

9.1

IBM MQ em contêineres

IBM

Nota

Antes de usar estas informações e o produto que elas suportam, leia as informações em [“Avisos” na página 53](#).

Esta edição se aplica à versão 9 liberação 1 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.....	5
Planejando-se para o IBM MQ em contêineres.....	5
Escolhendo como você deseja usar o IBM MQ em contêineres.....	5
Suporte para os contêineres certificados do IBM MQ.....	6
Suporte para a construção de suas próprias imagens de contêiner e gráficos do IBM MQ.....	8
Considerações de armazenamento para o IBM MQ Advanced certified container.....	9
Alta disponibilidade para IBM MQ Advanced certified container.....	10
Autenticação de usuário e autorização para o IBM MQ Advanced certified container.....	12
Instalando e desinstalando o IBM MQ Operator no OpenShift.....	12
Instalando o IBM MQ Operator usando o console da web do OpenShift.....	13
Instalando o IBM MQ Operator usando a CLI do OpenShift.....	14
Implementando contêineres certificados do IBM MQ.....	16
Preparando seu projeto OpenShift para o IBM MQ usando a CLI do OpenShift.....	16
Implementando um gerenciador de filas usando o IBM Cloud Pak for Integration Platform Navigator.....	17
Implementando um gerenciador de filas usando o console da web do OpenShift.....	18
Implementando um gerenciador de filas usando a CLI do OpenShift.....	19
Integrando-se ao IBM Cloud Pak for Integration Operations Dashboard.....	21
Construindo uma imagem com arquivos personalizados MQSC e INI, usando a CLI do OpenShift.....	22
Implementando contêineres certificados do IBM MQ usando o Helm.....	23
Implementando as liberações anteriores do CD do IBM MQ em um cluster do IBM Cloud Private.....	27
Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Private.....	29
Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Kubernetes Service.....	30
Conectando-se a um gerenciador de filas implementado em um cluster do OpenShift.....	30
Conectando-se ao IBM MQ Console implementado em um cluster OpenShift.....	32
Fazendo backup e restaurando a configuração do gerenciador de filas usando a CLI do OpenShift.....	33
Construindo seu próprio contêiner do IBM MQ.....	34
Planejando a sua própria imagem do gerenciador de filas do IBM MQ usando um contêiner.....	35
Construindo uma imagem do gerenciador de filas do IBM MQ de amostra usando o Docker.....	35
Executando aplicativos de ligação local em contêineres separados.....	38
Referência de API para o IBM MQ Operator.....	40
Referência da API para mq.ibm.com/v1beta1	40
Avisos.....	53
Informações sobre a Interface de Programação.....	54
Marcas comerciais.....	55

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 no contêiner predefinido fornecido no IBM MQ Advanced e no IBM MQ Advanced for Developers. Esse IBM MQ Advanced certified container oferece uma imagem e um gráfico do Helm suportados e pode ser usado para implementar uma imagem do IBM MQ pronta para produção no Red Hat® OpenShift®, no IBM Cloud Private ou no IBM Cloud Kubernetes Service.

Também é possível executar o IBM MQ em um contêiner do IBM Cloud Pak for Integration ou em um contêiner que você mesmo constrói.



Para obter mais informações sobre o IBM MQ Advanced certified container, consulte os links a seguir.

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 na arquitetura de contêineres, é necessário se familiarizar com ambos os conceitos básicos do IBM MQ (consulte [Visão geral técnica do IBM MQ](#)), assim como os conceitos básicos do Kubernetes/OpenShift (consulte [Arquitetura do OpenShift Container Platform](#)).

Procedimento

- [“Escolhendo como você deseja usar o IBM MQ em contêineres”](#) na página 5.
- [“Alta disponibilidade para IBM MQ Advanced certified container”](#) na página 10.
- [“Autenticação de usuário e autorização para o IBM MQ Advanced certified container”](#) na página 12.

Escolhendo como você deseja usar o IBM MQ em contêineres

Há diversas opções para usar o IBM MQ em contêineres: é possível optar por usar contêineres certificados predefinidos ou é possível construir suas próprias imagens e código de implementação.

Usando os contêineres certificados do IBM MQ Advanced

Se você estiver planejando implementar no Red Hat OpenShift Container Platform, provavelmente desejará usar os contêineres certificados. Há três variedades de contêineres certificados:

- IBM MQ Advanced certified container para IBM Cloud Pak for Integration. Esse é um produto IBM separado que inclui uma versão de um contêiner certificado.
- IBM MQ Advanced certified container
- Contêiner certificado do IBM MQ Advanced for Developers (não autorizado)

O IBM MQ 9.1.4 e as liberações anteriores do CD também eram suportados no IBM Cloud Private e no IBM Cloud Kubernetes Service.

Observe que os contêineres certificados estão evoluindo rapidamente e, portanto, são suportados apenas sob as liberações do [Continuous Delivery](#)

Os contêineres certificados incluem imagens de contêiner pré-construídas, bem como o código de implementação para execução no Red Hat OpenShift Container Platform. Do IBM MQ 9.1.5 em diante, os gerenciadores de filas são gerenciados usando um Operador do IBM MQ. Versões anteriores do IBM MQ, até e incluindo a versão 9.1.5, são gerenciadas usando gráficos Helm.

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

- Usar as APIs de REST para a administração ou o sistema de mensagens
- Usar qualquer um dos componentes do MQ a seguir:
 - Agentes do Managed File Transfer e seus recursos. No entanto, é possível usar os contêineres certificados para fornecer um ou mais gerenciadores de filas de Coordenação, Comando ou Agente.
 - AMQP
 - IBM MQ Bridge to Salesforce
 - IBM MQ Bridge to blockchain (não suportado em contêineres)
- Use o servidor da web quando estiver implementando usando gráficos do Helm (exceto para IBM Cloud Pak for Integration)
- Personalize opções usadas com `crtmqm`, `strmqm` e `endmqm`, como configurar logs de recuperação

Construindo suas próprias imagens e seus próprios gráficos

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

Estão disponíveis amostras para a construção de suas próprias imagens. Consulte o [“Construindo seu próprio contêiner do IBM MQ”](#) na página 34. Os gráficos do Helm fornecidos como parte dos contêineres certificados são publicados no GitHub e podem ser usados como amostras ao construir suas próprias imagens:

- [Gráfico do Helm para o IBM MQ Advanced certified container](#)
- [Gráfico do Helm para o contêiner certificado do IBM MQ Advanced for Developers](#)

Conceitos relacionados

[“Suporte para os contêineres certificados do IBM MQ”](#) na página 6


Os contêineres certificados do IBM MQ são suportados apenas em determinados ambientes do Kubernetes

[“Suporte para a construção de suas próprias imagens de contêiner e gráficos do IBM MQ”](#) na página 8
Informações a serem consideradas no caso de você estar usando contêineres em um sistema Linux.

Linux

Suporte para os contêineres certificados do IBM MQ

Os contêineres certificados do IBM MQ são suportados apenas em determinados ambientes do Kubernetes

 Para a liberação V9.1.4 do CD e mais recente, o IBM MQ Advanced certified container é suportado para uso com o Red Hat OpenShift. Consulte [“Implementando um gerenciador de filas usando a CLI do Helm”](#) na página 25.

As liberações do CD anteriores à V9.1.4 eram suportadas nos ambientes do Kubernetes a seguir:

- IBM Cloud Kubernetes Service
- IBM Cloud Private

- IBM Cloud Private com Red Hat OpenShift

Para obter versões suportadas específicas do Kubernetes, consulte os arquivos `qualification.yaml` e `Chart.yaml` dentro de um gráfico do Helm transferido por download do IBM MQ Advanced. Essas versões variam da liberação para liberação.

O IBM MQ Advanced certified container é suportado apenas quando implementado usando o IBM MQ Operator ou ao usar um dos gráficos Helm a seguir:

- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod` no IBM Cloud Pak for Integration

Nota: O uso de gráficos do Helm foi descontinuado, após a liberação do IBM MQ Operator.

Como a tecnologia de contêiner está evoluindo rapidamente, o IBM MQ Advanced certified container é suportado apenas na versão mais recente das plataformas que esse gráfico suporta no momento da liberação. Se você quiser usar uma versão de plataforma mais antiga, talvez precise usar uma versão mais antiga do IBM MQ Advanced certified container.

A imagem IBM MQ Advanced certified container é baseada em liberações de IBM MQ Continuous Delivery (CD). Elas são suportadas por até um ano, ou para duas liberações do CD, o que for mais longo. Long Term Support releases do IBM MQ não estão disponíveis como um contêiner certificado.

Do IBM MQ Advanced certified container V4.0 em diante, a imagem fornece uma instalação do IBM MQ em um Red Hat Universal Base Image (UBI), que inclui bibliotecas e utilitários essenciais do Linux usados pelo IBM MQ. A UBI é suportada pelo Red Hat quando executada em um host do Red Hat Enterprise Linux. Versões anteriores do IBM MQ Advanced certified container usavam uma imagem base não suportada do Ubuntu.

Conceitos relacionados

“Suporte para a construção de suas próprias imagens de contêiner e gráficos do IBM MQ” na página 8 Informações a serem consideradas no caso de você estar usando contêineres em um sistema Linux.

Linux > MQ Adv. > CD Suporte de versão para o IBM MQ Advanced certified container

Um conjunto de tabelas mostrando o mapeamento entre as versões suportadas do IBM MQ Advanced certified container, do IBM MQ, do IBM Cloud Kubernetes Service, do IBM Cloud Pak for Integration e do IBM Cloud Private.

Operador do IBM MQ

V 9.1.5

O IBM MQ Operator é suportado para uso como parte da IBM Cloud Pak for Integration versão 2020.2 ou independentemente, com IBM MQ versão 9.1.5 e acima.

O IBM MQ Operator é suportado na Red Hat OpenShift Container Platform versão 4.4 ou superior

IBM MQ Advanced certified container V 9.1.5 (gráfico Helm) -descontinuado

Inclui o gráfico do Helm `ibm-mqadvanced-server-prod`.

V 9.1.5

A partir do IBM MQ Advanced certified container V5.0.x, o gráfico do Helm, a imagem e as correções são enviados por meio do registro e catálogo autorizado da IBM. As versões anteriores foram enviadas por meio do Passport Advantage e as liberações de correção estão disponíveis em IBM Fix Central.

Tabela 1. Suporte para o IBM MQ Advanced certified container

Ve rs ão	Versão IBM MQ	Término de suporte	Plataformas suportadas
6.0.x	Liberção 9.1.5 do Continuous Delivery	Março de 2021	Requisitos do sistema detalhados
5.0.x	Liberção 9.1.4 do Continuous Delivery	Dezembro de 2020	Requisitos do sistema detalhados
4.1.x	Liberção 9.1.3 do Continuous Delivery	Julho de 2020	Requisitos do sistema detalhados

IBM MQ Advanced certified container software para o IBM Cloud Pak for Integration

V 9.1.5 (gráfico Helm)-descontinuado

Inclui o gráfico do Helm `ibm-mqadvanced-server-integration-prod`.

Tabela 2. Suporte de versão do software do IBM MQ Advanced certified container para o IBM Cloud Pak for Integration

Versã o	Versão IBM MQ	Versão IBM Cloud Pak for Integration
6.0.x	Liberção 9.1.4 do Continuous Delivery	2020.1.1 (Requisitos do sistema)
5.0.x	Liberção 9.1.3 do Continuous Delivery	2019.4.1 (Requisitos do sistema)
4.1.x	Liberção 9.1.3 do Continuous Delivery	2019.3.2.2 (Requisitos do sistema)
4.0.x	Liberção 9.1.3 do Continuous Delivery	2019.3.2 (Requisitos do sistema)
3.0.x	Liberção 9.1.3 do Continuous Delivery	2019.3.1 (Requisitos do sistema)

Consulte as [Notas sobre a liberação do IBM Cloud Pak for Integration](#) para obter informações sobre a versão suportada.

Linux Suporte para a construção de suas próprias imagens de contêiner e gráficos do IBM MQ

Informações a serem consideradas no caso de você estar usando contêineres em um sistema Linux.

- A imagem base usada pela imagem de contêiner deve usar um sistema operacional Linux que seja suportado.
- Deve-se usar os instaladores do IBM MQ para instalar o produto dentro da imagem do contêiner.
- Para obter uma lista de pacotes suportados, consulte [Componentes rpm do IBM MQ para sistemas Linux](#).
- V 9.1.0 Os pacotes a seguir não são suportados:
 - MQSeriesBCBridge

– MQSeriesRDQM

- O diretório de dados do gerenciador de filas (/var/mqm por padrão) deve ser armazenado em um volume de contêiner que mantém o estado persistente.

Importante: Não é possível usar o sistema de arquivos de união.

Deve-se montar um diretório de host como um volume de dados ou usar um contêiner de volume de dados. Para obter mais informações, veja [Gerenciar dados em contêineres](#).

- Deve-se ser capaz de executar comandos de controle do IBM MQ, como **endmqm**, dentro do contêiner.
- Deve-se ser capaz de obter arquivos e diretórios de dentro do contêiner para propósitos de diagnóstico.
- **V 9.1.0** É possível usar namespace para compartilhar os namespaces do contêiner para o gerenciador de filas com outros contêineres, a fim de vincular aplicativos localmente a um gerenciador de filas em execução em contêineres separados. Para obter informações adicionais, consulte [“Executando aplicativos de ligação local em contêineres separados”](#) na página 38.

Conceitos relacionados

“Suporte para os contêineres certificados do IBM MQ” na página 6

Os contêineres certificados do IBM MQ são suportados apenas em determinados ambientes do Kubernetes

Considerações de armazenamento para o IBM MQ Advanced certified container

O IBM MQ Advanced certified container é executado em dois modos de armazenamento:

- Para que seja possível usar o **armazenamento efêmero**, todos os estados de contêiner poderão ser descartado quando o contêiner for reiniciado. 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 operador do IBM MQ fornece a capacidade de personalizar 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 você usar o armazenamento efêmero, todos os estados do gerenciador de filas serão perdidos na reinicialização. 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 de 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. Além de descartar de todo o estado do sistema de mensagens no reinício, a configuração do gerenciador de filas também é descartada. 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 personalizados MQSC e INI, usando a CLI do OpenShift”](#) na

página 22). Se isso não for concluído, o IBM MQ precisará ser configurado cada vez que o contêiner for reiniciado.

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

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

Armazenamento persistente

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

- `spec.queueManager.availability` controla o modo de disponibilidade. Se você estiver usando `SingleInstance`, somente o armazenamento `ReadWriteOnce` será necessário. Para `MultiInstance`, no entanto, uma classe de armazenamento que suporte `ReadWriteMany` com as características de bloqueio de arquivo corretas será necessária. O IBM MQ fornece uma [instrução de suporte](#) e uma [instrução de teste](#). O modo de disponibilidade também influencia o layout de volume persistente. Para obter mais informações, consulte [“Alta disponibilidade para IBM MQ Advanced certified container”](#) na página 10.
- `spec.queueManager.storage` controla as configurações de armazenamento individuais. Um gerenciador de filas pode ser configurado para usar entre um e quatro volumes persistentes

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

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

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

```
spec:
  queueManager:
    availability:
      type: MultiInstance
    storage:
      queueManager:
        enabled: true
        class: ibmc-file-gold-gid
        persistedData:
          enabled: true
          class: ibmc-file-gold-gid
        recoveryLogs:
          enabled: true
          class: ibmc-file-gold-gid
      securityContext:
        supplementalGroups: [99]
```

Linux

MQ Adv.

CD

Alta disponibilidade para IBM MQ Advanced certified container

Você tem duas opções principais para alta disponibilidade com o IBM MQ Advanced certified container: **Gerenciador de filas de várias instâncias** (que é um par ativo/em espera, usando um sistema de

arquivos compartilhado, em rede) e **Gerenciador de filas resiliente único** (que oferece uma abordagem simples para alta disponibilidade usando armazenamento em rede).

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 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 se o contêiner está pronto. Isso é configurável no gráfico do Helm.
4. Quanto tempo leva para os clientes do IBM MQ se reconectarem.

Gerenciador de filas resiliente único

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

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

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

1. Quanto tempo leva para que a análise de atividade seja executada e quantas falhas são toleradas. Isso é configurável no gráfico do Helm.
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. Isso é configurável no gráfico do Helm.
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. Se o Nó principal do Kubernetes perder o contato com um nó 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.

Conceitos relacionados

[Configurações de alta disponibilidade](#)

Linux > MQ Adv. > CD **Autenticação de usuário e autorização para o IBM MQ Advanced certified container**

O IBM MQ pode ser configurado para usar usuários LDAP e grupos para autorização. Essa é a abordagem recomendada para o IBM MQ Advanced certified container.

Em um ambiente containerizado de diversos locatários, como o Red Hat OpenShift Container Platform, as restrições de segurança são estabelecidas para evitar potenciais problemas de segurança. Por exemplo, no Red Hat OpenShift Container Platform, o padrão `SecurityContextConstraints` (chamado `restricted`) usa um ID de usuário escolhido a esmo, desestimulando quaisquer usuários locais para o próprio contêiner. O IBM MQ geralmente usa a escalada de privilégio para verificar as senhas dos usuários, o que também não é recomendado em ambientes de contêiner de diversos locatários. Por essas razões, o uso de usuários definidos nas bibliotecas do sistema operacional dentro de um contêiner em execução não é suportado nos contêineres certificados do IBM MQ.

É necessário configurar o seu gerenciador de filas para usar o LDAP para a autenticação e a autorização do usuário. Para obter informações sobre como configurar o IBM MQ para fazer isso, consulte [Autenticação de conexão: repositórios de usuários e Autorização LDAP](#)

V 9.1.5 > Linux > MQ Adv. > CD **Instalando e desinstalando o IBM MQ Operator no OpenShift**

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

Antes de começar

Procedimento

- [“Instalando o IBM MQ Operator usando a CLI do OpenShift”](#) na página 14.

- “Instalando o IBM MQ Operator usando o console da web do OpenShift” na página 13.

V 9.1.5 Linux MQ Adv. CD Instalando o IBM MQ Operator usando o console da web do OpenShift

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

Antes de começar

Efetue login no console da web do cluster OpenShift.

Procedimento

1. Inclua os operadores do IBM Common Services na lista de operadores instaláveis

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

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

c) Clique em **Criar**.

2. Inclua os operadores IBM na lista de operadores instaláveis

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

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

c) Clique em **Criar**.

3. Crie um namespace a ser usado para o IBM MQ Operator

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

a) Na área de janela de navegação, clique em **Início > Projetos**.

A página de Projetos é exibida.

b) Clique em **Criar projeto**. Uma área Criar projeto é exibida.

c) Insira detalhes do namespace que você está criando. Por exemplo, é possível especificar "ibm-mq" como o nome.

d) Clique em **Criar**. O namespace para o seu IBM MQ Operator é criado.

4. Instale o operador do IBM MQ.

- a) Na área de janela de navegação, clique em **Operadores > OperatorHub**.
A página OperatorHub é exibida.
- b) No campo **Todos os itens**, insira "IBM MQ".
A entrada do catálogo do IBM MQ é exibida.
- c) Selecione **IBM MQ**.
A janela do IBM MQ é exibida.
- d) Clique em **Instalar**.
Você vê a página Criar assinatura do Operator.
- e) Configure o Modo de instalação para o namespace específico que você criou ou o escopo de abrangência do cluster
- f) Clique em **Subscrever**.
Você verá o IBM MQ na página Operadores instalados.
- g) Verifique o status do Operador na página Operadores instalados; ele mudará para Bem-sucedido quando a instalação estiver concluída.

Como proceder a seguir

[“Implementando contêineres certificados do IBM MQ” na página 16](#)

V 9.1.5 Linux MQ Adv CD Instalando o IBM MQ Operator usando a CLI do OpenShift

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

Antes de começar

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

Procedimento

1. Crie um `OperatorSource` para os operadores do IBM Common Services

- a) Crie um arquivo YAML definindo o recurso `OperatorSource`

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

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

- b) Aplique o `OperatorSource` no servidor.

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

2. Crie um `OperatorSource` para os operadores IBM

- a) Crie um arquivo YAML definindo o recurso `OperatorSource`

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

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

b) Aplique o `OperatorSource` no servidor.

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

3. Crie um namespace a ser usado para o IBM MQ Operator

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

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

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

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

5. Inspecione o IBM MQ Operator para verificar os InstallModes suportados e os canais disponíveis dele

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

6. Criar um arquivo YAML de objeto `OperatorGroup`

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

O namespace no qual você assina o Operador deve ter um `OperatorGroup` que corresponde ao `InstallMode` do Operador, seja no modo `AllNamespaces` ou no `SingleNamespace`. Se o Operador que você pretende instalar usar o `AllNamespaces`, o namespace `openshift-operators` já terá um `OperatorGroup` apropriado em vigor.

No entanto, se o Operador usar o modo `SingleNamespace` e você ainda não tiver um `OperatorGroup` apropriado em vigor, crie um.

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

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

b) Criar o objeto `OperatorGroup`

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

7. Crie um arquivo YAML de objeto de Assinatura para assinar um namespace para o MQ Operator

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

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-mq
  namespace: openshift-operators
spec:
```

```
channel:  
name: ibm-mq  
source: ibm-operator-catalog  
sourceNamespace: openshift-marketplace
```

Para o uso de AllNamespaces **InstallMode**, especifique o namespace openshift-operators. Caso contrário, especifique o namespace único relevante para o uso SingleNamespace **InstallMode**.

b) Criar o objeto Subscription

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

8. Verificar o status do Operador

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

Como proceder a seguir

[“Implementando contêineres certificados do IBM MQ” na página 16](#)

Linux > MQ Adv. > CD **Implementando contêineres certificados do IBM MQ**

IBM MQ versão 9.1.5 e acima podem ser implementados no Red Hat OpenShift usando o IBM MQ Operator. IBM MQ versões 9.1.5 e 9.1.4 podem ser implementadas no Red Hat OpenShift usando o Helm. Versões anteriores do CD podem ser implementadas em um cluster do IBM Cloud Private ou em um cluster do IBM Cloud Kubernetes Service, usando Helm.

Sobre esta tarefa

Procedimento

- [“Implementando um gerenciador de filas usando a CLI do Helm” na página 25.](#)
- [“Implementando as liberações anteriores do CD do IBM MQ em um cluster do IBM Cloud Private” na página 27.](#)
- [“Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Private” na página 29.](#)
- [“Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Kubernetes Service” na página 30.](#)

Linux > MQ Adv. > CD **Preparando seu projeto OpenShift para o IBM MQ usando a CLI do OpenShift**

Prepare seu cluster do Red Hat OpenShift Container Platform, para que ele esteja pronto para implementar um gerenciador de filas usando o IBM MQ Operator. Esta tarefa deve ser concluída por um administrador do projeto.

Antes de começar

Nota: Se você estiver planejando usar o IBM MQ em um projeto com outros componentes do IBM Cloud Pak for Integration já instalados, poderá não ser necessário seguir estas instruções.

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

Sobre esta tarefa

As imagens do IBM MQ Advanced certified container são extraídas de um registro de contêiner que executa uma verificação de titularidade de licença. Essa verificação requer uma chave de autorização que é armazenada em um segredo de extração de `docker-registry`. Se você ainda não tiver uma chave de autorização, siga estas instruções para obter uma chave de autorização e criar um segredo de extração.

Procedimento

1. Obtenha a chave de autorização designada para o seu ID.
 - a) Efetue login na [MyIBM Container Software Library](#) com o IBMid e a senha que estão associados ao software autorizado.
 - b) Na seção **Chaves de autorização**, selecione **Copiar chave** para copiar a chave de autorização para a área de transferência.
2. Crie um segredo contendo sua chave de autorização no projeto no qual você deseja implementar o seu gerenciador de filas.

Execute o seguinte comando, em que `<entitlement-key>` é a chave recuperada na etapa 1 e `<user-email>` é o ID IBM associado ao software intitulado.

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

Como proceder a seguir

[“Implementando um gerenciador de filas usando a CLI do OpenShift” na página 19](#)

V 9.1.5 Linux MQ Adv. CD Implementando um gerenciador de filas usando o IBM Cloud Pak for Integration Platform Navigator

Use o recurso customizado QueueManager para implementar um gerenciador de filas em um cluster de Plataforma de Contêineres Red Hat OpenShift usando o IBM Cloud Pak for Integration Platform Navigator. Essa tarefa deve ser concluída por um administrador do projeto

Antes de começar

Em um navegador, ative o IBM Cloud Pak for Integration Platform Navigator.

Se esta for a sua primeira implementação de um gerenciador de filas neste projeto do Red Hat OpenShift, siga as etapas para [“Preparando seu projeto OpenShift para o IBM MQ usando a CLI do OpenShift” na página 16](#).

Procedimento

1. Implementar um gerenciador de filas.

O exemplo a seguir implementa um gerenciador de filas de "iniciação rápida", que usa armazenamento efêmero (não persistente) e desliga a segurança do MQ. As mensagens não serão persistidas em reinicializações do gerenciador de filas. É possível ajustar a configuração para mudar muitas configurações do gerenciador de filas.

 - a) No IBM Cloud Pak for Integration Platform Navigator, clique em **Tempo de execução e instâncias**.
 - b) Clique em **Criar instância**.
 - c) Selecione **Gerenciador de Filas** e clique em **Avançar**.

O formulário para criar uma instância de um QueueManager é exibido.

Nota: Também é possível clicar em **Código** para visualizar ou mudar o YAML de configuração do QueueManager.

- d) Na seção **Detalhes**, verifique ou atualize o campo **Nome** e especifique o **Namespace** no qual a instância do gerenciador de filas será criada.
 - e) Se você aceitar o contrato de licença do IBM Cloud Pak for Integration, mude a **Aceitação da licença** para **Ativa**.
Deve-se aceitar a licença para implementar um gerenciador de filas.
 - f) Na seção **Configuração do Gerenciador de Filas**, verifique ou atualize o **Nome** do gerenciador de filas subjacente.
Por padrão, o nome do gerenciador de filas usado pelos aplicativos clientes do IBM MQ será o mesmo que o nome do QueueManager, mas com quaisquer caracteres inválidos (como hifens) removidos. Para forçar o uso de um nome específico, é possível realizar a edição aqui.
 - g) Clique em **Criar**.
A lista de gerenciadores de filas no projeto atual (namespace) agora é exibida. O novo QueueManager deve ter um status de Pending
2. Verifique se o gerenciador de filas está em execução
A criação terá sido concluída quando o status de QueueManager for Running.

Tarefas relacionadas

[“Conectando-se a um gerenciador de filas implementado em um cluster do OpenShift”](#) na página 30
Um conjunto de exemplos de configuração para a conexão com um gerenciador de filas implementado em um cluster do Red Hat OpenShift.

[“Conectando-se ao IBM MQ Console implementado em um cluster OpenShift”](#) na página 32
Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster Red Hat OpenShift Container Platform.

V 9.1.5 Linux MQ Adv. CD Implementando um gerenciador de filas usando o console da web do OpenShift

Use o recurso customizado do QueueManager para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform usando o console da web do Red Hat OpenShift. Essa tarefa deve ser concluída por um administrador do projeto

Antes de começar

Efetue login no console da web do cluster OpenShift. Será necessário selecionar um Projeto existente (namespace) para usar ou criar um novo.

Se esta for a sua primeira implementação de um gerenciador de filas neste projeto do Red Hat OpenShift, siga as etapas para [“Preparando seu projeto OpenShift para o IBM MQ usando a CLI do OpenShift”](#) na página 16.

Procedimento

1. Implementar um gerenciador de filas.
O exemplo a seguir implementa um gerenciador de filas de "iniciação rápida", que usa armazenamento efêmero (não persistente) e desliga a segurança do MQ. As mensagens não serão persistidas em reinicializações do gerenciador de filas. É possível ajustar a configuração para mudar muitas configurações do gerenciador de filas.
 - a) No console da web do OpenShift, na área de janela de navegação, clique em **Operadores > Operadores instalados**
 - b) Clique em **IBM MQ**.
 - c) Clique na guia **Gerenciador de Filas**.
 - d) Clique no botão **Criar QueueManager**.
Um editor YAML é exibido, contendo o YAML de exemplo para um recurso QueueManager.

Nota: Também é possível clicar em **Editar formulário** para visualizar ou mudar a configuração de QueueManager.

e) Se você aceitar o contrato de licença, mude **Aceitação de licença** para **Ativa**.

O IBM MQ está disponível sob várias licenças diferentes. Para obter mais informações sobre as licenças válidas, consulte [“Referência de licenciamento para mq.ibm.com/v1beta1”](https://www.ibm.com/docs/en/mq/v1beta1?topic=licensing) na página 40. Deve-se aceitar a licença para implementar um gerenciador de filas.

f) Clique em **Criar**.

A lista de gerenciadores de filas no projeto atual (namespace) agora é exibida. O novo QueueManager deve estar em um estado Pending.

2. Verifique se o gerenciador de filas está em execução

A criação terá sido concluída quando o status de QueueManager for Running.

Tarefas relacionadas

[“Conectando-se a um gerenciador de filas implementado em um cluster do OpenShift”](#) na página 30

Um conjunto de exemplos de configuração para a conexão com um gerenciador de filas implementado em um cluster do Red Hat OpenShift.

[“Conectando-se ao IBM MQ Console implementado em um cluster OpenShift”](#) na página 32

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

Implementando um gerenciador de filas usando a CLI do OpenShift

Use o recurso customizado QueueManager para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform usando a interface da linha de comandos (CLI). Essa tarefa deve ser concluída por um administrador do projeto

Antes de começar

É necessário instalar a [Interface da linha de comandos do Red Hat OpenShift Container Platform](#).

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

Se esta for a sua primeira implementação de um gerenciador de filas neste projeto do Red Hat OpenShift, siga as etapas para [“Preparando seu projeto OpenShift para o IBM MQ usando a CLI do OpenShift”](#) na página 16.

Procedimento

1. Implementar um gerenciador de filas.

O exemplo a seguir implementa um gerenciador de filas de "iniciação rápida", que usa armazenamento efêmero (não persistente) e desliga a segurança do MQ. As mensagens não serão persistidas em reinicializações do gerenciador de filas. É possível ajustar o conteúdo do YAML para mudar muitas configurações do gerenciador de filas.

a) Criar um arquivo YAML QueueManager

Por exemplo, para instalar um gerenciador de filas básico no IBM Cloud Pak for Integration, crie o arquivo "mq-quickstart.yaml" com o conteúdo a seguir:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
spec:
  version: 9.1.5.0-r2
  license:
    accept: false
    license: L-RJON-BN7PN3
```

```

    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"

```

Importante: se você aceitar o contrato de licença do IBM Cloud Pak for Integration, mude `accept: false` para `accept: true`. Consulte [“Referência de licenciamento para mq.ibm.com/v1beta1”](https://mq.ibm.com/v1beta1) na página 40 para obter detalhes sobre a licença.

Este exemplo também inclui um servidor da web implementado com o gerenciador de filas, com o console da web ativado com a Conexão única com o Cloud Pak Identity and Access Manager.

Para instalar um gerenciador de filas básico independentemente do IBM Cloud Pak for Integration, crie o arquivo `"mq-quickstart.yaml"` com o conteúdo a seguir:

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart
spec:
  version: 9.1.5.0-r2
  license:
    accept: false
    license: L-APIG-BM7GDH
    use: Development
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"

```

Importante: se você aceitar o contrato de licença MQ, altere `accept: false` para `accept: true`. Consulte [“Referência de licenciamento para mq.ibm.com/v1beta1”](https://mq.ibm.com/v1beta1) na página 40 para obter detalhes sobre a licença.

b) Criar o objeto QueueManager

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

2. Verifique se o gerenciador de filas está em execução

É possível validar a implementação executando

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

e, em seguida, verificando o status.

Por exemplo, execute

```
oc describe queuemanager quickstart
```

e confira se o campo `status.Phase` indica `Running`

Tarefas relacionadas

“Conectando-se a um gerenciador de filas implementado em um cluster do OpenShift” na página 30
Um conjunto de exemplos de configuração para a conexão com um gerenciador de filas implementado em um cluster do Red Hat OpenShift.

“Conectando-se ao IBM MQ Console implementado em um cluster OpenShift” na página 32
Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster Red Hat OpenShift Container Platform.

V 9.1.4 Linux MQ Adv. CD 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.

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 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 a implementação de um gerenciador de filas, consulte [“Implementando um gerenciador de filas usando o IBM Cloud Pak for Integration Platform Navigator”](#) na página 17

Se você estiver implementando usando a [CLI do OpenShift](#) ou o [console da web do OpenShift](#), será possível ativar o rastreamento com o fragmento YAML a seguir:

```
spec:
  tracing:
    enabled: true
    namespace: <Operations_Dashboard_Namespace
```

Se estiver implementando usando o Helm, será possível ativar o rastreamento configurando `odTracingConfig.enabled=true` e `odTracingConfig.odTracingNamespace=<Operations_Dashboard_Namespace`. Se você desejar ativar a integração do Painel de operações em um gerenciador de filas existente, será possível aplicar essa configuração durante o upgrade da liberação do Helm.

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 essa for a primeira vez que um gerenciador de filas com a integração do Painel de Operações foi implementado nesse namespace, será necessário [Registrar](#) com o Painel de Operações

O registro cria um objeto Segredo que o pod do gerenciador de filas precisa para ser iniciado com sucesso.

Construindo uma imagem com arquivos personalizados MQSC e INI, usando a CLI do 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 Segredo do OpenShift para o IBM Entitled Registry em seu projeto Red Hat OpenShift, siga as etapas para [“Preparando seu projeto OpenShift para o IBM MQ usando a CLI do OpenShift”](#) na página 16.

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:9.1.5.0-r2-amd64
      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:9.1.5.0-r2-amd64"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

Será necessário substituir os dois locais nos quais o IBM MQ base é mencionado, para apontar na imagem base correta para a versão e a correção que você deseja usar. 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 [“Implementando um gerenciador de filas usando a CLI do OpenShift”](#) na página 19, incluindo sua nova imagem personalizada no YAML.

Seria possível incluir o fragmento de YAML a seguir em seu YAML normal QueueManager, em que *my-namespace* é o projeto/namespace do OpenShift sendo usado e *image* é o nome da imagem criada anteriormente (por exemplo, "mymq:latest-amd64"):

```
spec:
  queueManager:
    image: image-registry.openshift-image-registry.svc:5000/my-namespace/my-image
```

Tarefas relacionadas

[“Implementando um gerenciador de filas usando a CLI do OpenShift”](#) na página 19

Use o recurso customizado QueueManager para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform usando a interface da linha de comandos (CLI). Essa tarefa deve ser concluída por um administrador do projeto

Linux

MQ Adv.

CD

Implementando contêineres certificados do IBM

MQ usando o Helm

No IBM MQ 9.1.5.0, a maneira recomendada de implementar um gerenciador de filas é usar o Operador do IBM MQ. IBM MQ 9.1.5.0 e liberações de CD anteriores podem ser implementadas usando Helm, usando as instruções a seguir.

Sobre esta tarefa

Procedimento

- [“Preparando seu cluster do OpenShift para IBM MQ no OpenShift usando o Helm”](#) na página 24.
- [“Implementando um gerenciador de filas usando a CLI do Helm”](#) na página 25.

MQ no OpenShift usando o Helm

Prepare o cluster do Red Hat OpenShift Container Platform para que ele esteja pronto para implementar um gerenciador de filas usando o Helm. Essa tarefa deve ser concluída por um administrador de cluster.

Antes de começar

Nota: Se você estiver usando o IBM Cloud Pak for Integration, o instalador deverá ter preparado um projeto do OpenShift (namespace) para você usar com o IBM MQ, portanto, talvez não seja necessário seguir essas instruções.

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

Procedimento

1. Certifique-se de ter incluído o repositório do Helm da IBM em sua cópia local do Helm.
Por exemplo, você pode executar o seguinte comando:

```
helm repo add ibm-entitled-charts https://raw.githubusercontent.com/IBM/charts/master/repo/entitled
```

2. Certifique-se de ter um servidor do Helm (chamado de "Tiller") instalado em seu cluster.
Siga as instruções em [Introdução ao Helm no OpenShift](#) para instalar o Helm em seu cluster.
3. Certifique-se de que as contas de serviço em seu projeto do OpenShift (namespace) estejam autorizadas a utilizar as restrições de contexto de segurança (SCCs) corretas.

V 9.1.5 O IBM MQ funciona sob a SCC padrão de "restrito", portanto, é normalmente possível ignorar essa etapa.

A aplicação de mudanças em SCCs precisa ser feita por um administrador de cluster do OpenShift. Há diferentes requisitos para as SCCs documentados no arquivo LEIA-ME individual de cada versão do gráfico do Helm:

```
helm inspect readme ibm-entitled-charts/ibm-mqadvanced-server-prod
```

Há instruções em cada LEIA-ME para a configuração da autorização para as SCCs. Observe que os gráficos do Helm do IBM MQ criam uma conta de serviço para uso próprio, o que significa que as permissões da SCC precisam ser aplicadas no nível "grupo" (para todas as contas de serviço no namespace).

4. Certifique-se de ter um "segredo de extração de imagem" válido para extrair imagens de seu registro de contêiner escolhido
As imagens do IBM MQ Advanced certified container são extraídas de um registro de contêiner que executa uma verificação de titularidade de licença. Essa verificação requer uma chave de autorização que é armazenada em um segredo de extração de `docker-registry`. Se você ainda não tiver uma chave de autorização, siga estas instruções para obter uma chave de autorização e criar um segredo de extração.
 - a) Obtenha a chave de autorização designada para o seu ID.
 - i) Efetue login na [MyIBM Container Software Library](#) com o IBMid e a senha que estão associados ao software autorizado.
 - ii) Na seção *Chaves de autorização*, selecione **Copiar chave** para copiar a chave de autorização para a área de transferência.
 - b) Crie o segredo no espaço de nomes no qual deseja implementar o seu gerenciador de filas.

- Execute o seguinte comando, em que `<entitlement-key>` é a chave recuperada na etapa 1 e `<user-email>` é o ID IBM associado ao software intitulado.

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

Como proceder a seguir

[“Implementando um gerenciador de filas usando a CLI do Helm”](#) na página 25

V 9.1.4 Linux MQ Adv. CD Implementando um gerenciador de filas usando a CLI do Helm

Use o Helm para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform. Esta tarefa deve ser concluída por um administrador do projeto.

Antes de começar

É necessário instalar o Helm V2 e a interface da linha de comandos do Red Hat OpenShift Container Platform. Se você não estiver usando o IBM Cloud Pak for Integration, siga as etapas para o [“Preparando seu cluster do OpenShift para IBM MQ no OpenShift usando o Helm”](#) na página 24.

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

Procedimento

1. Certifique-se de ter incluído o repositório do Helm da IBM em sua cópia local do Helm. Por exemplo, você pode executar o seguinte comando:

```
helm repo add ibm-entitled-charts https://raw.githubusercontent.com/IBM/charts/master/rep/entitled
```

2. Revise as opções de configuração para o seu gerenciador de filas

A etapa de implementação inclui as etapas de instalação e de configuração. Algumas configurações para o seu gerenciador de filas devem ser definidas no momento da implementação e, para mudá-las, será necessária uma reimplementação.

É possível visualizar o LEIA-ME do gráfico do Helm para obter detalhes de todas as opções de implementação disponíveis executando um dos comandos a seguir:

- Para o IBM MQ Advanced certified container no IBM Cloud Pak for Integration:

```
helm inspect readme ibm-entitled-charts/ibm-mqadvanced-server-integration-prod
```

- Para o IBM MQ Advanced certified container:

```
helm inspect readme ibm-entitled-charts/ibm-mqadvanced-server-prod
```

Geralmente, serão necessários pelo menos os parâmetros a seguir:

- a. Nome da liberação. Por exemplo: `my-release`
- b. Repositório remoto do Helm. Por exemplo: `ibm-entitled-charts`
- c. Gráfico do Helm: por exemplo, `ibm-mqadvanced-server-prod` ou `ibm-mqadvanced-server-integration-prod`
- d. Nome do segredo de extração de imagem. Por exemplo: `entitled-registry`. Note que isso não será necessário se você estiver implementando no projeto pré-definido para o MQ no IBM Cloud Pak for Integration

3. Implementar um gerenciador de filas.

Note que, por padrão, o gráfico do Helm assume que você tem um conjunto de classes de armazenamento padrão em seu cluster Red Hat OpenShift Container Platform.

Por exemplo, para instalar um gerenciador de filas básico no IBM Cloud Pak for Integration, execute o comando a seguir:

```
helm install \  
--tls \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-integration-prod \  
--set license=accept \  
--set tls.hostname=my.cluster \  
--set tls.generate=true
```

É possível inserir qualquer nome de host no campo `tls.hostname` (esse campo é obrigatório, mas não será usado porque estamos gerando um novo certificado autoassinado nesse exemplo)

Para instalar um gerenciador de filas básico independentemente do IBM Cloud Pak for Integration, é possível executar o comando a seguir:

```
helm install \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-prod \  
--set license=accept \  
--set image.pullSecret=ibm-entitlement-key
```

Tarefas relacionadas

[“Conectando-se a um gerenciador de filas implementado em um cluster do OpenShift”](#) na página 30

Um conjunto de exemplos de configuração para a conexão com um gerenciador de filas implementado em um cluster do Red Hat OpenShift.

[“Conectando-se ao IBM MQ Console implementado em um cluster OpenShift”](#) na página 32

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

Implementando um gerenciador de filas com o IBM Cloud File Storage usando a CLI do Helm

Cenário de exemplo para usar o Helm para implementar um gerenciador de filas em um cluster Red Hat OpenShift no IBM Cloud, usando IBM Cloud File Storage. Essa tarefa deve ser concluída por um administrador do projeto

Antes de começar

É necessário instalar o [Helm V2](#) e a [interface da linha de comandos do Red Hat OpenShift Container Platform](#). Se você não estiver usando o IBM Cloud Pak for Integration, siga as etapas para o [“Preparando seu cluster do OpenShift para IBM MQ no OpenShift usando o Helm”](#) na página 24.

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

Procedimento

1. Assegure-se de ter incluído o repositório do IBM Helm em sua cópia local do Helm.
Por exemplo, você pode executar o seguinte comando:

```
helm repo add ibm-entitled-charts https://raw.githubusercontent.com/IBM/charts/master/repo/entitled
```

2. Implementar um gerenciador de filas.

Ao usar o IBM Cloud File Storage, você geralmente verá os melhores resultados usando a classe de armazenamento `ibmc-file-gold-gid`. Essa classe de armazenamento ativa o armazenamento que pode ser gravado por usuários no grupo do sistema de arquivos correto

Por exemplo, para instalar um gerenciador de filas básico no IBM Cloud Pak for Integration, execute o comando a seguir:

```
helm install \
--tls \
--name my-release \
ibm-entitled-charts/ibm-mqadvanced-server-integration-prod \
--set license=accept \
--set tls.hostname=my.cluster \
--set tls.generate=true \
--set dataPVC.storageClassName=ibmc-file-gold-gid \
--set security.context.supplementalGroups={99}
```

É possível inserir qualquer nome do host no campo `tls.hostname` (este é um campo obrigatório, mas não é usado aqui porque, neste exemplo, estamos gerando um novo certificado autoassinado.)

Para instalar um gerenciador de filas básico independentemente do IBM Cloud Pak for Integration, é possível executar o comando a seguir:

```
helm install \
--name my-release \
ibm-entitled-charts/ibm-mqadvanced-server-prod \
--set license=accept \
--set image.pullSecret=ibm-entitlement-key \
--set dataPVC.storageClassName=ibmc-file-gold-gid \
--set security.context.supplementalGroups={99}
```

Tarefas relacionadas

“Conectando-se a um gerenciador de filas implementado em um cluster do OpenShift” na página 30
Um conjunto de exemplos de configuração para a conexão com um gerenciador de filas implementado em um cluster do Red Hat OpenShift.

“Conectando-se ao IBM MQ Console implementado em um cluster OpenShift” na página 32
Como conectar-se ao IBM MQ Console de um gerenciador de filas que foi implementado em um cluster Red Hat OpenShift Container Platform.

Linux

MQ Adv.

CD

Implementando as liberações anteriores do CD do IBM MQ em um cluster do IBM Cloud Private

Para versões de CD de IBM MQ anteriores à 9.1.4, use o console de gerenciamento do IBM Cloud Private para implementar um gerenciador de fila no IBM Cloud Private

Antes de começar



Atenção: **V 9.1.4** Essa implementação não é suportada no IBM MQ 9.1.4 nem em versões mais recentes.

Essa tarefa assume que você já tenha o [incluído como uma imagem do IBM MQ em um cluster do IBM Cloud Private](#).

O arquivo `README.md` do gráfico do Helm está disponível na entrada do catálogo do IBM Cloud Private, que é exibido após a conclusão [desta subetapa](#) ou por meio da linha de comandos [incluindo o repositório **local-charts**](#) do IBM Cloud Private como um repositório remoto do Helm e executando o comando a seguir:

```
helm inspect readme remote_repo_name/ibm-mqadvanced-server-prod
```

Deve-se ter um `PodSecurityPolicy` ou um `SecurityContextConstraint` (para o IBM Cloud Private no Red Hat OpenShift) que suporte o contexto de segurança necessário. Detalhes, incluindo exemplos, podem ser localizados no arquivo `README.md` do Gráfico do Helm.

Os detalhes sobre como configurar a liberação do Helm também podem ser localizados no arquivo `README.md` do gráfico do Helm.

Nota:

- Se você estiver implementando em um ambiente do IBM Cloud Private que não suporta as configurações de segurança necessárias por padrão, ative sua implementação seguindo as instruções em [Implementando gráficos do Helm que requerem privilégios elevados em um namespace não padrão](#) na documentação do produto IBM Cloud Private.
- Se você estiver usando o SELinux, deverá atender aos requisitos do IBM MQ descritos no [Suporte do IBM MQ para o SELinux no Red Hat Enterprise Linux](#).

Sobre esta tarefa

O IBM Cloud Private oferece uma plataforma para gerenciar aplicativos locais containerizados. Depois de incluir uma imagem do IBM MQ em um cluster do IBM Cloud Private, é possível usar o console de gerenciamento do IBM Cloud Private ou a linha de comandos para implementar um gerenciador de filas.

Procedimento

- Usando o console de gerenciamento do IBM Cloud Private
 - a) Abra o console de gerenciamento do IBM Cloud Private em um navegador da web e clique em **Catálogo**.
 Consulte [Acessando o cluster do IBM Cloud Private usando o console de gerenciamento](#) na documentação do produto IBM Cloud Private.
 - b) Selecione o gráfico `ibm-mqadvanced-server-prod` na lista.
 - c) Selecione **Configurar** e, em seguida, conclua as etapas de configuração a seguir:
 - a. Insira um nome de liberação.
 - b. Leia e aceite os contratos de licença.
 - c. Na seção **dataPVC**, configure **storageclass** para sua classe de armazenamento desejada. Deixe em branco para selecionar a classe de armazenamento padrão.
 - d. Na seção **imagem**, configure o repositório para o caminho de imagem completo. Por exemplo:


```
mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod
```
 - e. Na seção **imagem**, configure a tag para a tag de imagem. Por exemplo:


```
9.1.3.0-r1
```
 - f. Se for necessário que um segredo pull do Kubernetes acesse o registro da imagem, inclua-o como o **pullSecret**.
 - g. Na seção **queueManager**, configure o nome do gerenciador de filas.
 - d) Clique em **Instalar** para implementar seu gerenciador de filas como uma *Liberação do Helm*.
- Usando a linha de comandos
 - a) Configure **cloudctl** para acessar o cluster do IBM Cloud Private.
 Consulte [Instalando a CLI do IBM Cloud Private](#) na documentação do produto IBM Cloud Private.
 - b) Assegure-se de que tenha [incluído o repositório **local-charts** do IBM Cloud Private como um repositório remoto do Helm](#).
 - c) Instale o gráfico.
 Execute o comando a seguir, especificando estes parâmetros:
 - a. Nome da liberação (por exemplo, `my-release`)
 - b. Nome do repositório remoto do Helm que contém o gráfico `ibm-mqadvanced-server-prod` (por exemplo, `my-repo`)
 - c. Repositório de imagem (por exemplo, `mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod`)

d. Tag de imagem (por exemplo, 9.1.3.0-r1)

```
helm install --name my-release --repo my-repo ibm-mqadvanced-server-prod --set  
license=accept --set image.repository=mycluster.icp:8500/namespace_name/ibm-mqadvanced-  
server-prod --set image.tag=9.1.3.0-r1 --tls
```

Tarefas relacionadas

[“Implementando um gerenciador de filas usando a CLI do Helm” na página 25](#)

Use o Helm para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform. Esta tarefa deve ser concluída por um administrador do projeto.

[“Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Private” na página 29](#)

Para versões de CD de IBM MQ anteriores à 9.1.4, prepare seu cluster do IBM Cloud Private para implementar uma imagem pronta para produção para o IBM MQ

[“Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Kubernetes Service” na página 30](#)

Para versões de CD do IBM MQ anteriores à 9.1.4, importe uma imagem pronta para produção para o IBM MQ no IBM Cloud Kubernetes Service

Linux

MQ Adv.

CD

Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Private

Para versões de CD de IBM MQ anteriores à 9.1.4, prepare seu cluster do IBM Cloud Private para implementar uma imagem pronta para produção para o IBM MQ

Sobre esta tarefa



Atenção: **V 9.1.4** Essa importação não é suportada no IBM MQ 9.1.4 nem em versões mais recentes.

É possível fazer download de uma imagem do IBM MQ do Passport Advantage e importá-la para um contêiner do IBM Cloud Private.

Procedimento

1. Faça download da imagem do IBM MQ mais recente por meio do [Passport Advantage e Passport Advantage Express website](#).

Para obter detalhes dos downloads disponíveis, acesse [Fazendo download IBM MQ 9.1](#) e, então, clique na guia para a liberação que você deseja transferir por download. O nome e o número da peça para download são listados em uma tabela.

2. Importe o archive transferido por download para o IBM Cloud Private.

Consulte [Incluindo o software IBM no catálogo do IBM Cloud Private](#) na documentação do produto IBM Cloud Private.

Como proceder a seguir

Agora você está pronto para [Implementar um gerenciador de filas no IBM Cloud Private](#).

Tarefas relacionadas

[“Implementando um gerenciador de filas usando a CLI do Helm” na página 25](#)

Use o Helm para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform. Esta tarefa deve ser concluída por um administrador do projeto.

[“Implementando as liberações anteriores do CD do IBM MQ em um cluster do IBM Cloud Private” na página 27](#)

Para versões de CD de IBM MQ anteriores à 9.1.4, use o console de gerenciamento do IBM Cloud Private para implementar um gerenciador de fila no IBM Cloud Private

[“Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Kubernetes Service” na página 30](#)

Para versões de CD do IBM MQ anteriores à 9.1.4, importe uma imagem pronta para produção para o IBM MQ no IBM Cloud Kubernetes Service

Linux > MQ Adv. > CD > Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Kubernetes Service

Para versões de CD do IBM MQ anteriores à 9.1.4, importe uma imagem pronta para produção para o IBM MQ no IBM Cloud Kubernetes Service

Sobre esta tarefa



Atenção: **V 9.1.4** Essa importação não é suportada no IBM MQ 9.1.4 nem em versões mais recentes.

É possível fazer download de uma imagem do IBM MQ do Passport Advantage e importá-la para um cluster do IBM Cloud Kubernetes Service.

Procedimento

1. Faça download da imagem do IBM MQ mais recente por meio do [Passport Advantage e Passport Advantage Express website](#).

Para obter detalhes dos downloads disponíveis, acesse [Fazendo download IBM MQ 9.1](#) e, então, clique na guia para a liberação que você deseja transferir por download. O nome e o número da peça para download são listados em uma tabela.

2. Importe o archive transferido por download para o IBM Cloud Kubernetes Service.

Consulte [Executando as imagens do IBM Cloud Private em contêineres públicos do Kubernetes](#).

Tarefas relacionadas

[“Implementando um gerenciador de filas usando a CLI do Helm” na página 25](#)

Use o Helm para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform. Esta tarefa deve ser concluída por um administrador do projeto.

[“Implementando as liberações anteriores do CD do IBM MQ em um cluster do IBM Cloud Private” na página 27](#)

Para versões de CD de IBM MQ anteriores à 9.1.4, use o console de gerenciamento do IBM Cloud Private para implementar um gerenciador de fila no IBM Cloud Private

[“Incluindo liberações anteriores do CD de uma imagem do IBM MQ em um cluster do IBM Cloud Private” na página 29](#)

Para versões de CD de IBM MQ anteriores à 9.1.4, prepare seu cluster do IBM Cloud Private para implementar uma imagem pronta para produção para o IBM MQ

V 9.1.4 > Linux > MQ Adv. > CD > Conectando-se a um gerenciador de filas implementado em um cluster do OpenShift

Um conjunto de exemplos de configuração para a conexão com um gerenciador de filas implementado em um cluster do Red Hat OpenShift.

Sobre esta tarefa

É necessário uma [rota do OpenShift](#) para conectar o 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 do IBM MQ e no aplicativo cliente porque a Server Name Indication (SNI) está disponível apenas no protocolo TLS. O Red Hat OpenShift Container Platform Router usa SNI para rotear solicitações para o gerenciador de filas do IBM MQ.

A configuração necessária da rota do OpenShift depende do comportamento da SNI de seu aplicativo cliente.

Para configurar o cabeçalho SNI como TLS 1.2 ou superior, um CipherSpec ou CipherSuite deve ser usado para a sua comunicação TLS.

O SNI será configurado para o canal MQ se as condições a seguir forem atendidas:

- O IBM MQ C Client é V8 ou mais recente
- O Java/JMS Client é V9.1.1 ou mais recente e a instalação Java suporta a classe `javax.net.ssl.SNIHostName`
- O .NET Client está no modo não gerenciado

O SNI será configurado como o nome do host se um nome do host for fornecido como o nome da conexão e as condições a seguir forem atendidas:

- O .NET Client está no modo gerenciado
- O cliente AMQP ou XR é usado
- Os Clientes Java/JMS são usados com **AllowOutboundSNI** configurado como NO

O SNI não está configurado e está em branco nas seguintes condições:

- O IBM MQ C Client é V7.5 ou anterior..
- IBM MQ C Client é usado com **AllowOutboundSNI** configurado como NO.
- Os Clientes Java/JMS são usados com uma instalação Java que não suporta a classe `javax.net.ssl.SNIHostName`

Exemplo

Rotas do OpenShift baseadas em nome de host: para aplicativos clientes que configuram a SNI para o nome do host

Os seguintes gráficos do Helm criam automaticamente uma rota do OpenShift baseada no nome de host para conectar um aplicativo a um gerenciador de filas do IBM MQ. Os aplicativos clientes que configuram a SNI para o nome do host podem usar essa rota do OpenShift.

- `ibm-mqadvanced-server-dev`
- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod` no IBM Cloud Pak for Integration.

Se você não estiver usando esses gráficos e precisar criar a sua própria rota do OpenShift baseada no nome de host, será possível aplicar o `yaml` a seguir em seu cluster:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: <provide a unique name for the Route>
  namespace: <namespace of your MQ deployment>
spec:
  to:
    kind: Service
    name: <name of the Kubernetes Service for your MQ deployment (for example "<Helm Release>-ibm-mq")>
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Rotas do OpenShift baseadas em canais do MQ: para aplicativos clientes que configuram a SNI para o canal do MQ

Aplicativos clientes que configuram a SNI para o canal do MQ requerem que uma nova rota do OpenShift seja criada para cada canal com o qual você deseja se conectar. Também é preciso usar nomes de canais exclusivos em todo o cluster do Red Hat OpenShift para permitir o roteamento para o gerenciador de filas correto.

Para determinar o nome do host necessário para cada uma de suas novas rotas do OpenShift, é necessário mapear cada nome de canal para um endereço de SNI conforme documentado aqui: <https://www.ibm.com/support/pages/ibm-websphere-mq-how-does-mq-provide-multiple-certificates-certlbl-capability>

Deve-se, então, criar uma rota do OpenShift (para cada canal), aplicando o `yam1` a seguir 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 "<Helm Release>-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 "<Helm Release>-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 roteador do OpenShift Container Platform (OCP), normalmente a 443.

Tarefas relacionadas

“Implementando um gerenciador de filas usando a CLI do Helm” na página 25

Use o Helm para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform. Esta tarefa deve ser concluída por um administrador do projeto.

“Conectando-se ao IBM MQ Console implementado em um cluster OpenShift” na página 32

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

V 9.1.4

Linux

MQ Adv.

CD

Conectando-se ao IBM MQ

Console implementado em um cluster OpenShift

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

Sobre esta tarefa

Se você estiver usando o IBM MQ Operator, a URL IBM MQ Console poderá ser localizada na página de detalhes QueueManager no console da web do OpenShift ou no IBM Cloud Pak for Integration Platform Navigator. Como alternativa, ela pode ser localizada por meio da CLI do OpenShift executando o comando a seguir:

```
oc get queuemanager <QueueManager Name> -n <namespace of your MQ deployment> --output
jsonpath='{.status.adminUiUrl}'
```


Exemplo

Os seguintes gráficos do Helm criam automaticamente uma rota do OpenShift para acessar o IBM MQ Console

- `ibm-mqadvanced-server-dev`
- `ibm-mqadvanced-server-integration-prod` no IBM Cloud Pak for Integration.

É possível obter o nome do host da rota do OpenShift executando o comando a seguir:

```
oc get route <Route Name (for example "<Helm Release>-ibm-mq-web")>
-n <namespace of your MQ deployment> --output jsonpath='{.spec.host}'
```

É possível acessar o IBM MQ Console usando a URL a seguir:

```
https://<Route Hostname>/ibmmq/console
```

Tarefas relacionadas

[“Implementando um gerenciador de filas usando a CLI do Helm”](#) na página 25

Use o Helm para implementar um gerenciador de filas em um cluster do Red Hat OpenShift Container Platform. Esta tarefa deve ser concluída por um administrador do projeto.

[“Conectando-se a um gerenciador de filas implementado em um cluster do OpenShift”](#) na página 30

Um conjunto de exemplos de configuração para a conexão com um gerenciador de filas implementado em um cluster do Red Hat OpenShift.

Linux

MQ Adv.

CD

Fazendo backup e restaurando a configuração do gerenciador de filas usando a CLI do 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, se você estiver usando o Operator, será 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
```

Por exemplo, se você estiver usando o Helm, será possível executar o comando a seguir, em que `release_name` é o nome de sua liberação do Helm .

```
oc get pods --selector release=release_name
```

- b) Execute o comando **dmpmqcfig** no pod, direcionando a saída para um arquivo na máquina local.

dmpmqc:fg gera a configuração do MQSC do gerenciador de filas.

```
oc exec -it pod_name -- dmpmqc:fg > 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, se você estiver usando o Operator, será 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
```

Por exemplo, se você estiver usando o Helm, será possível executar o comando a seguir, em que `release_name` é o nome de sua liberação do Helm .

```
oc get pods --selector release=release_name
```

- b) Execute o comando **runmqsc** no pod, direcionando no conteúdo do arquivo `backup.mqsc`.

```
oc exec -i pod_name -- runmqsc < backup.mqsc
```

Construindo seu próprio contêiner do IBM MQ

Desenvolva um contêiner autoconstruído, anteriormente referido como a "Imagem de contêiner do Docker". Essa é a solução de contêiner mais flexível, mas requer que você tenha um bom conhecimento técnico sobre configuração de contêineres e que você seja o "proprietário" do contêiner resultante.


Antes de começar

Antes de desenvolver o seu próprio contêiner, considere se é possível, em vez disso, usar um dos contêineres predefinidos fornecidos pela IBM. Consulte [IBM MQ em contêineres](#)

Sobre esta tarefa

Ao empacotar o IBM MQ como uma imagem de contêiner, as mudanças em seu aplicativo podem ser implementadas para testar e preparar sistemas de forma rápida e fácil. Isso pode ser um grande benefício para a entrega contínua em sua empresa.

Procedimento

- Para obter informações sobre como construir uma imagem de contêiner do IBM MQ usando o Docker, consulte os subtópicos a seguir:
 -  [“Suporte para a construção de suas próprias imagens de contêiner e gráficos do IBM MQ” na página 8](#)
 - [“Planejando a sua própria imagem do gerenciador de filas do IBM MQ usando um contêiner” na página 35](#)
 - [“Construindo uma imagem do gerenciador de filas do IBM MQ de amostra usando o Docker” na página 35](#)
 - [“Executando aplicativos de ligação local em contêineres separados” na página 38](#)

Conceitos relacionados

[IBM MQ em contêineres](#)

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.

V9.1.0 Para resolver isso, quando o seu contêiner for iniciado, será possível usar o comando `crtmqdir` para preencher /var/mqm quando ele for executado pela primeira vez

Construindo uma imagem do gerenciador de filas do IBM MQ de amostra usando o Docker

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 36
 - [“Construindo uma imagem do gerenciador de filas do IBM MQ de amostra configurada”](#) na página 36

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](#).

Resultados

Agora você tem uma imagem de contêiner base com o IBM MQ instalada.

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, crie sua própria camada de imagem de contêiner usando a imagem genérica como um pai.

Antes de começar

Para uma imagem IBM MQ 9.1, não é possível configurar o acesso seguro usando o Red Hat OpenShift Container Platform "restricted" Security Context Constraint (SCC). O SCC "restrito" usa IDs de usuário aleatórios e evita a escalação de privilégios mudando para um usuário diferente. O instalador baseado em RPM IBM MQ 9.1 conta com um usuário e um grupo mqm e também usa bits setuid em programas executáveis.

Essa restrição é removida no IBM MQ 9.2

Procedimento

1. Crie um novo diretório e inclua um arquivo denominado `config.mqsc`, com o seguinte conteúdo:

```
DEFINE CHANNEL(PASSWORD.SVRCONN) CHLTYPE(SVRCONN)
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(BLOCKUSER) USERLIST('nobody') +
DESCR('Allow privileged users on this channel')
SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) DESCR('BackStop rule')
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(CHANNEL) CHCKCLNT(REQUIRED)
ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) ADOPTCTX(YES)
REFRESH SECURITY TYPE(CONNAUTH)
```

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
RUN useradd johndoe -G mqm && \
    echo johndoe:passw0rd | chpasswd
COPY config.mqsc /etc/mqm/
```

em que:

- johndoe é o ID do usuário que você deseja incluir
- passw0rd é a senha original

3. Construa a sua imagem de contêiner customizada usando o comando a seguir:

```
sudo 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.

O comando **RUN** inclui um usuário denominado `johndoe` com a senha `passwd` e o comando **COPY** inclui o arquivo `config.mqsc` em um local específico conhecido pela imagem pai.

Nota: No Red Hat Enterprise Linux (RHEL), use o comando **docker** (RHEL V7) ou **podman** (RHEL V7 ou RHEL V8). No caso de **podman**, o **sudo** não é necessário no início do comando.

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:

```
sudo docker run \  
  --env LICENSE=accept \  
  --env MQ_QMGR_NAME=QM1 \  
  --volume /var/example:/var/mqm \  
  --publish 1414:1414 \  
  --detach \  
  mymq
```

em que o:

Primeiro parâmetro env

Transmite uma variável de ambiente no contêiner, que reconhece sua aceitação da licença para IBM WebSphere MQ. Também é possível configurar a variável `LICENSE` para visualizar a licença.

Consulte [IBM MQ informações de licença](#) para obter detalhes adicionais sobre as licenças do IBM MQ.

Segundo parâmetro env

Configura o nome do gerenciador de filas que você está usando.

Parâmetro Volume

Diz ao contêiner que tudo o que MQ escreve para `/var/mqm` deve realmente ser gravado em `/var/example` no host.

Esta opção significa que é possível facilmente excluir o contêiner posteriormente e ainda manter quaisquer dados persistentes. Esta opção também torna mais fácil para visualizar arquivos de log.

Parâmetro Publish

Mapeia as portas no sistema host para portas no contêiner. O contêiner é executado por padrão com seu próprio endereço IP interno, o que significa que é necessário mapear especificamente quaisquer portas que você deseja expor.

Neste exemplo, isso significa a porta de mapeamento 1414 no host para a porta 1414 no contêiner.

Parâmetro Detach

Executa o contêiner em segundo plano.

Resultados

Você construiu uma imagem de contêiner configurada e pode visualizar os contêineres em execução usando o comando **ps** do Docker. 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

- Se seu contêiner não for mostrado ao usar o comando docker **ps**, o contêiner poderá ter falhado. É possível ver contêineres com falha usando o comando docker **ps -a**.
- Quando você usa o comando **ps -a** do docker, o ID do contêiner é exibido. Esse ID também era impresso quando você emitia o comando **run** do docker.
- É possível visualizar os logs de um contêiner usando o comando **logs \${CONTAINER_ID}** do docker.
- É possível configurar o número máximo de arquivos abertos usando o comando **sysctl fs.file-max=524288..**

V 9.1.0

Executando aplicativos de ligação local em contêineres separados

Com o compartilhamento do namespace do processo entre contêineres no Docker, é possível executar aplicativos que requeiram uma conexão de ligação local com o IBM MQ em contêineres separados do gerenciador de filas do IBM MQ.

Sobre esta tarefa

Essa funcionalidade é suportada no IBM MQ 9.0.3 e gerenciadores de filas mais recentes.

Deve-se aderir às restrições a seguir:

- Deve-se compartilhar o namespace de PID de contêineres usando o argumento **--pid**.
- Deve-se compartilhar o namespace de IPC de contêineres usando o argumento **--ipc**.
- Para isso:
 1. Compartilhe o namespace de UTS de contêineres com o host usando o argumento **--uts** ou
 2. Assegure que os contêineres tenham o mesmo nome do host usando o argumento **-h** ou **--hostname**.
- Deve-se montar o diretório de dados do IBM MQ em um volume que esteja disponível para todos os contêineres no diretório do **/var/mqm**.

É possível experimentar essa funcionalidade concluindo as etapas a seguir em um sistema Linux que já tenha o Docker instalado.

O exemplo a seguir usa a imagem de contêiner do IBM MQ de amostra. É possível localizar detalhes dessa imagem no [Github](#).

Procedimento

1. Crie um diretório temporário para agir como seu volume, emitindo o comando a seguir:

```
mkdir /tmp/dockerVolume
```

2. Crie um gerenciador de filas (QM1) em um contêiner, com o nome **sharedNamespace**, emitindo o comando a seguir:

```
docker run -d -e LICENSE=accept -e MQ_QMGR_NAME=QM1 --volume /tmp/dockerVol:/mnt/mqm
--uts host --name sharedNamespace ibmcom/mq
```

3. Inicie um segundo contêiner chamado `secondaryContainer`, com base em `ibmcom/mq`, mas não crie um gerenciador de filas, emitindo o comando a seguir:

```
docker run --entrypoint /bin/bash --volumes-from sharedNamespace --pid
container:sharedNamespace --ipc container:sharedNamespace --uts host --name
secondaryContainer -it --detach ibmcom/mq
```

4. Execute o comando **dspm** no segundo contêiner, para ver o status para ambos os gerenciadores de filas, emitindo o comando a seguir:

```
docker exec secondaryContainer dspmq
```

5. Execute o comando a seguir para processar comandos MQSC com relação ao gerenciador de filas em execução no outro contêiner:

```
docker exec -it secondaryContainer runmqsc QM1
```

Resultados

Agora você tem aplicativos locais em execução em contêineres separados e pode executar com sucesso comandos como **dspm**, **amqspu**, **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 39 para obter mais informações.

V 9.1.0 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 38 para obter uma lista de restrições que se deve seguir.

Se seu aplicativo não atender todas as restrições listadas, você poderá encontrar problemas em que o contêiner inicia, mas a funcionalidade esperada não funciona.

A lista a seguir descreve algumas causas comuns e o comportamento que você provavelmente verá se tiver esquecido de atender uma das restrições.

- Se você se esquecer de compartilhar o namespace (UTS/PID/IPC) ou o nome do host dos contêineres e montar o volume, seu contêiner poderá ver o gerenciador de filas, mas não interagirá com ele.
 - Para comandos **dspm**, você vê o seguinte:

```
docker exec container dspmq
QMNAME(QM1)                STATUS(Status not available)
```

- Para comandos **runmqsc** ou outros comandos que tentam se conectar ao gerenciador de filas, você provavelmente receberá uma mensagem de erro AMQ8146:

```
docker exec -it container runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 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, **dspm** não é capaz de ver seu gerenciador de filas e em seu lugar retorna uma resposta em branco:

```
docker exec container dspmq
```

- Se você compartilha todos os namespaces necessários mas você não monta um volume compartilhado para o diretório `/var/mqm` e você não tem um caminho de dados IBM MQ válido (ou nenhum caminho de dados IBM MQ), então você verá vários erros, pois o caminho de dados é um componente chave de uma instalação IBM MQ Sem o caminho de dados, o IBM MQ não pode operar.

Se você executar qualquer um dos comandos a seguir e vir respostas semelhantes às mostradas nesses exemplos, será necessário verificar se você montou o diretório ou criou um diretório de dados do IBM MQ:

```
docker exec container dspmq
'No such file or directory' from /var/mqm/mqs.ini
AMQ6090: IBM MQ was unable to display an error message FFFFFFFF.
AMQffff

docker exec container dspmqver
AMQ7047: An unexpected error was encountered by a command. Reason code is 0.

docker exec container mqrc
<file path>/mqrc.c[1152]
lpiObtainQMDetails --> 545261715

docker exec container crtmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container strmqm QM1
AMQ6239: Permission denied attempting to access filesystem location '/var/mqm'.
AMQ7002: An error occurred manipulating a file.

docker exec container endmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container dltmqm QM1
AMQ7002: An error occurred manipulating a file.

docker exec container strmqweb
<file path>/mqrc.c[1152]
lpiObtainQMDetails --> 545261715
```

[V 9.1.5](#) [Linux](#) [MQ Adv.](#) [CD](#) **Referência de API para o IBM MQ Operator**

O IBM MQ fornece um Operator do Kubernetes, que fornece integração nativa com o OpenShift Container Platform.

[V 9.1.5](#) [Linux](#) [MQ Adv.](#) [CD](#) **Referência da API para mq.ibm.com/v1beta1**

A API `v1beta1` pode ser usada para criar e gerenciar recursos do QueueManager.

[V 9.1.5](#) [Linux](#) [MQ Adv.](#) [CD](#) **Referência de licenciamento para mq.ibm.com/v1beta1**

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
L-RJON-BN7PN3	Production ou NonProduction	IBM Cloud Pak for Integration 2020.2
L-RJON-BPHL2Y		IBM Cloud Pak for Integration Edição Limitada 2020.2 ..
L-APIG-BJAKBF	Production ou Development	IBM MQ Advanced 9.1.5
L-APIG-BM7GDH	Development	IBM MQ Advanced for Developers 9.1.5

Observe que a *versão* da licença é especificada, que nem sempre é a mesma da versão do IBM MQ.

V 9.1.5 **Linux** **MQ Adv.** **CD** **Referência de API para QueueManager (mq.ibm.com/v1beta1)**

QueueManager

Um QueueManager é um Servidor do IBM MQ que fornece serviços de enfileiramento e de publicar/ assinar para aplicativos.

Campo	Descrição
Sequência de <code>apiVersion</code>	<code>APIVersion</code> define o esquema com versão dessa representação de um objeto. Os servidores devem converter os esquemas reconhecidos para o valor interno mais recente e podem rejeitar valores não reconhecidos. Mais informações: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources .
Sequência de <code>kind</code>	<code>Kind</code> é um valor de sequência que representa o recurso REST que este objeto representa. Os servidores podem inferir isso do terminal para o qual o cliente envia solicitações. Não é possível ser atualizado. In CamelCase. Mais informações: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds .
<code>metadata</code>	
<code>spec</code> “Especificação QueueManager” na página 45	O estado desejado do QueueManager.
<code>status</code> “QueueManagerStatus” na página 46	O estado observado do QueueManager.

Disponibilidade

Configurações de disponibilidade para o Gerenciador de Filas, como usar ou não um par ativo/em espera.

Aparece em:

- “[QueueManagerConfig](#)” na página 43

Campo	Descrição
Sequência de type	O tipo de disponibilidade para usar. Use "SingleInstance" para um único Pod, que será reiniciado automaticamente (em alguns casos) pelo Kubernetes. Use "MultiInstance" para um par de Pods, um dos quais é o Gerenciador de Filas "ativo" e o outro é uma espera. Consulte Alta disponibilidade para IBM MQ em contêineres na versão mais recente do IBM MQ..

Licença

Configurações que controlam sua aceitação da licença e quais as métricas de licença a serem usadas.

Aparece em:

- [“Especificação QueueManager” na página 45](#)

Campo	Descrição
Sequência de use	Configuração que controla como o software será usado, na qual a licença suporta diversos usos. Consulte https://ibm.biz/BdqvCF para obter os valores válidos.
Booleano accept	Indica se você aceita ou não a licença associada a este software (obrigatório).
Sequência de license	O identificador da licença que você está aceitando. Esse deve ser o identificador de licença correto para a versão do MQ que você está usando. Consulte https://ibm.biz/BdqvCF para obter os valores válidos.
Sequência de metric	Configuração que especifica qual métrica de licença usar. Por exemplo, "ProcessorValueUnit", "VirtualProcessorCore" ou "ManagedVirtualServer".

Limites

QueueManagerResourceList define configurações de CPU e memória.

Aparece em:

- [“Recursos” na página 49](#)

Campo	Descrição
cpu	
memory	

LocalObjectReferência

LocalObjectReference contém informações suficientes para deixar você localizar o objeto referenciado dentro do mesmo namespace.

Aparece em:

- [“Especificação QueueManager” na página 45](#)

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?.

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:

- “Especificação QueueManager” na página 45

Campo	Descrição
matriz do keys “PKISource” na página 43	Chaves privadas a serem adicionadas ao repositório de chaves do Queue Manager.
matriz do trust “PKISource” na página 43	Certificados a serem adicionados ao repositório de chaves do Queue Manager.

PKISource

O PKISource define uma origem de informações de Infraestrutura de chave pública, como chaves ou certificados.

Aparece em:

- “PKI” na página 43

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” na página 49	Forneça uma chave usando um Segredo do Kubernetes.

QueueManagerConfig

QueueManagerConfig define as configurações para o contêiner do Queue Manager e o Queue Manager subjacente.

Aparece em:

- “Especificação QueueManager” na página 45

Campo	Descrição
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.
metrics “Métricas do QueueManager” na página 44	Configurações para métricas de estilo Prometheus.
readinessProbe “QueueManagerReadinessProbe” na página 45	Configurações que controlam a análise de prontidão.
resources “Recursos” na página 49	Configurações que controlam os requisitos de recursos.
storage “QueueManagerArmazenamento” na página 48	Configurações de armazenamento para controlar o uso de volumes persistentes e classes de armazenamento pelo Gerenciador de Filas.

Campo	Descrição
availability “Disponibilidade” na página 41	Configurações de disponibilidade para o Gerenciador de Filas, como usar ou não um par ativo/em espera.
Sequência de imagePullPolicy	Configuração que controla quando o kubelet tenta extrair a imagem especificada.
livenessProbe “QueueManagerLivenessProbe” na página 44	Configurações que controlam a análise de vivacidade.
Booleano debug	Indica se mensagens de depuração do código específico do contêiner devem ou não ser registradas no log do contêiner.
Sequência de image	A imagem de contêiner que será usada.
Sequência de name	Nome do Gerenciador de Filas do MQ subjacente, se diferente de metadata.name. Use este campo se você quiser um nome do Gerenciador de Filas que não esteja em conformidade com as regras do Kubernetes para nomes (por exemplo, um nome que inclua letras maiúsculas).

QueueManagerLivenessProbe

Configurações que controlam a análise de vivacidade.

Aparece em:

- “QueueManagerConfig” na página 43

Campo	Descrição
Número inteiro failureThreshold	As falhas mínimas consecutivas para que a análise seja considerada com falha após ter sido bem-sucedida.
Número inteiro initialDelaySeconds	Número de segundos após o contêiner ter iniciado antes que as análises de vivacidade sejam iniciadas. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro periodSeconds	Com que frequência (em segundos) realizar a análise.
Número inteiro successThreshold	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida depois de ter falhado.
Número inteiro timeoutSeconds	Número de segundos após os quais a análise atingirá o tempo limite. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Métricas do QueueManager

Configurações para métricas de estilo Prometheus.

Aparece em:

- “QueueManagerConfig” na página 43

Campo	Descrição
Booleano enabled	Indica se um terminal de métricas compatível com o Prometheus deve ou não ser ativado.

QueueManagerOptionalVolume

Detalhes de persistentVolume para logs de recuperação do MQ. Necessário ao usar o Gerenciador de Filas de várias instâncias.

Aparece em:

- [“QueueManagerArmazenamento” na página 48](#)

Campo	Descrição
Sequência de class	Classe de armazenamento a usar para este volume. Válido apenas se "type" for "persistent-claim".
Booleano enabled	Se esse volume deve ou não ser ativado como um volume separado ou ser colocado no volume "queueManager" padrão.
Sequência de size	Tamanho do PersistentVolume para passar para Kubernetes. Válido apenas se "type" for "persistent-claim".
Sequência de sizeLimit	Limite de tamanho ao usar um volume "efêmero". Os arquivos ainda são gravados em um diretório temporário, portanto, é possível usar essa opção para limitar o tamanho. Válido apenas se type for "efêmero".
Sequência de type	Tipo de volume a ser usado. Escolha ephemeral para criar um volume "emptyDir" não persistente ou persistent-claim para usar um volume persistente.

QueueManagerReadinessProbe

Configurações que controlam a análise de prontidão.

Aparece em:

- [“QueueManagerConfig” na página 43](#)

Campo	Descrição
Número inteiro failureThreshold	As falhas mínimas consecutivas para que a análise seja considerada com falha após ter sido bem-sucedida.
Número inteiro initialDelaySeconds	Número de segundos após o contêiner ter iniciado antes que as análises de vivacidade sejam iniciadas. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro periodSeconds	Com que frequência (em segundos) realizar a análise.
Número inteiro successThreshold	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida depois de ter falhado.
Número inteiro timeoutSeconds	Número de segundos após os quais a análise atingirá o tempo limite. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Especificação QueueManager

O estado desejado do QueueManager.

Aparece em:

- [“QueueManager” na página 41](#)

Campo	Descrição
<code>license</code> “Licença” na página 42	Configurações que controlam sua aceitação da licença e quais as métricas de licença a serem usadas.
<code>pki</code> “PKI” na página 43	Configurações de Infraestrutura de chave pública, para definição de chaves e certificados para uso com a Segurança da Camada de Transporte (TLS) ou o MQ Advanced Message Security (AMS).
<code>queueManager</code> “QueueManagerConfig” na página 43	QueueManagerConfig define as configurações para o contêiner do Queue Manager e o Queue Manager subjacente.
<code>securityContext</code> “SecurityContext” na página 49	Configurações de segurança a serem adicionadas ao securityContext do pod do gerenciador de filas.
<code>tracing</code> “TracingConfig” na página 51	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 MQ versão 9.1.5.0, usando a segunda revisão da imagem de contêiner. Correções específicas do contêiner são frequentemente aplicadas em revisões, como correções para a imagem base.
<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>matriz do imagePullSecrets</code> “LocalObjectReferência” na página 42	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>template</code> “Template” na página 50	Modelagem avançada para recursos do Kubernetes. O modelo permite que os usuários substituam como o IBM MQ gera os recursos do Kubernetes subjacentes, como StatefulSet, Pods e Serviços. Destina-se somente a usuários avançados, pois terá o potencial de interromper a operação normal do MQ se for usado incorretamente. Quaisquer valores especificados em qualquer outro lugar no recurso QueueManager serão substituídos por configurações no modelo.
Número inteiro <code>terminationGracePeriod</code> Seconds	Duração opcional em segundos que o Pod precisa finalizar normalmente. O valor deve ser um número inteiro não negativo. O valor zero indica exclusão imediatamente. O horário alvo no qual o término do gerenciador de filas é tentado, escalando as fases de desconexão do aplicativo. Se necessário, tarefas de manutenção essenciais do gerenciador de filas serão interrompidas.
<code>web</code> “Configuração do WebServer” na página 52	Configurações para o servidor da web do MQ.

QueueManagerStatus

O estado observado do QueueManager.

Aparece em:

- “QueueManager” na página 41

Campo	Descrição
matriz do endpoints “ QueueManagerStatusEndpoint ” na página 47	Informações sobre os terminais que este Gerenciador de Filas está expondo, como terminais de API ou IU.
Sequência de name	O nome do Gerenciador de Filas.
versions “ QueueManagerStatusVersion ” na página 47	Versão do MQ que está sendo usada e outras versões disponíveis no IBM Entitled Registry.
Sequência de adminUiUrl	URL para a IU do Administrador.
matriz do conditions “ QueueManagerStatusCondition ” na página 47	As condições representam as últimas observações disponíveis do estado do Gerenciador de Filas.

QueueManagerStatusCondition

QueueManagerStatusCondition define as condições do Gerenciador de Filas.

Aparece em:

- “[QueueManagerStatus](#)” na página 46

Campo	Descrição
Sequência de message	Mensagem legível por humanos indicando detalhes sobre a última transição.
Sequência de type	Tipo de condição.
Sequência de lastTransitionTime	Última vez que a condição transitava de um status para outro.

QueueManagerStatusEndpoint

QueueManagerStatusEndpoint define os terminais para o QueueManager.

Aparece em:

- “[QueueManagerStatus](#)” na página 46

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.

QueueManagerStatusVersion

Versão do MQ que está sendo usada e outras versões disponíveis no IBM Entitled Registry.

Aparece em:

- “[QueueManagerStatus](#)” na página 46

Campo	Descrição
available “ QueueManagerStatusVersion Disponível” na página 48	Outras versões do MQ disponíveis no IBM Entitled Registry.

Campo	Descrição
Sequência de reconciled	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.

QueueManagerStatusVersionDisponível

Outras versões do MQ disponíveis no IBM Entitled Registry.

Aparece em:

- [“QueueManagerStatusVersion” na página 47](#)

Campo	Descrição
Matriz channels	Canais que estão disponíveis para atualizar automaticamente a versão do MQ.
matriz do versions “Versões” na página 52	Versões específicas do MQ que estão disponíveis.

QueueManagerArmazenamento

Configurações de armazenamento para controlar o uso de volumes persistentes e classes de armazenamento pelo Gerenciador de Filas.

Aparece em:

- [“QueueManagerConfig” na página 43](#)

Campo	Descrição
persistedData “QueueManagerOptionalVolume” na página 45	Detalhes de persistentVolume para dados persistidos do MQ, incluindo configuração, filas e mensagens. Necessário ao usar o Gerenciador de Filas de várias instâncias.
queueManager “Volume QueueManager” na página 48	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” na página 45	Detalhes de persistentVolume para logs de recuperação do MQ. Necessário ao usar o Gerenciador de Filas de várias instâncias.

Volume 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:

- [“QueueManagerArmazenamento” na página 48](#)

Campo	Descrição
Sequência de class	Classe de armazenamento a usar para este volume. Válido apenas se "type" for "persistent-claim".
Sequência de size	Tamanho do PersistentVolume para passar para Kubernetes. Válido apenas se "type" for "persistent-claim".
Sequência de sizeLimit	Limite de tamanho ao usar um volume "efêmero". Os arquivos ainda são gravados em um diretório temporário, portanto, é possível usar essa opção para limitar o tamanho. Válido apenas se type for "efêmero".

Campo	Descrição
Sequência de type	Tipo de volume a ser usado. Escolha ephemeral para criar um volume "emptyDir" não persistente ou persistent-claim para usar um volume persistente.

Solicitações

QueueManagerResourceList define configurações de CPU e memória.

Aparece em:

- [“Recursos” na página 49](#)

Campo	Descrição
memory	
cpu	

Recursos

Configurações que controlam os requisitos de recursos.

Aparece em:

- [“QueueManagerConfig” na página 43](#)

Campo	Descrição
limits “Limites” na página 42	QueueManagerResourceList define configurações de CPU e memória.
requests “Solicitações” na página 49	QueueManagerResourceList define configurações de CPU e memória.

Segredo

Forneça uma chave usando um Segredo do Kubernetes.

Aparece em:

- [“PKISource” na página 43](#)

Campo	Descrição
Matriz 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.

SecurityContext

Configurações de segurança a serem adicionadas ao securityContext do pod do gerenciador de filas.

Aparece em:

- [“Especificação QueueManager” na página 45](#)

Campo	Descrição
Matriz supplementalGroups	Uma lista de grupos aplicados ao primeiro processo executado em cada contêiner, além do GID principal do contêiner. Se não especificado, nenhum grupo será incluído em nenhum contêiner.

Campo	Descrição
Número inteiro fsGroup	Um grupo suplementar especial que se aplica a todos os contêineres em um pod. Alguns tipos de volume permitem que o Kubelet altere a propriedade desse volume para que seja de propriedade do pod: 1. O GOD proprietário será o FSGroup 2. O bit setgid está definido (novos arquivos criados no volume serão de propriedade do FSGroup) 3. Os bits de permissão são OR'd com rw-rw---- Se não estiver definido, o Kubelet não modificará a propriedade e as permissões de nenhum volume.
Booleano initVolumeAsRoot	Afeta o securityContext usado pelo contêiner que inicializa o PersistentVolume. Configure isso como "true" se estiver usando um provedor de armazenamento que requer que você seja o usuário raiz para acessar volumes recém-provisionados. Configurar isso como "true" afeta qual objeto SCC (Security Context Constraints) você pode usar e o Gerenciador de Filas pode falhar ao iniciar se você não estiver autorizado a usar um SCC que permita o usuário raiz. Para obter mais informações, consulte https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html .

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:

- “Especificação QueueManager” na página 45

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 .

TracingAgent

No Cloud Pak for Integration apenas, é possível definir configurações para o Agente de rastreamento opcional.

Aparece em:

- “TracingConfig” na página 51

Campo	Descrição
Sequência de imagem	A imagem de contêiner que será usada.
Sequência de imagePullPolicy	Configuração que controla quando o kubelet tenta extrair a imagem especificada.
livenessProbe “TracingProbe” na página 51	Configurações que controlam a análise de vivacidade.
readinessProbe “TracingProbe” na página 51	Configurações que controlam a análise de prontidão.

TracingCollector

No Cloud Pak for Integration apenas, é possível definir configurações para o Coletor de rastreamento opcional.

Aparece em:

- [“TracingConfig” na página 51](#)

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.
livenessProbe “TracingProbe” na página 51	Configurações que controlam a análise de vivacidade.
readinessProbe “TracingProbe” na página 51	Configurações que controlam a análise de prontidão.

TracingConfig

Configurações para integração de rastreamento com o Cloud Pak for Integration Operations Dashboard.

Aparece em:

- [“Especificação QueueManager” na página 45](#)

Campo	Descrição
agent “TracingAgent” na página 50	No Cloud Pak for Integration apenas, é possível definir configurações para o Agente de rastreamento opcional.
collector “TracingCollector” na página 50	No Cloud Pak for Integration apenas, é possível definir configurações para o Coletor de rastreamento opcional.
Booleano enabled	Indica se a integração com o Cloud Pak for Integration Operations Dashboard deve ou não ser ativada por meio do rastreamento.
Sequência de namespace	Namespace no qual o Cloud Pak for Integration Operations Dashboard está instalado.

TracingProbe

Configurações que controlam a análise de prontidão.

Aparece em:

- [“TracingCollector” na página 50](#)

Campo	Descrição
Número inteiro failureThreshold	As falhas mínimas consecutivas para que a análise seja considerada com falha após ter sido bem-sucedida.
Número inteiro initialDelaySeconds	Número de segundos após o contêiner ter iniciado antes que as análises de vivacidade sejam iniciadas. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Número inteiro periodSeconds	Com que frequência (em segundos) realizar a análise.
Número inteiro successThreshold	Sucessos mínimos consecutivos para que a análise seja considerada bem-sucedida depois de ter falhado.

Campo	Descrição
Número inteiro timeoutSeconds	Número de segundos após os quais a análise atingirá o tempo limite. Mais informações: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Versões

QueueManagerStatusVersion define uma versão do MQ.

Aparece em:

- “QueueManagerStatusVersionDisponível” na página 48

Campo	Descrição
Sequência de name	Versão "name" para esta versão do QueueManager. Estes são valores válidos para o campo spec.version.

Configuração do WebServer

Configurações para o servidor da web do MQ.

Aparece em:

- “Especificação QueueManager” na página 45

Campo	Descrição
Booleano enabled	Indica se o servidor da web deve ou não ser ativado.

Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos Estados Unidos.

É possível que a IBM não ofereça os produtos, serviços ou recursos discutidos nesta publicação em outros países. Consulte seu representante local do IBM para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a um IBM produto, programa ou serviço não se destina a estado ou significa que apenas esse produto IBM, programas ou serviços possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado em substituição. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou aplicativos de patentes pendentes relativas aos assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum sobre tais patentes. É possível enviar pedidos de licença, por escrito, para:

Relações Comerciais e Industriais da IBM
Av. Pasteur, 138-146
Botafogo
Rio, RJ 10504-1785
U.S.A.

Para pedidos de licença relacionados a informações de DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos de licença, por escrito, para:

licença de propriedade intelectual
IBM World Trade Asia Corporation Licensing
IBM Japan, Ltd.
Minato-ku
Tóquio 103-8510, Japão

O parágrafo a seguir não se aplica a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS A ELAS NÃO SE LIMITANDO, AS GARANTIAS IMPLÍCITAS DE NÃO INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO PROPÓSITO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, essa disposição pode não se aplicar ao Cliente.

Essas informações podem conter imprecisões técnicas ou erros tipográficos. Periodicamente, são feitas nas informações aqui contidas; essas alterações serão incorporadas em futuras edições desta publicação. IBM pode aperfeiçoar e/ou alterar no produto(s) e/ou programa(s) descritos nesta publicação a qualquer momento sem aviso prévio.

Referências nestas informações a websites não IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses websites. Os materiais contidos nesses websites não fazem parte dos materiais desse produto IBM e a utilização desses websites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Av. Pasteur, 138-146
Av. Pasteur, 138-146

Botafogo
Rio de Janeiro, RJ
U.S.A.

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriadas, incluindo em alguns casos o pagamento de uma taxa.

O programa licenciado descrito nesta publicação e todo o material licenciado disponível para ele são fornecidos pela IBM sob os termos do IBM Customer Agreement, IBM Contrato de Licença do Programa Internacional ou qualquer contrato equivalente entre as partes.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar significativamente. Algumas medidas podem ter sido tomadas em sistemas em nível de desenvolvimento e não há garantia de que estas medidas serão iguais em sistemas geralmente disponíveis. Além disto, algumas medidas podem ter sido estimadas através de extrapolação. Os resultados reais podem variar. usuários deste documento devem verificar os dados aplicáveis para seu ambiente específico.

As informações relativas a produtos não IBM foram obtidas junto aos fornecedores dos respectivos produtos, de seus anúncios publicados ou de outras fontes disponíveis publicamente. A IBM não testou estes produtos e não pode confirmar a precisão de seu desempenho, compatibilidade nem qualquer outra reivindicação relacionada a produtos não IBM. Dúvidas sobre os recursos de produtos não IBM devem ser encaminhadas diretamente a seus fornecedores.

Todas as declarações relacionadas aos objetivos e intenções futuras da IBM estão sujeitas a alterações ou cancelamento sem aviso prévio e representam somente metas e objetivos.

Essas informações contêm exemplos de dados e relatórios utilizados em operações diárias de negócios. Para ilustrá-los da forma mais completa possível, os exemplos incluem nomes de indivíduos, empresas, marcas e produtos. Todos estes nomes são fictícios e qualquer semelhança com os nomes e endereços utilizados por uma empresa real é mera coincidência.

LICENÇA DE COPYRIGHT :

Estas informações contêm programas de aplicativos de amostra na linguagem fonte, ilustrando as técnicas de programação em diversas plataformas operacionais. O Cliente pode copiar, modificar e distribuir estes programas de amostra sem a necessidade de pagar à IBM, com objetivos de desenvolvimento, uso, marketing ou distribuição de programas aplicativos em conformidade com a interface de programação de aplicativo para a plataforma operacional para a qual os programas de amostra são criados. Esses exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou implicar a confiabilidade, manutenção ou função destes programas.

Se estiver visualizando estas informações em formato eletrônico, as fotografias e ilustrações coloridas poderão não aparecer.

Informações sobre a Interface de Programação

As informações da interface de programação, se fornecidas, 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 comerciais da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Este produto inclui software desenvolvido pelo Projeto Eclipse (<http://www.eclipse.org/>).

Java e todas as marcas comerciais e logotipos baseados em Java são marcas comerciais ou marcas registradas da Oracle e/ou de suas afiliadas.



Part Number:

(1P) P/N: