VTPK-22YZPK	Production ou NonProduction	IBM Cloud Pak for Integration 2023.4.1	9.3.4 ou 9.3.5
L-QYQF-8UFZBN	Production ou NonProduction	IBM Cloud Pak for Integration Limited Edition 2023.4.1	9.3.4 ou 9.3.5
L-AMRD-XH6P3Q	Production	IBM MQ Advanced e IBM MQ Advanced para Ambiente de Não Produção 9.3 -05/2023	9.3.3, 9.3.4 ou 9.3.5
L-AXAF-JLZ53A	Development	IBM MQ Advanced for Developers (Não Warranted) 9.3 -05/2023	9.3.3, 9.3.4 ou 9.3.5

Valor de spec.license.license	Valor de spec.license.use	Informações sobre licença	Versões aplicáveis do IBM MQ
L-YBXJ-ADJNSM	Production ou NonProduction	IBM Cloud Pak for Integration 2023.2.1	9.3.3
L-PYRA-849GYQ	Production ou NonProduction	IBM Cloud Pak for Integration Edição Limitada 2023.2.1 ..	9.3.3
L-RJON-CJR2RX	Production ou NonProduction	IBM Cloud Pak for Integration 2022.4.1	9.3.1 ou 9.3.2
L-RJON-CJR2TC	Production ou NonProduction	IBM Cloud Pak for Integration Edição limitada 2022.4.1	9.3.1 ou 9.3.2
L-UPFX-8MW49T	Production	IBM MQ Advanced e IBM MQ Advanced para Ambiente de Não Produção 9.3-02/2023	9.3.2
L-APIG-CAUEQC	Development	IBM MQ Advanced for Developers (Não Warranted) 9.3	9.3.0, 9.3.1 ou 9.3.2
L-RJON-CD3JKX	Production ou NonProduction	IBM Cloud Pak for Integration 2022.2.1	9.3.0 ou 9.3.1
L-RJON-CD3JJU	Production ou NonProduction	IBM Cloud Pak for Integration Edição Limitada 2022.2.1 ..	9.3.0 ou 9.3.1
L-APIG-CAUEBE	Production	IBM MQ Advanced e IBM MQ Advanced para Ambiente de Não Produção 9.3	9.3.0 ou 9.3.1

Observe que a *versão* da licença é especificada, que nem sempre é a mesma da versão do IBM MQ.

Versões de licença mais antigas

Consulte [Versões de licença anteriores](#) na documentação do IBM MQ 9.2 .

  **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. IBM MQ Documentação: <https://ibm.biz/BdPZqj>. Referência da licença: <https://ibm.biz/BdPZfq>.

Campo	Descrição
Sequência de apiVersion	APIVersion 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 kind	Kind 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 .
metadata	

Campo	Descrição
<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 183

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> Anotações	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.
<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> <code>QueueManagerConfig</code>	Configurações para o contêiner do Gerenciador de Filas e o Gerenciador de Filas subjacente.
<code>securityContext</code> <code>SecurityContext</code>	Configurações de segurança a serem adicionadas ao securityContext do pod do gerenciador de filas.
<code>telemetry</code> Telemetria	Configurações para a configuração do Open Telemetry Requer MQ Operator 2.2.0 ou superior.
<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.

Campo	Descrição
Número inteiro terminationGracePeriod 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.
tracing 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.
web 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 184](#)

.spec.imagePullSecrets

LocalObjectReference contém informações suficientes para deixar você localizar o objeto referenciado dentro do mesmo namespace.

Aparece em:

- [“.spec” na página 184](#)

Campo	Descrição
Sequência de name	Nome do referente. Mais informações: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names PENDÊNCIA: incluir outros campos úteis. apiVersion, kind, 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 184](#)

.spec.license

Configurações que controlam sua aceitação da licença e quais as métricas de licença a serem usadas.

Aparece em:

- [“.spec” na página 184](#)

Campo	Descrição
Booleano <code>accept</code>	Indica se você aceita ou não a licença associada a este software (obrigatório).
Sequência de <code>license</code>	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 https://ibm.biz/BdPZfq para obter os valores válidos
Sequência de <code>metric</code>	Configuração que especifica qual métrica de licença usar. Por exemplo, <code>ProcessorValueUnit</code> , <code>VirtualProcessorCore</code> ou <code>ManagedVirtualServer</code> . O padrão é <code>ProcessorValueUnit</code> ao usar uma licença do MQ e <code>VirtualProcessorCore</code> ao usar uma licença do Cloud Pak for Integration.
Sequência de <code>use</code>	Configuração que controla como o software será usado, na qual a licença suporta diversos usos. Consulte https://ibm.biz/BdPZfq para obter os 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 184

Campo	Descrição
<code>keys</code> Matriz do PKISource	Chaves privadas a serem adicionadas ao repositório de chaves do Queue Manager.
<code>trust</code> Matriz do 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 186

Campo	Descrição
Sequência de <code>name</code>	O nome é usado como o rótulo para a chave ou o certificado. Deve ser uma sequência alfanumérica minúscula.
<code>secret</code> 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 186

Campo	Descrição
Matriz de <code>items</code>	Chaves dentro do segredo do Kubernetes que devem ser incluídas no contêiner do Gerenciador de Filas.
Sequência de <code>secretName</code>	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 186](#)

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 187](#)

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 184](#)

Campo	Descrição
availability 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 debug	Indica se mensagens de depuração do código específico do contêiner devem ou não ser registradas no log do contêiner. Usa false como padrão.
Sequência de image	A imagem de contêiner que será usada.
Sequência de imagePullPolicy	Configuração que controla quando o kubelet tenta extrair a imagem especificada. O padrão é IfNotPresent.
ini Matriz de INISource	Configurações para fornecer INI para o Gerenciador de Filas. Requer o MQ Operator 1.1.0 ou mais recente.
livenessProbe QueueManagerLivenessProbe	Configurações que controlam a análise de vivacidade.
Sequência de logFormat	Qual o formato de log a usar para este contêiner. Use JSON para logs formatados por JSON do contêiner. Use Basic para mensagens formatadas por texto. O padrão é Basic.
metrics QueueManagerMetrics	Configurações para métricas de estilo Prometheus.

Campo	Descrição
mqsc Matriz MQSCSource	Configurações para fornecer MQSC para o Gerenciador de Filas. Requer o MQ Operator 1.1.0 ou mais recente.
Sequência de name	Nome do Gerenciador de Filas do MQ subjacente, se diferente de metadata.name. Use esse campo se desejar um nome do Gerenciador de filas que não esteja em conformidade com as regras do Kubernetes para nomes (por exemplo, um nome que inclui letras maiúsculas).
readinessProbe QueueManagerReadinessProbe	Configurações que controlam a análise de prontidão.
recoveryLogs RecoveryLogs	Configurações para logs de recuperação do MQ .. Requer MQ Operator 2.4.0 ou superior.
resources Recursos	Configurações que controlam os requisitos de recursos.
route Rota	Configurações da rota do Gerenciador de Filas. Requer MQ Operator 1.4.0 ou mais recente.
startupProbe 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.
storage 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 187

Campo	Descrição
tls 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.
Sequência de type	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 do active e o outro 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.
Sequência de updateStrategy	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 188](#)

Campo	Descrição
Sequência de cipherSpec	O nome do CipherSpec para TLS NativeHA.
Sequência de secretName	O nome do segredo do Kubernetes.

.spec.queueManager.ini

Origem de arquivos de configuração INI.

Aparece em:

- [“.spec.queueManager” na página 187](#)

Campo	Descrição
configMap ConfigMapINISource	O ConfigMap representa um ConfigMap do Kubernetes que contém informações de INI.
secret SecretINISource	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 189](#)

Campo	Descrição
Matriz de items	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 189](#)

Campo	Descrição
Matriz de items	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 187](#)

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 SingleInstance. Padronizado para 0 segundos 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 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 187](#)

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 187](#)

Campo	Descrição
<code>configMap</code> ConfigMapMQSCSource	O ConfigMap representa um ConfigMap do Kubernetes que contém informações do MQSC.
<code>secret</code> 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 190](#)

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 190

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 187

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 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 periodSeconds	Com que frequência (em segundos) realizar a análise. O padrão é 5 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 3 segundos. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.recoveryLogs

Configurações para logs de recuperação do MQ .. Requer MQ Operator 2.4.0 ou superior.

Aparece em:

- [“.spec.queueManager”](#) na página 187

Campo	Descrição
Número inteiro logFilePages	Os dados do log de recuperação são retidos em uma série de arquivos. O tamanho do arquivo de log é especificado em unidades de páginas de 4 KB.

.spec.queueManager.resources

Configurações que controlam os requisitos de recursos.

Aparece em:

- [“.spec.queueManager”](#) na página 187

Campo	Descrição
<code>limits</code> <u>Limites</u>	Configurações de CPU e memória.
<code>requests</code> <u>Solicitações</u>	Configurações de CPU e memória.

.spec.queueManager.resources.limits

Configurações de CPU e memória.

Aparece em:

- [“.spec.queueManager.resources”](#) na página 191

Campo	Descrição
<code>cpu</code>	
<code>memory</code>	

.spec.queueManager.resources.requests

Configurações de CPU e memória.

Aparece em:

- [“.spec.queueManager.resources”](#) na página 191

Campo	Descrição
<code>cpu</code>	
<code>memory</code>	

.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 187

Campo	Descrição
Booleano <code>enabled</code>	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 187

Campo	Descrição
Número inteiro <code>failureThreshold</code>	Falhas mínimas consecutivas para que a análise seja considerada com falha. O padrão é 24.
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 .

Campo	Descrição
Número inteiro periodSeconds	Com que frequência (em segundos) realizar a análise. O padrão é 5 segundos.
Número inteiro successThreshold	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida. O padrão é 1.
Número inteiro timeoutSeconds	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 187](#)

Campo	Descrição
Sequência de defaultClass	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 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 defaultDeleteClaim	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 deleteClaim que substituirá essa configuração defaultDeleteClaim. Usa false como padrão.
persistedData QueueManagerOptionalVolume	Detalhes de persistentVolume para dados persistidos do MQ, incluindo configuração, filas e mensagens. Necessário ao usar o Gerenciador de Filas de várias instâncias.
queueManager QueueManagerVolume	PersistentVolume padrão para quaisquer dados normalmente em /var/mqm. Conterá todos os dados persistidos e logs de recuperação, se nenhum outro volume for especificado.
recoveryLogs QueueManagerOptionalVolume	Detalhes de volume persistente para logs de recuperação do MQ. Necessário ao usar o Gerenciador de Filas de várias instâncias.
scratch Rascunho	Configurações para o volume temporário do Gerenciador de Filas. Esse volume será montado como a pasta '/run' no contêiner Aplicável apenas se o sistema de arquivos raiz estiver configurado como somente leitura Requer MQ Operator 3.0.0 ou superior.
tmp Tmp	Configurações para o volume temporário de Tmp do Gerenciador de Filas. Esse volume será montado no contêiner como a Pasta '/tmp'. Os arquivos de dados de diagnóstico, como o arquivo zip produzido pelo comando runmqras, serão criados neste volume Aplicável apenas se o sistema de arquivos raiz estiver configurado como somente leitura Requer MQ Operator 3.0.0 ou superior.

.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 193](#)

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 do ephemeral. Os arquivos ainda são gravados em um diretório temporário, portanto, é possível usar essa opção para limitar o tamanho. Válido apenas se type for ephemeral e o sistema de arquivos raiz for configurado como somente leitura. Requer MQ Operator 3.0.0 ou superior.
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 193](#)

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.
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.

Campo	Descrição
Sequência de <code>sizeLimit</code>	Limite de tamanho ao usar um volume do <code>ephemeral</code> . Os arquivos ainda são gravados em um diretório temporário, portanto, é possível usar essa opção para limitar o tamanho. Válido apenas se <code>type</code> for <code>ephemeral</code> e o sistema de arquivos raiz for configurado como somente leitura. Requer MQ Operator 3.0.0 ou superior.
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 193](#)

Campo	Descrição
Sequência de <code>class</code>	Classe de armazenamento a usar para este volume. Apenas válido se <code>type</code> for <code>persistent-claim</code> . Se <code>type of availability</code> for <code>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</code> for <code>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 <code>PersistentVolume</code> para transmitir ao Kubernetes, incluindo unidades do SI. Apenas válido se <code>type</code> for <code>persistent-claim</code> . Por exemplo, <code>2Gi</code> . O padrão é <code>2Gi</code> .
Sequência de <code>sizeLimit</code>	Limite de tamanho ao usar um volume do <code>ephemeral</code> . Os arquivos ainda são gravados em um diretório temporário, portanto, é possível usar essa opção para limitar o tamanho. Válido apenas se <code>type</code> for <code>ephemeral</code> e o sistema de arquivos raiz for configurado como somente leitura. Requer MQ Operator 3.0.0 ou superior.
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.scratch

Configurações para o volume temporário do Gerenciador de Filas. Esse volume será montado como a pasta `"/run"` no contêiner. Aplicável apenas se o sistema de arquivos raiz estiver configurado como somente leitura. Requer MQ Operator 3.0.0 ou superior.

Aparece em:

- [“.spec.queueManager.storage” na página 193](#)

Campo	Descrição
Sequência de <code>sizeLimit</code>	Limite de tamanho do volume efêmero, incluindo as unidades SI Por exemplo, 2Gi. Válido apenas quando o sistema de arquivos raiz estiver configurado como somente leitura Requer MQ Operator 3.0.0 ou superior.

.spec.queueManager.storage.tmp

Configurações para o volume temporário de Tmp do Gerenciador de Filas. Esse volume será montado no contêiner como a Pasta '/tmp'. Os arquivos de dados de diagnósticos, como o arquivo zip produzido pelo comando `runmqras`, serão criados neste volume Aplicável apenas se o sistema de arquivos raiz estiver configurado como somente leitura Requer MQ Operator 3.0.0 ou superior.

Aparece em:

- [“.spec.queueManager.storage”](#) na página 193

Campo	Descrição
Sequência de <code>sizeLimit</code>	Limite de tamanho do volume efêmero, incluindo as unidades SI Por exemplo, 2Gi. Válido apenas quando o sistema de arquivos raiz estiver configurado como somente leitura Requer MQ Operator 3.0.0 ou superior.

.spec.securityContext

Configurações de segurança a serem adicionadas ao `securityContext` do pod do gerenciador de filas.

Aparece em:

- [“.spec”](#) na página 184

Campo	Descrição
Número inteiro <code>fsGroup</code>	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 <code>FSGroup</code> 2. O bit <code>setgid</code> está definido (novos arquivos criados no volume serão de propriedade do <code>FSGroup</code>) 3. Os bits de permissão são OR'd com <code>rw-rw----</code> Se não estiver definido, o Kubelet não modificará a propriedade e as permissões de nenhum volume.
Booleano <code>initVolumeAsRoot</code>	Afeta o <code>securityContext</code> usado pelo contêiner que inicializa o <code>PersistentVolume</code> . Configure como <code>true</code> se você estiver usando um provedor de armazenamento que requeira que você seja o usuário raiz para acessar volumes recém-provisionados. A configuração como <code>true</code> determina qual objeto de <code>Security Context Constraints (SCC)</code> 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 <code>false</code> como padrão. Para obter mais informações, consulte https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html .
Booleano <code>readOnlyRootFilesystem</code>	Se deve ou não ativar as configurações do sistema de arquivos raiz somente leitura para o Gerenciador de filas. Usa <code>false</code> como padrão. Requer MQ Operator 3.0.0 ou superior.
Matriz de <code>supplementalGroups</code>	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.telemetry

Configurações para a configuração do Open Telemetry Requer MQ Operator 2.2.0 ou superior.

Aparece em:

- [“.spec” na página 184](#)

Campo	Descrição
tracing Rastreio	Configurações para rastreio de Open Telemetry.

.spec.telemetry.tracing

Configurações para rastreio de Open Telemetry.

Aparece em:

- [“.spec.telemetry” na página 197](#)

Campo	Descrição
instana Instana	Configurações para o rastreio do Instana

.spec.telemetry.tracing.instana

Configurações para o rastreio do Instana

Aparece em:

- [“.spec.telemetry.tracing” na página 197](#)

Campo	Descrição
Sequência de agentHost	O nome do host do agente do Instana para o qual enviar dados de rastreio Isso não deve incluir um protocolo.
Booleano enabled	Se deve ou não ativar o rastreio do Instana. Usa false como padrão.
Sequência de protocol	O protocolo a ser usado na comunicação com o agente do Instana http e https são suportados.

.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 184](#)

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 rastreio com o Cloud Pak for Integration Operations Dashboard.

Aparece em:

- [“.spec” na página 184](#)

Campo	Descrição
agent TracingAgent	No Cloud Pak for Integration apenas, é possível definir configurações para o Agente de rastreo opcional.
collector TracingCollector	No Cloud Pak for Integration apenas, é possível definir configurações para o Coletor de rastreo opcional.
Booleano enabled	Indica se a integração com o Cloud Pak for Integration Operations Dashboard deve ou não ser ativada por meio do rastreo. Usa false como padrão.
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 197](#)

Campo	Descrição
Sequência de image	A imagem de contêiner que será usada.
Sequência de imagePullPolicy	Configuração que controla quando o kubelet tenta extrair a imagem especificada. O padrão é IfNotPresent.
livenessProbe TracingProbe	Configurações que controlam a análise de vivacidade.
readinessProbe TracingProbe	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 198](#)

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.tracing.agent.readinessProbe

Configurações que controlam a análise de prontidão.

Aparece em:

- [“.spec.tracing.agent” na página 198](#)

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 197](#)

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 199](#)

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.

Campo	Descrição
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.tracing.collector.readinessProbe

Configurações que controlam a análise de prontidão.

Aparece em:

- [“.spec.tracing.collector”](#) na página 199

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 184

Campo	Descrição
console Console	Configurações do console da web do MQ . Requer MQ Operator 3.0.0 ou superior.
Booleano enabled	Indica se o servidor da web deve ou não ser ativado. Usa false como padrão.
manualConfig ManualConfig	Configurações para fornecer a configuração XML do servidor da web Requer MQ Operator 3.0.0 ou superior.

.spec.web.console

Configurações do console da web do MQ . Requer MQ Operator 3.0.0 ou superior.

Aparece em:

- [“.spec.web” na página 200](#)

Campo	Descrição
authentication Autenticação	Configurações de autenticação para o MQ console da web. Requer MQ Operator 3.0.0 ou superior.
authorization Autorização	Configurações de autorização para o MQ console da web. Requer MQ Operator 3.0.0 ou superior.

.spec.web.console.authentication

Configurações de autenticação para o MQ console da web. Requer MQ Operator 3.0.0 ou superior.

Aparece em:

- [“.spec.web.console” na página 201](#)

Campo	Descrição
Sequência de provider	O provedor de autenticação para usar para o console da web do MQ . Use o <code>integration-keycloak</code> para usar a conexão única com a UI do Cloud Pak for Integration Platform (Keycloak). O padrão é <code>integration-keycloak</code> se você usar uma licença do Cloud Pak for Integration ou <code>manual</code> se usar uma licença do MQ . Use <code>manual</code> se você desejar fornecer sua própria configuração

.spec.web.console.authorization

Configurações de autorização para o MQ console da web. Requer MQ Operator 3.0.0 ou superior.

Aparece em:

- [“.spec.web.console” na página 201](#)

Campo	Descrição
Sequência de provider	O provedor de autorização para usar para o console da web do MQ . Use <code>integration-keycloak</code> para usar funções fornecidas pelo Cloud Pak for Integration Keycloak. Use <code>manual</code> se você desejar fornecer sua própria configuração O padrão é <code>integration-keycloak</code> se você usar uma licença do Cloud Pak for Integration ou <code>manual</code> se usar uma licença do MQ .

.spec.web.manualConfig

Configurações para fornecer a configuração XML do servidor da web Requer MQ Operator 3.0.0 ou superior.

Aparece em:

- [“.spec.web” na página 200](#)

Campo	Descrição
configMap ConfigMap	ConfigMap representa um Kubernetes ConfigMap que contém configuração XML do servidor da web.

Campo	Descrição
secret Segredo	O segredo representa um segredo do Kubernetes que contém a configuração XML do servidor da web. O uso de um Segredo protege quaisquer credenciais na camada Kubernetes , mas é possível que as ferramentas de monitoramento ou resolução de problemas possam expor o arquivo subjacente de forma insegura. Para segurança melhorada, codifique as credenciais usando "securityUtility".

.spec.web.manualConfig.configMap

ConfigMap representa um Kubernetes ConfigMap que contém configuração XML do servidor da web.

Aparece em:

- [“.spec.web.manualConfig”](#) na página 201

Campo	Descrição
Sequência de name	O nome da origem do Kubernetes.

.spec.web.manualConfig.secret

O segredo representa um segredo do Kubernetes que contém a configuração XML do servidor da web. O uso de um Segredo protege quaisquer credenciais na camada Kubernetes , mas é possível que as ferramentas de monitoramento ou resolução de problemas possam expor o arquivo subjacente de forma insegura. Para segurança melhorada, codifique as credenciais usando "securityUtility".

Aparece em:

- [“.spec.web.manualConfig”](#) na página 201

Campo	Descrição
Sequência de name	O nome da origem do Kubernetes.

.status

O estado observado do QueueManager.

Aparece em:

- [“QueueManager”](#) na página 183

Campo	Descrição
Sequência de adminUiUrl	URL para a IU do Administrador.
availability Disponibilidade	Status de disponibilidade para o Gerenciador de Filas.
conditions Matriz QueueManagerStatusCondi t	As condições representam as últimas observações disponíveis do estado do Gerenciador de Filas.
endpoints Matriz QueueManagerStatusEndpoin t	Informações sobre os terminais que este Gerenciador de Filas está expondo, como terminais de API ou IU.
metadata Metadados	Os metadados representam informações adicionais para o Gerenciador de Filas, incluindo Integração-statusKeycloak .
Sequência de name	O nome do Gerenciador de Filas.

Campo	Descrição
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 202](#)

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 202](#)

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 202](#)

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.metadata

Os metadados representam informações adicionais para o Gerenciador de Filas, incluindo `Integração-statusKeycloak`.

Aparece em:

- [“.status” na página 202](#)

Campo	Descrição
<code>integrationKeycloak</code> IntegrationKeycloak	QueueManagerStatusIntegrationKeycloak define o status de Integração-Keycloak para o QueueManager.

.status.metadata.integrationKeycloak

QueueManagerStatusIntegrationKeycloak define o status de Integração-Keycloak para o QueueManager.

Aparece em:

- [“.status.metadata” na página 203](#)

Campo	Descrição
Sequência de <code>clientName</code>	

.status.versions

Versão do MQ que está sendo usada e outras versões disponíveis no IBM Entitled Registry.

Aparece em:

- [“.status” na página 202](#)

Campo	Descrição
<code>available</code> QueueManagerStatusVersionAvailable	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 204](#)

Campo	Descrição
Matriz de <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 204](#)

Campo	Descrição
Matriz <code>licenses</code> Licenses	As licenças que são aplicáveis para esta versão do QueueManager
Sequência de <code>name</code>	name da versão para esta versão de QueueManager. Estes são valores válidos para o campo <code>spec.version</code> .

.status.versions.available.versions.licenses

QueueManagerStatusLicense define uma licença

Aparece em:

- [“.status.versions.available.versions”](#) na página 204

Campo	Descrição
Sequência de displayName	Nome de exibição da licença.
Sequência de link	Link para o conteúdo da licença.
Booleano matchesCurrentType	Se a licença corresponde ou não ao tipo de licença usado atualmente
Sequência de name	Nome da licença..

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 ⁹	Bloqueado	OperatorDependency	Para instalar, essa instância requer que o Keycloak seja configurado pela [IBM Cloud Pak for Integration]. Essa instância permanecerá no status [Pendente] até que o Keycloak seja relatado como [KeycloakReady] no recurso Cp4iServicesBinding para esse QueueManager.
			Para instalar, essa instância requer o operador [IBM IAM].. Essa instância permanecerá no status [Blocked] até que o operador seja instalado pelo [IBM Cloud Pak foundational services]..
	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	¹⁰ Um operando foi instalado por um operador que não é suportado na versão OCP <ocp_version>. Este operando não é suportado.
	Aviso	CP4I-LTS Suporte	¹¹ Um operando CP4I-LTS <mq_version> foi instalado, mas está sendo gerenciado por um operador que não se qualifica para a duração do suporte estendido.. Esse operando não se qualifica para a duração do suporte estendido.
	Aviso	CP4I-LTS Suporte	¹² Um operando CP4I-LTS <mq_version> foi instalado, mas o OCP versão 4< ocp_version> não se qualifica para a duração de suporte estendido. Esse operando não se qualifica para a duração do suporte estendido.
Aviso	CP4I-LTS Suporte	¹³ Um operando CP4I-LTS <mq_version> foi instalado, mas a versão do OCP < ocp_version> não se qualifica para a duração do suporte estendido Esse operando é suportado conforme uma liberação	

⁹ 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 estiver ativado, a condição OidcPending registrará o status do gerenciador de filas enquanto espera que o registro do cliente OIDC seja 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 ¹⁴	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 ¹⁵	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 seu próprio contêiner, considere se é possível usar o IBM MQ Operator. Consulte o ["Escolhendo como você deseja usar o IBM MQ em contêineres"](#) na página 5

Sobre esta tarefa

Procedimento

- ["Planejando a sua própria imagem do gerenciador de filas do IBM MQ usando um contêiner"](#) na página 208
- ["Construindo uma imagem de contêiner de gerenciador de filas do IBM MQ"](#) na página 208
- ["Executando aplicativos de ligação local em contêineres separados"](#) na página 211
- [Revise o IBM MQ gráfico Helm de amostra.](#)

¹⁰ Operador 1.4.0 e mais recente

¹¹ Operador 1.4.0 e mais recente

¹² Operador 1.4.0 e mais recente

¹³ Apenas Operador 1.3.0

¹⁴ 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

¹⁵ 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

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 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.

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.

Finalmente, você executa um contêiner usando esta imagem como seu sistema de arquivos, com o conteúdo de `/var/mqm` fornecido por um volume específico do 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 209
- [“Construindo uma imagem do gerenciador de filas do IBM MQ de amostra configurada”](#) na página 209

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](#).
- Opcional: Se você planeja configurar o acesso seguro usando o Red Hat OpenShift Container Platform "restricted" Security Context Constraint (SCC), use uma das imagens de não instalação do IBM MQ .
Os links para fazer download dessas imagens estão disponíveis na seção [Contêineres do IBM MQ downloads](#).

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

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. Na versão atual do IBM MQ, ao usar o pacote "Sem instalação" IBM MQ, não há mais nenhum usuário do 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 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 de namespace do processo entre contêineres, é possível executar aplicativos que requerem uma conexão de ligação local com o IBM MQ em contêineres separados do gerenciador de filas do IBM MQ

Sobre esta tarefa

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`.

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 **dspmqr** 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 **dspmqr**, **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 212 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 211 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 **dspmqr**, 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, 2024.
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 mensagem de erro AMQ8146.

No entanto, **dspmqr** 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 dlmqm QM1
AMQ7002: An error occurred manipulating a file.

docker exec container strmqweb
<file path>/mqrc.c[1152]
lpiObtainQMDetails --> 545261715
```

MQ Adv. 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.

Nota: Será possível criar um grupo de HA nativa somente em seus próprios contêineres dessa maneira se você estiver executando Kubernetes ou Red Hat OpenShift.

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 `ibmwebspheremqqueue_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 este 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 resolúvel 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)
```

MQ Adv.

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.

Kubernetes

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 REPLICAS 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 REPLICAS, 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 REPLICAS, 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.

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 **-lx** 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

[Comando dspmqr \(exibir gerenciadores de filas\)](#)

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 produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer outro produto, programa ou serviço, funcionalmente equivalente, poderá ser utilizado em substituição daqueles, desde que não infrinja nenhum direito de propriedade intelectual da IBM. 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:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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. São feitas alterações periódicas nas informações aqui contidas; tais 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.

Os licenciados deste programa que desejarem obter informações sobre este assunto com o propósito de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) o uso mútuo das informações trocadas, deverão 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, destinam-se a ajudá-lo a criar software aplicativo 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 essas informações de diagnóstico, modificação e ajuste como uma interface de programação, pois elas estão sujeitas a mudanças

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 registradas 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 registradas e logotipos baseados em Java são marcas ou marcas registradas da Oracle e/ou de suas afiliadas.



Part Number:

(1P) P/N: