

9.1

*IBM MQ en contenedores*

**IBM**

**Nota**

Antes de utilizar esta información y el producto al que da soporte, lea la información en [“Avisos” en la página 53](#).

Esta edición se aplica a la versión 9 release 1 de IBM® MQ y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

Cuando envía información a IBM, otorga a IBM un derecho no exclusivo para utilizar o distribuir la información de la forma que considere adecuada, sin incurrir por ello en ninguna obligación con el remitente.

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

---

# Contenido


<b>IBM MQ en contenedores.....</b>	<b>5</b>
Planificación de IBM MQ en contenedores.....	5
Cómo utilizar IBM MQ en contenedores.....	5
Soporte de contenedores certificados de IBM MQ.....	6
Soporte para la creación de imágenes y gráficos de contenedor de IBM MQ propios.....	8
Consideraciones sobre el almacenamiento para IBM MQ Advanced certified container.....	9
Alta disponibilidad para IBM MQ Advanced certified container.....	10
Autenticación de usuarios y autorización para IBM MQ Advanced certified container.....	12
Instalación y desinstalación del operador de IBM MQ en OpenShift.....	12
Instalación del operador de IBM MQ utilizando la consola web de OpenShift.....	13
Instalación del operador de IBM MQ utilizando la CLI de OpenShift.....	14
Despliegue de contenedores certificados de IBM MQ.....	16
Preparación del proyecto OpenShift para IBM MQ utilizando la CLI de OpenShift.....	16
Despliegue de un gestor de colas utilizando la IBM Cloud Pak for Integration Platform Navigator.....	17
Despliegue de un gestor de colas utilizando la consola web de OpenShift.....	18
Despliegue de un gestor de colas utilizando la CLI de OpenShift.....	19
Integración con el panel de control de operaciones de IBM Cloud Pak for Integration.....	21
Creación de una imagen con archivos MQSC e INI personalizados, utilizando la CLI de OpenShift.....	22
Despliegue de contenedores certificados de IBM MQ utilizando Helm.....	23
Despliegue de releases de CD anteriores de IBM MQ en un clúster IBM Cloud Private.....	27
Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Private.....	29
Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Kubernetes Service.....	30
Conexión con un gestor de colas desplegado en un clúster de OpenShift.....	30
Conexión con el IBM MQ Console desplegado en un clúster de OpenShift.....	32
Copia de seguridad y restauración de la configuración del gestor de colas utilizando la CLI de OpenShift.....	33
Creación de su propio contenedor de IBM MQ.....	34
Planificación de una imagen propia de gestor de colas de IBM MQ utilizando un contenedor.....	34
Creación de un ejemplo de imagen de gestor de colas de IBM MQ usando Docker.....	35
Ejecución de aplicaciones de enlaces locales en contenedores separados.....	38
Referencia de API para el operador IBM MQ.....	40
Referencia de API para <a href="https://mq.ibm.com/v1beta1">mq.ibm.com/v1beta1</a> .....	40
<b>Avisos.....</b>	<b>53</b>
Información acerca de las interfaces de programación.....	54
Marcas registradas.....	55



Los contenedores permiten empaquetar un gestor de colas de IBM MQ o una aplicación cliente de IBM MQ con todas sus dependencias, en una unidad estandarizada para el desarrollo de software.

Puede ejecutar IBM MQ en el contenedor empaquetado previamente que se proporciona en IBM MQ Advanced y en IBM MQ Advanced for Developers. Este IBM MQ Advanced certified container ofrece una imagen soportada y un gráfico Helm, y se puede utilizar para desplegar una imagen de IBM MQ lista para producción en Red Hat® OpenShift®, IBM Cloud Private o IBM Cloud Kubernetes Service.

También se puede ejecutar IBM MQ en un contenedor de IBM Cloud Pak for Integration o en un contenedor creado por usted mismo.

 Puede obtener información adicional sobre el IBM MQ Advanced certified container consultando los enlaces siguientes.

Al planificar IBM MQ en contenedores, tenga en cuenta el soporte que proporciona IBM MQ para diversas opciones de arquitectura como, por ejemplo, cómo gestionar la alta disponibilidad y cómo proteger los gestores de colas.

### Acerca de esta tarea

Antes de planificar la arquitectura de IBM MQ en contenedores, hay que familiarizarse con los conceptos básicos de IBM MQ (consulte [Descripción general técnica de IBM MQ](#)), así como los conceptos básicos de Kubernetes/OpenShift (consulte [Arquitectura de plataforma de contenedores de OpenShift](#)).

### Procedimiento

- [“Cómo utilizar IBM MQ en contenedores”](#) en la página 5.
- [“Alta disponibilidad para IBM MQ Advanced certified container”](#) en la página 10.
- [“Autenticación de usuarios y autorización para IBM MQ Advanced certified container”](#) en la página 12.

Existen varias opciones para utilizar IBM MQ en contenedores: puede optar por utilizar contenedores certificados preempaquetados o puede crear sus propias imágenes y código de despliegue.

### Utilización de los contenedores certificados de IBM MQ Advanced

Si tiene previsto desplegar en Red Hat OpenShift Container Platform, es probable que le interese utilizar los contenedores certificados. Existen tres tipos de contenedores certificados:

- IBM MQ Advanced certified container para IBM Cloud Pak for Integration. Se trata de un producto IBM aparte que incluye una versión de un contenedor certificado.
- IBM MQ Advanced certified container
- Contenedor certificado de IBM MQ Advanced for Developers (sin garantía)

IBM MQ 9.1.4 y los releases anteriores de CD también estaban soportados en IBM Cloud Private y IBM Cloud Kubernetes Service.

Tenga en cuenta que los contenedores certificados evolucionan con rapidez y, por tanto, solo están soportados en los releases de [Continuous Delivery](#).

Los contenedores certificados incluyen imágenes de contenedor precompiladas, así como código de despliegue para su ejecución en Red Hat OpenShift Container Platform. A partir de IBM MQ 9.1.5, los gestores de colas se gestionan utilizando un operador IBM MQ. Las versiones anteriores de IBM MQ, hasta la versión 9.1.5 incluida, se gestionan utilizando diagramas de Helm.

Algunas características de IBM MQ no están soportadas cuando se utilizan contenedores certificados. Tendrá que crear sus propias imágenes y gráficas si desea hacer cualquiera de las cosas siguientes:

- Utilizar las API REST para la administración o la mensajería.
- Utilizar cualquiera de los siguientes componentes de MQ:
  - Managed File Transfer Agentes y sus recursos. Sin embargo, puede utilizar los contenedores certificados para proporcionar uno o varios gestores de colas de Coordinación, Mandato o Agente.
  - AMQP
  - IBM MQ Bridge to Salesforce
  - IBM MQ Bridge to blockchain (no soportado en contenedores)
- Utilice el servidor web cuando realice el despliegue utilizando diagramas de Helm (excepto para IBM Cloud Pak for Integration)
- Personalizar opciones utilizadas con `crtmqm`, `strmqm` y `endmqm`, como por ejemplo configurar registros de recuperación

## Creación de imágenes y gráficas propias

Esta es la solución de contenedor más flexible, pero requiere tener sólidos conocimientos técnicos en configuración de contenedores y "tener en propiedad" el contenedor resultante. Si no tiene previsto utilizar Red Hat OpenShift Container Platform, tendrá que crear su propio código de despliegue e imágenes.

Hay ejemplos de compilación de imágenes propias. Consulte [“Creación de su propio contenedor de IBM MQ” en la página 34](#). Las gráficas de Helm proporcionadas como parte de los contenedores certificados están publicadas en GitHub y se pueden utilizar como ejemplo cuando se construyan imágenes propias:

- [Gráfica de Helm de IBM MQ Advanced certified container](#)
- [Gráfica de Helm de un contenedor certificado de IBM MQ Advanced for Developers](#)

## Conceptos relacionados

[“Soporte de contenedores certificados de IBM MQ” en la página 6](#)


Los contenedores certificados de IBM MQ solo están soportados en determinados entornos de Kubernetes

[“Soporte para la creación de imágenes y gráficos de contenedor de IBM MQ propios” en la página 8](#)  
Información por tener en cuenta si se están utilizando contenedores en un sistema Linux.

Linux

## Soporte de contenedores certificados de IBM MQ

Los contenedores certificados de IBM MQ solo están soportados en determinados entornos de Kubernetes

 En el caso de los releases de CD V9.1.4 y posteriores, se soporta IBM MQ Advanced certified container para su uso con Red Hat OpenShift. Consulte [“Despliegue de un gestor de colas utilizando la CLI de Helm” en la página 25](#)

Los releases de CD anteriores a V9.1.4 estuvieron soportados en los entornos de Kubernetes siguientes:

- IBM Cloud Kubernetes Service
- IBM Cloud Private
- IBM Cloud Private con Red Hat OpenShift

Para ver las versiones soportadas concretas de Kubernetes, consulte los archivos `qualification.yaml` y `Chart.yaml` dentro de un diagrama de Helm de IBM MQ Advanced descargado. Estas versiones varían de un release a otro.

El IBM MQ Advanced certified container sólo está soportado cuando se despliega utilizando el operador IBM MQ o cuando se utiliza uno de los siguientes diagramas de Helm :

- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod` en IBM Cloud Pak for Integration

**Nota:** El uso de los diagramas de Helm está en desuso, después del release del operador de IBM MQ .

Puesto que la tecnología de contenedores evoluciona con rapidez, el IBM MQ Advanced certified container solo está soportado en la última versión de las plataformas soportadas por este gráfico en el momento de la publicación. Si desea utilizar una versión de plataforma anterior, puede que tenga que utilizar una versión anterior de IBM MQ Advanced certified container.

La imagen IBM MQ Advanced certified container se basa en los releases de IBM MQ Continuous Delivery (CD). Estos se soportan durante un máximo de un año o durante dos releases de CD, lo que sea más largo. Los releases de Long Term Support de IBM MQ no están disponibles como contenedor certificado.

A partir de IBM MQ Advanced certified container V4.0 inclusive, la imagen proporciona una instalación de IBM MQ en una Red Hat Universal Base Image (UBI), que incluye bibliotecas clave de Linux y utilidades usadas por IBM MQ. La UBI está soportada por Red Hat cuando se ejecuta en un host Red Hat Enterprise Linux. Las versiones anteriores de IBM MQ Advanced certified container utilizaban una imagen base Ubuntu no soportada.

### Conceptos relacionados

“Soporte para la creación de imágenes y gráficos de contenedor de IBM MQ propios” en la [página 8](#) Información por tener en cuenta si se están utilizando contenedores en un sistema Linux.

## Linux > MQ Adv. > CD Soporte de versiones del IBM MQ Advanced certified container

Conjunto de tablas que muestra la correlación entre las versiones soportadas del IBM MQ Advanced certified container, IBM MQ, IBM Cloud Kubernetes Service, IBM Cloud Pak for Integration e IBM Cloud Private.

### Operador de IBM MQ

> V 9.1.5

El operador IBM MQ está soportado para su uso como parte de IBM Cloud Pak for Integration versión 2020.2, o de forma independiente, con IBM MQ versión 9.1.5 y superior.

El operador IBM MQ está soportado en Red Hat OpenShift Container Platform versión 4.4 o superior.

### IBM MQ Advanced certified container > V 9.1.5 (diagrama deHelm )-en desuso

Incluye la gráfica de Helm `ibm-mqadvanced-server-prod`.

> V 9.1.5 A partir de IBM MQ Advanced certified container V5.0.x inclusive, la gráfica de Helm, la imagen y los arreglos se suministran a través de IBM Entitled Catalog and Registry. Las versiones anteriores se suministraban a través de Passport Advantage y los releases de arreglo están disponibles en IBM Fix Central.

Tabla 1. Soporte del IBM MQ Advanced certified container

Ve rsi ón	Versión de IBM MQ	Fin de soporte	Plataformas soportadas
6.0.x	Release 9.1.5 de entrega continua	Marzo de 2021	<a href="#">Requisitos detallados del sistema</a>
5.0.x	Release 9.1.4 de entrega continua	Diciembre de 2020	<a href="#">Requisitos detallados del sistema</a>
4.1.x	Release 9.1.3 de entrega continua	Julio de 2020	<a href="#">Requisitos detallados del sistema</a>

## Software de IBM MQ Advanced certified container para IBM Cloud Pak for Integration **V 9.1.5** (diagrama deHelm )-en desuso

Incluye la gráfica de Helm `ibm-mqadvanced-server-integration-prod`.

Tabla 2. Soporte de versiones de software de IBM MQ Advanced certified container para IBM Cloud Pak for Integration

Versión	Versión de IBM MQ	Versión de IBM Cloud Pak for Integration
6.0.x	Release 9.1.4 de entrega continua	2020.1.1 ( <a href="#">requisitos del sistema</a> )
5.0.x	Release 9.1.3 de entrega continua	2019.4.1 ( <a href="#">requisitos del sistema</a> )
4.1.x	Release 9.1.3 de entrega continua	2019.3.2.2 ( <a href="#">requisitos del sistema</a> )
4.0.x	Release 9.1.3 de entrega continua	2019.3.2 ( <a href="#">requisitos del sistema</a> )
3.0.x	Release 9.1.3 de entrega continua	2019.3.1 ( <a href="#">requisitos del sistema</a> )

Consulte las [notas del release de IBM Cloud Pak for Integration](#) para obtener información sobre la versión soportada.

## Linux Soporte para la creación de imágenes y gráficos de contenedor de IBM MQ propios

Información por tener en cuenta si se están utilizando contenedores en un sistema Linux.

- La imagen base utilizada por la imagen de contenedor debe utilizar un sistema operativo Linux que esté soportado.
- Hay que usar los instaladores de IBM MQ para instalar el producto dentro de la imagen de contenedor.
- Para obtener la lista de paquetes soportados, consulte [componentes rpm de IBM MQ para sistemas Linux](#).
- **V 9.1.0** Los paquetes siguientes no están soportados:
  - MQSeriesBCBridge
  - MQSeriesRDQM
- El directorio de datos del gestor de colas (`/var/mqm` de forma predeterminada) tiene que estar almacenado en un volumen de contenedor que mantenga un estado persistente.

**Importante:** No puede utilizar el sistema de archivos de unión.



Debe montar un directorio de host como un volumen de datos o utilizar un contenedor de volúmenes de datos. Para obtener más información, consulte [Gestionar datos en contenedores](#).

- Debe ser capaz de ejecutar mandatos de control de IBM MQ, como por ejemplo, **endmqm**, dentro del contenedor.
- Hay que ser capaz de obtener archivos y directorios desde dentro del contenedor a efectos de diagnóstico.
- **V 9.1.0** El uso de espacios de nombres permite compartir espacios de nombres del contenedor del gestor de colas con otros contenedores para vincular localmente aplicaciones con un gestor de colas que ejecute en contenedores independientes. Para obtener más información, consulte [“Ejecución de aplicaciones de enlaces locales en contenedores separados”](#) en la página 38.

### Conceptos relacionados

“Soporte de contenedores certificados de IBM MQ” en la página 6

Los contenedores certificados de IBM MQ solo están soportados en determinados entornos de Kubernetes

## **V 9.1.5** **Linux** **MQ Adv.** **CD** **Consideraciones sobre el almacenamiento para IBM MQ Advanced certified container**

El IBM MQ Advanced certified container se ejecuta en dos modalidades de almacenamiento:

- El **almacenamiento efímero** se utiliza cuando se puede desechar todo el estado del contenedor cuando se reinicia el contenedor. Esto se suele utilizar cuando se crean entornos para demostración o cuando se desarrollan con gestores de colas autónomos.
- El **almacenamiento persistente** es la configuración común para IBM MQ y garantiza que si se reinicia el contenedor, la configuración existente, los registros y los mensajes persistentes estarán disponibles en el contenedor reiniciado.

El operador IBM MQ proporciona la posibilidad de personalizar las características de almacenamiento que pueden diferir considerablemente en función del entorno y de la modalidad de almacenamiento deseada.

### Almacenamiento efímero

IBM MQ es una aplicación con estado y persiste este estado en el almacenamiento para la recuperación en el caso de un reinicio. Si se utiliza almacenamiento efímero, se perderá todo el estado del gestor de colas durante el reinicio. Esto incluye:

- Todos los mensajes
- Estado de comunicación de todo el gestor de colas con el gestor de colas (números de secuencia de mensajes de canal)
- La identidad del clúster de MQ del gestor de colas
- Todos los estados de transacción
- Configuración de todos los gestores de colas
- Todos los datos de diagnóstico locales

Por este motivo, debe tener en cuenta si el almacenamiento efímero es un enfoque adecuado para un escenario de producción, prueba o desarrollo. Por ejemplo, donde se sabe que todos los mensajes no son persistentes y el gestor de colas no es miembro de un clúster MQ. Además de desechar todo el estado de mensajería durante el reinicio, la configuración del gestor de colas también se descarta. Para habilitar un contenedor completamente efímero, la configuración de IBM MQ debe añadirse a la propia imagen del contenedor (para obtener más información, consulte [“Creación de una imagen con archivos MQSC e INI personalizados, utilizando la CLI de OpenShift”](#) en la página 22). Si esto no se completa, será necesario configurar IBM MQ cada vez que se reinicie el contenedor.

Por ejemplo, para configurar IBM MQ con almacenamiento efímero, el tipo de almacenamiento de QueueManager debe incluir lo siguiente:

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

## Almacén persistente

IBM MQ normalmente se ejecuta con almacenamiento de persistencia para asegurarse de que el gestor de colas conserva sus mensajes persistentes y la configuración después de un reinicio. Por lo tanto, este es el comportamiento predeterminado. Debido a los distintos proveedores de almacenamiento y a las distintas prestaciones de cada soporte, esto a menudo significa que es necesaria la personalización de la configuración. A continuación se describen los campos comunes que personalizan la configuración de almacenamiento de MQ en la API v1beta1 :

- `spec.queueManager.availability` controla la modalidad de disponibilidad. Si utiliza `SingleInstance` sólo necesita almacenamiento `ReadWriteOnce` , mientras que `MultiInstance` requiere una clase de almacenamiento que admita `ReadWriteMany` con las características de bloqueo de archivos correctas. IBM MQ proporciona una [declaración de soporte](#) y una [declaración de prueba](#). La modalidad de disponibilidad también influye en el diseño del volumen persistente. Para obtener más información, consulte [“Alta disponibilidad para IBM MQ Advanced certified container”](#) en la [página 10](#)
- `spec.queueManager.storage` controla los valores de almacenamiento individuales. Un gestor de colas se puede configurar para utilizar entre uno y cuatro volúmenes persistentes

El ejemplo siguiente muestra un fragmento de una configuración simple utilizando un gestor de colas de una sola instancia:

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

El ejemplo siguiente muestra un fragmento de código de una configuración de gestor de colas de varias instancias, con una clase de almacenamiento no predeterminada, y con almacenamiento de archivos que requiere grupos suplementarios:

```
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 disponibilidad para IBM MQ Advanced certified container

Hay dos opciones principales de alta disponibilidad en IBM MQ Advanced certified container: **Gestor de colas multiinstancia** (que es un par 'activo-en espera', que utiliza un sistema de archivos compartido en red) y **Gestor de colas único resiliente** (que ofrece un método simple de HA utilizando un almacenamiento en red).

Debe considerar por separado la disponibilidad de **mensajes** y de **servicio**. En IBM MQ for Multiplatforms, un mensaje se almacena en un único gestor de colas. Por lo tanto, si dicho gestor de colas deja de estar disponible, se perderá temporalmente el acceso a los mensajes que contenga. Para lograr una alta disponibilidad de mensajes, hay que poder recuperar un gestor de colas lo más rápidamente posible. Se puede conseguir la disponibilidad de **servicio** teniendo múltiples instancias de colas para que las utilicen las aplicaciones cliente, por ejemplo, utilizando un clúster uniforme de IBM MQ.

Se puede pensar en un gestor de colas como dividido en dos partes: los datos almacenados en el disco y los procesos en ejecución que dan acceso a los datos. Cualquier gestor de colas se puede mover a un nodo Kubernetes diferente siempre que mantenga los mismos datos (proporcionados por Volúmenes persistentes de Kubernetes) y las aplicaciones clientes sigan pudiendo direccionarlo a través de la red. En Kubernetes, se utiliza un servicio para proporcionar una identidad de red coherente.

IBM MQ se apoya en la disponibilidad de los datos de volúmenes persistentes. Por lo tanto, la disponibilidad del almacenamiento que proporcionen los volúmenes persistentes es de vital importancia para la disponibilidad del gestor de colas, porque IBM MQ no puede estar más disponible que el almacenamiento que usa. Si desea tolerar la caída de una zona de disponibilidad completa, tendrá que usar un proveedor de volúmenes que replique las escrituras de disco en otra zona.

## Gestor de colas multiinstancia

Los gestores de colas de varias instancias implican un pod **activo** y un pod **en espera** Kubernetes, que se ejecutan como parte de un conjunto con estado Kubernetes con exactamente dos réplicas y un conjunto de volúmenes persistentes de Kubernetes. Los datos y registros de transacciones del gestor de colas se conservan en dos volúmenes persistentes utilizando un sistema de archivos compartido.

Los gestores de colas multiinstancia requieren que ambos Pods, **activo** y **en espera**, tengan acceso simultáneo al volumen persistente. Para configurarlo, utilice Kubernetes Volúmenes persistentes con **access mode** establecido en ReadWriteMany. Los volúmenes también tienen que cumplir los IBM MQ requisitos para sistemas de archivos compartidos, porque IBM MQ se basa en el desbloqueo automático de archivos para provocar una migración tras error del gestor de colas. IBM MQ genera una lista de sistemas de archivos probados.

Los tiempos de recuperación de un gestor de colas multiinstancia están controlados por los factores siguientes:

1. Cuánto tiempo tarda el sistema de archivos compartido, tras producirse un fallo, en liberar los bloqueos obtenidos inicialmente por la instancia activa.
2. Cuánto tiempo tarda la instancia en espera en adquirir los bloqueos e iniciarse.
3. Cuánto tiempo tarda la el sondeo de preparación del Pod de Kubernetes en detectar que el contenedor está listo. Esto se puede configurar en la gráfica de Helm.
4. Cuánto tiempo tardan los clientes de IBM MQ en reconectarse.

## Gestor de colas único resiliente

Un único gestor de colas resiliente es una única instancia de gestor de colas que ejecuta en un único Pod de Kubernetes, donde Kubernetes supervisa el gestor de colas y sustituye al Pod según sea necesario.

Los requisitos de IBM MQ para los sistemas de archivos compartidos también se aplican cuando se utiliza un único gestor de colas resiliente (excepto para el bloqueo basado en alquiler), pero no tiene que utilizar un sistema de archivos compartidos. Se puede utilizar un almacenamiento en bloques con un sistema de archivos adecuado en la parte superior. Por ejemplo, *xfs* o *ext4*.

Los tiempos de recuperación de un único gestor de colas resiliente están controlados por los factores siguientes:

1. Cuánto tiempo tarda en ejecutar el sondeo de vida, y cuántos errores tolera. Esto se puede configurar en la gráfica de Helm.
2. Cuánto tiempo tarda el Planificador de Kubernetes en replanificar el pod que ha fallado a un nuevo nodo.

3. Cuánto tiempo se tarda en descargar la imagen del contenedor en el nuevo nodo. Si se utiliza una **imagePullPolicy** de valor `IfNotPresent`, puede que la imagen ya esté disponible en ese nodo.
4. Cuánto tiempo tarda en iniciarse la nueva instancia del gestor de colas.
5. Cuánto tiempo tarda la el sondeo de preparación del Pod de Kubernetes en detectar que el contenedor está listo. Esto se puede configurar en la gráfica de Helm.
6. Cuánto tiempo tardan los clientes de IBM MQ en reconectarse.

### Importante:

Aunque el patrón de gestor de colas resiliente único tiene algunas ventajas, hay que tener claro si se pueden alcanzar los objetivos de disponibilidad con las limitaciones relativas a fallos de nodo.

En Kubernetes, un pod que ha fallado suele recuperarse con rapidez, pero el fallo de un nodo entero se maneja de forma diferente. Si el nodo maestro de Kubernetes pierde el contacto con un nodo trabajador, no podrá determinar si dicho nodo ha fallado o si simplemente se ha perdido la conectividad de red. Por lo tanto, Kubernetes no llevará a cabo **ninguna acción** en este caso mientras no se produzca uno de los sucesos siguientes:

1. El nodo recupera un estado en el que el nodo maestro de Kubernetes se puede comunicar con él.
2. Se ha realizado una acción administrativa para eliminar explícitamente el pod en el nodo maestro de Kubernetes. Esto no implica necesariamente que el pod deje de ejecutar, sino que se limita a borrarlo del almacén de Kubernetes. Por lo tanto, esta acción administrativa tiene que hacerse con mucho cuidado.

### Conceptos relacionados

[Configuraciones de alta disponibilidad](#)

## Linux > MQ Adv. > CD **Autenticación de usuarios y autorización para IBM MQ Advanced certified container**

IBM MQ se puede configurar para que utilice usuarios y grupos LDAP en las autorizaciones. Este es el método recomendado para IBM MQ Advanced certified container.

En un entorno contenerizado de varios arrendatarios como, por ejemplo, Red Hat OpenShift Container Platform, se ponen en marcha restricciones de seguridad para evitar posibles problemas de seguridad. Por ejemplo, en Red Hat OpenShift Container Platform el valor predeterminado `SecurityContextConstraints` (denominado `restricted`) utiliza un ID de usuario aleatorizado, lo que desaconseja a los usuarios locales del propio contenedor. IBM MQ suele utilizar el escalamiento de privilegios para comprobar las contraseñas de los usuarios, lo que tampoco se recomienda en entornos de contenedor de varios arrendatarios. Por estas razones, el uso de usuarios definidos en las bibliotecas del sistema operativo dentro de un contenedor en ejecución no está soportado en los contenedores certificados de IBM MQ .

Debe configurar el gestor de colas para utilizar LDAP para la autenticación y autorización de usuarios. Para obtener información sobre cómo configurar IBM MQ para hacer esto, consulte [Autenticación de conexiones: Repositorios de usuario](#) y [Autorización LDAP](#)

## V 9.1.5 > Linux > MQ Adv. > CD **Instalación y desinstalación del operador de IBM MQ en OpenShift**

El operador IBM MQ se puede instalar en OpenShift utilizando el concentrador del operador.

### Antes de empezar

### Procedimiento

- [“Instalación del operador de IBM MQ utilizando la CLI de OpenShift”](#) en la página 14.
- [“Instalación del operador de IBM MQ utilizando la consola web de OpenShift”](#) en la página 13.

## utilizando la consola web de OpenShift

El operador IBM MQ se puede instalar en OpenShift utilizando el concentrador del operador.

### Antes de empezar

Inicie sesión en la consola web del clúster de OpenShift .

### Procedimiento

1. Añada los operadores de IBM Common Services a la lista de operadores instalables

- Pulse el icono de signo más. Verá el recuadro de diálogo **Importar YAML**.
- Pegue la siguiente definición de recurso en el recuadro 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) Pulse **Crear**.

2. Añada los operadores de IBM a la lista de operadores instalables

- Pulse el icono de signo más. Verá el recuadro de diálogo **Importar YAML**.
- Pegue la siguiente definición de recurso en el recuadro 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) Pulse **Crear**.

3. Cree un espacio de nombres para utilizarlo para el operador IBM MQ

El operador IBM MQ se puede instalar en un único espacio de nombres o en todos los espacios de nombres. Este paso sólo es necesario si desea realizar la instalación en un espacio de nombres determinado que todavía no existe.

- En el panel de navegación, pulse **Inicio > Proyectos**.  
Se visualiza la página Proyectos.
  - Pulse **Crear proyecto**. Se visualiza un área Crear proyecto.
  - Especifique los detalles del espacio de nombres que está creando. Por ejemplo, puede especificar "ibm-mq" como nombre.
  - Pulse **Crear**. Se crea el espacio de nombres para el operador de IBM MQ .
4. Instale el operador de IBM MQ.
- En el panel de navegación, pulse **Operadores > OperatorHub**.

- Se visualiza la página OperatorHub.
- b) En el campo **Todos los elementos** , especifique "IBM MQ".  
Se visualiza la entrada de catálogo IBM MQ .
- c) Seleccione **IBM MQ**.  
Se visualiza la ventana IBM MQ .
- d) Pulse **Instalar**.  
Verá la página Crear suscripción de operador.
- e) Establezca la modalidad de instalación en el espacio de nombres específico que ha creado o en el ámbito de todo el clúster
- f) Pulse **Suscribir**.  
Verá IBM MQ en la página Operadores instalados.
- g) Compruebe el estado del operador en la página Operadores instalados, el estado cambiará a Satisfactorio cuando se complete la instalación.

## Qué hacer a continuación

[“Despliegue de contenedores certificados de IBM MQ” en la página 16](#)

## V 9.1.5 Linux MQ Adv. CD Instalación del operador de IBM MQ utilizando la CLI de OpenShift

El operador IBM MQ se puede instalar en OpenShift utilizando el concentrador del operador.

### Antes de empezar

Inicie sesión en la interfaz de línea de comandos (CLI) de OpenShift utilizando **oc login**. Para estos pasos, tendrá que ser administrador del clúster.

### Procedimiento

1. Cree un `OperatorSource` para los operadores de IBM Common Services

- a) Crear un archivo YAML que defina el recurso `OperatorSource`

Cree un archivo denominado "operator-source-cs.yaml" con el contenido siguiente:

```
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 `OperatorSource` al servidor.

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

2. Cree un `OperatorSource` para los operadores de IBM

- a) Crear un archivo YAML que defina el recurso `OperatorSource`

Cree un archivo denominado "operator-source-ibm.yaml" con el contenido siguiente:

```
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 OperatorSource al servidor.

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

### 3. Cree un espacio de nombres para utilizarlo para el operador IBM MQ

El operador de IBM MQ se puede instalar en un único espacio de nombres o en todos los espacios de nombres. Este paso sólo es necesario si desea realizar la instalación en un espacio de nombres determinado que todavía no existe.

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

### 4. Vea la lista de operadores disponibles para el clúster desde OperatorHub

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

### 5. Inspeccione el operador de IBM MQ para verificar sus InstallModes soportados y los canales disponibles

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

### 6. Crear un archivo YAML de objeto OperatorGroup

Un OperatorGroup es un recurso OLM que selecciona espacios de nombres de destino en los que generar el acceso RBAC necesario para todos los operadores en el mismo espacio de nombres que el OperatorGroup.

El espacio de nombres al que se suscribe el operador debe tener un OperatorGroup que coincida con el InstallModedel operador, ya sea la modalidad AllNamespaces o SingleNamespace . Si el operador que desea instalar utiliza AllNamespaces, el espacio de nombres openshift-operators ya tiene un OperatorGroup adecuado.

Sin embargo, si el operador utiliza la modalidad SingleNamespace y todavía no tiene un OperatorGroup adecuado en su lugar, debe crear uno.

a) Cree un archivo denominado "mq-operator-group.yaml" con el contenido siguiente:

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

b) Crear el objeto OperatorGroup

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

### 7. Crear un archivo YAML de objeto de suscripción para suscribir un espacio de nombres al operador de MQ

a) Cree un archivo denominado "mq-sub.yaml" con el contenido siguiente:

```
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 el uso de `AllNamespaces InstallMode`, especifique el espacio de nombres `openshift-operators`. De lo contrario, especifique el espacio de nombres único relevante para el uso de `SingleNamespace InstallMode`.

b) Crear el objeto Subscription

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

8. Comprobar el estado del operador

Una vez que la instalación del operador se ha realizado correctamente, el estado del pod se muestra como *En ejecución*. Para el uso de `AllNamespaces InstallMode`, especifique `openshift-operators` como espacio de nombres. De lo contrario, especifique el espacio de nombres único relevante para el uso de `SingleNamespace InstallMode`.

## Qué hacer a continuación

[“Despliegue de contenedores certificados de IBM MQ” en la página 16](#)

## Linux > MQ Adv. > CD Despliegue de contenedores certificados de IBM MQ

IBM MQ versión 9.1.5 y superior se puede desplegar en Red Hat OpenShift utilizando el operador IBM MQ. Las versiones de IBM MQ 9.1.5 y 9.1.4 se pueden desplegar en Red Hat OpenShift utilizando Helm. Las versiones anteriores de CD se pueden desplegar en un clúster de IBM Cloud Private o en un clúster de IBM Cloud Kubernetes Service, utilizando Helm.

### Acerca de esta tarea

#### Procedimiento

- [“Despliegue de un gestor de colas utilizando la CLI de Helm” en la página 25.](#)
- [“Despliegue de releases de CD anteriores de IBM MQ en un clúster IBM Cloud Private” en la página 27.](#)
- [“Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Private” en la página 29.](#)
- [“Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Kubernetes Service” en la página 30.](#)

## Linux > MQ Adv. > CD Preparación del proyecto OpenShift para IBM MQ utilizando la CLI de OpenShift

Prepare el clúster de Red Hat OpenShift Container Platform para que esté preparado para desplegar un gestor de colas utilizando el operador IBM MQ. Esta tarea debe ser completada por un administrador de proyectos.

### Antes de empezar

**Nota:** Si tiene previsto utilizar IBM MQ en un proyecto con otros componentes de IBM Cloud Pak for Integration ya instalados, es posible que no tenga que seguir estas instrucciones.

Inicie sesión en el clúster con `cloudctl login` (en IBM Cloud Pak for Integration) o `oc login`.



## Acerca de esta tarea

Las imágenes de IBM MQ Advanced certified container se extraen de un registro de contenedor que efectúa una comprobación de titularidad de licencia. Esta comprobación requiere una clave de titularidad que se almacena en un registro de extracción `docker-registry`. Si aún no tiene una clave de titularidad, siga estas instrucciones para obtenerla y crear un secreto de extracción.

## Procedimiento

1. Obtenga la clave de titularidad asignada a su ID.
  - a) Inicie sesión en [MyIBM Container Software Library](#) con el ID de IBM y la contraseña asociados al software autorizado.
  - b) En la sección **Entidades de titularidad**, seleccione **Copiar clave** para copiar la clave de titularidad en el portapapeles.
2. Cree un secreto que contenga la clave de titularidad, en el proyecto donde desea desplegar el gestor de colas.

Ejecute el mandato siguiente, donde `<entitlement-key>` es la clave recuperada en el paso 1, y `<user-email>` es el ID de IBM asociado con el software autorizado.

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

## Qué hacer a continuación

[“Despliegue de un gestor de colas utilizando la CLI de OpenShift” en la página 19](#)

## Despliegue de un gestor de colas utilizando la IBM Cloud Pak for Integration Platform Navigator

Utilice el recurso personalizado QueueManager para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform utilizando IBM Cloud Pak for Integration Platform Navigator. Esta tarea la debe completar un administrador de proyectos

## Antes de empezar

En un navegador, inicie IBM Cloud Pak for Integration Platform Navigator.

Si es la primera vez que despliega un gestor de colas en este proyecto de Red Hat OpenShift, siga los pasos para [“Preparación del proyecto OpenShift para IBM MQ utilizando la CLI de OpenShift” en la página 16](#).

## Procedimiento

1. Despliegue un gestor de colas.

El ejemplo siguiente despliega un gestor de colas de "inicio rápido", que utiliza almacenamiento efímero (no persistente) y desactiva la seguridad de MQ. Los mensajes no se conservarán en los reinicios del gestor de colas. Puede ajustar la configuración para cambiar muchos valores del gestor de colas.

  - a) En IBM Cloud Pak for Integration Platform Navigator, pulse **Tiempo de ejecución e instancias**.
  - b) Pulse **Crear instancia**.
  - c) Seleccione **Gestor de colas** y pulse **Siguiente**.

Se muestra el formulario para crear una instancia de un QueueManager.

**Nota:** También puede pulsar **Código** para ver o cambiar el YAML de configuración de QueueManager.

- d) En la sección **Detalles** , compruebe o actualice el campo **Nombre** y especifique el **Espacio de nombres** en el que desea crear la instancia del gestor de colas.
  - e) Si acepta el acuerdo de licencia de IBM Cloud Pak for Integration , cambie **Aceptación de licencia** a **Activado**.  
Debe aceptar la licencia para desplegar un gestor de colas.
  - f) En la sección **Configuración del gestor de colas** , compruebe o actualice el **Nombre** del gestor de colas subyacente.  
De forma predeterminada, el nombre del gestor de colas utilizado por las aplicaciones cliente de IBM MQ será el mismo que el nombre del QueueManager, pero con los caracteres no válidos (como los guiones) eliminados. Si desea forzar el uso de un nombre en particular, puede editarlo aquí.
  - g) Pulse **Crear**  
Ahora se visualiza la lista de gestores de colas en el proyecto actual (espacio de nombres). El nuevo QueueManager debe tener un estado de Pending
2. Comprobar que el gestor de colas se está ejecutando  
La creación se completa cuando el estado de QueueManager es Running.

### Tareas relacionadas

[“Conexión con un gestor de colas desplegado en un clúster de OpenShift” en la página 30](#)

Conjunto de ejemplos de configuración para conectarse con un gestor de colas desplegado en un clúster de Red Hat OpenShift.

[“Conexión con el IBM MQ Console desplegado en un clúster de OpenShift” en la página 32](#)

Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

V 9.1.5

Linux

MQ Adv.

CD

## Despliegue de un gestor de colas utilizando la consola web de OpenShift

Utilice el recurso personalizado QueueManager para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform utilizando la consola web de Red Hat OpenShift . Esta tarea la debe completar un administrador de proyectos

### Antes de empezar

Inicie sesión en la consola web del clúster de OpenShift . Deberá seleccionar un proyecto existente (espacio de nombres) para utilizarlo o crear uno nuevo.

Si es la primera vez que despliega un gestor de colas en este proyecto de Red Hat OpenShift , siga los pasos para [“Preparación del proyecto OpenShift para IBM MQ utilizando la CLI de OpenShift” en la página 16.](#)

### Procedimiento

1. Despliegue un gestor de colas.  
El ejemplo siguiente despliega un gestor de colas de "inicio rápido", que utiliza almacenamiento efímero (no persistente) y desactiva la seguridad de MQ . Los mensajes no se conservarán en los reinicios del gestor de colas. Puede ajustar la configuración para cambiar muchos valores del gestor de colas.
  - a) En la consola web de OpenShift , en el panel de navegación, pulse **Operadores > Operadores instalados**
  - b) Pulse **IBM MQ**.
  - c) Pulse el separador **Gestor de colas** .
  - d) Pulse el botón **Crear QueueManager** .  
Se visualiza un editor YAML, que contiene el YAML de ejemplo para un recurso QueueManager .

**Nota:** También puede pulsar **Editar formulario** para ver o cambiar la configuración de QueueManager .

- e) Si acepta el acuerdo de licencia, cambie **Aceptación de licencia** a **Activado**.

IBM MQ está disponible bajo varias licencias diferentes. Para obtener más información sobre las licencias válidas, consulte [“Referencia de licencia para mq.ibm.com/v1beta1”](https://www.ibm.com/docs/es/mq/v1beta1?topic=licensing) en la página 40. Debe aceptar la licencia para desplegar un gestor de colas.

- f) Pulse **Crear**

Ahora se visualiza la lista de gestores de colas en el proyecto actual (espacio de nombres). El nuevo QueueManager debe estar en un estado Pending .

2. Comprobar que el gestor de colas se está ejecutando

La creación se completa cuando el estado de QueueManager es Running.

### Tareas relacionadas

[“Conexión con un gestor de colas desplegado en un clúster de OpenShift”](#) en la página 30

Conjunto de ejemplos de configuración para conectarse con un gestor de colas desplegado en un clúster de Red Hat OpenShift.

[“Conexión con el IBM MQ Console desplegado en un clúster de OpenShift”](#) en la página 32

Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

## Despliegue de un gestor de colas utilizando la CLI de OpenShift

Utilice el recurso personalizado QueueManager para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform utilizando la interfaz de línea de mandatos (CLI). Esta tarea la debe completar un administrador de proyectos

### Antes de empezar

Debe instalar la [interfaz de línea de mandatos de Red Hat OpenShift Container Platform](#).

Inicie sesión en el clúster con **cloudctl login** (en IBM Cloud Pak for Integration) o **oc login**.

Si es la primera vez que despliega un gestor de colas en este proyecto de Red Hat OpenShift , siga los pasos para [“Preparación del proyecto OpenShift para IBM MQ utilizando la CLI de OpenShift”](#) en la página 16.

### Procedimiento

1. Despliegue un gestor de colas.

El ejemplo siguiente despliega un gestor de colas de "inicio rápido", que utiliza almacenamiento efímero (no persistente) y desactiva la seguridad de MQ . Los mensajes no se conservarán en los reinicios del gestor de colas. Puede ajustar el contenido del archivo YAML para cambiar muchos valores del gestor de colas.

- a) Crear un archivo YAML de QueueManager

Por ejemplo, para instalar un gestor de colas básico en IBM Cloud Pak for Integration, cree el archivo "mq-quickstart.yaml" con el contenido siguiente:

```
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:** Si acepta el acuerdo de licencia de IBM Cloud Pak for Integration, cambie `accept: false` por `accept: true`. Consulte [“Referencia de licencia para mq.ibm.com/v1beta1”](https://mq.ibm.com/v1beta1) en la página 40 para obtener detalles sobre la licencia.

Este ejemplo también incluye un servidor web desplegado con el gestor de colas, con la consola web habilitada con inicio de sesión único con Cloud Pak Identity and Access Manager.

Para instalar un gestor de colas básico independientemente de IBM Cloud Pak for Integration, cree el archivo `"mq-quickstart.yaml"` con el contenido siguiente:

```

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:** Si acepta el acuerdo de licencia de MQ, cambie `accept: false` por `accept: true`. Consulte [“Referencia de licencia para mq.ibm.com/v1beta1”](https://mq.ibm.com/v1beta1) en la página 40 para obtener detalles sobre la licencia.

#### b) Crear el objeto QueueManager

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

#### 2. Comprobar que el gestor de colas se está ejecutando

Puede validar el despliegue ejecutando

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

y, a continuación, comprobando el estado.

Por ejemplo, ejecute

```
oc describe queuemanager quickstart
```

y compruebe que el campo `status.Phase` indica `Running`

#### Tareas relacionadas

[“Conexión con un gestor de colas desplegado en un clúster de OpenShift”](#) en la página 30

Conjunto de ejemplos de configuración para conectarse con un gestor de colas desplegado en un clúster de Red Hat OpenShift.

“[Conexión con el IBM MQ Console desplegado en un clúster de OpenShift](#)” en la página 32  
Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

## Linux MQ Adv. V 9.1.4 CD Integración con el panel de control de operaciones de IBM Cloud Pak for Integration

La capacidad de rastrear transacciones a través de IBM Cloud Pak for Integration la proporciona el panel de control de operaciones.

### Acerca de esta tarea

La habilitación de la integración con el panel de control de operaciones instala una salida de API de MQ en el gestor de colas. La salida de API envía los datos de rastreo de los mensajes que fluyen a través del gestor de colas al almacén de datos del panel de control de operaciones.

Tenga en cuenta que sólo se rastrean los mensajes que se envían utilizando enlaces de cliente MQ .

### Procedimiento

#### 1. Desplegar un gestor de colas con el rastreo habilitado

De forma predeterminada, la característica de rastreo está inhabilitada.

Si está desplegando utilizando IBM Cloud Pak for Integration Platform Navigator, puede habilitar el rastreo durante el despliegue, estableciendo **Habilitar rastreo** en **Activado** y estableciendo el **Espacio de nombres de rastreo** en el espacio de nombres donde está instalado el panel de control de operaciones. Para obtener más información sobre el despliegue de un gestor de colas, consulte “[Despliegue de un gestor de colas utilizando la IBM Cloud Pak for Integration Platform Navigator](#)” en la página 17

Si está realizando el despliegue utilizando la [CLI de OpenShift](#) o la [consola web de OpenShift](#), puede habilitar el rastreo con el siguiente fragmento de código YAML:

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

Si está desplegando utilizando Helm, puede habilitar el rastreo estableciendo `odTracingConfig.enabled=true` y `odTracingConfig.odTracingNamespace=<Operations_Dashboard_Namespace`. Si desea habilitar la integración del panel de control de operaciones en un gestor de colas existente, puede aplicar esta configuración al actualizar el release de Helm.

**Importante:** El gestor de colas no se iniciará mientras MQ no se haya registrado en el panel de control de operaciones (consulte el paso siguiente).

Observe que, cuando esta característica está habilitada, ejecutará dos contenedores sidecar ("Agent" y "Collector") además del contenedor del gestor de colas. Las imágenes de estos contenedores sidecar estarán disponibles en el mismo registro que la imagen de MQ principal y usarán la misma política de extracción y el mismo secreto de extracción. Existen dos parámetros adicionales disponibles para configurar los límites de memoria y CPU.

#### 2. Si es la primera vez que se despliega un gestor de colas con integración del panel de control de operaciones en este espacio de nombres, es necesario [Registro](#) con el panel de control de operaciones.

El registro crea un objeto Secreto que el Pod de gestor de colas necesita para iniciarse correctamente.

## MQSC e INI personalizados, utilizando la CLI de OpenShift

Utilice una interconexión de Red Hat OpenShift Container Platform para crear una nueva imagen de contenedor de IBM MQ, con archivos MQSC e INI que desea que se apliquen a los gestores de colas utilizando esta imagen. Esta tarea la debe completar un administrador de proyectos

### Antes de empezar

Debe instalar la [interfaz de línea de mandatos de Red Hat OpenShift Container Platform](#).

Inicie sesión en el clúster con **cloudctl login** (en IBM Cloud Pak for Integration) o **oc login**.

Si no tiene un secreto de OpenShift para el registro autorizado de IBM en el proyecto Red Hat OpenShift, siga los pasos para [“Preparación del proyecto OpenShift para IBM MQ utilizando la CLI de OpenShift”](#) en la [página 16](#).

### Procedimiento

#### 1. Cree una ImageStream

Una secuencia de imágenes y sus etiquetas asociadas proporcionan una abstracción para hacer referencia a imágenes de contenedor desde dentro de Red Hat OpenShift Container Platform. La secuencia de imágenes y sus etiquetas le permiten ver qué imágenes están disponibles y asegurarse de que está utilizando la imagen específica que necesita incluso si la imagen en el repositorio cambia.

```
oc create imagestream mymq
```

#### 2. Crear un BuildConfig para la nueva imagen

Un BuildConfig permitirá compilaciones para la nueva imagen, que se basará en las imágenes oficiales de IBM, pero añadirá los archivos MQSC o INI que desee que se ejecuten en el inicio del contenedor.

##### a) Crear un archivo YAML que defina el recurso BuildConfig

Por ejemplo, cree un archivo denominado "mq-build-config.yaml" con el contenido siguiente:

```
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'
```

Tendrá que sustituir los dos lugares en los que se menciona la base IBM MQ, para que apunte a la imagen base correcta para la versión y el arreglo que desea utilizar. A medida que se apliquen los arreglos, tendrá que repetir estos pasos para volver a crear la imagen.

Este ejemplo crea una imagen nueva basada en la imagen oficial de IBM y añade archivos denominados "my.mqsc" y "my.ini" en el directorio /etc/mqm. Cualquier archivo MQSC o INI que se encuentre en este directorio será aplicado por el contenedor durante el inicio. Los archivos

INI se aplican utilizando la opción `crtmqm -ii` y se fusionan con los archivos INI existentes. Los archivos MQSC se aplican en orden alfabético.

Es importante que los mandatos MQSC sean repetibles, ya que se ejecutarán *cada vez* que se inicie el gestor de colas. Esto normalmente significa añadir el parámetro REPLACE en cualquier mandato DEFINE y añadir el parámetro IGNSTATE (YES) a cualquier mandato START o STOP .

b) Aplique BuildConfig al servidor.

```
oc apply -f mq-build-config.yaml
```

3. Ejecutar una compilación para crear la imagen

a) Iniciar la compilación

```
oc start-build mymq
```

Debería ver una salida similar a la esta:

```
build.build.openshift.io/mymq-1 started
```

b) Comprobar el estado de la compilación

Por ejemplo, puede ejecutar el mandato siguiente, utilizando el identificador de compilación devuelto en el paso anterior:

```
oc describe build mymq-1
```

4. Desplegar un gestor de colas, utilizando la nueva imagen

Siga los pasos descritos en [“Despliegue de un gestor de colas utilizando la CLI de OpenShift”](#) en la [página 19](#), añadiendo la nueva imagen personalizada al YAML.

Puede añadir el siguiente fragmento de YAML al QueueManager YAML normal, donde *my-namespace* es el proyecto/espacio de nombres OpenShift que está utilizando y *image* es el nombre de la imagen que ha creado anteriormente (por ejemplo, "mymq:latest-amd64"):

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

### Tareas relacionadas

[“Despliegue de un gestor de colas utilizando la CLI de OpenShift”](#) en la [página 19](#)

Utilice el recurso personalizado QueueManager para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform utilizando la interfaz de línea de mandatos (CLI). Esta tarea la debe completar un administrador de proyectos

Linux

MQ Adv.

CD

## Despliegue de contenedores certificados de IBM

### MQ utilizando Helm

A partir de IBM MQ 9.1.5.0, la forma recomendada de desplegar un gestor de colas es utilizar el operador IBM MQ . IBM MQ 9.1.5.0 y releases CD anteriores se pueden desplegar utilizando Helm, utilizando las instrucciones siguientes.

### Acerca de esta tarea

### Procedimiento

- [“Preparación del clúster de OpenShift para IBM MQ en OpenShift utilizando Helm”](#) en la [página 24](#).
- [“Despliegue de un gestor de colas utilizando la CLI de Helm”](#) en la [página 25](#).

## MQ en OpenShift utilizando Helm

Prepare el clúster de Red Hat OpenShift Container Platform para que esté preparado para desplegar un gestor de colas utilizando Helm. Esta tarea tiene que completarla un administrador de clúster.

### Antes de empezar

**Nota:** Si está utilizando IBM Cloud Pak for Integration, el instalador debería haber preparado un proyecto OpenShift (espacio de nombres) para que lo utilice con IBM MQ, por lo que es posible que no tenga que seguir estas instrucciones.

Inicie sesión en el clúster con **cloudctl login** (en IBM Cloud Pak for Integration) o **oc login**.

### Procedimiento

1. Asegúrese de haber añadido el repositorio de IBM Helm a la copia local de Helm.  
Por ejemplo, puede ejecutar el comando siguiente:

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

2. Asegúrese de que dispone de un servidor Helm (llamado "Tiller") instalado en el clúster.  
Siga las instrucciones indicadas en [Cómo empezar a utilizar Helm en OpenShift](#) para instalar Helm en el clúster.
3. Asegúrese de que las cuentas de servicio del proyecto OpenShift (espacio de nombres) estén autorizadas a utilizar las restricciones de contexto de seguridad (SCC) adecuadas.

**V 9.1.5** IBM MQ funciona bajo el SCC predeterminado "restringido", por lo que normalmente este paso se puede omitir.

La aplicación de cambios a los SCC tiene que hacerla un administrador de clústeres de OpenShift. Cada versión de gráfica de Helm tiene requisitos diferentes para los SCC, documentados en el archivo README individual de la gráfica de Helm:

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

En cada README hay instrucciones para configurar la autorización de los SCC. Observe que las gráficas de Helm de IBM MQ crean una cuenta de servicio para su propio uso, lo que significa que hay que aplicar los permisos SCC a nivel de "grupo" (en todas las cuentas de servicio del espacio de nombres).

4. Asegúrese de tener un "secreto de extracción de imágenes" válido para extraer las imágenes del registro de contenedores elegido

Las imágenes de IBM MQ Advanced certified container se extraen de un registro de contenedor que efectúa una comprobación de titularidad de licencia. Esta comprobación requiere una clave de titularidad que se almacena en un registro de extracción `docker-registry`. Si aún no tiene una clave de titularidad, siga estas instrucciones para obtenerla y crear un secreto de extracción.

- a) Obtenga la clave de titularidad asignada a su ID.
  - i) Inicie sesión en [MyIBM Container Software Library](#) con el ID de IBM y la contraseña asociados al software autorizado.
  - ii) En la sección *Entidades de titularidad*, seleccione **Copiar clave** para copiar la clave de titularidad en el portapapeles.
- b) Cree el secreto en el espacio de nombres en el que desee desplegar el gestor de colas.



- Ejecute el mandato siguiente, donde *<entitlement-key>* es la clave recuperada en el paso 1, y *<user-email>* es el ID de IBM asociado con el software autorizado.

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

## Qué hacer a continuación

[“Despliegue de un gestor de colas utilizando la CLI de Helm” en la página 25](#)

## Linux > MQ Adv. > V 9.1.4 > CD **Despliegue de un gestor de colas utilizando la CLI de Helm**

Utilice Helm para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform. Esta tarea debe ser completada por un administrador de proyectos.

### Antes de empezar

Hay que instalar Helm V2 y la interfaz de línea de comandos de Red Hat OpenShift Container Platform. Si no está utilizando IBM Cloud Pak for Integration, siga los pasos de [“Preparación del clúster de OpenShift para IBM MQ en OpenShift utilizando Helm” en la página 24](#).

Inicie sesión en el clúster con **cloudctl login** (en IBM Cloud Pak for Integration) o **oc login**.

### Procedimiento

1. Asegúrese de haber añadido el repositorio de IBM Helm a la copia local de Helm.

Por ejemplo, puede ejecutar el comando siguiente:

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

2. Revise las opciones de configuración del gestor de colas

El paso de despliegue incluye los pasos de instalación y configuración. Algunos valores del gestor de colas tienen que configurarse durante el despliegue y, si se modifican, habrá que volver a desplegar.

Puede consultar el archivo README de la gráfica de Helm para obtener los detalles de todas las opciones de despliegue disponibles ejecutando uno de los comandos siguientes:

- En IBM MQ Advanced certified container in IBM Cloud Pak for Integration:

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

- En IBM MQ Advanced certified container:

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

Se suelen necesitar al menos estos parámetros:

- a. Nombre del release. Por ejemplo: *my-release*
  - b. Repositorio Helm remoto. Por ejemplo: *ibm-entitled-charts*
  - c. Gráfica de Helm: Por ejemplo, *ibm-mqadvanced-server-prod* o *ibm-mqadvanced-server-integration-prod*
  - d. Nombre del secreto de extracción de la imagen. Por ejemplo: *entitled-registry*. Tenga en cuenta que esto no es necesario si va a desplegar en el proyecto predefinido de MQ en IBM Cloud Pak for Integration
3. Despliegue un gestor de colas.

Tenga en cuenta que, de forma predeterminada, la gráfica de Helm presupone que se tiene configurada una Clase de almacenamiento predeterminada en el clúster de Red Hat OpenShift Container Platform.

Por ejemplo, para instalar un gestor de colas básico en IBM Cloud Pak for Integration, ejecute este comando:

```
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
```

Se puede especificar cualquier nombre de host en el campo `tls.hostname` (es un campo obligatorio, pero no se usa porque en este ejemplo se genera un nuevo certificado autofirmado)

Para instalar un gestor de colas básico independientemente de IBM Cloud Pak for Integration, se puede ejecutar este comando:

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

### Tareas relacionadas

[“Conexión con un gestor de colas desplegado en un clúster de OpenShift” en la página 30](#)

Conjunto de ejemplos de configuración para conectarse con un gestor de colas desplegado en un clúster de Red Hat OpenShift.

[“Conexión con el IBM MQ Console desplegado en un clúster de OpenShift” en la página 32](#)

Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

## **Despliegue de un gestor de colas con IBM Cloud File Storage, utilizando la CLI de Helm**

Escenario de ejemplo para utilizar Helm para desplegar un gestor de colas en un clúster Red Hat OpenShift on IBM Cloud , utilizando IBM Cloud File Storage. Esta tarea la debe completar un administrador de proyectos

### Antes de empezar

Hay que instalar Helm V2 y la interfaz de línea de comandos de Red Hat OpenShift Container Platform. Si no está utilizando IBM Cloud Pak for Integration, siga los pasos de [“Preparación del clúster de OpenShift para IBM MQ en OpenShift utilizando Helm” en la página 24.](#)

Inicie sesión en el clúster con **cloudctl login** (en IBM Cloud Pak for Integration) o **oc login**.

### Procedimiento

1. Asegúrese de que ha añadido el repositorio de IBM Helm a la copia local de Helm.  
Por ejemplo, puede ejecutar el comando siguiente:

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

2. Despliegue un gestor de colas.

Al utilizar IBM Cloud File Storage, normalmente verá los mejores resultados utilizando la clase de almacenamiento `ibmc-file-gold-gid` . Esta clase de almacenamiento habilita el almacenamiento que pueden grabar los usuarios en el grupo de sistemas de archivos correcto.

Por ejemplo, para instalar un gestor de colas básico en IBM Cloud Pak for Integration, ejecute este comando:

```
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}
```

Puede especificar cualquier nombre de host en el campo `tls.hostname` (es un campo obligatorio pero no se utiliza aquí, porque en este ejemplo estamos generando un nuevo certificado autofirmado).

Para instalar un gestor de colas básico independientemente de IBM Cloud Pak for Integration, se puede ejecutar este comando:

```
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}
```

### Tareas relacionadas

[“Conexión con un gestor de colas desplegado en un clúster de OpenShift” en la página 30](#)

Conjunto de ejemplos de configuración para conectarse con un gestor de colas desplegado en un clúster de Red Hat OpenShift.

[“Conexión con el IBM MQ Console desplegado en un clúster de OpenShift” en la página 32](#)

Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

Linux

MQ Adv.

CD

## Despliegue de releases de CD anteriores de IBM MQ en un clúster IBM Cloud Private

Para las versiones de CD de IBM MQ anteriores a 9.1.4, utilice la consola de gestión de IBM Cloud Private para desplegar un gestor de colas en IBM Cloud Private.

### Antes de empezar



**Atención:** **V 9.1.4** Este despliegue no está soportado en IBM MQ 9.1.4 ni posteriores.

En esta tarea se presupone que ya ha [añadido una imagen de IBM MQ a un clúster de IBM Cloud Private](#).

El archivo `README.md` de la gráfica Helm está disponible en la entrada de catálogo de IBM Cloud Private, que se visualiza tras completar [este subpaso](#), o por línea de comandos [añadiendo el repositorio \*\*local-charts\*\* de IBM Cloud Private como un repositorio Helm remoto](#) y ejecutando este comando:

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

Hay que tener una [PodSecurityPolicy](#) o una [SecurityContextConstraint](#) (para IBM Cloud Private en Red Hat OpenShift) que soporte el contexto de seguridad necesario. Los detalles, incluidos ejemplos, se pueden encontrar en el archivo `README.md` de la gráfica de Helm.

Los detalles de cómo configurar el release de Helm también se pueden encontrar en el archivo de gráfica de Helm `README.md`.

### Nota:

- Si va a desplegar en un entorno IBM Cloud Private que no soporta de forma predeterminada los valores de seguridad necesarios, habilite el despliegue siguiendo las instrucciones en [Despliegue de](#)

diagramas de Helm que requieren privilegios elevados en un espacio de nombres no predeterminado en la documentación de producto de IBM Cloud Private.

- Si utiliza SELinux, hay que cumplir los requisitos de IBM MQ descritos en [Soporte de IBM MQ para SELinux en Red Hat Enterprise Linux](#).

## Acerca de esta tarea

IBM Cloud Private ofrece una plataforma para la gestión de aplicaciones contenerizadas locales. Una vez añadida una imagen de IBM MQ a un clúster de IBM Cloud Private, se puede utilizar la consola de gestión de IBM Cloud Private o la línea de comando para desplegar un gestor de colas.

## Procedimiento

- Utilización de la consola de gestión de IBM Cloud Private
  - a) Abra la consola de gestión de IBM Cloud Private en un navegador web y pulse **Catálogo**.  
Consulte [Acceso al clúster de IBM Cloud Private utilizando la consola de gestión](#) en la documentación del producto IBM Cloud Private.
  - b) Seleccione el gráfico `ibm-mqadvanced-server-prod` en la lista.
  - c) Seleccione **Configurar** y siga estos pasos de configuración:
    - a. Especifique un nombre de release.
    - b. Lea y acepte los acuerdos de licencia.
    - c. En la sección **dataPVC**, establezca **storageclass** a la clase de almacenamiento deseada. Déjelo en blanco para seleccionar la clase de almacenamiento predeterminada.
    - d. En la sección **image**, establezca el repositorio en la vía de acceso de la imagen completa. Por ejemplo:

```
mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod
```
    - e. En la sección **image**, establezca el código en la imagen de código. Por ejemplo:

```
9.1.3.0-r1
```
    - f. Si necesita un secreto de extracción de Kubernetes para acceder al registro de imágenes, añádale en **pullSecret**.
    - g. En la sección **queueManager**, establezca el nombre del gestor de colas.
  - d) Pulse **Instalar** para desplegar el gestor de colas como un *Release de Helm*.
- Utilización de la línea de mandatos
  - a) Configure **cloudctl** para acceder al clúster de IBM Cloud Private.  
Consulte [Instalación de la CLI de IBM Cloud Private](#) en la documentación de producto de IBM Cloud Private.
  - b) Asegúrese de haber [añadido el repositorio \*\*local-charts\*\* de IBM Cloud Private como repositorio Helm remoto](#).
  - c) Instale la gráfica.  
Ejecute el comando siguiente, especificando estos parámetros:
    - a. Nombre de release (por ejemplo, `my-release`)
    - b. Nombre del repositorio de helm remoto que contiene la gráfica de `ibm-mqadvanced-server-prod` (por ejemplo, `my-repo`)
    - c. Repositorio de imágenes (por ejemplo, `mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod`)

d. Etiqueta de imagen (por ejemplo 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
```

### Tareas relacionadas

“Despliegue de un gestor de colas utilizando la CLI de Helm” en la [página 25](#)

Utilice Helm para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform. Esta tarea debe ser completada por un administrador de proyectos.

“Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Private” en la [página 29](#)

Para las versiones de CD de IBM MQ anteriores a 9.1.4, prepare el clúster de IBM Cloud Private para desplegar una imagen preparada para producción para IBM MQ.

“Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Kubernetes Service” en la [página 30](#)

Para las versiones de CD de IBM MQ anteriores a 9.1.4, importe una imagen preparada para producción para IBM MQ en IBM Cloud Kubernetes Service.

Linux

MQ Adv.

CD

## Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Private

Para las versiones de CD de IBM MQ anteriores a 9.1.4, prepare el clúster de IBM Cloud Private para desplegar una imagen preparada para producción para IBM MQ.

### Acerca de esta tarea



**Atención:** **V 9.1.4** Esta importación no está soportada en IBM MQ 9.1.4 ni posteriores.

Puede descargar una imagen de IBM MQ en Passport Advantage e importarla a un contenedor de IBM Cloud Private.

### Procedimiento

1. Descargue la imagen de IBM MQ más reciente en [Passport Advantage y Passport Advantage Express sitio web](#).

Para obtener más información sobre las descargas disponibles, vaya a [Descarga de IBM MQ 9.1](#), a continuación, pulse la pestaña del release que desee descargar. El nombre y el número del componente que se descarga se listan en una tabla.

2. Importe el archivo de archivado descargado a IBM Cloud Private.

Consulte [Adición de software de IBM al catálogo de IBM Cloud Private Catalog](#) en la documentación de producto de IBM Cloud Private.

### Qué hacer a continuación

Ahora ya está listo para [Desplegar un gestor de colas en IBM Cloud Private](#).

### Tareas relacionadas

“Despliegue de un gestor de colas utilizando la CLI de Helm” en la [página 25](#)

Utilice Helm para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform. Esta tarea debe ser completada por un administrador de proyectos.

“Despliegue de releases de CD anteriores de IBM MQ en un clúster IBM Cloud Private” en la [página 27](#)

Para las versiones de CD de IBM MQ anteriores a 9.1.4, utilice la consola de gestión de IBM Cloud Private para desplegar un gestor de colas en IBM Cloud Private.

“Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Kubernetes Service” en la [página 30](#)

Para las versiones de CD de IBM MQ anteriores a 9.1.4, importe una imagen preparada para producción para IBM MQ en IBM Cloud Kubernetes Service.

## Linux > MQ Adv. > CD Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Kubernetes Service

Para las versiones de CD de IBM MQ anteriores a 9.1.4, importe una imagen preparada para producción para IBM MQ en IBM Cloud Kubernetes Service.

### Acerca de esta tarea



**Atención:** **V 9.1.4** Esta importación no está soportada en IBM MQ 9.1.4 ni posteriores.

Puede descargar una imagen IBM MQ desde Passport Advantage e importarla a un clúster de IBM Cloud Kubernetes Service.

### Procedimiento

1. Descargue la imagen de IBM MQ más reciente en [Passport Advantage y Passport Advantage Express sitio web](#).

Para obtener más información sobre las descargas disponibles, vaya a [Descarga de IBM MQ 9.1](#), a continuación, pulse la pestaña del release que desee descargar. El nombre y el número del componente que se descarga se listan en una tabla.

2. Importe el archivo de archivado descargado a IBM Cloud Kubernetes Service.

Consulte [Ejecución de imágenes de IBM Cloud Private en contenedores públicos de Kubernetes](#).

### Tareas relacionadas

“Despliegue de un gestor de colas utilizando la CLI de Helm” en la página 25

Utilice Helm para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform. Esta tarea debe ser completada por un administrador de proyectos.

“Despliegue de releases de CD anteriores de IBM MQ en un clúster IBM Cloud Private” en la página 27

Para las versiones de CD de IBM MQ anteriores a 9.1.4, utilice la consola de gestión de IBM Cloud Private para desplegar un gestor de colas en IBM Cloud Private.

“Adición de anteriores releases de CD de una imagen de IBM MQ en un clúster IBM Cloud Private” en la página 29

Para las versiones de CD de IBM MQ anteriores a 9.1.4, prepare el clúster de IBM Cloud Private para desplegar una imagen preparada para producción para IBM MQ.

## Linux > MQ Adv. > V 9.1.4 > CD Conexión con un gestor de colas desplegado en un clúster de OpenShift

Conjunto de ejemplos de configuración para conectarse con un gestor de colas desplegado en un clúster de Red Hat OpenShift.

### Acerca de esta tarea

Se necesita una Ruta OpenShift para conectar una aplicación con un gestor de colas de IBM MQ desde fuera de un clúster de Red Hat OpenShift.

Hay que habilitar TLS en el gestor de colas IBM MQ y en la aplicación cliente, porque la Indicación de Nombre de Servidor (SNI) solo está disponible en el protocolo TLS. El direccionador de Red Hat OpenShift Container Platform utiliza SNI para direccionar solicitudes al gestor de colas de IBM MQ .

La configuración necesaria de la Ruta OpenShift dependerá del comportamiento SNI de la aplicación cliente.

Para establecer la cabecera SNI como TLS 1.2 o superior, se debe utilizar una CipherSpec o CipherSuite para la comunicación TLS.

SNI se establece en el canal MQ si se cumplen las condiciones siguientes:

- El cliente C de IBM MQ es V8 o posterior.
- El cliente Java/JMS es V9.1.1 o posterior, y la instalación de Java da soporte a la clase `javax.net.ssl.SNIHostName`.
- El cliente .NET está en modalidad no gestionada.

SNI se establece en el nombre de host si se proporciona un nombre de host como nombre de conexión y se cumplen las condiciones siguientes:

- El cliente .NET está en modalidad gestionada.
- Se utiliza el cliente AMQP o XR.
- Los clientes Java/JMS se utilizan con **AllowOutboundSNI** establecido en NO.

El SNI no está establecido y está en blanco en las condiciones siguientes:

- El cliente C de IBM MQ es V7.5 o anterior.
- IBM MQ C Client se utiliza con **AllowOutboundSNI** establecido en NO.
- Los clientes Java/JMS se utilizan con una instalación Java que no da soporte a la clase `javax.net.ssl.SNIHostName`.

## Ejemplo

### **Rutas OpenShift basadas en nombre de host: En aplicaciones cliente que establecen la SNI al nombre de host**

Los siguientes diagramas de Helm crean automáticamente una ruta OpenShift basada en el nombre de host para conectar una aplicación con un gestor de colas de IBM MQ. Las aplicaciones cliente que establecen la SNI al nombre de host pueden utilizar esta ruta OpenShift.

- `ibm-mqadvanced-server-dev`
- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod` en IBM Cloud Pak for Integration.

Si no utiliza estos diagramas y necesita crear su propio nombre de host basado en una ruta OpenShift, puede aplicar el siguiente `yaml` en el clúster:

```
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
```

### **Rutas OpenShift basadas en canal MQ: En aplicaciones cliente que establecen la SNI al canal MQ**

Las aplicaciones cliente que establecen la SNI al canal MQ requieren crear una ruta OpenShift por cada canal con el que deseen conectarse. También hay que usar nombres de canal exclusivos en el clúster Red Hat OpenShift para permitir el direccionamiento al gestor de colas que corresponda.

Para determinar el nombre de host necesario para cada una de las nuevas rutas OpenShift, hay que correlacionar cada nombre de canal con una dirección SNI, tal como se documenta aquí: <https://www.ibm.com/support/pages/ibm-websphere-mq-how-does-mq-provide-multiple-certificates-certlbl-capability>

Luego hay que crear una ruta OpenShift (por cada canal) aplicando los siguientes yaml en el clúster:

```
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
```

### **Configuración de los detalles de conexión de la aplicación cliente**

Se puede determinar el nombre de host que hay que usar en la conexión cliente con este comando:

```
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}"
```

El puerto de la conexión cliente tiene que establecerse al puerto utilizado por el direccionador de OpenShift Container Platform (OCP), que suele ser 443.

### **Tareas relacionadas**

[“Despliegue de un gestor de colas utilizando la CLI de Helm”](#) en la página 25

Utilice Helm para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform. Esta tarea debe ser completada por un administrador de proyectos.

[“Conexión con el IBM MQ Console desplegado en un clúster de OpenShift”](#) en la página 32

Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

Linux

MQ Adv.

V 9.1.4

CD

## **Conexión con el IBM MQ Console desplegado en un clúster de OpenShift**

Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

### **Acerca de esta tarea**

Si está utilizando el operador IBM MQ , el URL IBM MQ Console se puede encontrar en la página de detalles de QueueManager en la consola web de OpenShift o en IBM Cloud Pak for Integration Platform Navigator. De forma alternativa, se puede encontrar en la CLI de OpenShift ejecutando el mandato siguiente:

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

### **Ejemplo**

Los siguientes diagramas de Helm crean automáticamente una ruta OpenShift para acceder a IBM MQ Console

- ibm-mqadvanced-server-dev
- ibm-mqadvanced-server-integration-prod en IBM Cloud Pak for Integration.

Puede obtener el nombre de host de la ruta de OpenShift con este comando:

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



Puede acceder a IBM MQ Console a través de este URL:

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

### Tareas relacionadas

“Despliegue de un gestor de colas utilizando la CLI de Helm” en la página 25

Utilice Helm para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform. Esta tarea debe ser completada por un administrador de proyectos.

“Conexión con un gestor de colas desplegado en un clúster de OpenShift” en la página 30

Conjunto de ejemplos de configuración para conectarse con un gestor de colas desplegado en un clúster de Red Hat OpenShift.

Linux

MQ Adv.

CD

## Copia de seguridad y restauración de la configuración del gestor de colas utilizando la CLI de OpenShift

La copia de seguridad de la configuración del gestor de colas puede ayudar a regenerar un gestor de colas a partir de sus definiciones si se pierde su configuración. Este procedimiento no hace copia de seguridad de los datos de registro cronológico del gestor de colas. Debido a la naturaleza temporal de los mensajes, lo más probable es que los datos del registro histórico sean irrelevantes en el momento de la restauración.

### Antes de empezar

Inicie sesión en el clúster con **cloudctl login** (en IBM Cloud Pak for Integration) o **oc login**.

### Procedimiento

- Haga una copia de seguridad de la configuración del gestor de colas.

Puede utilizar el comando **dmpmqcfig** para volcar la configuración de un gestor de colas de IBM MQ.

- Obtenga el nombre del pod para el gestor de colas.

Por ejemplo, si utiliza el operador, puede ejecutar el mandato siguiente, donde *queue\_manager\_name* es el nombre del recurso QueueManager :

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

Por ejemplo, si utiliza Helm, podría ejecutar el mandato siguiente, donde *release\_name* es el nombre del release de Helm .

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

- Ejecute el comando **dmpmqcfig** en el pod, dirigiendo la salida a un archivo de la máquina local.

**dmpmqcfig** genera la configuración de MQSC del gestor de colas.

```
oc exec -it pod_name -- dmpmqcfig > backup.mqsc
```

- Restablezca la configuración del gestor de colas.

Tras haber seguido el procedimiento de copia de seguridad descrito en el paso anterior, debería tener el archivo `backup.mqsc`, que contiene la configuración del gestor de colas. Puede restaurar la configuración aplicando este archivo a un nuevo gestor de colas.

- Obtenga el nombre del pod para el gestor de colas.

Por ejemplo, si utiliza el operador, puede ejecutar el mandato siguiente, donde *queue\_manager\_name* es el nombre del recurso QueueManager :

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

Por ejemplo, si utiliza Helm, podría ejecutar el mandato siguiente, donde *release\_name* es el nombre del release de Helm .

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

- b) Ejecute el comando **runmqsc** en el pod, redirigiendo a la entrada el contenido del archivo `backup.mqsc`.

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

## Creación de su propio contenedor de IBM MQ

Desarrolle un contenedor autocreado, conocido anteriormente como "Imagen de contenedor Docker". Esta es la solución de contenedor más flexible, pero requiere tener sólidos conocimientos técnicos en configuración de contenedores y "tener en propiedad" el contenedor resultante.


### Antes de empezar

Antes de desarrollar su propio contenedor, considere si puede utilizar alguno de los contenedores preempaquetados que proporciona IBM. Consulte [IBM MQ en contenedores](#)

### Acerca de esta tarea

Cuando se empaqueta IBM MQ como una imagen de contenedor, los cambios en las aplicaciones se pueden desplegar en sistemas de prueba y de preproducción con rapidez y facilidad. Esto puede ser una gran ventaja para la entrega continua de las empresas.

### Procedimiento

- Para obtener información sobre cómo crear una imagen de contenedor de IBM MQ utilizando Docker, consulte los subtemas siguientes:
  -  ["Soporte para la creación de imágenes y gráficos de contenedor de IBM MQ propios" en la página 8](#)
  - ["Planificación de una imagen propia de gestor de colas de IBM MQ utilizando un contenedor" en la página 34](#)
  - ["Creación de un ejemplo de imagen de gestor de colas de IBM MQ usando Docker" en la página 35](#)
  - ["Ejecución de aplicaciones de enlaces locales en contenedores separados" en la página 38](#)

### Conceptos relacionados

[IBM MQ en contenedores](#)

## Planificación de una imagen propia de gestor de colas de IBM MQ utilizando un contenedor

Hay varios requisitos por tener en cuenta cuando se ejecuta un gestor de colas de IBM MQ en un contenedor. La imagen de contenedor de ejemplo proporciona una forma de manejar estos requisitos, pero si desea utilizar su propia imagen, habrá de tener en cuenta cómo se manejan dichos requisitos.

### Supervisión del proceso

Al ejecutar un contenedor, básicamente está ejecutando un proceso único (PID 1 dentro del contenedor), que luego puede generar procesos hijo.


Si el proceso principal finaliza, el entorno de ejecución del contenedor para el contenedor. Un gestor de colas de IBM MQ requiere la ejecución de varios procesos en segundo plano.

Por este motivo, debe asegurarse de que el proceso principal permanece activo mientras el gestor de colas se está ejecutando. Es aconsejable comprobar que el gestor de colas está activo desde este proceso, por ejemplo realizando consultas administrativas.

## Llenar /var/mqm

Los contenedores tienen que configurarse con el volumen /var/mqm.

Al hacerlo, el directorio del volumen está vacío cuando el contenedor se inicia por primera vez. Este directorio suele llenarse en tiempo de instalación, pero la instalación y el entorno de ejecución son entornos independientes cuando se utiliza un contenedor.

 Para resolver esto, cuando se inicia el contenedor, puede utilizar el mandato `crtmqdir` para llenar /var/mqm cuando se ejecuta por primera vez.

## Creación de un ejemplo de imagen de gestor de colas de IBM MQ usando Docker

Utilice esta información para crear un ejemplo de imagen de contenedor para ejecutar un gestor de colas de IBM MQ en un contenedor.

### Acerca de esta tarea

En primer lugar, se crea una imagen base que contenga un sistema de archivos de Red Hat Universal Base Image y una instalación limpia de IBM MQ.

En segundo lugar, se crea otra capa de imagen de contenedor encima de la base, que añade cierta configuración de IBM MQ para permitir la seguridad básica de ID de usuario y contraseña.

Por último, se ejecuta un contenedor usando esta imagen como sistema de archivos, con el contenido de /var/mqm proporcionado por un volumen específico de contenedor en el sistema de archivos anfitrión.

### Procedimiento

- Para obtener información sobre de cómo crear un ejemplo de imagen de contenedor para ejecutar un gestor de colas de IBM MQ en un contenedor, consulte los subtemas siguientes:
  - [“Creación de una imagen de gestor de colas de IBM MQ base de muestra”](#) en la página 35
  - [“Creación de una imagen de gestor de colas de IBM MQ configurada de muestra”](#) en la página 36

## Creación de una imagen de gestor de colas de IBM MQ base de muestra

Para poder utilizar IBM MQ en su propia imagen de contenedor, primero tiene que crear una imagen base con una instalación de IBM MQ limpia. Los pasos siguientes muestran cómo crear un ejemplo de imagen base utilizando el código de ejemplo alojado en GitHub.

### Procedimiento

- Utilice los archivos make proporcionados en el repositorio [mq-container GitHub](#) para crear la imagen de contenedor de producción.
  - Siga las instrucciones de [Creación de una imagen de contenedor](#) en GitHub.

### Resultados

Ahora tiene una imagen de contenedor base con IBM MQ instalado.

## Creación de una imagen de gestor de colas de IBM MQ configurada de muestra

Después de crear la imagen de contenedor IBM MQ base genérica, debe aplicar su propia configuración para permitir el acceso seguro. Para ello, cree su propia capa de imagen de contenedor utilizando la imagen genérica como padre.

### Antes de empezar

Para una imagen de IBM MQ 9.1, no puede configurar el acceso seguro utilizando la restricción de contexto de seguridad (SCC) de Red Hat OpenShift Container Platform "restringido". La SCC "restringida" utiliza ID de usuario aleatorios e impide el escalamiento de privilegios cambiando a un usuario diferente. El instalador basado en RPM de IBM MQ 9.1 se basa en un usuario y grupo de mqm y también utiliza bits de setuid en programas ejecutables.

Esta restricción se elimina en IBM MQ 9.2.

### Procedimiento

1. Cree un directorio y añádale un archivo llamado `config.mqsc` con el siguiente contenido:

```
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)
```

Tenga en cuenta que el ejemplo anterior simple la autenticación simple de ID de usuario y contraseña. No obstante, puede aplicar cualquier configuración de seguridad que su empresa necesite.

2. Cree un archivo denominado `Dockerfile` con el siguiente contenido:

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

donde:

- `johndoe` es el ID de usuario que desea añadir
- `passwd` es la contraseña original

3. Cree su propia imagen de contenedor personalizada con este comando:

```
sudo docker build -t mymq .
```

donde `.` es el directorio que contiene los dos archivos que acaba de crear.

A continuación, Docker crea un contenedor temporal utilizando esa imagen y ejecuta los mandatos restantes.

El mandato **RUN** añade un usuario llamado `johndoe` con la contraseña `passwd` y el mandato **COPY** añade el archivo `config.mqsc` en una ubicación específica conocida por la imagen padre.

**Nota:** En Red Hat Enterprise Linux (RHEL) se usa el comando **docker** (RHEL V7) o **podman** (RHEL V7 o RHEL V8). En el caso de **podman**, no es necesario **sudo** al principio del comando.

4. Ejecute la nueva imagen personalizada para crear un contenedor, con la imagen de disco que acaba de crear.

La capa de imagen nueva no especificaba ningún mandato determinado a ejecutar, de modo que se ha heredado de la imagen padre. El punto de entrada del padre (el código está disponible en GitHub):

- Crea un gestor de colas
- Inicia el gestor de colas

- Crea un escucha predeterminado
- A continuación, ejecuta los mandatos MQSC desde `/etc/mqm/config.mqsc`.

Emita los mandatos siguientes para ejecutar la nueva imagen personalizada:

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

donde:

#### Primer parámetro env

Pasa una variable de entorno en el contenedor, que reconoce la aceptación por parte del usuario de la licencia para IBM IBM WebSphere MQ. También puede establecer la variable `LICENSE` para ver la licencia.

Consulte [Información de licencia de IBM MQ](#) para obtener más detalles acerca de las licencias de IBM MQ.

#### Segundo parámetro env

Establece el nombre del gestor de colas que está utilizando.

#### Parámetro volume

Indica al contenedor que lo que MQ grabe en `/var/mqm` debe grabarse en realidad en `/var/example` en el host.

Esta opción significa que puede suprimir fácilmente el contenedor posteriormente y seguir conservando los datos persistentes. Esta opción también hace que sea más fácil ver los archivos de registro.

#### Parámetro publish

Correlaciona los puertos del host con los puertos del contenedor. El contenedor se ejecuta de forma predeterminada con su propia dirección IP interna, lo que significa que debe correlacionar específicamente los puertos que desea exponer.

En este ejemplo, eso implica correlacionar el puerto 1414 del host con el puerto 1414 del contenedor.

#### Parámetro detach

Ejecuta el contenedor en segundo plano.

## Resultados

Ha creado una imagen de contenedor configurada y podrá ver los contenedores en ejecución con el comando `docker ps`. Puede ver los procesos de IBM MQ que se ejecutan en el contenedor mediante el mandato `top` de docker.



#### Atención:

Puede ver los registros de un contenedor mediante el mandato `docker logs ${CONTAINER_ID}`.

## Qué hacer a continuación

- si el contenedor no está cerrado cuando se utiliza el mandato `ps` de docker, es posible que el contenedor haya fallado. Puede ver los contenedores anómalos utilizando el mandato `docker ps -a`.
- Cuando se utiliza el comando `docker ps -a`, se muestra el ID de contenedor. Este ID también se muestra cuando se ejecuta el comando `docker run`.
- Puede ver los registros de un contenedor utilizando el mandato `docker logs ${CONTAINER_ID}`.
- Puede establecer el número máximo de archivos abiertos utilizando el mandato `sysctl fs.file-max=524288`.

## V 9.1.0 Ejecución de aplicaciones de enlaces locales en contenedores separados

Con la compartición de espacios de nombres de proceso entre contenedores en Docker, se pueden ejecutar aplicaciones que necesiten una conexión de enlace local con IBM MQ en contenedores separados desde el gestor de colas de IBM MQ.

### Acerca de esta tarea

Esta funcionalidad está soportada en gestores de colas IBM MQ 9.0.3 y posteriores.

Hay que atenerse a las restricciones siguientes:

- Hay que compartir el espacio de nombres de PID del proceso con el argumento `--pid`.
- Hay que compartir el espacio de nombres de IPC del proceso con el argumento `--ipc`.
- Hay que:
  1. Compartir el espacio de nombres de UTS del contenedor con el host usando el argumento `--uts`, o bien
  2. asegurarse de que los contenedores tengan el mismo nombre de host con los argumentos `-h` o `--hostname`.
- Debe montar el directorio de datos IBM MQ en un volumen que esté disponible para todos los contenedores bajo el directorio `/var/mqm`.

Puede probar esta funcionalidad siguiendo los pasos siguientes en un sistema Linux que ya tenga Docker instalado.

El ejemplo siguiente utiliza la imagen de contenedor de IBM MQ de ejemplo. Puede encontrar detalles de esta imagen en [Github](#).

### Procedimiento

1. Cree un directorio temporal para que actúe como volumen ejecutando el siguiente mandato:

```
mkdir /tmp/dockerVolume
```

2. Cree el gestor de colas (QM1) en un contenedor de nombre `sharedNamespace` ejecutando el mandato siguiente:

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

3. Inicie un segundo contenedor denominado `secondaryContainer`, basado en `ibmcom/mq`, pero no cree un gestor de colas, emitiendo el mandato siguiente:

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

4. Ejecute el mandato **dspm** en el segundo contenedor, para ver el estado de ambos gestores de colas, ejecutando el siguiente mandato:

```
docker exec secondaryContainer dspmq
```

5. Ejecute el mandato siguiente para procesar mandatos MQSC contra el gestor de colas que se ejecuta en el otro contenedor:

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

## Resultados

Ahora tiene aplicaciones locales que se ejecutan en contenedores separados y ahora puede ejecutar correctamente mandatos como **dspmq**, **amqsput**, **amqsgety** **runmqsc** como enlaces locales al gestor de colas QM1 desde el contenedor secundario.

Si no ve el resultado esperado, consulte [“Resolución de problemas en las aplicaciones de espacio de nombres”](#) en la página 39 para obtener más información.

## V 9.1.0 Resolución de problemas en las aplicaciones de espacio de nombres

Al utilizar espacios de nombres compartidos, debe asegurarse de que comparte todos los espacios de nombres (IPC, PID y UTS/nombre de host) y los volúmenes montados; de lo contrario, las aplicaciones no funcionarán.

Consulte [“Ejecución de aplicaciones de enlaces locales en contenedores separados”](#) en la página 38 para obtener la lista de restricciones a la que hay que atenerse.

Si la aplicación no cumple todas las restricciones listadas, podrían surgir problemas al iniciarse el contenedor y no se tendrá la funcionalidad que cabe esperar.

La lista siguiente describe algunas de las causas comunes y el comportamiento que probablemente se observe si no se cumple alguna de las restricciones.

- Si olvida compartir el espacio de nombres (UTS/PID/IPC) o el nombre de host de los contenedores y monta el volumen, el contenedor podrá ver el gestor de colas pero no interactuar con el gestor de colas.
  - En los mandatos **dspmq** se ve lo siguiente:

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

- En los mandatos **runmqsc** u otros mandatos que intenten conectar con el gestor de colas, es probable que se reciba un mensaje de error 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
```

- Si comparte todos los espacios de nombres necesarios pero no monta un volumen compartido en el directorio `/var/mqm` y tiene una vía de acceso de datos IBM MQ válida, los mandatos también reciben mensajes de error AMQ8146 .

Sin embargo, **dspmq** no podrá ver el gestor de colas y devolverá una respuesta en blanco:

```
docker exec container dspmq
```

- Si comparte todos los espacios de nombres necesarios pero no monta un volumen compartido en el directorio `/var/mqm` y no tiene una vía de acceso de datos de IBM MQ válida (o ninguna vía de acceso de datos de IBM MQ ), verá varios errores ya que la vía de acceso de datos es un componente clave de una instalación de IBM MQ . Sin la ruta de datos, IBM MQ no puede funcionar.

Si ejecuta alguno de los mandatos siguientes y ve respuestas similares a las que se muestran en estos ejemplos, debe verificar que ha montado el directorio o ha creado un directorio de datos 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 **Referencia de API para el operador IBM MQ**

IBM MQ proporciona un operador Kubernetes , que proporciona integración nativa con OpenShift Container Platform.

V 9.1.5 Linux MQ Adv. CD **Referencia de API para mq.ibm.com/v1beta1**

La API v1beta1 se puede utilizar para crear y gestionar recursos de QueueManager .

V 9.1.5 Linux MQ Adv. CD **Referencia de licencia para mq.ibm.com/v1beta1**

El campo `spec.license.license` debe contener el identificador de licencia para la licencia que está aceptando. Los valores válidos son los siguientes:

Valor de <code>spec.license.license</code>	Valor de <code>spec.license.use</code>	Información de licencia
L-RJON-BN7PN3	Production o NonProduction	<a href="#">IBM Cloud Pak for Integration 2020.2</a>
L-RJON-BPHL2Y		<a href="#">IBM Cloud Pak for Integration Edición limitada 2020.2</a>
L-APIG-BJAKBF	Production o Development	<a href="#">IBM MQ Advanced 9.1.5</a>
L-APIG-BM7GDH	Development	<a href="#">IBM MQ Advanced for Developers 9.1.5</a>

Tenga en cuenta que se ha especificado la licencia *versión* , que no siempre es la misma que la versión de IBM MQ.



**QueueManager (mq.ibm.com/v1beta1)****QueueManager**

Un QueueManager es un servidor IBM MQ que proporciona servicios de cola y publicación/suscripción a aplicaciones.

Campo	Descripción
Serie apiVersion	APIVersion define el esquema con versión de esta representación de un objeto. Los servidores deben convertir los esquemas reconocidos al valor interno más reciente y pueden rechazar valores no reconocidos. Más información: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a> .
Serie kind	El tipo es un valor de serie que representa el recurso REST que representa este objeto. Los servidores pueden inferirlo desde el punto final al que el cliente envía las solicitudes. No se puede actualizar. En CamelCase. Más información: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-classes">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-classes</a> .
metadata	
spec “Especificación de QueueManager” en la página 45	El estado deseado de QueueManager.
status “Estado del gestor de colas” en la página 46	El estado observado de QueueManager.

**Disponibilidad**

Valores de disponibilidad para el gestor de colas, como por ejemplo si se debe utilizar o no un par activo-en espera.

Aparece en:

- “Configuración de QueueManager” en la página 43

Campo	Descripción
Serie type	El tipo de disponibilidad a utilizar. Utilice "SingleInstance" para un único pod, que Kubernetes reiniciará automáticamente (en algunos casos). Utilice "MultiInstance" para un par de Pods, uno de los cuales es el gestor de colas "activo" y el otro está en espera. Consulte <a href="#">Alta disponibilidad para IBM MQ en contenedores</a> en la versión más reciente de IBM MQ.

**Licencia**

Valores que controlan la aceptación de la licencia y qué métricas de licencia utilizar.

Aparece en:

- “Especificación de QueueManager” en la página 45

Campo	Descripción
Serie use	Valor que controla cómo se utilizará el software, donde la licencia admite varios usos. Consulte <a href="https://ibm.biz/BdqvCF">https://ibm.biz/BdqvCF</a> para ver los valores válidos.
accept boolean	Si acepta o no la licencia asociada con este software (obligatorio).

Campo	Descripción
Serie license	El identificador de la licencia que está aceptando. Debe ser el identificador de licencia correcto para la versión de MQ que está utilizando. Consulte <a href="https://ibm.biz/BdqvCF">https://ibm.biz/BdqvCF</a> para ver los valores válidos.
Serie metric	Valor que especifica qué métrica de licencia se debe utilizar. Por ejemplo, "ProcessorValueUnit", "VirtualProcessorCore" o "ManagedVirtualServer".

## Límites

QueueManagerResourceList define los valores de CPU y memoria.

Aparece en:

- [“Recursos” en la página 49](#)

Campo	Descripción
cpu	
memory	

## Referencia de LocalObject

LocalObjectReference incluye información suficiente para que pueda localizar el objeto de referencia en el mismo espacio de nombres.

Aparece en:

- [“Especificación de QueueManager” en la página 45](#)

Campo	Descripción
Serie name	Nombre del referente. Más información: <a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names</a> HACER: Añadir otros campos útiles. apiVersion, kind, uid?.

## PKI

Valores de infraestructura de claves públicas, para definir claves y certificados para utilizarlos con TLS (Transport Layer Security) o AMS (MQ Advanced Message Security).

Aparece en:

- [“Especificación de QueueManager” en la página 45](#)

Campo	Descripción
Matriz de keys <a href="#">“PKISource” en la página 42</a>	Claves privadas para añadir al repositorio de claves del gestor de colas.
Matriz de trust <a href="#">“PKISource” en la página 42</a>	Certificados para añadir al repositorio de claves del gestor de colas.

## PKISource

PKISource define una fuente de información de infraestructura de claves públicas, como claves o certificados.

Aparece en:

- [“PKI” en la página 42](#)

Campo	Descripción
Serie name	El nombre se utiliza como etiqueta para la clave o el certificado. Debe ser una serie alfanumérica en minúsculas.
secret <a href="#">“Secreto” en la página 49</a>	Proporcione una clave utilizando un secreto de Kubernetes .

## Configuración de QueueManager

La configuración de QueueManager define los valores para el contenedor del gestor de colas y el gestor de colas subyacente.

Aparece en:

- [“Especificación de QueueManager” en la página 45](#)

Campo	Descripción
Serie logFormat	El formato de registro que se debe utilizar para este contenedor. Utilice "JSON" para los registros con formato JSON del contenedor. Utilice "Basic" para mensajes con formato de texto.
metrics <a href="#">“Métricas de QueueManager” en la página 44</a>	Valores para métricas de estilo Prometheus.
readinessProbe <a href="#">“QueueManagerReadinessProbe” en la página 44</a>	Valores que controlan el sondeo de preparación.
resources <a href="#">“Recursos” en la página 49</a>	Valores que controlan los requisitos de recursos.
storage <a href="#">“QueueManagerAlmacenamiento” en la página 48</a>	Valores de almacenamiento para controlar el uso del gestor de colas de volúmenes persistentes y clases de almacenamiento.
availability <a href="#">“Disponibilidad” en la página 41</a>	Valores de disponibilidad para el gestor de colas, como por ejemplo si se debe utilizar o no un par activo-en espera.
Serie imagePullPolicy	Valor que controla cuándo el kubelet intenta extraer la imagen especificada.
livenessProbe <a href="#">“QueueManagerLivenessProbe” en la página 43</a>	Valores que controlan el análisis de actividad.
debug boolean	Indica si se deben registrar o no los mensajes de depuración del código específico del contenedor en el registro del contenedor.
Serie image	La imagen de contenedor que se utilizará.
Serie name	Nombre del gestor de colas de MQ subyacente, si es distinto de metadata.name. Utilice este campo si desea un nombre de gestor de colas que no se ajuste a las reglas de Kubernetes para los nombres (por ejemplo, un nombre que incluya letras mayúsculas).

## QueueManagerLivenessProbe

Valores que controlan el análisis de actividad.

Aparece en:

- [“Configuración de QueueManager” en la página 43](#)

<b>Campo</b>	<b>Descripción</b>
failureThreshold entero	Número mínimo de anomalías consecutivas para que se considere que el sondeo ha fallado después de haber sido satisfactorio.
initialDelaySeconds entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicien los sondeos de actividad. Más información: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds entero	Con qué frecuencia (en segundos) se realizará el análisis.
successThreshold entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio después de haber fallado.
timeoutSeconds entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. Más información: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

### Métricas de QueueManager

Valores para métricas de estilo Prometheus.

Aparece en:

- [“Configuración de QueueManager” en la página 43](#)

<b>Campo</b>	<b>Descripción</b>
enabled boolean	Indica si se debe habilitar o no un punto final de métricas compatible con Prometheus.

### QueueManagerOptionalVolume

Detalles de PersistentVolume para los registros de recuperación de MQ . Necesario cuando se utiliza el gestor de colas de varias instancias.

Aparece en:

- [“QueueManagerAlmacenamiento” en la página 48](#)

<b>Campo</b>	<b>Descripción</b>
Serie class	Clase de almacenamiento a utilizar para este volumen. Sólo es válido si "type" es "persistent-claim".
enabled boolean	Indica si este volumen se debe habilitar o no como un volumen independiente o se debe colocar en el volumen "queueManager" predeterminado.
Serie size	Tamaño del PersistentVolume para pasar a Kubernetes. Sólo es válido si "type" es "persistent-claim".
Serie sizeLimit	Límite de tamaño cuando se utiliza un volumen "efímero". Los archivos se siguen grabando en un directorio temporal, por lo que puede utilizar esta opción para limitar el tamaño. Sólo es válido si type es "efímero".
Serie type	Tipo de volumen a utilizar. Elija ephemeral para crear un volumen "emptyDir" no persistente o persistent-claim para utilizar un volumen persistente.

### QueueManagerReadinessProbe

Valores que controlan el sondeo de preparación.

Aparece en:

- [“Configuración de QueueManager” en la página 43](#)

<b>Campo</b>	<b>Descripción</b>
failureThreshold entero	Número mínimo de anomalías consecutivas para que se considere que el sondeo ha fallado después de haber sido satisfactorio.
initialDelaySeconds entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicien los sondeos de actividad. Más información: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds entero	Con qué frecuencia (en segundos) se realizará el análisis.
successThreshold entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio después de haber fallado.
timeoutSeconds entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. Más información: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

## Especificación de QueueManager

El estado deseado de QueueManager.

Aparece en:

- [“QueueManager” en la página 41](#)

<b>Campo</b>	<b>Descripción</b>
license <a href="#">“Licencia” en la página 41</a>	Valores que controlan la aceptación de la licencia y qué métricas de licencia utilizar.
pki <a href="#">“PKI” en la página 42</a>	Valores de infraestructura de claves públicas, para definir claves y certificados para utilizarlos con TLS (Transport Layer Security) o AMS (MQ Advanced Message Security).
queueManager <a href="#">“Configuración de QueueManager” en la página 43</a>	La configuración de QueueManager define los valores para el contenedor del gestor de colas y el gestor de colas subyacente.
securityContext <a href="#">“SecurityContext” en la página 49</a>	Valores de seguridad para añadir al securityContext del pod del gestor de colas.
tracing <a href="#">“TracingConfig” en la página 51</a>	Valores para rastrear la integración con el panel de control de operaciones de Cloud Pak for Integration.
Serie version	Valor que controla la versión de MQ que se utilizará (necesario). Por ejemplo: "9.1.5.0-r2" especificaría MQ versión 9.1.5.0, utilizando la segunda revisión de la imagen de contenedor. Los arreglos específicos de contenedor se aplican a menudo en revisiones, como por ejemplo arreglos en la imagen base.
affinity	Reglas de afinidad Kubernetes estándar. Para obtener más información, consulte <a href="https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core">https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core</a> .

<b>Campo</b>	<b>Descripción</b>
Matriz de <code>imagePullSecrets</code> “Referencia de LocalObject” en la página 42	Una lista opcional de referencias a secretos en el mismo espacio de nombres para utilizar para extraer cualquiera de las imágenes utilizadas por este QueueManager. Si se especifica, estos secretos se pasarán a implementaciones de extractor individuales para que los utilicen. Por ejemplo, en el caso de docker, solo se respetan los secretos de tipo DockerConfig. Para obtener más información, consulte <a href="https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod">https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod</a> .
<code>template</code> “Plantilla” en la página 50	Plantillas avanzadas para recursos de Kubernetes. La plantilla permite a los usuarios alterar temporalmente cómo IBM MQ genera los recursos de Kubernetes subyacentes, como por ejemplo StatefulSet, Pods y Servicios. Esto es sólo para usuarios avanzados, ya que tiene el potencial de interrumpir el funcionamiento normal de MQ si se utiliza incorrectamente. Los valores especificados en cualquier otro lugar del recurso QueueManager se alterarán temporalmente mediante los valores de la plantilla.
<code>terminationGracePeriod</code> Seconds entero	Duración opcional en segundos que el pod necesita para terminar correctamente. El valor debe ser un entero no negativo. El valor cero indica suprimir inmediatamente. La hora de destino en la que se intenta finalizar el gestor de colas, escalando las fases de la desconexión de la aplicación. Las tareas esenciales de mantenimiento del gestor de colas se interrumpen si es necesario.
<code>web</code> “Configuración de WebServer” en la página 52	Valores para el servidor web de MQ.

## Estado del gestor de colas

El estado observado de QueueManager.

Aparece en:

- “QueueManager” en la página 41

<b>Campo</b>	<b>Descripción</b>
Matriz de endpoints “QueueManagerStatusEndpoint” en la página 47	Información sobre los puntos finales que este gestor de colas está exponiendo, como por ejemplo puntos finales de interfaz de usuario o API.
Serie name	Nombre del gestor de colas.
<code>versions</code> “QueueManagerStatusVersion” en la página 47	Versión de MQ que se está utilizando y otras versiones disponibles en IBM Entitled Registry.
Serie adminUiUrl	URL para la interfaz de usuario de administración.
Matriz de conditions “QueueManagerStatusCondition” en la página 46	Las condiciones representan las últimas observaciones disponibles del estado del gestor de colas.

## QueueManagerStatusCondition

QueueManagerStatusCondition define las condiciones del gestor de colas.

Aparece en:

- “Estado del gestor de colas” en la página 46

Campo	Descripción
Serie message	Mensaje que puede leer el usuario que incluye detalles sobre la última transición.
Serie type	Tipo de condición.
Serie lastTransitionTime	La última vez que la condición ha pasado de un estado a otro.

### QueueManagerStatusEndpoint

QueueManagerStatusEndpoint define los puntos finales para QueueManager.

Aparece en:

- [“Estado del gestor de colas” en la página 46](#)

Campo	Descripción
Serie name	Nombre del punto final.
Serie type	El tipo de punto final, por ejemplo 'UI' para un punto final de interfaz de usuario, 'API' para un punto final de API, 'OpenAPI' para la documentación de la API.
Serie uri	URI para el punto final.

### QueueManagerStatusVersion

Versión de MQ que se está utilizando y otras versiones disponibles en IBM Entitled Registry.

Aparece en:

- [“Estado del gestor de colas” en la página 46](#)

Campo	Descripción
available <a href="#">“QueueManagerStatusVersion Disponible” en la página 47</a>	Otras versiones de MQ disponibles en IBM Entitled Registry.
Serie reconciled	La versión específica de IBM MQ que se está utilizando. Si se especifica una imagen personalizada, es posible que no coincida con la versión de MQ que se está utilizando realmente.

### QueueManagerStatusVersionDisponible

Otras versiones de MQ disponibles en IBM Entitled Registry.

Aparece en:

- [“QueueManagerStatusVersion” en la página 47](#)

Campo	Descripción
channels matriz	Canales que están disponibles para actualizar automáticamente la versión de MQ .
Matriz de versiones <a href="#">“Versiones” en la página 51</a>	Versiones específicas de MQ que están disponibles.

## QueueManagerAlmacenamiento

Valores de almacenamiento para controlar el uso del gestor de colas de volúmenes persistentes y clases de almacenamiento.

Aparece en:

- [“Configuración de QueueManager” en la página 43](#)

Campo	Descripción
<code>persistedData</code> <a href="#">“QueueManagerOptionalVolume” en la página 44</a>	Detalles de PersistentVolume para datos persistentes de MQ , incluida la configuración, las colas y los mensajes. Necesario cuando se utiliza el gestor de colas de varias instancias.
<code>queueManager</code> <a href="#">“Volumen de QueueManager” en la página 48</a>	PersistentVolume predeterminado para los datos que se encuentran normalmente en <code>/var/mqm</code> . Contendrá todos los datos persistentes y registros de recuperación, si no se especifica ningún otro volumen.
<code>recoveryLogs</code> <a href="#">“QueueManagerOptionalVolume” en la página 44</a>	Detalles de PersistentVolume para los registros de recuperación de MQ . Necesario cuando se utiliza el gestor de colas de varias instancias.

## Volumen de QueueManager

PersistentVolume predeterminado para los datos que se encuentran normalmente en `/var/mqm`. Contendrá todos los datos persistentes y registros de recuperación, si no se especifica ningún otro volumen.

Aparece en:

- [“QueueManagerAlmacenamiento” en la página 48](#)

Campo	Descripción
<code>Serie class</code>	Clase de almacenamiento a utilizar para este volumen. Sólo es válido si "type" es "persistent-claim".
<code>Serie size</code>	Tamaño del PersistentVolume para pasar a Kubernetes. Sólo es válido si "type" es "persistent-claim".
<code>Serie sizeLimit</code>	Límite de tamaño cuando se utiliza un volumen "efímero". Los archivos se siguen grabando en un directorio temporal, por lo que puede utilizar esta opción para limitar el tamaño. Sólo es válido si type es "efímero".
<code>Serie type</code>	Tipo de volumen a utilizar. Elija <code>ephemeral</code> para crear un volumen "emptyDir" no persistente o <code>persistent-claim</code> para utilizar un volumen persistente.

## Solicitudes

QueueManagerResourceList define los valores de CPU y memoria.

Aparece en:

- [“Recursos” en la página 49](#)

Campo	Descripción
<code>memory</code>	
<code>cpu</code>	



## Recursos

Valores que controlan los requisitos de recursos.

Aparece en:

- [“Configuración de QueueManager” en la página 43](#)

Campo	Descripción
limits <a href="#">“Límites” en la página 42</a>	QueueManagerResourceList define los valores de CPU y memoria.
requests <a href="#">“Solicitudes” en la página 48</a>	QueueManagerResourceList define los valores de CPU y memoria.

## Secreto

Proporcione una clave utilizando un secreto de Kubernetes .

Aparece en:

- [“PKISource” en la página 42](#)

Campo	Descripción
items matriz	Claves dentro del secreto de Kubernetes que se debe añadir al contenedor del gestor de colas.
Serie secretName	El nombre del secreto de Kubernetes .

## SecurityContext

Valores de seguridad para añadir al securityContextdel pod del gestor de colas.

Aparece en:

- [“Especificación de QueueManager” en la página 45](#)

Campo	Descripción
supplementalGroups matriz	Una lista de grupos aplicados al primer proceso ejecutado en cada contenedor, además del GID primario del contenedor. Si no se especifica, no se añadirán grupos a ningún contenedor.
fsGroup entero	Un grupo suplementario especial que se aplica a todos los contenedores de un pod. Algunos tipos de volumen permiten que Kubelet cambie la propiedad de ese volumen para que sea propiedad del pod: 1. El GID propietario será el FSGroup 2. El bit setgid está establecido (los nuevos archivos creados en el volumen serán propiedad de FSGroup) 3. Los bits de permiso son OR ' d con rw-rw ---- Si no se establece, Kubelet no modificará la propiedad ni los permisos de ningún volumen.
initVolumeAsRoot boolean	Esto afecta al securityContext utilizado por el contenedor que inicializa el PersistentVolume. Establézcalo en "true" si está utilizando un proveedor de almacenamiento que requiere que sea el usuario root para acceder a los volúmenes recién suministrados. Establecer este valor en "true" afecta al objeto de restricciones de contexto de seguridad (SCC) que puede utilizar, y es posible que el gestor de colas no se pueda iniciar si no está autorizado a utilizar un SCC que permita al usuario root. Para obtener más información, consulte <a href="https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html">https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html</a> .

## Plantilla

Plantillas avanzadas para recursos de Kubernetes . La plantilla permite a los usuarios alterar temporalmente cómo IBM MQ genera los recursos de Kubernetes subyacentes, como por ejemplo StatefulSet, Pods y Servicios. Esto es sólo para usuarios avanzados, ya que tiene el potencial de interrumpir el funcionamiento normal de MQ si se utiliza incorrectamente. Los valores especificados en cualquier otro lugar del recurso QueueManager se alterarán temporalmente mediante los valores de la plantilla.

Aparece en:

- [“Especificación de QueueManager” en la página 45](#)

Campo	Descripción
pod	Alteraciones temporales para la plantilla utilizada para el pod. Consulte <a href="https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core">https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core</a> .

## TracingAgent

Solo en Cloud Pak for Integration , puede configurar valores para el agente de rastreo opcional.

Aparece en:

- [“TracingConfig” en la página 51](#)

Campo	Descripción
Serie image	La imagen de contenedor que se utilizará.
Serie imagePullPolicy	Valor que controla cuándo el kubelet intenta extraer la imagen especificada.
livenessProbe <a href="#">“TracingProbe” en la página 51</a>	Valores que controlan el análisis de actividad.
readinessProbe <a href="#">“TracingProbe” en la página 51</a>	Valores que controlan el sondeo de preparación.

## TracingCollector

Solo en Cloud Pak for Integration , puede configurar valores para el recopilador de rastreo opcional.

Aparece en:

- [“TracingConfig” en la página 51](#)

Campo	Descripción
Serie image	La imagen de contenedor que se utilizará.
Serie imagePullPolicy	Valor que controla cuándo el kubelet intenta extraer la imagen especificada.
livenessProbe <a href="#">“TracingProbe” en la página 51</a>	Valores que controlan el análisis de actividad.
readinessProbe <a href="#">“TracingProbe” en la página 51</a>	Valores que controlan el sondeo de preparación.

## TracingConfig

Valores para rastrear la integración con el panel de control de operaciones de Cloud Pak for Integration .

Aparece en:

- [“Especificación de QueueManager” en la página 45](#)

Campo	Descripción
agent <a href="#">“TracingAgent” en la página 50</a>	Solo en Cloud Pak for Integration , puede configurar valores para el agente de rastreo opcional.
collector <a href="#">“TracingCollector” en la página 50</a>	Solo en Cloud Pak for Integration , puede configurar valores para el recopilador de rastreo opcional.
enabled boolean	Indica si se debe habilitar o no la integración con el panel de control de operaciones de Cloud Pak for Integration , a través del rastreo.
Serie namespace	Espacio de nombres donde está instalado el panel de control de operaciones de Cloud Pak for Integration .

## TracingProbe

Valores que controlan el sondeo de preparación.

Aparece en:

- [“TracingCollector” en la página 50](#)

Campo	Descripción
failureThreshold entero	Número mínimo de anomalías consecutivas para que se considere que el sondeo ha fallado después de haber sido satisfactorio.
initialDelaySeconds entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicien los sondeos de actividad. Más información: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds entero	Con qué frecuencia (en segundos) se realizará el análisis.
successThreshold entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio después de haber fallado.
timeoutSeconds entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. Más información: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

## Versiones

QueueManagerStatusVersion define una versión de MQ.

Aparece en:

- [“QueueManagerStatusVersionDisponible” en la página 47](#)

Campo	Descripción
Serie name	Versión "name" para esta versión de QueueManager. Son valores válidos para el campo spec . version .

## Configuración de WebServer

Valores para el servidor web de MQ .

Aparece en:

- [“Especificación de QueueManager” en la página 45](#)

Campo	Descripción
enabled boolean	Indica si se debe habilitar o no el servidor web.

## Avisos

---

Esta información se ha desarrollado para productos y servicios ofrecidos en los Estados Unidos.

Es posible que IBM no ofrezca los productos, servicios o las características que se tratan en este documento en otros países. Consulte al representante local de IBM para obtener información sobre los productos y servicios disponibles actualmente en su zona. Las referencias a programas, productos o servicios de IBM no pretenden indicar ni implicar que sólo puedan utilizarse los productos, programas o servicios de IBM. En su lugar podrá utilizarse cualquier producto, programa o servicio equivalente que no infrinja ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio no IBM.

IBM puede tener patentes o solicitudes de patentes pendientes que cubran el tema principal descrito en este documento. El suministro de este documento no le otorga ninguna licencia sobre estas patentes. Puede enviar consultas sobre licencias, por escrito, a:

IBM Director  
of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Para consultas sobre licencias relacionadas con información de doble byte (DBCS), póngase en contacto con el Departamento de propiedad intelectual de IBM de su país o envíe las consultas por escrito a:

Licencias de Propiedad Intelectual  
Ley de Propiedad intelectual y legal  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokio 103-8510, Japón

**El párrafo siguiente no se aplica al Reino Unido ni a ningún otro país donde estas disposiciones contradigan la legislación vigente:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN NINGÚN TIPO DE GARANTÍA, YA SEA EXPLÍCITA O IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO INCUMPLIMIENTO, COMERCIALIZABILIDAD O IDONEIDAD PARA UNA FINALIDAD DETERMINADA. Algunas legislaciones no contemplan la exclusión de garantías, ni implícitas ni explícitas, en determinadas transacciones, por lo que puede haber usuarios a los que no les afecte dicha norma.

Esta información puede contener imprecisiones técnicas o errores tipográficos. La información aquí contenida está sometida a cambios periódicos; tales cambios se irán incorporando en nuevas ediciones de la publicación. IBM puede efectuar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento y sin previo aviso.

Cualquier referencia en esta información a sitios web que no son de IBM se realiza por razones prácticas y de ninguna manera sirve como un respaldo de dichos sitios web. Los materiales de dichos sitios web no forman parte de este producto de IBM y la utilización de los mismos será por cuenta y riesgo del usuario.

IBM puede utilizar o distribuir cualquier información que el usuario le proporcione del modo que considere apropiado sin incurrir por ello en ninguna obligación con respecto al usuario.

Los titulares de licencias de este programa que deseen información del mismo con el fin de permitir: (i) el intercambio de información entre los programas creados de forma independiente y otros programas (incluido este) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Corporation  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N

Rochester, MN 55901  
U.S.A.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluyendo, en algunos casos, el pago de una cantidad.

El programa bajo licencia que se describe en esta información y todo el material bajo licencia disponible para el mismo lo proporciona IBM bajo los términos del Acuerdo de cliente de IBM, el Acuerdo de licencia de programas internacional de IBM o cualquier acuerdo equivalente entre las partes.

Los datos de rendimiento incluidos en este documento se han obtenido en un entorno controlado. Por consiguiente, los resultados obtenidos en otros entornos operativos pueden variar de manera significativa. Es posible que algunas mediciones se hayan realizado en sistemas en nivel de desarrollo y no existe ninguna garantía de que estas mediciones serán las mismas en sistemas disponibles generalmente. Además, algunas mediciones pueden haberse estimado por extrapolación. Los resultados reales pueden variar. Los usuarios de este documento deben verificar los datos aplicables a su entorno específico.

La información relativa a productos que no son de IBM se obtuvo de los proveedores de esos productos, sus anuncios publicados u otras fuentes de disponibilidad pública. IBM no ha comprobado estos productos y no puede confirmar la precisión de su rendimiento, compatibilidad o alguna reclamación relacionada con productos que no sean de IBM. Las preguntas relacionadas con las posibilidades de los productos que no sean de IBM deben dirigirse a los proveedores de dichos productos.

Todas las declaraciones relacionadas con una futura intención o tendencia de IBM están sujetas a cambios o se pueden retirar sin previo aviso y sólo representan metas y objetivos.

Este documento contiene ejemplos de datos e informes que se utilizan diariamente en la actividad de la empresa. Para ilustrar los ejemplos de la forma más completa posible, éstos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con los nombres y direcciones utilizados por una empresa real es puramente casual.

#### LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente que ilustran técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo de cualquier forma sin pagar ninguna cuota a IBM para fines de desarrollo, uso, marketing o distribución de programas de aplicación que se ajusten a la interfaz de programación de aplicaciones para la plataforma operativa para la que se han escrito los programas de ejemplo. Los ejemplos no se han probado minuciosamente bajo todas las condiciones. IBM, por tanto, no puede garantizar la fiabilidad, servicio o funciones de estos programas.

Puede que si visualiza esta información en copia software, las fotografías e ilustraciones a color no aparezcan.

## Información acerca de las interfaces de programación

---

La información de interfaz de programación, si se proporciona, está pensada para ayudarle a crear software de aplicación para su uso con este programa.

Este manual contiene información sobre las interfaces de programación previstas que permiten al cliente escribir programas para obtener los servicios de WebSphere MQ.

Sin embargo, esta información puede contener también información de diagnóstico, modificación y ajustes. La información de diagnóstico, modificación y ajustes se proporciona para ayudarle a depurar el software de aplicación.

**Importante:** No utilice esta información de diagnóstico, modificación y ajuste como interfaz de programación porque está sujeta a cambios.

## Marcas registradas

---

IBM, el logotipo de IBM , ibm.com, son marcas registradas de IBM Corporation, registradas en muchas jurisdicciones de todo el mundo. Hay disponible una lista actual de marcas registradas de IBM en la web en "Copyright and trademark information"[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml). Otros nombres de productos y servicios pueden ser marcas registradas de IBM o de otras empresas.

Microsoft y Windows son marcas registradas de Microsoft Corporation en EE.UU. y/o en otros países.

UNIX es una marca registrada de Open Group en Estados Unidos y en otros países.

Linux es una marca registrada de Linus Torvalds en Estados Unidos y en otros países.

Este producto incluye software desarrollado por Eclipse Project (<http://www.eclipse.org/>).

Java y todas las marcas registradas y logotipos son marcas registradas de Oracle o sus afiliados.









Número Pieza:

(1P) P/N: