

9.4

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 179](#).

Esta edición se aplica a la versión 9 release 4 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


IBM MQ en contenedores y IBM Cloud Pak for Integration.....	5
Acerca de.....	5
Historial de releases de IBM MQ Operator.....	5
Planificación.....	7
Cómo utilizar IBM MQ en contenedores.....	8
Soporte para IBM MQ en contenedores.....	8
Planificación de licencias de IBM MQ en contenedores.....	16
Planificación del almacenamiento para IBM MQ Operator.....	16
Planificación de la alta disponibilidad para IBM MQ en contenedores.....	19
Recuperación tras desastre para IBM MQ en contenedores.....	24
Planificación de la seguridad para IBM MQ en contenedores.....	24
Planificación de la escalabilidad y el rendimiento para IBM MQ en contenedores.....	30
Preparación, instalación y actualización.....	31
Instalación y actualización de IBM MQ Operator.....	31
Preparación para IBM MQ creando su propia imagen de contenedor.....	56
Despliegue y configuración.....	64
Despliegue y configuración de gestores de colas utilizando IBM MQ Operator.....	64
Despliegue y configuración de gestores de colas utilizando Helm.....	105
Migración a IBM MQ Operator.....	106
Comprobación de que las funciones necesarias están disponibles.....	107
Extracción de la configuración del gestor de colas.....	107
Opcional: Extracción y adquisición de las claves y certificados del gestor de colas.....	108
Opcional: Configuración de LDAP.....	110
Opcional: Cambio de las direcciones IP y los nombres de host en la configuración de IBM MQ....	118
Actualización de la configuración del gestor de colas para un entorno de contenedor.....	119
Selección de la arquitectura HA de destino para IBM MQ que se ejecuta en contenedores.....	122
Creación de los recursos para el gestor de colas.....	123
Creación del nuevo gestor de colas en Red Hat OpenShift.....	124
Verificación del nuevo despliegue de contenedor.....	128
Operativo.....	130
Funcionamiento de IBM MQ utilizando la IBM MQ Operator.....	130
Visualización del estado de los gestores de colas HA nativa.....	137
Finalización manual de instancias de gestor de colas de HA nativa.....	140
Referencia.....	140
Referencia de API para IBM MQ Operator.....	140
Anotaciones de licencia al crear su propia imagen de contenedor de IBM MQ.....	165
IBM MQ Advanced for Developers imagen de contenedor.....	170
Resolución de problemas.....	173
Resolución de problemas de reinicios no planificados de IBM MQ en contenedores.....	173
Resolución de problemas con IBM MQ Operator.....	174
Avisos.....	179
Información acerca de las interfaces de programación.....	180
Marcas registradas.....	181

IBM MQ en contenedores y IBM Cloud Pak for Integration

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 utilizando IBM MQ Operator en Red Hat® OpenShift®. Esto se puede realizar utilizando IBM Cloud Pak for Integration, IBM MQ Advanced o IBM MQ Advanced for Developers.

También puede ejecutar IBM MQ en un contenedor que cree usted mismo.

 Puede obtener información adicional sobre el IBM MQ Operator consultando los enlaces siguientes.

Acerca de IBM MQ en contenedores

Información introductoria para ayudarle a empezar a utilizar IBM MQ en contenedores.

Los contenedores son una tecnología para permitir el empaquetado y aislamiento de código con su entorno de ejecución, que se puede ejecutar de una forma aislada de otro software en la misma infraestructura. Esto hace que sea fácil mover un gestor de colas o una aplicación entre entornos (como dev, test y production). Los modernos orquestadores de contenedores, como Red Hat OpenShift Container Platform y Kubernetes, pueden ejecutar muchos tipos de contenedores en la misma máquina, cada uno aislado entre sí en términos de recursos, seguridad y anomalías.

Puede ejecutar gestores de colas IBM MQ o sus aplicaciones IBM MQ en contenedores.

Información relacionada

[¿Qué son los contenedores?](#)

Historial de releases de IBM MQ Operator

Notas:

- Para obtener información sobre operadores de IBM MQ anteriores, consulte [Historial de releases para IBM MQ Operator](#) en la documentación de IBM MQ 9.3.
- Para obtener información sobre futuras actualizaciones de IBM MQ, consulte la página general [IBM MQ Arreglos recomendados y fechas de release de mantenimiento planificado](#).

IBM MQ Operator 3.2.1



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 16.1.0

Canal de operador

v3.2-sc2

Valores permitidos para `.spec.version`

[9.4.0.0-r1](#)

Valores permitidos para `.spec.version` durante la migración

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

Versiones de Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 y superior.

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versión 4.6 únicamente.

Novedades

- Resuelve un problema en OpenShift Container Platform 4.12 donde la actualización al canal v3.2-sc2 podría provocar un comportamiento inesperado para los usuarios de IBM Cloud Pak for Integration . Para obtener más información, consulte [Actualización desde 2023.4](#) en la documentación de IBM Cloud Pak for Integration .

IBM MQ Operator 3.2.0



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 16.1.0

Canal de operador

v3.2-sc2

Valores permitidos para .spec.version

9.4.0.0-r1

Valores permitidos para .spec.version durante la migración

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

Versiones de Red Hat OpenShift Container Platform

OpenShift Container Platform 4.12 y superior.

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services versión 4.6 únicamente.

Novedades

- Ahora se admite [“Expansión de volúmenes persistentes”](#) en la página 101.
- Ahora los gestores de colas se pueden detener añadiendo la anotación `mq.ibm.com/stop` y estableciéndola en `true`. Consulte [“Detención de un gestor de colas \(mq.ibm.com/stop\)”](#) en la página 105

Notas:

- Un gestor de colas detenido tiene el campo `.replicas` en su `StatefulSet` establecido en 0.
- Puesto que el IBM MQ Operator ahora gestiona activamente el campo `.replicas` en el `StatefulSet`, si modifica este campo, el operador lo revierte inmediatamente.
- Las versiones anteriores de IBM MQ entran en un estado 'Fallido' si modifica el campo `.replicas`, pero conservan el valor modificado. Si los procedimientos operativos existentes se basan en este comportamiento, desde IBM MQ 9.4 debe utilizar la anotación `mq.ibm.com/stop`.

Novedades

- Ahora se da soporte a los releases con números impares de Red Hat OpenShift Container Platform .
- IBM MQ La imagen de catálogo se ha movido al formato de catálogo basado en archivo desde el formato de base de datos SQLite .
- Se basa en [Red Hat Universal Base Image 9.4-949.1716471857](#). **Nota:** UBI 9 tiene pendiente la certificación FIPS 140-3. UBI 9 no está soportado en la arquitectura de Power 8.
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

Historial de releases para imágenes de contenedor de gestor de colas para su uso con IBM MQ Operator

Nota: Para obtener información sobre las imágenes anteriores del contenedor del gestor de colas, consulte [Historial de releases para IBM MQ Operator](#) en la documentación de IBM MQ 9.3 .

9.4.0.0-r1

CP4I-SC2 > CD

Versión de operador necesaria

[3.2.0](#) o superior

Arquitecturas soportadas

amd64, s390x, ppc64le

Imágenes

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.0-r1](#)
- [icr.io/ibm-messaging/mq:9.4.0.0-r1](#)

Novedades

- [Novedades en IBM MQ 9.4.0 for Multiplatforms-titularidad base y avanzada](#)

Novedades

- [Qué ha cambiado en IBM MQ 9.4.0](#)
- **Deprecated** Cuando se utiliza IBM MQ Advanced for Developers, el establecimiento de las contraseñas para los usuarios admin y app a través de una variable de entorno está en desuso. En su lugar, utilice Secretos.
- Se ha añadido un nuevo valor opcional mqsc para la variable de entorno `MQ_LOGGING_CONSOLE_SOURCE`. Esta opción se puede utilizar para reflejar el contenido de `autocfgmqsc.LOG` en el registro de contenedor.
- Se basa en [Red Hat Universal Base Image 9.4-949.1716471857](#). **Nota:** UBI 9 tiene pendiente la certificación FIPS 140-3. UBI 9 no está soportado en la arquitectura de Power 8.

Planificación de IBM MQ en contenedores

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 IBM MQ en la arquitectura de contenedores, debe familiarizarse con los conceptos básicos de IBM MQ (consulte [IBM MQ](#)) así como con los conceptos básicos de Kubernetes/Red Hat OpenShift (consulte [Arquitectura de OpenShift Container Platform](#)).

Procedimiento

- [“Cómo utilizar IBM MQ en contenedores”](#) en la página 8.
- [“Soporte para IBM MQ en contenedores”](#) en la página 8.
- [“Planificación del almacenamiento para IBM MQ Operator”](#) en la página 16.
- [“Planificación de la alta disponibilidad para IBM MQ en contenedores”](#) en la página 19.
- [“Recuperación tras desastre para IBM MQ en contenedores”](#) en la página 24.
- [“Autenticación de usuario y autorización para IBM MQ en contenedores”](#) en la página 25.

Cómo utilizar IBM MQ en contenedores

Existen varias opciones para utilizar IBM MQ en contenedores: puede elegir utilizar IBM MQ Operator, que utiliza imágenes de contenedor empaquetadas previamente, o puede crear sus propias imágenes y código de despliegue.

Utilización de IBM MQ Operator



Si tiene previsto realizar el despliegue en Red Hat OpenShift Container Platform, es probable que desee utilizar IBM MQ Operator.

IBM MQ Operator amplía la API Red Hat OpenShift Container Platform para añadir un nuevo recurso personalizado QueueManager. El operador observa las nuevas definiciones de gestor de colas y, a continuación, las convierte en recursos de bajo nivel necesarios, como los recursos StatefulSet y Service. En el caso de HA nativa, el operador también puede realizar la actualización continua compleja de instancias de gestor de colas. Consulte [“Consideraciones para realizar su propia actualización continua de un gestor de colas de HA nativa”](#) en la [página 22](#)

Algunas características de IBM MQ no están soportadas cuando se utiliza IBM MQ Operator. Consulte [“Soporte para IBM MQ en contenedores”](#) en la [página 8](#) para obtener detalles de lo que está soportado cuando se utiliza IBM MQ Operator.

Creación de sus propias imágenes y código de despliegue

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 sus propios código de despliegue e imágenes.

Hay ejemplos de compilación de imágenes propias. Consulte [“Preparación para IBM MQ creando su propia imagen de contenedor”](#) en la [página 56](#).

Consulte [“Soporte para IBM MQ en contenedores”](#) en la [página 8](#) para obtener detalles de lo que está soportado al crear su propia imagen y código de despliegue.

Referencia relacionada

[“Soporte para IBM MQ en contenedores”](#) en la [página 8](#)

No todas las características de IBM MQ están disponibles y soportadas de la misma forma en contenedores.

Soporte para IBM MQ en contenedores

No todas las características de IBM MQ están disponibles y soportadas de la misma forma en contenedores.

A continuación se muestra una tabla que muestra en detalle cómo se soportan las características de IBM MQ con IBM MQ Operator, o cuando se crean sus propios contenedores y código de despliegue.

Notas:

- Las imágenes de contenedor de IBM MQ precompiladas en IBM Container Registry (icr.io y cp.icr.io) solo están soportadas y son elegibles para arreglos si se utilizan con IBM MQ Operator.
- Desde IBM MQ Operator channel v3.2, Long Term Support (LTS) se ha renombrado a Support Cycle 2 (SC2). Esto se debe a que la única vía de acceso de LTS disponible para IBM MQ en contenedores es el soporte de dos años bajo la titularidad de IBM Cloud Pak for Integration y IBM Cloud Pak for Integration ha adoptado el término SC2. Aquí está la imagen completa de la titularidad:
 - Con la titularidad de IBM MQ, el IBM MQ Operator solo puede desplegar las imágenes de IBM MQ Continuous Delivery (CD).

- Con la titularidad de IBM Cloud Pak for Integration , el IBM MQ Operator puede desplegar imágenes de CD o SC2 (formerly LTS) .

No es posible "actualizar" la licencia de la imagen de IBM MQ Advanced for Developers precompilada a una licencia diferente. El IBM MQ Operator desplegará distintas imágenes, en función de la licencia seleccionada.

En esta tabla, se aplican los términos siguientes:

"Código de habilitación de contenedor"

Los ejecutables **runmqserver**, **runmqintegrationserver**, **chkmqhealthy**, **chkmqready** y **chkmqstarted**. Este código se proporciona como ejemplo y solo se soporta como parte de los contenedores precompilados cuando se utiliza con IBM MQ Operator.

	Utilización de IBM MQ Operator y una licencia de IBM Cloud Pak for Integration	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced for Developers	Imagen de IBM MQ Advanced for Developers precompilada	Cree su propio contenedor
Plataformas compatibles	<p>Soportado solo en Red Hat OpenShift Container Platform. Los releases de Red Hat OpenShift Container Platform ya no están soportados por IBM MQ una vez que Red Hat detiene el soporte.</p> <p>Consulte “Soporte de versiones del IBM MQ Operator” en la página 14 para obtener más detalles.</p>		<p>Disponible sólo en Red Hat OpenShift Container Platform , pero no soportado.</p>	<p>Funciona en cualquier plataforma Docker, containerd o cri-o, pero no está soportada. Consulte Requisitos del sistema para IBM MQ para obtener detalles.</p>	<p>Cualquier plataforma Docker, containerd o cri-o. Consulte Requisitos del sistema para IBM MQ para obtener detalles. La HA nativa sólo está soportada en Kubernetes o Red Hat OpenShift Container Platform. La imagen de contenedor de ejemplo utiliza un Red Hat Universal Base Image (UBI), que incluye bibliotecas y programas de utilidad de Linux® utilizados por IBM MQ. El UBI está soportado por Red Hat cuando se ejecuta en Red Hat OpenShift. El <i>código de habilitación de contenedor</i> no está soportado.</p>
Arquitecturas de CPU	<p>Soportado en amd64 y s390x z/Linux. También se admite en sistemas ppc64le Power Systems versión 9 y posteriores. Tenga en cuenta que todos los nodos del clúster Red Hat OpenShift Container Platform deben utilizar la misma arquitectura de CPU.</p>		<p>Disponible en amd64 y s390x z/Linux, pero no soportado. También está disponible en sistemas ppc64le Power Systems versión 9 y posteriores, pero no está soportado. Tenga en cuenta que todos los nodos del clúster Red Hat OpenShift Container Platform deben utilizar la misma arquitectura de CPU.</p>		<p>Según el software de IBM MQ .</p>

	Utilización de IBM MQ Operator y una licencia de IBM Cloud Pak for Integration	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced for Developers	Imagen de IBM MQ Advanced for Developers precompilada	Cree su propio contenedor
Duración del soporte	<p>IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) o Continuous Delivery.¹</p> <p>El operador CD y los gestores de colas están soportados hasta el siguiente release de IBM Cloud Pak for Integration CD o CP4I-SC2 .</p> <p>Los gestores de colas y operadores de CP4I-SC2 están soportados hasta el siguiente release de IBM Cloud Pak for Integration CP4I-SC2 , además de un periodo de gracia para permitir la actualización.</p>	<p>Sólo corriente de Continuous Delivery , tanto para el IBM MQ Operator, como para los gestores de colas.</p> <p>Cada versión de IBM MQ Operator y del gestor de colas sólo está soportada hasta el siguiente release de CD .</p>	No soportado		Según el software de IBM MQ . Consulte IBM MQ FAQ for Long Term Support and Continuous Delivery releases . El <i>código de habilitación de contenedor</i> no está soportado.

¹ El IBM MQ Operator está soportado como un release de IBM MQ CD o como un release de IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) :

- Las imágenes de contenedor de IBM MQ 9.4.0.x desplegadas con IBM MQ Operator 3.2.x, cuando se utilizan como parte de IBM Cloud Pak for Integration 16.1.0, son elegibles para el soporte de CP4I-LTS . El último release de Support Cycle 2 (SC2) de IBM MQ Operator es 3.2.1, y la última imagen de contenedor de SC2 es 9.4.0.0-r1.
- Las imágenes de contenedor de IBM MQ 9.4.0.x desplegadas con IBM MQ Operator 3.2.x, cuando se utilizan como parte de IBM Cloud Pak for Integration 16.1.0, son elegibles para el soporte de CD . El último release de Continuous Delivery (CD) de IBM MQ Operator es 3.2.1, y la última imagen de contenedor de CD es 9.4.0.0-r1.

	Utilización de IBM MQ Operator y una licencia de IBM Cloud Pak for Integration	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced for Developers	Imagen de IBM MQ Advanced for Developers precompilada	Cree su propio contenedor
Disponibilidad de arreglos de seguridad	Arreglos periódicos disponibles como imágenes de contenedor en IBM Container Registry				Los arreglos para el software de IBM MQ están disponibles como software en Fix Central . El <i>código de habilitación de contenedor</i> no está soportado.
Disponibilidad de arreglos temporales	Los arreglos del gestor de colas están disponibles como software y es necesaria una compilación de imagen personalizada. Los arreglos de IBM MQ Operator no están disponibles como arreglos temporales.	No hay arreglos temporales disponibles.			Los arreglos para el software de IBM MQ están disponibles como software en Fix Central o a través del soporte de IBM . El <i>código de habilitación de contenedor</i> no está soportado.
Característica: Advanced Message Security	Soportado. Tenga en cuenta que no es fácil utilizar el cifrado del lado del servidor, porque IBM MQ Operator no le permite directamente especificar su propio almacén de claves para Advanced Message Security.		Disponible pero no soportado.		Soportado según el software de IBM MQ , pero no hay ningún ejemplo disponible.
Característica: Managed File Transfer	No disponible y no soportado. Sin embargo, puede utilizar IBM MQ Operator para proporcionar uno o varios gestores de colas de Coordinación, Mandato o Agente.			No disponible y no soportado.	Soportado según el software de IBM MQ , con sample para el agente.
Característica: MQTT	No disponible y no soportado.				Soportado según el software de IBM MQ , pero no hay ningún ejemplo disponible.

	Utilización de IBM MQ Operator y una licencia de IBM Cloud Pak for Integration	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced for Developers	Imagen de IBM MQ Advanced for Developers precompilada	Cree su propio contenedor
Característica: AMQP	No disponible y no soportado.				Soportado según el software de IBM MQ , pero no hay ningún ejemplo disponible.
Característica: REST API	Disponible y soportado.				Disponible y soportado según el software de IBM MQ .
Característica: Gestores de colas de datos replicados	No disponible y no soportado. Los gestores de colas de datos replicados (RDQM) están estrechamente emparejados con el kernel de Linux y no están soportados en contenedores.				
Característica: HA nativa	Disponible y soportado.	Disponible, pero no soportado.		Sólo está disponible en Kubernetes y Red Hat OpenShift Container Platform. Soportado según el software de IBM MQ .	
Característica: Gestores de colas multiinstancia	Disponible y soportado.	Disponible, pero no soportado.		Disponible y soportado según el software de IBM MQ .	
Característica: Tipos de registros de recuperación	Sólo registros circulares o replicados. El registro lineal no está soportado.				Disponible y soportado según el software de IBM MQ . Es necesario configurar las opciones de <code>crtmqm</code> .

	Utilización de IBM MQ Operator y una licencia de IBM Cloud Pak for Integration	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced	Utilización de IBM MQ Operator y una licencia de IBM MQ Advanced for Developers	Imagen de IBM MQ Advanced for Developers precompilada	Cree su propio contenedor
Característica: especificación de opciones de línea de mandatos personalizadas para <code>crtmqdir</code>, <code>crtmqm</code>, <code>strmqm</code> y <code>endmqm</code>	No disponible y no soportado. La mayoría de las opciones se pueden configurar utilizando un archivo INI, pero algunas no se pueden configurar, como por ejemplo el uso del registro lineal.				Opcional, en función de cómo implemente el código de habilitación de contenedor.
Característica: Usuarios del sistema operativo (SO)	No disponible y no soportado.				Posible y soportado según el software de IBM MQ, si instala IBM MQ utilizando RPM, pero no hay ningún ejemplo disponible. No se recomienda debido a un riesgo de seguridad.

Nota: La frase "soportado según el software de IBM MQ" significa que el soporte técnico de IBM está limitado al software principal de IBM MQ que se ejecuta dentro del contenedor.

Conceptos relacionados

[Preguntas más frecuentes de IBM MQ para soporte a largo plazo y releases de entrega continua](#)

Referencia relacionada

[IBM Cloud Pak for Integration Software Support Lifecycle Addendum](#)

OpenShift > CP4I-SC2 > CP4I > CD Soporte de versiones del IBM MQ

Operator


Una correlación entre versiones soportadas de IBM MQ, OpenShift Container Platform y IBM Cloud Pak for Integration.

- [“Versiones de IBM MQ disponibles” en la página 15](#)
- [“Versiones de Red Hat OpenShift Container Platform compatibles” en la página 15](#)
- [“Versiones de IBM Cloud Pak for Integration” en la página 15](#)
- [“Versiones de IBM MQ disponibles en operadores más antiguos” en la página 15](#)
- [“Versiones compatibles de OpenShift Container Platform para operadores más antiguos” en la página 15](#)

Versiones de IBM MQ disponibles

Canal de operador	Versión del operador	Versiones de IBM MQ						
		9.4.0	9.3.5	9.3.4	9.3.3	9.3.2	9.3.1	9.3.0
v32-sc2	3.2	CD y SC2	DEP	DEP	DEP	DEP	DEP	MIG

Clave:

- CD: el soporte de Continuous Delivery está disponible.
- SC2: IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) está disponible.
- MIG: Sólo está disponible durante la migración desde un operando IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) a un operando Continuous Delivery .
- DEP:  En desuso. A medida que los releases de IBM MQ dejan de tener soporte, es posible que se puedan configurar en el operador, pero ya no son elegibles para el soporte y pueden eliminarse en futuros releases.

Consulte “Historial de releases de IBM MQ Operator” en la página 5 para obtener detalles completos de cada versión, incluidas características detalladas, cambios y arreglos en cada versión.

Versiones de Red Hat OpenShift Container Platform compatibles

Canal de operador	Versión del operador	Versiones de OpenShift Container Platform ²		
		4.15	4.14	4.12
v3.2-sc2	3.2.0 en adelante	SC2	SC2	SC2

Clave:

- CD: el soporte de Continuous Delivery está disponible.
- SC2: IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) está disponible.
- EOS: Ya no está soportado. Migre a una versión posterior de OpenShift Container Platform .

Versiones de IBM Cloud Pak for Integration

Soportado para su uso como parte de IBM Cloud Pak for Integration versión 16.1.0, o de forma independiente:

- IBM MQ Operator 3.2.x

Versiones de IBM MQ disponibles en operadores más antiguos

Consulte [Versiones de IBM MQ disponibles](#) en la documentación de IBM MQ 9.3 .

Versiones compatibles de OpenShift Container Platform para operadores más antiguos

Consulte [Compatible OpenShift Container Platform versiones](#) en la documentación de IBM MQ 9.3 .

Edición de recursos creados por IBM MQ Operator

El IBM MQ Operator reconcilia un recurso personalizado QueueManager creando y gestionando recursos nativos de Kubernetes . Estos recursos gestionados **no** se deben editar directamente.

² Las versiones de OpenShift Container Platform están sujetas a sus propias fechas de soporte. Consulte [Política de ciclo de vida de OpenShift Container Platform](#) para obtener más información.

Normalmente puede determinar si un recurso es propiedad de otro recurso de nivel superior, consultando `ownerReferences`. Por ejemplo, los metadatos siguientes tomados de un `StatefulSet` muestran que es propiedad del recurso `QueueManager` "qm1":

```
metadata:
  ownerReferences:
  - apiVersion: mq.ibm.com/v1beta1
    kind: QueueManager
    name: qm1
    uid: 60fda34c-9f7c-42d2-a293-78fec4315c62
    controller: true
    blockOwnerDeletion: true
```

Tenga en cuenta que no todos los recursos tienen estos metadatos.

Es responsabilidad del IBM MQ Operator gestionar los recursos subyacentes, como `StatefulSet`, `Service` y `Route`. Si cambia alguno de estos recursos subyacentes, IBM MQ Operator los volverá a cambiar y puede experimentar un tiempo de inactividad si ese cambio requiere una actualización continua.

La mayoría de los valores importantes para los gestores de colas están disponibles en el recurso `QueueManager`. Sin embargo, si detecta que necesita un control completo de los recursos subyacentes, existen algunas opciones:

- Si necesita alterar temporalmente los valores en el pod creado por IBM MQ Operator, puede añadir una plantilla de alteración temporal de pod en la sección `.spec.template` del YAML de `QueueManager`.
- Si necesita alterar temporalmente los valores en el gestor de colas `Route` creado por el IBM MQ Operator, debe inhabilitar la ruta por completo estableciendo el valor `.spec.route.enabled` en "false" y, a continuación, creando su propia ruta.
- Valores como etiquetas y anotaciones, así como valores de Pod como `securityContext`, se pueden establecer en el recurso `QueueManager`.
- En otros casos, es posible que IBM MQ Operator no sea adecuado para su caso de uso si necesita un control completo.

Planificación de licencias de IBM MQ en contenedores

Las licencias de contenedor le permiten obtener una licencia sólo de la capacidad disponible de los contenedores individuales de IBM MQ, en lugar de exigirle que obtenga una licencia de todo el servidor en el que se ejecutan los contenedores. Para aprovechar las licencias de contenedor, se debe utilizar el License Service de IBM para realizar un seguimiento del uso de licencia y determinar la titularidad necesaria.

Referencia relacionada

[“Anotaciones de licencia al crear su propia imagen de contenedor de IBM MQ” en la página 165](#)

Las anotaciones de licencia le permiten realizar un seguimiento del uso según los límites definidos en el contenedor, en lugar de hacerlo según la máquina subyacente. Los clientes se configuran para desplegar el contenedor con anotaciones específicas que el IBM License Service utiliza para realizar el seguimiento del uso.

Información relacionada

[Licencias de contenedor de IBM](#)

[Preguntas más frecuentes sobre licencias de contenedor](#)

[Instalación de License Service](#)

[Visualización y seguimiento del uso de licencias](#)

Planificación del almacenamiento para IBM MQ Operator

El IBM MQ Operator se ejecuta en dos modalidades de almacenamiento:

- El **almacenamiento efímero** se utiliza cuando toda la información de estado del contenedor se puede descartar 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.



IBM MQ Operator 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, toda la información de estado del gestor de colas se pierde 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 todos los estados 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 Red Hat OpenShift” en la página 94](#)). 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 persistente para asegurarse de que el gestor de colas conserva sus mensajes persistentes y la configuración después de un reinicio. Éste es el comportamiento predeterminado. Debido a que hay varios proveedores de almacenamiento, cada uno de los cuales da soporte a prestaciones diferentes, esto a menudo significa que es necesaria la personalización de la configuración. El ejemplo siguiente describe los campos comunes que personalizan la configuración de almacenamiento de IBM MQ en la API v1beta1 :

- **spec.queueManager.availability** controla la modalidad de disponibilidad. Si está utilizando SingleInstance o NativeHA, sólo necesita almacenamiento ReadWriteOnce. Para multiInstance necesita una clase de almacenamiento que dé soporte a 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 [“Planificación de la alta disponibilidad para IBM MQ en contenedores” en la página 19](#).

- **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:
        class: ibmc-file-gold-gid
      persistedData:
        enabled: true
        class: ibmc-file-gold-gid
      recoveryLogs:
        enabled: true
        class: ibmc-file-gold-gid
    securityContext:
      supplementalGroups: [65534] # Change to 99 for clusters with RHEL7 or earlier worker nodes
```

Para obtener información sobre las consideraciones de almacenamiento para los gestores de colas de HA nativa, consulte [“HA nativa”](#) en la página 21.

Nota: También puede configurar grupos suplementarios con gestores de colas de una sola instancia.

Capacidad de almacenamiento



Cuando utilice IBM MQ Operator , debe intentar asegurarse de que solicita volúmenes lo suficientemente grandes para sus necesidades en curso. Sin embargo, si necesita aumentar la capacidad de almacenamiento de uno o más volúmenes, estos volúmenes se pueden ampliar si la clase de almacenamiento da soporte a la expansión de volúmenes. Los volúmenes se pueden expandir mediante un procedimiento en línea o fuera de línea. Un procedimiento fuera de línea requiere que se reinicien los pods de QueueManager , mientras que un procedimiento en línea no lo hace. Para determinar si la clase de almacenamiento da soporte a la expansión de volumen y qué procedimiento sigue la expansión de volumen, consulte la documentación del proveedor de almacenamiento. Debe tener en cuenta esta información al seleccionar una clase de almacenamiento. Para obtener una guía sobre la expansión de volúmenes, consulte [“Expansión de volúmenes persistentes”](#) en la página 101.

Cifrado



IBM MQ no cifra activamente los datos en reposo. Por lo tanto, debe utilizar almacenamiento cifrado de forma pasiva, o IBM MQ Advanced Message Security, o ambos, para cifrar los mensajes. En IBM Cloud , tanto el almacenamiento en bloque como el almacenamiento de archivos están disponibles con cifrado pasivo en reposo.

Planificación de la alta disponibilidad para IBM MQ en contenedores

Hay tres opciones para la alta disponibilidad con IBM MQ Operator: **Gestor de colas HA nativa** (que tiene una réplica activa y dos réplicas en espera), **Gestor de colas multiinstancia** (que es un par activo-en espera, que utiliza un sistema de archivos compartido en red) o **Gestor de colas resiliente único** (que ofrece un enfoque sencillo para HA utilizando almacenamiento en red). Los dos últimos se basan en el sistema de archivos para garantizar la disponibilidad de los datos recuperables, sin embargo, la HA nativa no lo hace. Por lo tanto, cuando no se utiliza HA nativa, la disponibilidad del sistema de archivos es fundamental para la disponibilidad del gestor de colas. Cuando la recuperación de datos es importante, el sistema de archivos debe garantizar la redundancia a través de la réplica.

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 HA nativo

MQ Adv.

Los gestores de colas de alta disponibilidad nativos implican un **activo** y dos **réplica** Kubernetes Pods, que se ejecutan como parte de un Kubernetes StatefulSet con exactamente tres réplicas cada uno con su propio conjunto de volúmenes persistentes de Kubernetes. Los requisitos de IBM MQ para sistemas de archivos compartidos también se aplican cuando se utiliza un gestor de colas HA nativo (excepto para el bloqueo basado en arrendamiento), pero no es necesario 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 para un gestor de colas HA nativo se controlan mediante los factores siguientes:

1. El tiempo que tardan las instancias de réplica en detectar que la instancia activa ha fallado. Esto es configurable.
2. El tiempo que tarda el sondeo de preparación de pod de Kubernetes en detectar que el contenedor preparado ha cambiado y redirigir el tráfico de red. Esto es configurable.
3. Cuánto tiempo tardan los clientes de IBM MQ en volver a conectarse.

Para obtener más información, consulte [“HA nativa” en la página 21](#).

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. El tiempo que tarda el sondeo de preparación de pod de Kubernetes en detectar que el contenedor preparado ha cambiado y redirigir el tráfico de red. Esto es configurable.
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 es configurable.
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 es configurable.
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. Cuando se utiliza una carga de trabajo con estado como IBM MQ con un Kubernetes StatefulSet, si un nodo maestro de Kubernetes pierde el contacto con un nodo trabajador, no puede determinar si el nodo ha fallado o si simplemente 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.

Nota: El cambio de los detalles de StatefulSet de un gestor de colas de IBM MQ , incluido el número de réplicas, no está soportado cuando el gestor de colas se crea a través de IBM MQ Operator.

Conceptos relacionados

[Configuraciones de alta disponibilidad](#)

Tareas relacionadas

[“Configuración de la alta disponibilidad para gestores de colas utilizando la IBM MQ Operator” en la página 76](#)

CP4I

MQ Adv.

HA nativa

La alta disponibilidad nativa es una solución de alta disponibilidad nativa (incorporada) para IBM MQ que es adecuada para su uso con el almacenamiento en bloque en la nube.

Una configuración de HA nativa proporciona un gestor de colas altamente disponible donde los datos de MQ recuperables (por ejemplo, los mensajes) se replican en varios conjuntos de almacenamiento, lo que impide la pérdida de anomalías de almacenamiento. El gestor de colas consta de varias instancias en ejecución, una es la que lidera, las otras están preparadas para asumir rápidamente el control en caso de una anomalía, maximizando el acceso al gestor de colas y sus mensajes.

Una configuración de HA nativa consta de tres pods de Kubernetes, cada uno con una instancia del gestor de colas. Una instancia es el gestor de colas activo, procesando mensajes y grabando en su registro de recuperación. Siempre que se graba el registro de recuperación, el gestor de colas activo envía los datos a las otras dos instancias, conocidas como réplicas. Cada réplica graba en su propio registro de recuperación, reconoce los datos y, a continuación, actualiza sus propios datos de cola del registro de recuperación replicado. Si el pod que ejecuta el gestor de colas activo falla, una de las instancias de réplica del gestor de colas toma el control del rol activo y tiene datos actuales con los que operar.

El tipo de registro se conoce como 'registro replicado'. Un registro replicado es esencialmente un registro lineal, con la gestión automática de registros y las imágenes de soporte automáticas habilitadas. Consulte [Tipos de registro](#). Utilice las mismas técnicas para gestionar el registro replicado que utiliza para gestionar un registro lineal.

Se utiliza un Kubernetes Service para direccionar las conexiones de cliente TCP/IP a la instancia activa actual, que se identifica como el único pod que está preparado para el tráfico de red. Esto sucede sin necesidad de que la aplicación cliente tenga en cuenta las distintas instancias.

Se utilizan tres vainas para reducir en gran medida la posibilidad de que surja una situación de cerebro dividido. En un sistema de alta disponibilidad de dos pods, el cerebro dividido podría producirse cuando se rompe la conectividad entre los dos pods. Sin conectividad, ambos pods podrían ejecutar el gestor de colas al mismo tiempo, acumulando datos diferentes. Cuando se restaura la conexión, habría dos versiones diferentes de los datos (un 'split-brain'), y se requiere una intervención manual para decidir qué conjunto de datos conservar, y cuál descartar.

La HA nativa utiliza un sistema de tres pods con quórum para evitar la situación de cerebro dividido. Los pods que pueden comunicarse con al menos uno de los otros pods forman un quórum. Un gestor de colas sólo puede convertirse en la instancia activa en un pod que tenga quórum. El gestor de colas no puede activarse en un pod que no esté conectado al menos a otro pod, por lo que nunca puede haber dos instancias activas al mismo tiempo:

- Si un único pod falla, el gestor de colas de uno de los otros dos pods puede tomar el control. Si dos pods fallan, el gestor de colas no puede convertirse en la instancia activa en el pod restante porque el pod no tiene quórum (el pod restante no puede indicar si los otros dos pods han fallado, o si todavía se están ejecutando y ha perdido la conectividad).
- Si un único pod pierde la conectividad, el gestor de colas no puede activarse en este pod porque el pod no tiene quórum. El gestor de colas en uno de los dos pods restantes puede tomar el control, que tienen quórum. Si todos los pods pierden la conectividad, el gestor de colas no puede activarse en ninguno de los pods, porque ninguno de los pods tiene quórum.

Si un pod activo falla y posteriormente se recupera, puede volver a unir el grupo en un rol de réplica.

Para el rendimiento y la fiabilidad, se recomienda utilizar el almacenamiento persistente RWO (ReadWriteOnce) con una configuración de HA nativa. Los volúmenes RWO de cualquier proveedor de almacenamiento están soportados si cumplen las condiciones siguientes:

- Se obtiene de un proveedor de almacenamiento en bloques.
- Formateado como ext4 o XFS (que garantiza la conformidad con POSIX).
- Admite el suministro de volúmenes dinámicos y la modalidad "volumeBinding: WaitForFirstConsumer".

Los proveedores siguientes están explícitamente prohibidos:

- NFS
- GlusterFS
- Otros proveedores que no son de bloque.

La figura siguiente muestra un despliegue típico con tres instancias de un gestor de colas desplegadas en tres contenedores.

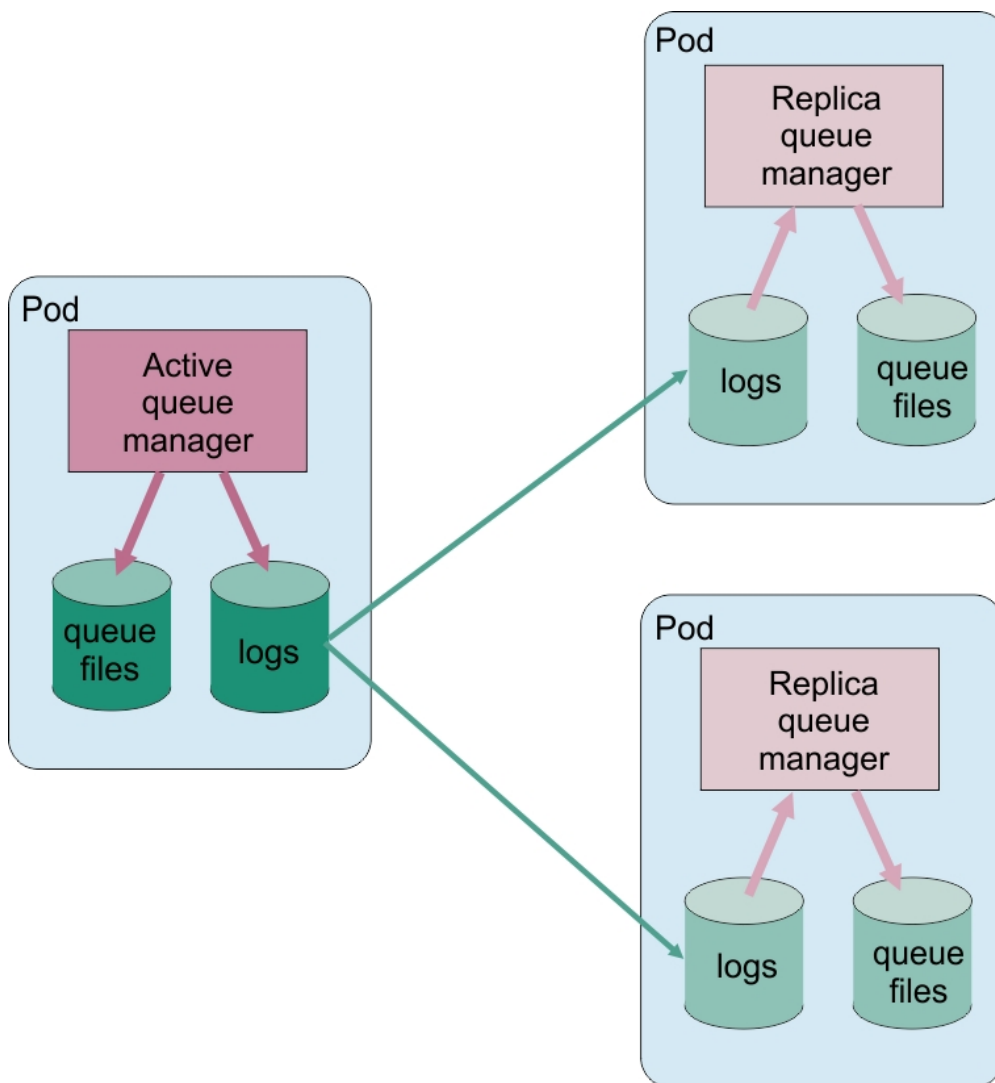


Figura 1. Ejemplo de configuración de HA nativa

MQ Adv. Consideraciones para realizar su propia actualización continua de un gestor de colas de HA nativa

Cualquier actualización de la versión de IBM MQ o de la especificación de pod para un gestor de colas de HA nativa, requerirá que realice una actualización continua de las instancias del gestor de colas. El IBM MQ Operator lo maneja automáticamente, pero si está creando su propio código de despliegue, hay algunas consideraciones importantes.

Nota: El diagrama de [ejemplo Helm](#) incluye un script de shell para realizar una actualización continua, pero el script **no** es adecuado para su uso en producción, ya que no aborda las consideraciones de este tema.

Kubernetes En Kubernetes, los recursos de `StatefulSet` se utilizan para gestionar actualizaciones continuas y de inicio ordenadas. Parte del procedimiento de arranque es iniciar cada Pod de forma individual, esperar a que esté listo y, a continuación, pasar al siguiente Pod. Esto no funcionará para la HA nativa, ya que todos los Pods necesitan ser iniciados para que puedan llevar a cabo una elección de líder. Por lo tanto, el campo `.spec.podManagementPolicy` en `StatefulSet` debe establecerse en `Parallel`. Esto también significa que todos los Pods se actualizarán en paralelo también, lo que es particularmente indeseable. Por este motivo, `StatefulSet` también debe utilizar la estrategia de actualización de `OnDelete`.

La incapacidad de utilizar las unidades de código de actualización continua de `StatefulSet` requiere un código de actualización continua personalizado, que debe tener en cuenta lo siguiente:

- Procedimiento de actualización continua general
- Minimización del tiempo de inactividad actualizando los pods en el mejor orden
- Manejo de cambios en el estado de clúster
- Manejo de errores
- Manejo de problemas de temporización

Procedimiento de actualización continua general

El código de actualización continua debe esperar a que cada instancia muestre un estado de `REPLICA` de `dspm`. Esto significa que la instancia ha realizado algún nivel de inicio (por ejemplo, el contenedor se ha iniciado y los procesos de MQ se están ejecutando), pero no ha conseguido necesariamente hablar con las otras instancias todavía. Por ejemplo: el pod A se reinicia y, en cuanto está en estado `REPLICA`, el pod B se reinicia. Una vez que el Pod B empieza con la nueva configuración, debe poder hablar con el Pod A, y puede formar quórum, y A o B se convertirán en la nueva instancia activa.

Como parte de esto, es útil tener un retardo después de que cada pod haya alcanzado el estado `REPLICA`, para permitirle conectarse a sus iguales y establecer quórum.

Minimización del tiempo de inactividad actualizando los pods en el mejor orden

El código de actualización continua debe suprimir los Pods de uno en uno, empezando por los Pods que están en un estado de error conocido, seguidos de los Pods que no se han iniciado correctamente. Generalmente, el pod del gestor de colas activo debe actualizarse en último lugar.

También es importante pausar la supresión de Pods si la última actualización ha dado como resultado un pod que entra en un estado de error conocido. Esto impide el despliegue de una actualización interrumpida en todos los pods. Por ejemplo, esto puede suceder si el Pod se actualiza para utilizar una nueva imagen de contenedor que no es accesible (o contiene un error tipográfico).

Manejo de cambios en el estado de clúster

El código de actualización continua debe reaccionar adecuadamente a los cambios en tiempo real en el estado del clúster. Por ejemplo, uno de los pods del gestor de colas se puede desalojar debido a un rearranque de nodo o debido a la presión del nodo. Es posible que un pod desalojado no se vuelva a planificar inmediatamente si el clúster está ocupado. En este caso, el código de actualización continua tendría que esperar adecuadamente antes de reiniciar cualquier otro pod.

Manejo de errores

El código de actualización continua debe ser sólido para los errores al llamar a la API Kubernetes y a otro comportamiento de clúster inesperado.

Además, el propio código de actualización continua debe ser tolerante a que se reinicie. Una actualización continua puede ser de larga ejecución y es posible que sea necesario reiniciar el código.

Manejo de problemas de temporización

El código de actualización continua debe comprobar las revisiones de actualización del pod, para que pueda asegurarse de que el pod se ha reiniciado. Esto evita problemas de temporización en los que un Pod puede indicar que está "Iniciado", pero de hecho todavía no ha terminado.

Conceptos relacionados

[“Cómo utilizar IBM MQ en contenedores” en la página 8](#)

Existen varias opciones para utilizar IBM MQ en contenedores: puede elegir utilizar IBM MQ Operator, que utiliza imágenes de contenedor empaquetadas previamente, o puede crear sus propias imágenes y código de despliegue.

Recuperación tras desastre para IBM MQ en contenedores

Es necesario considerar para qué tipo de desastre se está preparando. En entornos de nube, el uso de zonas de disponibilidad proporciona un determinado nivel de tolerancia ante desastres y es mucho más fácil de utilizar. Si tiene un número impar de centros de datos (para quórum) y un enlace de red de latencia baja, podría ejecutar potencialmente un único clúster Red Hat OpenShift Container Platform o Kubernetes con varias zonas de disponibilidad, cada uno en una ubicación física independiente. En este tema se tratan las consideraciones para la recuperación tras desastre en las que no se pueden cumplir estos criterios: es decir, un número par de centros de datos o un enlace de red de latencia alta.

Para la recuperación tras desastre, debe tener en cuenta lo siguiente:

- Réplica de datos de IBM MQ (contenidos en uno o varios recursos de PersistentVolume) en la ubicación de recuperación tras desastre
- Volver a crear el gestor de colas utilizando los datos replicados
- El ID de red del gestor de colas que es visible para las aplicaciones cliente de IBM MQ y otros gestores de colas. Este ID podría ser una entrada DNS, por ejemplo.

Los datos persistentes deben replicarse, de forma síncrona o asíncrona, en el sitio de recuperación tras desastre. Esto suele ser específico del proveedor de almacenamiento, pero también se puede realizar utilizando un VolumeSnapshot. Consulte [Instantáneas de volumen CSI](#) para obtener más información sobre las instantáneas de volumen.

Al recuperarse de un desastre, tendrá que volver a crear la instancia del gestor de colas en el nuevo clúster de Kubernetes , utilizando los datos replicados. Si está utilizando IBM MQ Operator, necesitará el YAML de QueueManager , así como el YAML para otros recursos de soporte como ConfigMap o Secret.

Información relacionada

[ha_for_ctr.dita](#)

Planificación de la seguridad para IBM MQ en contenedores

Consideraciones de seguridad al planificar IBM MQ en la configuración de contenedores.

Procedimiento

- [“Autenticación de usuario y autorización para IBM MQ en contenedores” en la página 25](#)
 - [“Restricciones de seguridad en el uso de usuarios del sistema operativo en contenedores” en la página 25](#)
- [“Consideraciones para restringir el tráfico de red a IBM MQ en contenedores” en la página 26](#)

Autenticación de usuario y autorización para IBM MQ en contenedores

IBM MQ en contenedores se puede configurar para autenticar usuarios a través de LDAP, TLS mutuo o un plugin personalizado de MQ .

Tenga en cuenta que el operador IBM MQ no permite el uso de usuarios y grupos del sistema operativo dentro de la imagen de contenedor. Para obtener más información, consulte [“Restricciones de seguridad en el uso de usuarios del sistema operativo en contenedores”](#) en la página 25.

LDAP

Para obtener información sobre cómo configurar IBM MQ para utilizar un repositorio de usuarios LDAP, consulte [Autenticación de conexión: Repositorios de usuarios](#) y [Autorización LDAP](#).

TLS mutuo

Si configura conexiones entrantes a un gestor de colas para requerir un certificado TLS (TLS mutuo), puede correlacionar el nombre distinguido del certificado con un nombre de usuario. Usted necesita hacer dos cosas:

- Configure un registro de autenticación de canal para crear la correlación con un nombre de usuario, utilizando SSLPEER. Para obtener más información, consulte [Correlación de un nombre distinguido SSL o TLS con un ID de usuario MCAUSER](#).
- Configure el gestor de colas para que le permita definir registros de autorización para un nombre de usuario que el sistema no conoce. Para obtener más información, consulte [Stanza de servicio del archivo qm.ini](#).

Señales web JSON

Para obtener información sobre cómo configurar IBM MQ para utilizar JSON Web Tokens (JWT), consulte [Trabajar con señales de autenticación](#).

Plug-in MQ personalizado

Esta es una técnica avanzada, y requiere mucho más trabajo. Para obtener más información, consulte [Utilización de un servicio de autorización personalizado](#).

Tareas relacionadas

[“Ejemplo: Configuración de un gestor de colas con autenticación TLS mutua”](#) en la página 70
Este ejemplo despliega un gestor de colas en OpenShift Container Platform utilizando IBM MQ Operator. Se utiliza TLS mutuo para la autenticación, para correlacionar desde un certificado TLS con una identidad en el gestor de colas.

Restricciones de seguridad en el uso de usuarios del sistema operativo en contenedores

No se recomienda utilizar usuarios del sistema operativo en contenedores, y está prohibido con el operador IBM MQ .

En un entorno contenerizado de varios arrendatarios, las restricciones de seguridad se suelen aplicar para evitar posibles problemas de seguridad, por ejemplo:

- **Impedir el uso del usuario "root" dentro de un contenedor**
- **Forzar el uso de un UID aleatorio.** Por ejemplo, en Red Hat OpenShift Container Platform el valor predeterminado SecurityContextConstraints (denominado `restricted`) utiliza un ID de usuario aleatorizado para cada contenedor.
- **Impedir el uso del escalamiento de privilegios.** IBM MQ on Linux utiliza el escalamiento de privilegios para comprobar las contraseñas de los usuarios-utiliza un programa "setuid" para convertirse en el usuario "root" para hacerlo.

Para garantizar la conformidad con estas medidas de seguridad, el IBM MQ Operator no permite el uso de los ID definidos en las bibliotecas del sistema operativo dentro de un contenedor. No hay ningún ID de usuario o grupo de mqm definido en el contenedor.

Consideraciones para restringir el tráfico de red a IBM MQ en contenedores

Puede definir políticas de red para restringir el tráfico a los pods del clúster en OpenShift Container Platform y Kubernetes. En este tema se describen algunas consideraciones sobre cómo se pueden aplicar las políticas de red a IBM MQ.

Para la entrada de red a un gestor de colas, hay varios puertos a tener en cuenta:

- Puerto 1414 para el tráfico del gestor de colas
- Puerto 9414 para HA nativa
- Puerto 9157 para medidas
- Puerto 9443 para la consola web y las API REST

La salida de red es más compleja. Ejemplos de salida de red que puede tener en cuenta:

- DNS-si tiene canales u otra configuración que utilizan nombres DNS
- Otros gestores de colas
- Protocolo de estado de certificados en línea (OCSP) y listas de revocación de certificados (CRL)-determinado por el proveedor de certificados.
- Proveedores de autenticación:
 - LDAP
 - Open ID Connect u otro proveedor de inicio de sesión configurado para el servidor web de IBM MQ . Esto incluye IBM Cloud Pak Keycloak.
- Proveedores de rastreo:
 - IBM Instana

Nota: Para versiones anteriores de IBM MQ , el panel de control de operaciones de IBM Cloud Pak for Integration también estaba disponible como proveedor de rastreo. Sin embargo, el panel de control de operaciones se ha eliminado en IBM MQ 9.3.3 CD y IBM MQ 9.4.0 LTS.

Ejemplo de ingress NetworkPolicy

A continuación se muestra una política de red de ejemplo para controlar la entrada para un gestor de colas denominado "myqm", para su uso en Red Hat OpenShift Container Platform.

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

```

- from:
  - namespaceSelector:
      matchLabels:
        network.openshift.io/policy-group: monitoring
  ports:
    - protocol: TCP
      port: 9157
# Allow access to web server via Route
- from:
  - namespaceSelector:
      matchLabels:
        network.openshift.io/policy-group: ingress
  ports:
    - protocol: TCP
      port: 9443

```

Conformidad con FIPS para IBM MQ en contenedores

Durante el inicio, IBM MQ en contenedores detecta si el sistema operativo en el que se está iniciando el contenedor es compatible con FIPS y (si es así) configura el soporte de FIPS automáticamente. Aquí se anotan los requisitos y limitaciones.

Federal Information Processing Standards

El gobierno de EE.UU. ofrece asesoramiento técnico en cuanto a sistemas de TI y seguridad, que incluye el cifrado de datos. El National Institute for Standards and Technology (NIST) es un organismo gubernamental que se ocupa de los sistemas de TI y la seguridad. El NIST propone recomendaciones y genera estándares, entre los que se incluye FIPS (Federal Information Processing Standards).

Un estándar FIPS significativo es FIPS 140-2, que requiere el uso de algoritmos criptográficos fuertes. FIPS 140-2 también especifica los requisitos para que algoritmos de hash se puedan utilizar para proteger los paquetes contra su modificación mientras están en tránsito.

IBM MQ proporciona soporte para FIPS 140-2 si se ha configurado para ello.

Nota: En AIX, Linux, and Windows, IBM MQ proporciona conformidad con FIPS 140-2 a través del módulo criptográfico IBM Crypto for C (ICC) . El certificado para este módulo se ha movido al estado Histórico. Los clientes deben ver el [certificado de IBM Crypto for C \(ICC\)](#) y tener en cuenta cualquier consejo proporcionado por NIST. [Un módulo FIPS 140-3 de sustitución está actualmente en curso](#) y su estado se puede ver buscándolo en los [módulos CMVP de NIST](#) en la lista de procesos.

La imagen de contenedor de IBM MQ Operator 3.2.0 y el gestor de colas 9.4.0.0 en adelante se basan en UBI 9. La conformidad con FIPS 140-3 está pendiente actualmente y su estado se puede visualizar buscando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" en los [módulos CMVP de NIST](#) en la lista de procesos.

Requisitos

Para conocer los requisitos relacionados con la configuración del clúster y otras consideraciones, consulte [FIPS Wall: enfoque actual de IBM a la conformidad con FIPS](#).

IBM MQ en contenedores se puede ejecutar en modalidad de conformidad con FIPS 140-2. Durante el inicio, IBM MQ en contenedores detecta si el sistema operativo de host en el que se está iniciando el contenedor es compatible con FIPS. Si el sistema operativo de host es compatible con FIPS y se han proporcionado claves privadas y certificados, el contenedor IBM MQ configura el gestor de colas, el servidor web de IBM MQ y la transferencia de datos entre los nodos en un despliegue de alta disponibilidad nativa, para que se ejecute en modalidad de conformidad con FIPS.

Cuando se utiliza IBM MQ Operator para desplegar gestores de colas, el operador crea una ruta con un tipo de terminación de **Passthrough**. Esto significa que el tráfico se envía directamente al destino sin que el direccionador proporcione la terminación TLS. El gestor de colas de IBM MQ y el servidor web de IBM MQ son los destinos en este caso y ya proporcionan una comunicación segura compatible con FIPS.

Requisitos clave:

1. Una clave privada y certificados, proporcionados en un secreto al gestor de colas y al servidor web, que permiten a los clientes externos conectarse de forma segura al gestor de colas y al servidor web.
2. Una clave privada y certificados para la transferencia de datos entre distintos nodos en una configuración de alta disponibilidad nativa.

Limitaciones

Para un despliegue compatible con FIPS de IBM MQ en contenedores, tenga en cuenta lo siguiente:

- IBM MQ en contenedores proporciona un punto final para la recopilación de métricas. Actualmente, este punto final es sólo HTTP. Puede desactivar el punto final de métricas para que el resto de IBM MQ sea compatible con FIPS.
- IBM MQ en contenedores permite alteraciones temporales de imagen personalizadas. Es decir, puede crear imágenes personalizadas utilizando la imagen de contenedor IBM MQ como imagen base. Es posible que la conformidad con FIPS no se aplique a dichas imágenes personalizadas.
- Para el seguimiento de mensajes utilizando IBM Instana, la comunicación entre IBM MQ y IBM Instana es HTTP o HTTPS, sin conformidad con FIPS.
- El acceso de IBM MQ Operator a los servicios de gestión de accesos e identidades (IAM) /Zen de IBM no es compatible con FIPS.

Cómo se detecta la conformidad con FIPS y se configura automáticamente el soporte de FIPS

Si el sistema operativo en el que se inicia el contenedor es compatible con FIPS, el soporte de FIPS se configura automáticamente.

Nota: En AIX, Linux, and Windows, IBM MQ proporciona conformidad con FIPS 140-2 a través del módulo criptográfico IBM Crypto for C (ICC) . El certificado para este módulo se ha movido al estado Histórico. Los clientes deben ver el [certificado de IBM Crypto for C \(ICC\)](#) y tener en cuenta cualquier consejo proporcionado por NIST. Un módulo FIPS 140-3 de sustitución está actualmente en curso y su estado se puede ver buscándolo en los [módulos CMVP de NIST en la lista de procesos](#).

La imagen de contenedor de IBM MQ Operator 3.2.0 y el gestor de colas 9.4.0.0 en adelante se basan en UBI 9. La conformidad con FIPS 140-3 está pendiente actualmente y su estado se puede visualizar buscando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" en los [módulos CMVP de NIST en la lista de procesos](#).

Durante el inicio, IBM MQ en contenedores detecta si el sistema operativo en el que se está iniciando el contenedor es compatible con FIPS. Si es así, las acciones siguientes se realizan automáticamente:

Gestor de colas

Si el sistema operativo de host es compatible con FIPS y se proporcionan la clave privada y los certificados, el atributo de gestor de colas **SSLFIPS** se establece en YES. De lo contrario, el atributo **SSLFIPS** se establece en NO.

IBM MQ servidor web

El servidor web de IBM MQ proporciona una interfaz HTTP/HTTPS para administrar IBM MQ. Si el sistema operativo de host es compatible con FIPS, las opciones de JVM se actualizan para que el servidor web utilice criptografía compatible con FIPS. Para poder utilizar FIPS, se deben proporcionar la clave privada y los certificados durante el inicio del contenedor.

HA nativa

La seguridad de los datos replicados entre nodos se controla mediante la stanza

NativeHALocalInstance del archivo `qm.ini` . Por ejemplo:

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

Si FIPS está habilitado, el atributo **SSLFipsRequired** se añade a la stanza, con el valor establecido en Yes:

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

Si el contenedor se ejecuta en un clúster de OpenShift sin soporte de FIPS, el gestor de colas, el servidor web de IBM MQ y los componentes de HA nativa no tienen su soporte de FIPS habilitado automáticamente. Actualmente solo la arquitectura x86-64 está soportada por la plataforma OpenShift para FIPS. Para las arquitecturas Power y Linux for IBM Z, OpenShift no ofrece soporte FIPS. Para habilitar explícitamente el soporte de FIPS en los componentes de IBM MQ para estas arquitecturas, establezca la variable de entorno `MQ_ENABLE_FIPS` en `true` en el YAML del gestor de colas. El siguiente fragmento de código YAML describe el uso de la variable de entorno `MQ_ENABLE_FIPS`:

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

Alteración temporal de la modalidad FIPS automática para IBM MQ en contenedores

Utilice la variable de entorno `MQ_ENABLE_FIPS` para habilitar o inhabilitar explícitamente la modalidad FIPS para los componentes de IBM MQ en el contenedor.

Antes de empezar

Nota: En AIX, Linux, and Windows, IBM MQ proporciona conformidad con FIPS 140-2 a través del módulo criptográfico IBM Crypto for C (ICC). El certificado para este módulo se ha movido al estado Histórico. Los clientes deben ver el [certificado de IBM Crypto for C \(ICC\)](#) y tener en cuenta cualquier consejo proporcionado por NIST. Un módulo FIPS 140-3 de sustitución está actualmente en curso y su estado se puede ver buscándolo en los [módulos CMVP de NIST en la lista de procesos](#).

La imagen de contenedor de IBM MQ Operator 3.2.0 y el gestor de colas 9.4.0.0 en adelante se basan en UBI 9. La conformidad con FIPS 140-3 está pendiente actualmente y su estado se puede visualizar buscando "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" en los [módulos CMVP de NIST en la lista de procesos](#).

Acerca de esta tarea

`MQ_ENABLE_FIPS` da soporte a tres valores:

Automática

Éste es el valor predeterminado.

Si el sistema operativo de host está habilitado para FIPS, todos los componentes (gestor de colas, servidor web de IBM MQ y HA nativa) se ejecutan en modalidad FIPS.

Si el sistema operativo de host no está habilitado para FIPS, todos los componentes no se ejecutan en modalidad FIPS.

true

Este valor activa FIPS para los componentes seleccionados en el contenedor.

El atributo de gestor de colas **SSLFIPS** se establece en YES incluso si IBM MQ en contenedores se ejecuta en un sistema operativo de host que no es compatible con FIPS. Es decir, si el gestor de colas de IBM MQ, el servidor web y la HA nativa son compatibles con FIPS, pero el sistema operativo del contenedor no lo es.

falso

Este valor desactiva la conformidad con FIPS.

El atributo de gestor de colas **SSLFIPS** se establece en NO, incluso si IBM MQ en contenedores se ejecuta en una máquina host compatible con FIPS. Sin embargo, IBM MQ sigue protegiendo las conexiones si se proporcionan la clave privada y los certificados.

Las opciones de JVM no se actualizan para el servidor web de IBM MQ . Sin embargo, el servidor web de IBM MQ sigue ejecutando un punto final HTTPS si se proporcionan la clave privada y los certificados.

La réplica de datos en HA nativa no utiliza la criptografía FIPS.

Ejemplo

A continuación se muestra un YAML de gestor de colas de ejemplo que describe la habilitación de TLS y FIPS para el componente de gestor de colas:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  namespace: ibm-mq-fips
  name: ibm-mq-qm-ppcle
spec:
  license:
    accept: true
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: PPCLEQM
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQ_ENABLE_FIPS
              value: "true"
          name: qmgr
  version: 9.4.0.0-r1
  web:
    enabled: false
  pki:
    keys:
      - name: ibm-mq-tls-certs
        secret:
          secretName: ibm-mq-tls-secret
          items:
            - tls.key
            - tls.crt
```

Planificación de la escalabilidad y el rendimiento para IBM MQ en contenedores

En la mayoría de los casos, el escalado y el rendimiento de IBM MQ en contenedores es el mismo que IBM MQ for Multiplatforms. Sin embargo, existen algunos límites adicionales que puede imponer la plataforma de contenedor.

Acerca de esta tarea

Al planificar la escalabilidad y el rendimiento para IBM MQ en contenedores, tenga en cuenta las opciones siguientes:

Procedimiento

- **Limitar el número de hebras y procesos.**

IBM MQ utiliza hebras para gestionar la simultaneidad. En Linux, las hebras se implementan como procesos, por lo que puede encontrar límites impuestos por la plataforma de contenedor o el sistema operativo, en el número máximo de procesos. A partir de Red Hat OpenShift Container Platform 4.11, hay un límite predeterminado de 4096 procesos por contenedor. Aunque esto es adecuado

para la gran mayoría de escenarios, puede haber casos en los que esto pueda afectar al número de conexiones de cliente para un gestor de colas.

El límite de proceso en Kubernetes lo puede configurar un administrador del clúster utilizando el valor de configuración de kubelet **podPidsLimit**. Consulte [Límites y reservas de ID de proceso](#) en la documentación de Kubernetes . En Red Hat OpenShift Container Platform, también puede [crear un recurso personalizado ContainerRuntimeConfig](#) para editar los parámetros CRI-O.

En la configuración de IBM MQ , también puede establecer el número máximo de conexiones de cliente para un gestor de colas. Consulte [Límites de canal de conexión de servidor](#) para aplicar límites a un canal de conexión de servidor individual y el atributo [MAXCHANNELS INI](#) para aplicar límites a todo el gestor de colas.

- **Limitar el número de volúmenes.**

En los sistemas de nube y contenedores, los volúmenes de almacenamiento conectados a la red se utilizan con frecuencia. Hay límites en el número de volúmenes que se pueden conectar a nodos Linux . Por ejemplo, [AWS EC2 limita a no más de 30 volúmenes por máquina virtual](#). Red Hat OpenShift Container Platform [tiene un límite similar](#), al igual que Microsoft Azure y Google Cloud Platform.

Un gestor de colas de HA nativa requiere un volumen para cada una de las tres instancias e impone que las instancias se repartan entre nodos. Sin embargo, puede configurar el gestor de colas para que utilice tres volúmenes por instancia (datos del gestor de colas, registros de recuperación y datos persistentes).

- **Utilizar técnicas de escalado de IBM MQ .**

En lugar de un pequeño número de gestores de colas grandes, puede ser beneficioso utilizar técnicas de escalado de IBM MQ como, por ejemplo, clústeres uniformes de IBM MQ para ejecutar varios gestores de colas con la misma configuración. Esto tiene la ventaja añadida de que el impacto de un único reinicio de contenedor (por ejemplo, como parte del mantenimiento de la plataforma de contenedor) se reduce.

Preparación, instalación y actualización del entorno para IBM MQ en contenedores

Puede realizar una serie de tareas para preparar el entorno para IBM MQ

Acerca de esta tarea

Si utiliza IBM MQ Operator, prepare el clúster de Red Hat OpenShift Container Platform instalando el operador. Consulte [“Instalación y actualización de IBM MQ Operator”](#) en la página 31

De lo contrario, puede preparar el entorno de contenedor creando sus propias imágenes de contenedor. Consulte [“Preparación para IBM MQ creando su propia imagen de contenedor”](#) en la página 56

Instalación y actualización de IBM MQ Operator

Puede realizar una serie de tareas para instalar, desinstalar y actualizar el IBM MQ Operator.

Acerca de esta tarea

Para empezar a instalar y actualizar IBM MQ Operator en Red Hat OpenShift Container Platform, consulte los temas siguientes.

Procedimiento

- [“Dependencias para IBM MQ Operator”](#) en la página 32
- [“Permisos con ámbito de clúster necesarios para IBM MQ Operator”](#) en la página 32
- [“Verificación de firmas de imagen”](#) en la página 33

- [“Instalación del IBM MQ Operator”](#) en la página 33
- [“Actualización de IBM MQ Operator y gestores de colas”](#) en la página 44
- [“Desinstalación de IBM MQ Operator”](#) en la página 54

Dependencias para IBM MQ Operator

No se instalan otros operadores automáticamente al instalar IBM MQ Operator.

El operador de licencias de IBM debe instalarse por separado para realizar un seguimiento del uso de licencia. Consulte [Despliegue de License Service](#) en la documentación de IBM Cloud Pak for Integration .

Al crear un QueueManager utilizando una licencia de IBM Cloud Pak for Integration , puede elegir si desea utilizar o no el inicio de sesión único con la instancia de IBM Cloud Pak for Integration de Keycloak. El uso de Keycloak está habilitado de forma predeterminada con una licencia de IBM Cloud Pak for Integration , pero si no está instalado, QueueManager entrará en un estado "Bloqueado" hasta que se instalen las dependencias correctas. Consulte [“Instalación del IBM MQ Operator”](#) en la página 33 para obtener más detalles sobre las dependencias.

Permisos con ámbito de clúster necesarios para IBM MQ Operator

IBM MQ Operator requiere permisos con ámbito de clúster para gestionar webhooks y ejemplos de admisión, y para leer información de clase de almacenamiento y versión de clúster.

IBM MQ Operator requiere los siguientes permisos con ámbito de clúster:

- Permiso para gestionar webhooks de admisión. Esto permite crear, recuperar y actualizar webhooks específicos que se utilizan en el proceso de creación y gestión de contenedores proporcionados por el operador.
 - Grupos de API: **admissionregistration.k8s.io**
 - resources: **validatingwebhookconfigurations**
 - verbs: **get, delete**
- Permiso para crear y gestionar recursos que se utilizan en la consola de Red Hat OpenShift para proporcionar ejemplos y fragmentos de código al crear recursos personalizados.
 - Grupos de API: **console.openshift.io**
 - resources: **consoleyamlsamples**
 - verbs: **create, get, update, delete**
- Permiso para leer la versión del clúster. Esto permite al operador retroalimentar cualquier problema con el entorno de clúster.
 - Grupos de API: **config.openshift.io**
 - resources: **clusterversions**
 - verbs: **get, list, watch**
- Permiso para leer clases de almacenamiento en el clúster. Esto permite al operador retroalimentar cualquier problema con las clases de almacenamiento seleccionadas en contenedores.
 - Grupos de API: **storage.k8s.io**
 - resources: **storageclasses**
 - verbs: **get, list**

Nota: IBM MQ Operator también requiere permisos con ámbito de espacio de nombres. Si el IBM MQ Operator está instalado en un ámbito de clúster, los permisos con ámbito de espacio de nombres están presentes en todos los espacios de nombres.

Las imágenes de contenedor del gestor de colas IBM MQ Operator y IBM MQ están firmadas digitalmente.

Acerca de esta tarea

Las firmas digitales proporcionan una forma para que los consumidores de contenidos se aseguren de que lo que descargan es tanto auténtico (se originó de la fuente esperada) como tiene integridad (es lo que esperamos que sea).

Procedimiento

- Verifique las firmas de las imágenes de contenedor del gestor de colas IBM MQ Operator y IBM MQ :
 - Consulte [Verificación de firmas de imagen](#) en la documentación de IBM Cloud Pak for Integration (CP4I) 16.1.0 .

El IBM MQ Operator se puede instalar en Red Hat OpenShift utilizando la consola de OpenShift o la interfaz de línea de mandatos (CLI).

Antes de empezar

Importante:

- Este tema es para instalar IBM MQ Operator para uso autónomo **solo**. Si tiene previsto utilizar el inicio de sesión único de IBM Cloud Pak for Integration o Keycloak para uno o varios gestores de colas, consulte [“Instalación de IBM MQ Operator para su uso con CP4I”](#) en la página 40.
- Revise la guía sobre [estructuración del despliegue](#) antes de instalar IBM MQ Operator.

Para asegurarse de que la instalación se realiza de la forma más fluida posible, asegúrese de que comprende todos los requisitos previos y requisitos antes de iniciar la instalación. Consulte [“Planificación de IBM MQ en contenedores”](#) en la página 7.

Acerca de esta tarea

Los pasos siguientes representan el flujo de tareas típico para instalar IBM MQ Operator:

1. [Instale Red Hat OpenShift Container Platform](#).
2. [Configurar el almacenamiento](#).
3. [Imágenes duplicadas \(solo espacio vacío\)](#).
4. [Añada el catálogo de IBM MQ Operator](#).
5. [Instale IBM MQ Operator](#).
6. [Crear el secreto de clave de titularidad \(solo instalaciones en línea\)](#).
7. [Despliegue el License Service](#).
8. [Desplegar un gestor de colas](#).

Procedimiento

1. Instale Red Hat OpenShift Container Platform.

Para ver los pasos detallados para instalar OpenShift, consulte [Instalación del software de Red Hat 4.6 o posterior](#).

Importante: Asegúrese de instalar una versión soportada de OpenShift Container Platform. Por ejemplo, para utilizar IBM MQ Operator 3.2 o posterior, debe instalar OpenShift Container Platform

4.12 o posterior. Para obtener más información, consulte [Compatibilidad de IBM Cloud Pak y Red Hat OpenShift Container Platform](#).

Para los pasos que utilizan la CLI de Red Hat OpenShift Container Platform, debe haber iniciado sesión en el clúster de OpenShift con `oc login`. Para instalar la CLI, consulte [Iniciación a la CLI de OpenShift](#).

Después de instalar OpenShift, puede verificar y obtener acceso al software de contenedor utilizando la clave de titularidad de IBM que ha creado en [Crear el secreto de clave de titularidad](#).

2. Configure el almacenamiento.

Debe definir clases de almacenamiento en Red Hat OpenShift Container Platform y establecer la configuración de almacenamiento para satisfacer los requisitos de dimensionamiento.

Importante: Los gestores de colas de una sola instancia y HA nativa de IBM MQ pueden utilizar la modalidad de acceso RWO, mientras que los gestores de colas de varias instancias requieren RWX tal como se describe en [“Planificación del almacenamiento para IBM MQ Operator”](#) en la [página 16](#). Los gestores de colas multiinstancia de IBM MQ requieren características de sistema de archivos específicas, que se pueden verificar utilizando las instrucciones de [Prueba de un sistema de archivos compartido para IBM MQ](#).

Puede encontrar una lista de sistemas de archivos compatibles y no conformes conocidos, así como notas sobre otros límites o restricciones, en la [Declaración de prueba para sistemas de archivos IBM MQ](#).

Los proveedores de almacenamiento recomendados se pueden encontrar en la [página CP4I Consideraciones sobre el almacenamiento](#).

3. Imágenes de espejo (solo espacio de aire).

Si el clúster está en un entorno de red restringido (aislado), debe duplicar las imágenes de IBM MQ utilizando los valores siguientes:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.2.1
```

Para crear imágenes duplicadas, consulte [Duplicación de imágenes para un clúster aislado](#).

4. Añada el origen de catálogo de IBM MQ Operator.

Añada el origen de catálogo que hace que los operadores estén disponibles para el clúster. Consulte [“Adición del origen de catálogo de IBM MQ Operator”](#) en la [página 35](#).

5. Instale el IBM MQ Operator.

Elija una de las dos opciones siguientes (utilice la consola o utilice la CLI):

- Opción 1: [Instalar IBM MQ Operator utilizando la consola de OpenShift](#).
- Opción 2: [Instalar IBM MQ Operator utilizando la CLI de OpenShift](#).

6. Cree el secreto de clave de titularidad (solo instalaciones en línea).

El IBM MQ Operator despliega imágenes de gestor de colas que se extraen de un registro de contenedor que realiza una comprobación de titularidad de licencia. Esta comprobación requiere una clave de titularidad que se almacena en un registro de extracción `docker-registry`. Si todavía no tiene una clave de titularidad en el espacio de nombres en el que instalará gestores de colas, siga estas instrucciones para obtener una clave de titularidad y crear un secreto de extracción.

Nota: La clave de titularidad no es necesaria si sólo se van a desplegar los gestores de colas IBM MQ Advanced for Developers (sin garantía).

Puede crear el secreto de clave de titularidad utilizando la consola de OpenShift o la CLI. El ejemplo siguiente utiliza la CLI:

- a. Obtenga la clave de titularidad asignada a su ID de IBM. 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.
- c. Desde la CLI de OpenShift , ejecute el mandato siguiente para crear un secreto de extracción de imágenes denominado `ibm-entitlement-key`.

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

Donde *entitlement_key* es la clave de titularidad que ha copiado en el paso b, *user_email* es el ID de IBM asociado con el software autorizado y *namespace* es el espacio de nombres en el que ha instalado IBM MQ Operator .

7. Despliegue el License Service.

Esto es necesario para supervisar el uso de licencias de los gestores de colas. Siga las instrucciones de [Despliegue de License Service](#).

8. Despliegue un gestor de colas.

Para obtener instrucciones sobre cómo desplegar un gestor de colas de "inicio rápido" de ejemplo, consulte [“Despliegue de un gestor de colas simple utilizando la IBM MQ Operator”](#) en la página 64.

Tareas relacionadas

[“Desinstalación de IBM MQ Operator”](#) en la página 54

Puede utilizar la consola o la CLI de Red Hat OpenShift para desinstalar el IBM MQ Operator de Red Hat OpenShift.

Adición del origen de catálogo de IBM MQ Operator

Añada el origen de catálogo de IBM MQ Operator al clúster de OpenShift para que el IBM MQ Operator esté disponible para la instalación. Esta tarea también es necesaria si está aplicando fixpacks de origen de catálogo antes de completar una actualización.

Acerca de esta tarea

Un catálogo de operadores es un índice de operadores disponibles para ampliar la API de un clúster de Red Hat OpenShift Container Platform para habilitar los productos de software de IBM .

Están disponibles los siguientes orígenes de catálogo:

Opción 1: Origen de catálogo específico para IBM MQ Operator.

Mediante el uso de un origen de catálogo de IBM MQ Operator específico, obtiene el control completo del mantenimiento de versiones de software en un clúster y cuándo se producen las actualizaciones. Una nueva versión de IBM MQ Operator pasa a estar disponible en un clúster de OpenShift **sólo** después de actualizar el origen de catálogo. Este proceso le proporciona un control manual de las actualizaciones, por lo que no es necesario utilizar la opción `Manual` para el valor **Update approval** para los operadores. La opción **Manual** fuerza que se realicen todas las actualizaciones posibles al mismo tiempo y puede bloquear las actualizaciones, por lo tanto, utilice sólo la opción **Automático** . Para obtener más información, consulte la sección "Restricción de actualizaciones automáticas con una estrategia de aprobación" de [Instalación de los operadores utilizando la consola de Red Hat OpenShift](#).

Elija esta opción si está completando una actualización y necesita añadir el origen de catálogo de IBM MQ Operator de una versión más reciente.

Para utilizar esta opción, vaya a [Opción 1: Añadir orígenes de catálogo específicos para el IBM MQ Operator](#).

Opción 2: Catálogo de operadores de IBM .

Con esta opción, las nuevas versiones de operador pasan a estar disponibles y se aplican **sin** ninguna intervención del usuario. Por lo tanto, utilice esta opción **solo** para instalaciones en línea en las que desee actualizaciones **automáticas** de IBM MQ Operator en las que no sean necesarias instalaciones deterministas.

Nota: Esta opción puede ser útil para entornos de prueba de concepto, pero **no es adecuada para entornos de producción**.

Para utilizar esta opción, vaya a [Opción 2: Añadir el IBM Catálogo de operadores](#).

Procedimiento

• Opción 1: Añadir orígenes de catálogo específicos para IBM MQ Operator.

Esta tarea presupone que ha completado los 3 primeros pasos de ["Instalación del IBM MQ Operator"](#) en la [página 33](#).

Esta tarea la debe realizar un administrador del clúster y se debe realizar utilizando la CLI.

a) Sólo actualización: si está aplicando fixpacks de origen de catálogo antes de una actualización, realice los pasos siguientes:

- Confirme que los operadores se están ejecutando correctamente.
- Si hay actualizaciones de IBM MQ Operator pendientes que requieren aprobación manual, apruébelas antes de iniciar este procedimiento. Para obtener más información, consulte "Restricción de actualizaciones automáticas con una estrategia de aprobación" en [Instalación de los operadores utilizando la consola de Red Hat OpenShift](#).

b) Si todavía no lo ha instalado, o si es necesario actualizarlo, [descargue el plug-in de IBM Catalog Management \(versión 1.6.0 o posterior\) desde GitHub](#).

Este plugin le permite ejecutar mandatos **oc ibm-pak** en el clúster.

c) Inicie sesión en el clúster utilizando el mandato **oc login** y las credenciales de usuario:

```
oc login openshift_url -u username -p password -n namespace
```

d) Exporte las variables de entorno siguientes para IBM MQ Operator:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.2.1
export ARCH=ARCHITECTURE
```

donde *ARCHITECTURE* hace referencia a la arquitectura del sistema en el que está desplegando el IBM MQ Operator tiene un valor de amd64, ppc64le o s390x.

Importante: Si va a pasar del catálogo de operadores de IBM al origen de catálogo específico para el IBM MQ Operator, establezca *OPERATOR_VERSION* en la versión del despliegue del IBM MQ Operator.

e) Descargue los archivos del operador de IBM MQ .

Nota: Si está completando una instalación **aislado** , ya debería tener los archivos que necesita después de completar el paso "Imágenes duplicadas" de "Instalación de IBM MQ Operator", en cuyo caso puede saltar al paso ["8"](#) en la [página 37](#) "Aplicar el origen de catálogo de IBM MQ Operator al clúster".

```
oc ibm-pak get ${OPERATOR_PACKAGE_NAME} --version ${OPERATOR_VERSION}
```

f) Genere el origen de catálogo necesario para IBM MQ Operator:

```
oc ibm-pak generate mirror-manifests ${OPERATOR_PACKAGE_NAME} icr.io --version $
{OPERATOR_VERSION}
```

g) Opcional: Genere los orígenes de catálogo y guárdelos en otro directorio.

a. Obtener el origen de catálogo:

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

b. (Opcional) Vaya al directorio en el navegador de archivos para copiar estos artefactos en archivos que puede conservar para su reutilización o para interconexiones.

h) Aplique el origen de catálogo de IBM MQ Operator al clúster.

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

i) Confirme que el origen de catálogo de IBM MQ Operator se ha creado en el espacio de nombres openshift-marketplace :

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

Salida de ejemplo:

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

NAME	DISPLAY	TYPE	PUBLISHER	AGE
ibmmq-operator-catalogsource	ibm-mq-3.1.3	grpc	IBM	23h

Ahora está preparado para completar el [Paso 5 de la instalación de IBM MQ Operator](#).

- **Opción 2: Añadir el catálogo de operadores de IBM .**

Importante: Utilice el IBM Catálogo de operadores **sólo** para las instalaciones en línea en las que desee actualizaciones **automáticas** de IBM MQ Operator en las que no sean necesarias instalaciones deterministas. Esta opción puede ser útil para entornos de prueba de concepto, pero **no es adecuada para entornos de producción**.

El catálogo de operadores de IBM es un índice de operadores disponibles para ampliar la API de un clúster de Red Hat OpenShift Container Platform para habilitar los productos de software de IBM . Al añadir los orígenes de catálogo al clúster de OpenShift se añaden los operadores de IBM a la lista de operadores que puede instalar.

Esta tarea presupone que ha completado los 3 primeros pasos de [“Instalación del IBM MQ Operator”](#) en la [página 33](#).

Esta tarea se puede realizar utilizando la CLI o utilizando la consola web de OpenShift .

Utilización de la CLI

1. Copie la siguiente definición de recurso para los operadores de IBM en un archivo local del sistema:

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

2. Ejecute el mandato siguiente. Sustituya *filename.yaml* por el nombre del archivo que ha creado en el paso anterior:

```
oc apply -f filename.yaml
```

Utilización de la consola web de OpenShift

1. Inicie sesión en la consola web de OpenShift con las credenciales de administrador de clúster de OpenShift .

2. En el banner, pulse el signo más ("+") para abrir el recuadro de diálogo **Importar YAML** .

Nota: No es necesario seleccionar un valor para **Proyecto**. El código YAML del paso siguiente ya incluye el valor correcto para metadata : namespace, lo que garantiza que el origen de catálogo esté instalado en el proyecto correcto (espacio de nombres).

3. 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
  image: 'icr.io/cpopen/ibm-operator-catalog:latest'
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

4. Pulse **Crear**.

Ahora está preparado para completar el [Paso 5 de la instalación de IBM MQ Operator](#).

Instalación de IBM MQ Operator utilizando la consola de OpenShift

El IBM MQ Operator se puede instalar en Red Hat OpenShift utilizando OperatorHub.

Antes de empezar

Esta tarea presupone que ha completado los pasos 1-4 de [“Instalación del IBM MQ Operator”](#) en la [página 33](#).

Procedimiento

1. Inicie una sesión en la consola del clúster de Red Hat OpenShift.
2. En el panel de navegación, pulse **Operadores > OperatorHub**.
Se visualiza la página OperatorHub.
3. En el campo **Todos los elementos** , especifique "IBM MQ".
Se visualiza la entrada de catálogo IBM MQ .
4. Seleccione **IBM MQ**.
Se visualiza la ventana IBM MQ .
5. Pulse **Instalar**.
Se visualiza la página Instalar operador.
6. Especifique los valores siguientes:
 - a) Establezca **Canal** en la versión elegida.
Revise [“Soporte de versiones del IBM MQ Operator”](#) en la [página 14](#) para determinar qué canal de operador elegir.
 - b) Establezca **Modalidad de instalación** en "un espacio de nombres específico en el clúster" (que puede crear en el paso siguiente) o en el ámbito de todo el clúster.
Se recomienda elegir el ámbito de todo el clúster, porque la instalación de distintas versiones de un operador en distintos espacios de nombres puede provocar problemas. Los operadores están diseñados para ser extensiones del plano de control.
 - c) Opcional: Si elige "un espacio de nombres específico en el clúster", establezca el **Espacio de nombres** en el valor del proyecto (espacio de nombres) en el que desea instalar el operador.
Nota: Al utilizar la consola para instalar el operador, puede utilizar un espacio de nombres existente, el espacio de nombres predeterminado proporcionado por el operador, o crear un

espacio de nombres nuevo. Si desea crear un nuevo espacio de nombres, puede crearlo desde este formulario, como se indica a continuación: En el panel de navegación, pulse **Inicio > Proyectos**, seleccione **Crear proyecto**, especifique el **Nombre** del proyecto (el espacio de nombres) que desea crear y, a continuación, pulse **Crear**.

d) Establezca **Estrategia de aprobación** en Automática.

7. Pulse **Instalar** y espere a que se instale el operador.

Se le proporcionará una confirmación cuando se complete la instalación.

Para verificar la instalación, vaya a **Operadores > Operadores instalados** y seleccione el proyecto en la lista desplegable **Proyectos**. El estado del operador cambia a Satisfactorio cuando la instalación se ha completado.

Qué hacer a continuación

Ahora está preparado para [Crear el secreto de clave de titularidad](#) (paso 6 de [“Instalación del IBM MQ Operator”](#) en la página 33).

Instalación de IBM MQ Operator utilizando la CLI de Red Hat OpenShift

IBM MQ Operator se puede instalar en Red Hat OpenShift utilizando la interfaz de línea de mandatos (CLI).

Antes de empezar

Esta tarea presupone que ha completado los pasos 1-4 de [“Instalación del IBM MQ Operator”](#) en la [página 33](#).

Procedimiento

1. Inicie sesión en la interfaz de línea de mandatos (CLI) de Red Hat OpenShift utilizando **oc login**.
2. Opcional: Cree un espacio de nombres para utilizarlo para IBM MQ Operator.

El ámbito de IBM MQ Operator 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.

Para crear un nuevo espacio de nombres en la CLI, ejecute el mandato siguiente:

```
oc create namespace namespace_name
```

Donde *nombre_espacio_nombres* es el nombre del espacio de nombres que desea crear.

3. Vea la lista de operadores disponibles para el clúster desde OperatorHub:

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

4. Inspeccione el IBM MQ Operator para verificar su **InstallModes** soportado y su **Channels** disponible.

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

5. Opcional: Cree un **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 del 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 **InstallMode** del operador, ya sea la modalidad **AllNamespaces** o **SingleNamespace**.

Si el operador que desea instalar utiliza la modalidad **AllNamespaces**, el espacio de nombres **openshift-operators** ya tiene un **OperatorGroup** adecuado en su lugar y puede omitir este paso.

Si el operador utiliza la modalidad `SingleNamespace` y todavía no tiene un **OperatorGroup** adecuado en su lugar, cree uno ejecutando el mandato siguiente:

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

6. Revise [“Soporte de versiones del IBM MQ Operator”](#) en la página 14 para determinar qué canal de operador elegir.
7. Instale el operador.

Utilice el mandato siguiente, cambiando *ibm-mq-operator-channel* para que coincida con el canal de la versión del operador de IBM MQ que desea instalar, y cambiando *namespace_name* a **openshift-operators** si está utilizando la modalidad "AllNamespaces", o al espacio de nombres en el que desea desplegar el operador de IBM MQ si está utilizando la modalidad "SingleNamespace".

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-mq
  namespace: namespace_name
spec:
  channel: ibm-mq-operator-channel
  installPlanApproval: Automatic
  name: ibm-mq
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

8. Después de unos minutos, se instala el operador. Ejecute el mandato siguiente para verificar que todos los componentes están en estado Satisfactorio:

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

Donde *nombre_espacio_nombres* es **openshift-operators** si está utilizando la modalidad "AllNamespaces", o el nombre del proyecto (espacio de nombres) si está utilizando la modalidad "SingleNamespace".

Qué hacer a continuación

Ahora está preparado para [Crear el secreto de clave de titularidad](#) (paso 6 de [“Instalación del IBM MQ Operator”](#) en la página 33).

Instalación de IBM MQ Operator para su uso con CP4I

Para su uso con IBM Cloud Pak for Integration (CP4I), IBM MQ Operator se puede instalar en Red Hat OpenShift a través de la consola de OpenShift o la interfaz de línea de mandatos (CLI).

Antes de empezar

Importante:

- Este tema es para instalar IBM MQ Operator para su uso con CP4I, o si tiene previsto desplegar al menos uno de los gestores de colas utilizando una CP4I licencia **sólo**. Para obtener instrucciones sobre cómo instalar IBM MQ Operator para uso autónomo, consulte [“Instalación del IBM MQ Operator”](#) en la [página 33](#).
- Revise la guía sobre [estructuración del despliegue](#) antes de instalar IBM MQ Operator.

Para asegurarse de que la instalación se realiza de la forma más fluida posible, asegúrese de que comprende todos los requisitos previos y requisitos antes de iniciar la instalación. Consulte [“Planificación de IBM MQ en contenedores”](#) en la página 7.

Acerca de esta tarea

Los pasos siguientes representan el flujo de tareas típico para instalar IBM MQ Operator:

1. [Instale Red Hat OpenShift Container Platform.](#)
2. [Configurar el almacenamiento.](#)
3. [Imágenes duplicadas \(solo espacio vacío\).](#)
4. [Añada el catálogo de IBM MQ Operator y prepare el clúster.](#)
5. [Instale el IBM MQ Operator.](#)
6. [Crear el secreto de clave de titularidad \(solo instalaciones en línea\).](#)
7. [Opcional: instale IBM Cloud Pak for Integration \(CP4I\) y sus dependencias.](#)
8. [Despliegue el License Service.](#)
9. [Desplegar un gestor de colas.](#)

Procedimiento

1. Instale Red Hat OpenShift Container Platform.

Para ver los pasos detallados para instalar OpenShift, consulte [Instalación del software de Red Hat 4.6 o posterior](#).

Importante: Asegúrese de instalar una versión soportada de OpenShift Container Platform. Por ejemplo, para utilizar IBM MQ Operator 3.2 o posterior, debe instalar OpenShift Container Platform 4.12 o posterior. Para obtener más información, consulte [Compatibilidad de IBM Cloud Pak y Red Hat OpenShift Container Platform](#).

Para los pasos que utilizan la CLI de Red Hat OpenShift Container Platform, debe haber iniciado sesión en el clúster de OpenShift con `oc login`. Para instalar la CLI, consulte [Iniciación a la CLI de OpenShift](#).

Después de instalar OpenShift, puede verificar y obtener acceso al software de contenedor utilizando la clave de titularidad de IBM que cree en [Crear el secreto de clave de titularidad](#).

2. Configure el almacenamiento.

Debe definir clases de almacenamiento en Red Hat OpenShift Container Platform y establecer la configuración de almacenamiento para satisfacer los requisitos de dimensionamiento.

Importante: Los gestores de colas de instancia única y HA nativa de IBM MQ pueden utilizar la modalidad de acceso RWO, mientras que los gestores de colas de varias instancias requieren RWX tal como se describe en [“Planificación del almacenamiento para IBM MQ Operator”](#) en la página 16. Los gestores de colas multiinstancia de IBM MQ requieren características de sistema de archivos específicas, que se pueden verificar utilizando las instrucciones de [Prueba de un sistema de archivos compartidos para IBM MQ](#).

Puede encontrar una lista de sistemas de archivos compatibles y no conformes conocidos, así como notas sobre otros límites o restricciones, en la [Declaración de prueba para sistemas de archivos IBM MQ](#).

Los proveedores de almacenamiento recomendados se pueden encontrar en la página [CP4I Consideraciones sobre el almacenamiento](#).

3. Imágenes de espejo (solo espacio de aire).

Si el clúster está en un entorno de red restringido (aislado), debe duplicar las imágenes de IBM MQ. En función de la configuración, es posible que también tenga que duplicar algunos componentes

adicionales. Lea la siguiente información y, a continuación, duplique las imágenes según sea necesario.

- Debe duplicar las imágenes de IBM MQ . Utilice los valores siguientes:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.2.1
```

- También debe duplicar algunos componentes necesarios adicionales si tiene previsto desplegar al menos un gestor de colas donde se cumplen **todas** las sentencias siguientes:
 - Está utilizando una licencia de CP4I .
 - El IBM MQ Console está habilitado.
 - Está utilizando el servicio de IBM Cloud Pak for Integration Keycloak para la autenticación y autorización de inicio de sesión único (SSO) de IBM MQ Console (el valor predeterminado).

Si todas las sentencias anteriores son verdaderas, Keycloak proporciona SSO. Por lo tanto, así como para el origen de catálogo de IBM MQ Operator , también debe repetir los pasos para cada uno de estos componentes necesarios adicionales:

- IBM Cloud Pak foundational services
- IBM Cloud Pak for Integration
- Keycloak (operador de Red Hat OpenShift)

Para crear imágenes duplicadas, consulte [Duplicación de imágenes para un clúster aislado](#).

4. Añada el origen de catálogo de IBM MQ Operator .

Añada el origen de catálogo que hace que el IBM MQ Operator esté disponible para el clúster utilizando los valores siguientes:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.2.1
export ARCH=ARCHITECTURE
```

donde *ARCHITECTURE* hace referencia a la arquitectura del sistema y tiene un valor de amd64, ppc64le o s390x.

Existen algunos componentes necesarios adicionales al desplegar al menos un gestor de colas donde se cumplen **todas** las sentencias siguientes:

- Está utilizando una licencia de CP4I .
- El IBM MQ Console está habilitado.
- Está utilizando el servicio de IBM Cloud Pak for Integration Keycloak para la autenticación y autorización de inicio de sesión único (SSO) de IBM MQ Console (el valor predeterminado).

Si todas las sentencias anteriores son verdaderas, Keycloak proporciona SSO. Por lo tanto, así como para el origen de catálogo de IBM MQ Operator , también debe repetir los pasos para cada uno de estos componentes necesarios adicionales:

- IBM Cloud Pak foundational services
- IBM Cloud Pak for Integration
- Keycloak (operador de Red Hat OpenShift)

Siga los pasos para los orígenes de catálogo necesarios en [Adición de orígenes de catálogo a un clúster](#).

5. Instale el IBM MQ Operator.

Elija una de las dos opciones siguientes (utilice la consola o utilice la CLI):

- Opción 1: [Instalar IBM MQ Operator utilizando la consola de OpenShift](#).
- Opción 2: [Instalar IBM MQ Operator utilizando la CLI de OpenShift](#).

6. Cree el secreto de clave de titularidad (solo instalaciones en línea).

El IBM MQ Operator despliega imágenes de gestor de colas que se extraen de un registro de contenedor que realiza una comprobación de titularidad de licencia. Esta comprobación requiere una clave de titularidad que se almacena en un registro de extracción `docker-registry`. Si todavía no tiene una clave de titularidad en el espacio de nombres en el que instalará gestores de colas, siga estas instrucciones para obtener una clave de titularidad y crear un secreto de extracción.

Nota: La clave de titularidad no es necesaria si solo se van a desplegar los gestores de colas IBM MQ Advanced for Developers (sin garantía).

Puede crear el secreto de clave de titularidad utilizando la consola de OpenShift o la CLI. El ejemplo siguiente utiliza la CLI:

- a. Obtenga la clave de titularidad asignada a su ID de IBM . 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.
- c. Desde la CLI de OpenShift , ejecute el mandato siguiente para crear un secreto de extracción de imágenes denominado `ibm-entitlement-key`.

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

Donde `entitlement_key` es la clave de titularidad que ha copiado en el paso b, `user_email` es el ID de IBM asociado con el software autorizado y `namespace` es el espacio de nombres en el que ha instalado IBM MQ Operator .

7. Opcional: Instale CP4I y sus dependencias.

Existen algunos componentes necesarios adicionales al desplegar al menos un gestor de colas donde se cumplen **todas** las sentencias siguientes:

- Está utilizando una licencia de CP4I .
- El IBM MQ Console está habilitado.
- Está utilizando el servicio de CP4I Keycloak para la autenticación y autorización de inicio de sesión único (SSO) de IBM MQ Console (el valor predeterminado).

Si todas las sentencias anteriores son verdaderas, el inicio de sesión único lo proporciona Keycloak y debe completar los pasos adicionales siguientes:

- Instale el operador de IBM Cloud Pak foundational services en la misma modalidad de instalación que el operador de CP4I . Para ver las versiones soportadas, consulte [Versiones de canal de operador para este release](#).
- [Instale el operador de CP4I](#).
- Opcional: Desplegar la interfaz de usuario de plataforma.
 - a. Cree el espacio de nombres `ibm-common-services` . Cuando haya iniciado sesión en el clúster de OpenShift a través de la CLI, ejecute el mandato siguiente:

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

- b. [Desplegar la interfaz de usuario de la plataforma](#).

8. Despliegue el License Service.

Esto es necesario para supervisar el uso de licencias de los gestores de colas. Siga las instrucciones de [Despliegue de License Service](#).

9. Despliegue un gestor de colas.

Para obtener instrucciones sobre cómo desplegar un gestor de colas de "inicio rápido" de ejemplo, consulte [“Despliegue de un gestor de colas simple utilizando la IBM MQ Operator”](#) en la página 64.

Tareas relacionadas

“Desinstalación de IBM MQ Operator” en la [página 54](#)

Puede utilizar la consola o la CLI de Red Hat OpenShift para desinstalar el IBM MQ Operator de Red Hat OpenShift.

Actualización de IBM MQ Operator y gestores de colas

Existen diferentes procesos de actualización para los usuarios de IBM MQ Operator, en función de si utiliza licencias de IBM MQ o licencias de IBM Cloud Pak for Integration (CP4I). Complete el paso de actualización para el tipo de despliegue.

Acerca de esta tarea

Para actualizar IBM MQ Operator y los gestores de colas, realice uno de los pasos siguientes:

Procedimiento

- Opción 1: **Actualizar los despliegues a la versión más reciente en el canal de operador actual.**
Para actualizar los despliegues de IBM MQ Operator a la versión más reciente en el canal de operador actual, consulte [“Actualización a un release de seguridad de canal de IBM MQ Operator más reciente”](#) en la [página 45](#).
- Opción 2: **Actualizar las licencias de IBM MQ Operator for IBM MQ .**
Para actualizar los despliegues de IBM MQ Operator donde **solo** IBM MQ se utilizan licencias, consulte [“Actualización de IBM MQ Operator”](#) en la [página 44](#).
- Opción 3: **Actualizar IBM MQ Operator para usuarios de CP4I .**
Actualice los despliegues de IBM MQ Operator para los usuarios de IBM Cloud Pak for Integration. Esto incluye si ha desplegado al menos uno de los gestores de colas bajo una licencia de CP4I . Consulte [“Actualización de IBM MQ Operator para usuarios de CP4I”](#) en la [página 49](#).

Actualización de IBM MQ Operator

Actualice los despliegues de IBM MQ Operator donde **solo** IBM MQ se utilizan licencias.

Antes de empezar

Importante: Esta tarea es para los usuarios de las licencias IBM MQ Operator y **solo** IBM MQ . Si es un usuario de IBM Cloud Pak for Integration (CP4I), o si ha desplegado al menos uno de los gestores de colas utilizando una licencia de CP4I , consulte [“Actualización de IBM MQ Operator para usuarios de CP4I”](#) en la [página 49](#).

Acerca de esta tarea

Realice cualquiera de los pasos siguientes que coincidan con la actualización que necesita.

Nota: La versión 3.2.x de IBM MQ Operator se ha publicado como un release CD y un release SC2 .

Procedimiento

- Opción 1: [“Actualización a un release de seguridad de canal de IBM MQ Operator más reciente”](#) en la [página 45](#)
- Opción 2: [“Actualización de un canal 2.0.x LTS IBM MQ Operator a 3.2.x SC2/CD”](#) en la [página 46](#)
- Opción 3: [“Actualización de un CD IBM MQ Operator al canal 3.2.x SC2/CD”](#) en la [página 47](#)

La actualización de IBM MQ Operator le permite actualizar los gestores de colas.

Antes de empezar

Importante: Este tema es para actualizar los despliegues de IBM MQ Operator al release de seguridad más reciente en el canal del despliegue. Si esto no se aplica al despliegue, consulte las vías de acceso de actualización alternativas que se describen en [“Actualización de IBM MQ Operator y gestores de colas”](#) en la página 44.

Acerca de esta tarea

En primer lugar, actualice el origen de catálogo y, a continuación, actualice los gestores de colas. Hay dos opciones, en función del origen de catálogo utilizado para desplegar el IBM MQ Operator que se está actualizando.

Opción 1: Origen de catálogo específico para IBM MQ Operator

Una nueva versión de IBM MQ Operator pasa a estar disponible en un clúster de OpenShift **solo** después de actualizar el origen de catálogo. Este proceso le proporciona un control manual de las actualizaciones, por lo que no es necesario utilizar la opción **Manual** para el valor **Update approval** para los operadores. La opción **Manual** fuerza que se realicen todas las actualizaciones posibles al mismo tiempo y puede bloquear las actualizaciones, por lo tanto, utilice sólo la opción **Automático**. Para obtener más información, consulte la sección "Restricción de actualizaciones automáticas con una estrategia de aprobación" de [Instalación de los operadores utilizando la consola de Red Hat OpenShift](#).

Para utilizar esta opción, vaya a [Actualizar con el origen de catálogo específico para el IBM MQ Operator](#).

Opción 2: Catálogo de operadores de IBM

Con esta opción, las nuevas versiones de operador pasan a estar disponibles y se aplican **sin** ninguna intervención del usuario. Por lo tanto, utilice esta opción **solo** para instalaciones en línea en las que desee actualizaciones **automáticas** de IBM MQ Operator en las que no sean necesarias instalaciones deterministas. Esta opción puede ser útil para entornos de prueba de concepto, pero **no es adecuada para entornos de producción**.

Para utilizar esta opción, vaya a [Actualizar con el catálogo de operadores de IBM](#).

Para pasar de utilizar el catálogo de operadores de IBM a utilizar el origen de catálogo específico para el IBM MQ Operator, que le proporciona un mayor control sobre las actualizaciones, consulte [“Traslado al origen de catálogo específico para IBM MQ Operator”](#) en la página 48.

Procedimiento

• Actualización con el origen de catálogo específico para el IBM MQ Operator

a) Aplique el origen de catálogo más reciente.

Siga las instrucciones en ["Añadir orígenes de catálogo específicos para el IBM MQ Operator" en Adición del IBM MQ Operator origen de catálogo](#).

b) Si tiene el estado **Actualizar aprobación** para IBM MQ Operator establecido en **Automático**, el operador se actualiza. Si tiene la **Aprobación de actualización** establecida en **Manual**, siga estos pasos para actualizar IBM MQ Operator:

a. En el panel de navegación, pulse **Operadores > Operadores instalados**.

Se visualizan todos los operadores instalados en el proyecto especificado.

b. Seleccione el operador **IBM MQ**

c. Vaya a la pestaña **Suscripción**

- d. Pulse **Actualizar disponible**
- e. Pulse **Vista previa de InstallPlan**
- f. Pulse **Aprobar** para completar la actualización.

El operador actualiza a la nueva versión.

- c) Actualice los gestores de colas de IBM MQ .

Continúe con las instrucciones de [Actualizar gestores de colas de IBM MQ](#).

- **Actualizar con el catálogo de operadores de IBM**

- a) Actualice IBM MQ Operator a una versión más reciente.

Si tiene establecidas actualizaciones automáticas, cuando se publique un nuevo release de seguridad, IBM MQ Operator completará una actualización. Si no tiene establecidas las actualizaciones automáticas, apruebe manualmente la actualización de IBM MQ Operator :

- Si hay una actualización disponible, **Upgrade Status** podría ser "Actualización disponible".
- En este caso, puede haber un control disponible que puede utilizar para aprobar el **InstallPlan** que actualiza el IBM MQ Operator.

- b) Actualizar cualquier gestor de colas de IBM MQ

Continúe con las instrucciones de [Actualizar gestores de colas de IBM MQ](#).

- **Actualice los gestores de colas de IBM MQ.**

Debe actualizar los gestores de colas de IBM MQ a una versión más reciente después de actualizar el IBM MQ Operator.

La tabla siguiente describe la versión más reciente del gestor de colas IBM MQ para cada canal de operador activo. Utilizando la versión relevante, siga el procedimiento de ["Actualización de un gestor de colas de IBM MQ utilizando Red Hat OpenShift"](#) en la página 52.

Canal de operador	Gestor de colas IBM MQ más reciente
v3.2 (SC2/CD)	9.4.0.0-r1



La actualización de IBM MQ Operator le permite actualizar los gestores de colas.

Antes de empezar

Importante:

- Esta tarea es para los usuarios de las licencias IBM MQ Operator y **solo** IBM MQ . Si es un usuario de IBM Cloud Pak for Integration (CP4I), o si ha desplegado al menos uno de los gestores de colas utilizando una licencia de CP4I , consulte ["Actualización de IBM MQ Operator para usuarios de CP4I"](#) en la página 49.
- Este tema es para actualizar despliegues de 2.0.x Long Term Support (LTS) IBM MQ Operator al canal Support Cycle 2 (SC2) de IBM MQ Operator 3.2.x **únicamente**. Si esto no se aplica al despliegue, consulte las vías de acceso de actualización alternativas descritas en ["Actualización de IBM MQ Operator y gestores de colas"](#) en la página 44.

Para actualizar a IBM MQ Operator 3.2.1 , debe ejecutar Red Hat OpenShift Container Platform 4.12 o posterior. Para verificar las versiones compatibles para cada canal de IBM MQ Operator , consulte ["Versiones de Red Hat OpenShift Container Platform compatibles"](#) en la página 15. Para actualizar la plataforma, consulte [Actualización de Red Hat OpenShift](#).

Procedimiento

1. Imágenes duplicadas (solo espacio vacío).

Debe duplicar las imágenes de IBM MQ . Complete los pasos en el enlace siguiente, utilizando sólo estos valores:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.2.1
```

Debe omitir la sección 3.5 "Configurar el clúster", porque la conexión con el registro de imágenes debería haberse configurado durante las instalaciones o actualizaciones anteriores.


Enlace: [Duplicación de imágenes para un clúster aislado.](#)

2. Actualice el IBM MQ Operator a 3.2.1.

Consulte ["Actualización de IBM MQ Operator utilizando Red Hat OpenShift"](#) en la página 50.

3. Actualice las instancias.

Para recibir las últimas características y arreglos de seguridad, actualice el operando de IBM MQ (imagen del contenedor del gestor de colas) a la última versión de CD (9.4.0.0-r1). Consulte ["Actualización de un gestor de colas de IBM MQ utilizando Red Hat OpenShift"](#) en la página 52.

 *Actualización de un CD IBM MQ Operator al canal 3.2.x SC2/CD*
La actualización de IBM MQ Operator le permite actualizar los gestores de colas.

Antes de empezar

Importante:

- Esta tarea es para los usuarios de las licencias IBM MQ Operator y **solo** IBM MQ . Si es un usuario de IBM Cloud Pak for Integration (CP4I), o si ha desplegado al menos uno de los gestores de colas utilizando una licencia de CP4I , consulte ["Actualización de IBM MQ Operator para usuarios de CP4I"](#) en la página 49.
- Este tema es para actualizar los despliegues de Continuous Delivery (CD) de la IBM MQ Operator anterior a la versión 3.2.0, a la versión 3.2.1 **solamente**. Si esto no se aplica al despliegue, consulte las vías de acceso de actualización alternativas descritas en ["Actualización de IBM MQ Operator y gestores de colas"](#) en la página 44.

Para actualizar a IBM MQ Operator 3.2.1 , debe ejecutar Red Hat OpenShift Container Platform 4.12 o posterior. Para verificar las versiones compatibles para cada canal de IBM MQ Operator , consulte ["Versiones de Red Hat OpenShift Container Platform compatibles"](#) en la página 15. Para actualizar la plataforma, consulte [Actualización de Red Hat OpenShift](#).

Procedimiento

1. Opcional: Actualice un IBM MQ Operator que esté actualmente en una versión de CD anterior a 3.0.0.

Si su IBM MQ Operator está actualmente en una versión de CD anterior a 3.0.0, siga los pasos relevantes en [Migración al canal de CD actual del operador de IBM MQ \(documentación de IBM MQ 9.3 \)](#) y, a continuación, vuelva aquí para actualizar a la última versión de CD . Tenga en cuenta que este es un paso de requisito previo obligatorio antes de actualizar a la versión 3.2.1.

2. Imágenes duplicadas (solo espacio vacío).

Debe duplicar las imágenes de IBM MQ . Complete los pasos en el enlace siguiente, utilizando sólo estos valores:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.2.1
```


Debe omitir la sección 3.5 "Configurar el clúster", porque la conexión con el registro de imágenes debería haberse configurado durante las instalaciones o actualizaciones anteriores.

Enlace: [Duplicación de imágenes para un clúster aislado.](#)

3. Actualice el IBM MQ Operator a 3.2.1.

Consulte ["Actualización de IBM MQ Operator utilizando Red Hat OpenShift"](#) en la página 50.

4. Actualice las instancias.

Para recibir las últimas características y arreglos de seguridad, actualice el operando de IBM MQ (imagen del contenedor del gestor de colas) a la última versión de CD (9.4.0.0-r1). Consulte ["Actualización de un gestor de colas de IBM MQ utilizando Red Hat OpenShift"](#) en la página 52.



Si tiene una instalación de IBM MQ Operator desde un release anterior y está utilizando el catálogo de operadores de IBM, la aplicación del origen de catálogo específico es la forma más eficaz de controlar completamente el mantenimiento de versiones de software en un clúster.

Antes de empezar

Importante: Esta tarea debe realizarla un administrador del clúster. Consulte [Roles y permisos deOpenShift.](#)

Los pasos siguientes se completan utilizando la CLI.

Acerca de esta tarea

El catálogo de operadores de IBM es un índice de operadores disponibles para ampliar la API de un clúster de Red Hat OpenShift Container Platform para habilitar los productos de software de IBM.

Este procedimiento mueve una instalación del IBM MQ Operator desde el catálogo de operadores de IBM para que pueda utilizar el origen de catálogo específico para el IBM MQ Operator.

Procedimiento

1. Añada el catálogo de IBM MQ Operator .

Siga las instrucciones en ["Añadir orígenes de catálogo específicos para el IBM MQ Operator" en Adición del IBM MQ Operator origen de catálogo.](#)

2. Confirme que el origen de catálogo de IBM MQ Operator se ha creado en el espacio de nombres openshift-marketplace .

Ejecute el siguiente mandato:

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

Salida de ejemplo:

```
oc get catalogsource -n openshift-marketplace
NAME                                DISPLAY                                TYPE    PUBLISHER  AGE
ibm-operator-catalog                IBM Operator Catalog                  grpc   IBM        23h
ibmmq-operator-catalogsource        ibm-mq-3.1.3                          grpc   IBM        23h
```

3. Opcional: Suprima el origen del catálogo del operador de IBM .

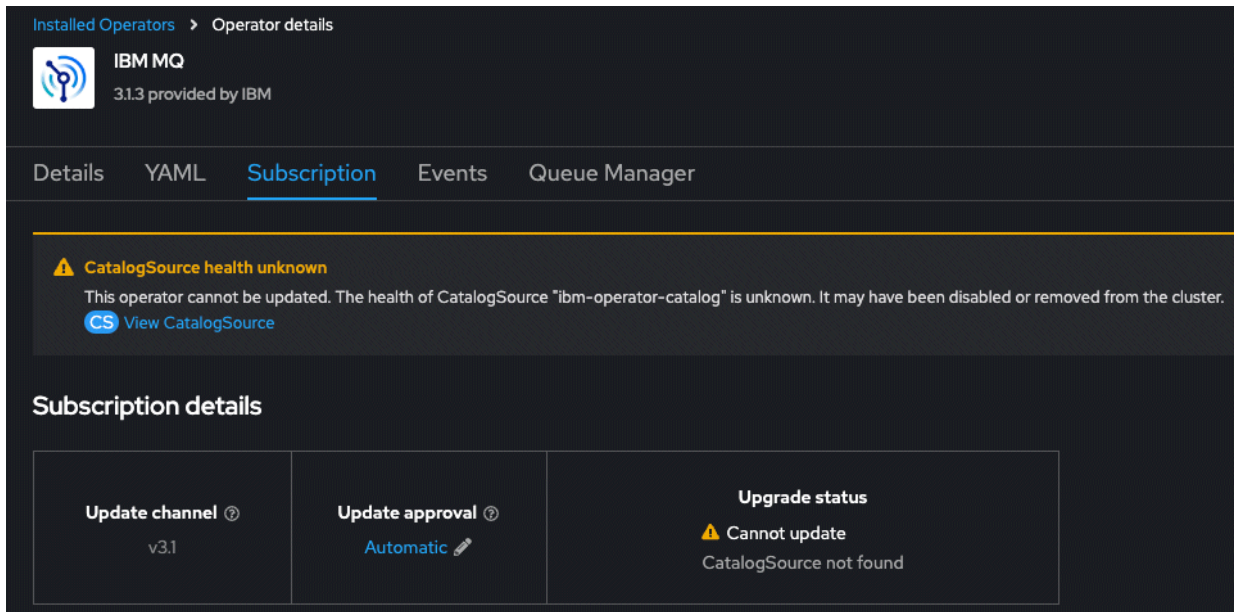


Aviso: Sólo debe completar este paso si está seguro de que no hay otros operadores que utilicen el catálogo de operadores de IBM .

Ejecute el siguiente mandato:

```
oc delete catalogsource ibm-operator-catalog -n openshift-marketplace
```


El estado de IBM MQ Operator cambia a `CatalogSource not found`. Es lo esperado.



Installed Operators > Operator details

IBM MQ
3.1.3 provided by IBM

Details | YAML | **Subscription** | Events | Queue Manager

⚠ CatalogSource health unknown
This operator cannot be updated. The health of CatalogSource "ibm-operator-catalog" is unknown. It may have been disabled or removed from the cluster.
[View CatalogSource](#)

Subscription details

Update channel ⓘ v3.1	Update approval ⓘ Automatic ✎	Upgrade status ⚠ Cannot update CatalogSource not found
---------------------------------	---	---

4. Cambie la suscripción de IBM MQ Operator para que apunte al nuevo origen de catálogo de IBM MQ Operator específico.

a) Edite la suscripción.

Ejecute el mandato siguiente, sustituyendo `OPERATOR-NAMESPACE` por `openshift-operators` para instalaciones de todo el clúster de IBM MQ Operator o el espacio de nombres específico en el que se despliega IBM MQ Operator :

```
oc edit subscription ibm-mq -n OPERATOR-NAMESPACE
```

b) Cambie el valor `spec.source` de `ibm-operator-catalog` por el nombre del origen de catálogo creado en el paso "1" en la página 48.

Por ejemplo:

```
spec:
  channel: v3.1
  installPlanApproval: Automatic
  name: ibm-mq
  source: ibm-operator-catalog # CHANGE --> ibmmq-operator-catalogsource
  sourceNamespace: openshift-marketplace
```

c) Guarde los cambios.

La instalación de IBM MQ Operator ahora apunta al origen de catálogo de IBM MQ Operator . Si ha suprimido el catálogo de operadores de IBM , el estado se revierte de "CatalogSource no encontrado" a "Satisfactorio".

Resultados

La instalación del IBM MQ Operator ahora apunta al origen de catálogo específico para el IBM MQ Operator. Esto le proporciona un control total sobre las actualizaciones del operador.

Actualización de IBM MQ Operator para usuarios de CP4I

Actualice los despliegues del IBM MQ Operator donde se utiliza una licencia de IBM Cloud Pak for Integration (CP4I).

Antes de empezar

Importante: Esta tarea es para usuarios de CP4I . Esto incluye si ha desplegado al menos uno de los gestores de colas bajo una licencia de CP4I . Si esto no se aplica a usted, consulte [“Actualización de IBM MQ Operator”](#) en la página 44.

Acerca de esta tarea

Complete una de las opciones siguientes:

Procedimiento

- **Opción 1:** Actualizar despliegues de 2.0.x Long Term Support (LTS) IBM MQ Operator
Siga los pasos de [Actualización desde 2022.2](#) generando un plan de actualización.
- **Opción 2:** Actualizar un despliegue de 3.0.x o 3.1.x de IBM MQ Operator
Siga los pasos de [Actualización desde 2023.4](#) generando un plan de actualización.
- **Opción 3:** Actualizar otros despliegues del IBM MQ Operator
Siga los pasos relevantes de [Migración al canal de CD actual de la documentación de IBM MQ Operator \(IBM MQ 9.3\)](#) y, a continuación, vuelva aquí y continúe con la **Opción 2**. Tenga en cuenta que este es un paso de requisito previo obligatorio.

Actualización de IBM MQ Operator utilizando Red Hat OpenShift

Puede actualizar IBM MQ Operator utilizando la consola web o la CLI de Red Hat OpenShift .

Procedimiento

Para actualizar el IBM MQ Operator utilizando Red Hat OpenShift, realice una de las tareas siguientes:

- [“Actualización de IBM MQ Operator utilizando la consola de Red Hat OpenShift”](#) en la página 50
- [“Actualización de IBM MQ Operator utilizando la CLI de Red Hat OpenShift”](#) en la página 51

Actualización de IBM MQ Operator utilizando la consola de Red Hat OpenShift

El IBM MQ Operator se puede actualizar utilizando el concentrador del operador.

Antes de empezar

Nota: La última versión de CD de IBM MQ Operator es 3.2.1, y es una versión de SC2 y CD . Para obtener las últimas notas del release de IBM MQ Operator , consulte [Historial del release de IBM MQ Operator](#).

Inicie una sesión en la consola del clúster de Red Hat OpenShift.

Procedimiento

1. Revise [“Soporte de versiones del IBM MQ Operator”](#) en la página 14 para determinar a qué canal de operador se debe actualizar.
2. Aplicar origen de catálogo más reciente.

Si está utilizando el origen de catálogo específico para IBM MQ Operator, en lugar de `ibm-operator-catalog`, debe aplicar el origen de catálogo para la nueva versión de IBM MQ .

Para pasar de utilizar el catálogo de operadores de IBM a utilizar el origen de catálogo específico para el IBM MQ Operator obtener un mayor control sobre las actualizaciones, consulte los pasos de [“Traslado al origen de catálogo específico para IBM MQ Operator”](#) en la página 48 antes de volver a completar el paso “3” en la página 51.

Si está utilizando el catálogo del operador de IBM (sólo algunas instalaciones en línea), continúe con el paso “3” en la página 51.

Siga las instrucciones de [“Adición del origen de catálogo de IBM MQ Operator”](#) en la página 35.

3. Actualice el IBM MQ Operator. Las nuevas versiones mayores/menores de IBM MQ Operator se entregan a través de nuevos canales de suscripción. Para actualizar el operador a una nueva versión principal/menor, tendrá que actualizar el canal seleccionado en la suscripción de IBM MQ Operator .

- a) En el panel de navegación, pulse **Operadores > Operadores instalados**.

Se visualizan todos los operadores instalados en el proyecto especificado.

- b) Seleccione el operador **IBM MQ**



- c) Vaya a la pestaña **Suscripción**

- d) Pulse el **Canal**

Se visualiza la ventana **Cambiar canal de actualización de suscripción** .

- e) Seleccione el canal deseado y pulse **Guardar**.

El operador actualizará a la última versión disponible para el nuevo canal. Consulte [“Soporte de versiones del IBM MQ Operator”](#) en la página 14.

  *Actualización de IBM MQ Operator utilizando la CLI de Red Hat OpenShift*
IBM MQ Operator se puede actualizar desde la línea de mandatos.

Antes de empezar

Nota: La última versión de CD de IBM MQ Operator es 3.2.1, y es una versión de SC2 y CD . Para obtener las últimas notas del release de IBM MQ Operator , consulte [Historial del release de IBM MQ Operator](#).

Inicie sesión en el clúster utilizando **oc login**.

Procedimiento

1. Revise [“Soporte de versiones del IBM MQ Operator”](#) en la página 14 para determinar a qué canal de operador se debe actualizar.
2. Aplicar origen de catálogo más reciente.

Si está utilizando el origen de catálogo específico para IBM MQ Operator, en lugar de `ibm-operator-catalog`, debe aplicar el origen de catálogo para la nueva versión de IBM MQ .

Para pasar de utilizar el catálogo de operadores de IBM a utilizar el origen de catálogo específico para el IBM MQ Operator, obtener un mayor control sobre las actualizaciones, consulte los pasos de [“Traslado al origen de catálogo específico para IBM MQ Operator”](#) en la página 48 antes de volver a completar el paso [“3”](#) en la página 51.

Si está utilizando el catálogo del operador de IBM (sólo algunas instalaciones en línea), continúe con el paso [“3”](#) en la página 51.

Siga las instrucciones de [“Adición del origen de catálogo de IBM MQ Operator”](#) en la página 35.

3. Actualice el IBM MQ Operator. Las nuevas versiones mayores/menores de IBM MQ Operator se entregan a través de nuevos canales de suscripción. Para actualizar el operador a una nueva versión principal o menor, tendrá que actualizar el canal seleccionado en la suscripción de IBM MQ Operator .

- a) Asegúrese de que el canal de actualización de IBM MQ Operator necesario esté disponible.

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

- b) Aplique un parche en Subscription para pasar al canal de actualización deseado (donde vX.Y es el canal de actualización deseado identificado en el paso anterior.

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

Antes de empezar

Como parte del proceso para actualizar los gestores de colas de IBM MQ , es posible que se le haya enviado a este tema desde la documentación de IBM Cloud Pak for Integration .

Procedimiento

Para actualizar el gestor de colas de IBM MQ utilizando Red Hat OpenShift, realice una de las tareas siguientes:

- [“Actualización de un gestor de colas de IBM MQ utilizando la consola de Red Hat OpenShift” en la página 52](#)
- [“Actualización de un gestor de colas de IBM MQ utilizando la CLI de Red Hat OpenShift” en la página 53](#)
- [“Actualización de un gestor de colas de IBM MQ en Red Hat OpenShift utilizando la interfaz de usuario de Platform” en la página 53](#)

Qué hacer a continuación

Para completar una actualización de IBM Cloud Pak for Integration , es posible que tenga que volver a la documentación de IBM Cloud Pak for Integration .

Un gestor de colas de IBM MQ , desplegado utilizando IBM MQ Operator, se puede actualizar en Red Hat OpenShift utilizando el concentrador del operador.

Antes de empezar

Nota: La versión más reciente del gestor de colas de IBM MQ es 9.4.0.0-r1y es a la vez una versión de SC2 y CD . Para obtener las últimas notas del release del gestor de colas de IBM MQ , consulte [Historial de releases para imágenes de contenedor de gestor de colas para su uso con IBM MQ Operator](#).

- Inicie sesión en la consola web del clúster de Red Hat OpenShift .
- Asegúrese de que el IBM MQ Operator está utilizando el canal de actualización deseado. Consulte [“Actualización de IBM MQ Operator utilizando Red Hat OpenShift” en la página 50](#).

Para poder actualizar el gestor de colas en un entorno aislado, debe duplicar las imágenes de IBM Cloud Pak for Integration más recientes a través de los pasos específicos de espacio aislado en [Actualización de un CD IBM MQ Operator al canal 3.2.x SC2/CD](#).

Procedimiento

1. En el panel de navegación, pulse **Operadores > Operadores instalados**.
Se visualizan todos los operadores instalados en el proyecto especificado.
2. Seleccione el operador **IBM MQ**.
Se visualiza la ventana **Operador de IBM MQ** .
3. Vaya a la pestaña **Gestor de colas** .
Se visualiza la ventana **Detalles del gestor de colas** .
4. Seleccione el gestor de colas que desea actualizar.
5. Vaya a la pestaña **YAML**.
6. Actualice los campos siguientes, cuando sea necesario, para que coincidan con la actualización de la versión deseada del gestor de colas de IBM MQ .

- spec.version
- spec.license.licence

Consulte [“Historial de releases para imágenes de contenedor de gestor de colas para su uso con IBM MQ Operator”](#) en la [página 7](#) para obtener una correlación de versiones de IBM MQ Operator e imágenes de contenedor de gestor de colas de IBM MQ .

7. Guarde el archivo YAML del gestor de colas actualizado.

Actualización de un gestor de colas de IBM MQ utilizando la CLI de Red Hat OpenShift

Un gestor de colas de IBM MQ , desplegado utilizando la IBM MQ Operator, se puede actualizar en Red Hat OpenShift utilizando la línea de mandatos.

Antes de empezar

Nota: La versión más reciente del gestor de colas de IBM MQ es 9.4.0.0-r1y es a la vez una versión de SC2 y CD . Para obtener las últimas notas del release del gestor de colas de IBM MQ , consulte [Historial de releases para imágenes de contenedor de gestor de colas para su uso con IBM MQ Operator](#).

Debe ser un administrador del clúster para completar estos pasos.

- Inicie sesión en la interfaz de línea de mandatos (CLI) de Red Hat OpenShift utilizando `oc login`.
- Asegúrese de que el IBM MQ Operator está utilizando el canal de actualización deseado. Consulte [“Actualización de IBM MQ Operator y gestores de colas”](#) en la [página 44](#).

Para poder actualizar el gestor de colas en un entorno aislado, debe duplicar las imágenes de IBM Cloud Pak for Integration más recientes a través de los pasos específicos de espacio aislado en [Actualización de un CD IBM MQ Operator al canal 3.2.x SC2/CD](#).

Procedimiento

Edite el recurso **QueueManager** para actualizar los campos siguientes, cuando sea necesario, para que coincidan con la actualización de la versión deseada del gestor de colas de IBM MQ .

- spec.version
- spec.license.licence

Consulte [“Soporte de versiones del IBM MQ Operator”](#) en la [página 14](#) para obtener una correlación de canales con versiones de IBM MQ Operator y versiones de gestor de colas de IBM MQ .

Utilice el mandato siguiente:

```
oc edit queuemanager my_qmgr
```

donde `my_qmgr` es el nombre del recurso QueueManager que desea actualizar.

Actualización de un gestor de colas de IBM MQ en Red Hat OpenShift utilizando la interfaz de usuario de Platform

Un gestor de colas de IBM MQ , desplegado utilizando la IBM MQ Operator, se puede actualizar en Red Hat OpenShift utilizando la IBM Cloud Pak for Integration Platform UI.

Antes de empezar

Nota: La versión más reciente del gestor de colas de IBM MQ es 9.4.0.0-r1y es a la vez una versión de SC2 y CD . Para obtener las últimas notas del release del gestor de colas de IBM MQ , consulte [Historial de releases para imágenes de contenedor de gestor de colas para su uso con IBM MQ Operator](#).

- Inicie sesión en IBM Cloud Pak for Integration Platform UI en el espacio de nombres que contiene el gestor de colas que desea actualizar.

- Asegúrese de que el IBM MQ Operator está utilizando el canal de actualización deseado. Consulte [“Actualización de IBM MQ Operator y gestores de colas”](#) en la página 44.

Para poder actualizar el gestor de colas en un entorno aislado, debe duplicar las imágenes de IBM Cloud Pak for Integration más recientes a través de los pasos específicos de espacio aislado en [Actualización de un CD IBM MQ Operator al canal 3.2.x SC2/CD](#).

Procedimiento

1. En la página de inicio de IBM Cloud Pak for Integration Platform UI , pulse el separador **Tiempos de ejecución** .
2. Los gestores de colas con actualizaciones disponibles tienen un **i** azul junto a la **Versión**. Pulse **i** para mostrar **Nueva versión disponible**.
3. Pulse los tres puntos en el extremo derecho del gestor de colas que desea actualizar y, a continuación, pulse **Cambiar versión**.
4. En **Seleccionar un nuevo canal o versión**, seleccione la versión de actualización necesaria.
5. Pulse **Cambiar versión**.

Resultados

El gestor de colas se ha actualizado.

Desinstalación de IBM MQ Operator

Puede utilizar la consola o la CLI de Red Hat OpenShift para desinstalar el IBM MQ Operator de Red Hat OpenShift.

Procedimiento

- Opción 1: Desinstale IBM MQ Operator con la consola de OpenShift .
 - Nota:** Si el IBM MQ Operator se instala en todos los proyectos/espacios de nombres del clúster, repita los pasos 2-6 del procedimiento siguiente para cada proyecto en el que desee suprimir gestores de colas.
 - a) Inicie sesión en la consola web de Red Hat OpenShift Container Platform con las credenciales de administración del clúster de Red Hat OpenShift Container Platform .
 - b) Cambie **Proyecto** por el espacio de nombres del que desea desinstalar el IBM MQ Operator. Seleccione el espacio de nombres en la lista desplegable **Proyecto** .
 - c) En el panel de navegación, pulse **Operadores > Operadores instalados**.
 - d) Pulse el operador **IBM MQ** .
 - e) Pulse el separador **Gestores de colas** para ver los gestores de colas gestionados por este IBM MQ Operator.
 - f) Suprima uno o más gestores de colas.

Tenga en cuenta que, aunque estos gestores de colas continúan ejecutándose, es posible que no funcionen como se esperaba sin un IBM MQ Operator.
 - g) Opcional: Si procede, repita los pasos del 2 al 6 para cada proyecto en el que desee suprimir gestores de colas.
 - h) Vuelva a **Operadores > Operadores instalados**.
 - i) Junto al operador **IBM MQ** , pulse el menú de tres puntos y seleccione **Desinstalar operador**.
- Opción 2: Desinstalar IBM MQ Operator con la CLI de OpenShift
 - a) Inicie sesión en el clúster de Red Hat OpenShift utilizando `oc login`.
 - b) Si el IBM MQ Operator está instalado en un único espacio de nombres, realice los subpasos siguientes:

- a. Asegúrese de que está en el proyecto que contiene el IBM MQ Operator que se va a desinstalar:

```
oc project project_name
```

- b. Vea los gestores de colas instalados en el proyecto:

```
oc get qmgr
```

- c. Suprima uno o varios gestores de colas:

```
oc delete qmgr qmgr_name
```

Tenga en cuenta que, aunque estos gestores de colas continúan ejecutándose, es posible que no funcionen como se esperaba sin un IBM MQ Operator.

- d. Vea las instancias de **ClusterServiceVersion** :

```
oc get csv
```

- e. Suprima el IBM MQ **ClusterServiceVersion**:

```
oc delete csv ibm_mq_csv_name
```

- f. Ver las suscripciones:

```
oc get subscription
```

- g. Suprimir todas las suscripciones:

```
oc delete subscription ibm_mq_subscription_name
```

- h. Si nada más está utilizando servicios comunes, es posible que desee desinstalar el operador de servicios comunes y suprimir el grupo de operadores:

i) Desinstale el operador de servicios comunes, siguiendo las instrucciones de [Desinstalación de servicios básicos](#) en la documentación del producto IBM Cloud Pak foundational services .

- ii) Vea el grupo de operadores:

```
oc get operatorgroup
```

- iii) Suprima el grupo de operadores:

```
oc delete OperatorGroup operator_group_name
```

- c) Si el IBM MQ Operator está instalado y disponible para todos los espacios de nombres del clúster, realice los subpasos siguientes:

- a. Ver todos los gestores de colas instalados:

```
oc get qmgr -A
```

- b. Suprima uno o varios gestores de colas:

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

Tenga en cuenta que, aunque estos gestores de colas continúan ejecutándose, es posible que no funcionen como se esperaba sin un IBM MQ Operator.

- c. Vea las instancias de **ClusterServiceVersion** :

```
oc get csv -A
```

- d. Suprima el IBM MQ **ClusterServiceVersion** del clúster:

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

- e. Ver las suscripciones:

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

f. Suprime las suscripciones:

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

g. Opcional: Si nada más está utilizando servicios comunes, es posible que desee desinstalar el operador de servicios comunes. Para ello, siga las instrucciones de [Desinstalación de servicios básicos](#) en la documentación del producto IBM Cloud Pak foundational services .

Preparación para IBM MQ creando su propia imagen de contenedor

Desarrolle un contenedor autoconstruido. 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 IBM MQ Operator. Consulte [“Cómo utilizar IBM MQ en contenedores”](#) en la página 8

Acerca de esta tarea

Procedimiento

- [“Consideraciones generales al crear su propia imagen de gestor de colas”](#) en la página 56
- [“Creación de una imagen de contenedor de gestor de colas de IBM MQ de ejemplo”](#) en la página 57
- [“Ejecución de aplicaciones de enlaces locales en contenedores separados”](#) en la página 59
- [Revise el diagrama Helm de ejemplo de IBM MQ.](#)

Consideraciones generales al crear su propia imagen de gestor de colas

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.

seguridad del contenedor

Para minimizar los requisitos de seguridad de tiempo de ejecución, las imágenes de contenedor de ejemplos se instalan utilizando la instalación descomprimible de IBM MQ . Esto garantiza que no se hayan establecido `setuid bits` y que el contenedor no tenga que utilizar el escalamiento de privilegios. Algunos sistemas de contenedor definen qué ID de usuario puede utilizar y la instalación descomprimible no presupone ningún usuario del sistema operativo disponible.

Creación de una imagen de contenedor de gestor de colas de IBM MQ de ejemplo

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 57
 - [“Creación de una imagen de gestor de colas de IBM MQ configurada de muestra”](#) en la página 58

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.
- Opcional: Si tiene previsto configurar el acceso seguro utilizando la restricción de contexto de seguridad (SCC) de Red Hat OpenShift Container Platform "restringido", utilice una de las imágenes no de instalación de IBM MQ .

Los enlaces para descargar estas imágenes están disponibles en la sección [Contenedores de descargas de IBM MQ](#).

Resultados

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

Ahora está preparado para [crear un ejemplo de IBM MQ imagen de gestor de colas configurada](#).

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

Esta tarea presupone que, al crear la imagen del gestor de colas base de IBM MQ de ejemplo, ha utilizado el paquete "No-Install" IBM MQ . De lo contrario, no puede configurar el acceso seguro utilizando la restricción de contexto de seguridad (SCC) de Red Hat OpenShift Container Platform "restringido" . La SCC "restringida" , que se utiliza de forma predeterminada, utiliza ID de usuario aleatorios e impide el escalamiento de privilegios cambiando a un usuario diferente. El instalador basado en RPM tradicional de IBM MQ se basa en un usuario y grupo de mqm , y también utiliza bits de `setuid` en programas ejecutables. En la versión actual de IBM MQ, cuando se utiliza el paquete "No-Install" IBM MQ , ya no hay ningún usuario mqm ni un grupo mqm .

Procedimiento

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

```
DEFINE QLOCAL(EXAMPLE.QUEUE.1) REPLACE
```

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
COPY config.mqsc /etc/mqm/
```

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

```
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.

Nota: En Red Hat Enterprise Linux (RHEL) se usa el comando **docker** (RHEL V7) o **podman** (RHEL V7 o RHEL V8). En Linux, tendrá que ejecutar mandatos **docker** con **sudo** al principio del mandato, para obtener privilegios adicionales.

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:

```
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 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 puede ver los contenedores en ejecución utilizando el mandato **docker ps**. Puede ver los procesos de IBM MQ que se ejecutan en el contenedor utilizando el mandato **docker top**.



Atención:

Puede ver los registros de un contenedor utilizando el mandato **docker logs \$ {CONTAINER_ID}**.

Qué hacer a continuación

- Si el contenedor no se muestra cuando se utiliza el mandato **docker ps**, es posible que el contenedor haya fallado. Puede ver los contenedores anómalos utilizando el mandato **docker ps -a**.
- Cuando se utiliza el mandato **docker ps -a**, se visualiza el ID de contenedor. Este ID también se imprimió cuando se emitió el mandato **docker run**.
- Puede ver los registros de un contenedor utilizando el mandato **docker logs \$ {CONTAINER_ID}**.

Ejecución de aplicaciones de enlaces locales en contenedores separados

Con la compartición de espacios de nombres de proceso entre contenedores, puede ejecutar aplicaciones que requieran una conexión de enlace local con IBM MQ en contenedores separados del gestor de colas de IBM MQ.

Acerca de esta tarea

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`.

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 **dspm**, **amqsput**, **amqsget** y **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 60 para obtener más información.

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 59 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 crtqm QM1
AMQ8101: IBM MQ error (893) has occurred.

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

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

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

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

MQ Adv.

Creación del grupo HA nativo si crea sus propios contenedores

Debe crear, configurar e iniciar tres gestores de colas para crear el grupo HA nativo.

Acerca de esta tarea

El método recomendado para crear una solución de HA nativa es utilizar el operador IBM MQ (consulte [HA nativa](#)). De forma alternativa, si crea sus propios contenedores, puede seguir estas instrucciones.

Para crear un grupo HA nativo, cree tres gestores de colas en tres nodos con su tipo de registro establecido en `log replication`. A continuación, edite el archivo `qm.ini` para cada gestor de colas para añadir los detalles de conexión para cada uno de los tres nodos para que puedan replicar los datos de registro entre sí.

A continuación, debe iniciar los tres gestores de colas para que puedan comprobar que las tres instancias pueden comunicarse entre sí y determinar cuál de ellas será la instancia activa y cuáles serán las réplicas.

Nota: Sólo puede crear un grupo HA nativo en sus propios contenedores de esta forma si ejecuta Kubernetes o Red Hat OpenShift.

Procedimiento

1. En cada uno de los tres nodos, cree un gestor de colas, especificando un tipo de registro de réplica de registro y proporcionando un nombre exclusivo para cada instancia de registro. Cada gestor de colas tiene el mismo nombre:

```
crtmqm -lr instance_name qmname
```

Por ejemplo:

```
node 1> crtmqm -lr qm1_inst1 qm1
node 2> crtmqm -lr qm1_inst2 qm1
node 3> crtmqm -lr qm1_inst3 qm1
```

2. Al crear correctamente cada gestor de colas, se añade una stanza adicional denominada `NativeHALocalInstance` al archivo de configuración del gestor de colas, `qm.ini`. Se añade un atributo `Name` a la stanza que especifica el nombre de instancia proporcionado.

Puede añadir opcionalmente los atributos siguientes a la stanza `NativeHALocalInstance` en el archivo `qm.ini`:

KeyRepository

La ubicación del repositorio de claves que contiene el certificado digital que se debe utilizar para la protección del tráfico de réplica de registro. La ubicación se proporciona en formato raíz, es decir, incluye la vía de acceso completa y el nombre de archivo sin una extensión. Si se omite el atributo de stanza `KeyRepository`, los datos de réplica de registro se intercambian entre instancias en texto sin formato.

CertificateLabel

Etiqueta de certificado que identifica el certificado digital que se debe utilizar para la protección del tráfico de réplica de registro. Si se proporciona `KeyRepository` pero se omite `CertificateLabel`, se utiliza un valor predeterminado de `ibmwebsphermqueue_manager`.

CipherSpec

La `CipherSpec` de MQ que se utilizará para proteger el tráfico de réplica de registro. Si se proporciona este atributo de stanza, también se debe proporcionar `KeyRepository`. Si se proporciona `KeyRepository` pero se omite `CipherSpec`, se utiliza un valor predeterminado de `ANY`.

LocalAddress

La dirección de la interfaz de red local que acepta el tráfico de réplica de registro. Si se proporciona este atributo de stanza, identifica la interfaz de red local y/o el puerto utilizando el formato "[addr] [(port)]". La dirección de red se puede especificar como un nombre de host, IPv4 decimal con puntos o formato hexadecimal IPv6. Si se omite este atributo, el gestor de colas intenta enlazar con todas las interfaces de red, utiliza el puerto especificado en `ReplicationAddress` en la stanza `NativeHAInstances` que coincide con el nombre de instancia local.

HeartbeatInterval

El intervalo de latidos define la frecuencia en milisegundos a la que una instancia activa de un gestor de colas de HA nativo envía una pulsación de red. El rango válido del valor del intervalo de pulsaciones es de 500 (0.5 segundos) a 60000 (1 minuto), un valor fuera de este rango hace que el gestor de colas no se pueda iniciar. Si se omite este atributo, se utiliza un valor predeterminado de 5000 (5 segundos). Cada instancia debe utilizar el mismo intervalo de pulsaciones.

HeartbeatTimeout

El tiempo de espera de latido define cuánto tiempo espera una instancia de réplica de un gestor de colas de HA nativo antes de decidir que la instancia activa no responde. El rango válido del valor de tiempo de espera de intervalo de pulsaciones es de 500 (0.5 segundos) a 120000 (2 minutos). El valor del tiempo de espera de pulsaciones debe ser mayor o igual que el intervalo de pulsaciones.

Un valor no válido hace que el gestor de colas no se inicie. Si se omite este atributo, una réplica espera 2 x HeartbeatInterval antes de iniciar el proceso para seleccionar una nueva instancia activa. Cada instancia debe utilizar el mismo tiempo de espera de latido.

RetryInterval

El intervalo de reintento define la frecuencia en milisegundos a la que un gestor de colas HA nativo debe reintentar un enlace de réplica anómalo. El rango válido del intervalo de reintento es de 500 (0.5 segundos) a 120000 (2 minutos). Si se omite este atributo, una réplica espera 2 x HeartbeatInterval antes de reintentar un enlace de réplica fallido.

3. Edite el archivo `qm.ini` para cada gestor de colas y añada detalles de conexión. Añada tres stanzas `NativeHAInstance`, una para cada instancia de gestor de colas en el grupo HA nativo (incluida la instancia local). Añada los atributos siguientes:

Nombre

Especifique el nombre de instancia que ha utilizado al crear la instancia del gestor de colas.

ReplicationAddress

Especifique el nombre de host, IPv4 decimal con puntos o IPv6 dirección de formato hexadecimal de la instancia. Puede especificar la dirección como un nombre de host, IPv4 decimal con puntos o dirección en formato hexadecimal IPv6. La dirección de réplica debe poder resolverse y direccionarse desde cada instancia del grupo. El número de puerto que se debe utilizar para la réplica de registro debe especificarse entre corchetes, por ejemplo:

```
ReplicationAddress=host1.example.com(4444)
```

Nota: Las stanzas `NativeHAInstance` son idénticas en cada instancia y se pueden proporcionar utilizando la configuración automática (`crtmqm -ii`).

4. Inicie cada una de las tres instancias:

```
strmqm QMgrName
```

Cuando se inician las instancias, se comunican para comprobar que las tres instancias se están ejecutando y, a continuación, decidir cuál de las tres es la instancia activa, mientras que las otras dos instancias continúan ejecutándose como réplicas.

Ejemplo

El ejemplo siguiente muestra la sección de un archivo `qm.ini` que especifica los detalles de HA nativa necesarios para una de las tres instancias:

```
NativeHALocalInstance:
  LocalName=node-1

NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
```

Despliegue y configuración de gestores de colas en contenedores

Puede realizar una serie de tareas para desplegar y configurar gestores de colas de IBM MQ .

Acerca de esta tarea

Para empezar a desplegar y configurar gestores de colas, consulte los temas siguientes.

Procedimiento

- [“Despliegue y configuración de gestores de colas utilizando IBM MQ Operator”](#) en la página 64
- [“Despliegue y configuración de gestores de colas utilizando Helm”](#) en la página 105

Despliegue y configuración de gestores de colas utilizando IBM MQ Operator

Ejemplos de configuración; configuración de HA; conexión desde fuera de un clúster OpenShift ; integración con el panel de control CP4i ; integración con el rastreo de Instana; creación de una imagen con archivos MQSC e INI personalizados; adición de anotaciones y etiquetas personalizadas.

Acerca de esta tarea

Procedimiento

- [“Ejemplos para configurar un gestor de colas”](#) en la página 67.
- [“Configuración de la alta disponibilidad para gestores de colas utilizando la IBM MQ Operator”](#) en la página 76.
- [“Configuración de una ruta para conectarse a un gestor de colas desde fuera de un clúster de Red Hat OpenShift”](#) en la página 85.
- [“Integración de IBM MQ con el rastreo de IBM Instana”](#) en la página 87.
- [“Creación de una imagen con archivos MQSC e INI personalizados, utilizando la CLI de Red Hat OpenShift”](#) en la página 94.
- [“Adición de anotaciones y etiquetas personalizadas a los recursos del gestor de colas”](#) en la página 96.
- [“Inhabilitación de comprobaciones de webhook en tiempo de ejecución”](#) en la página 96.
- [“Inhabilitación de las actualizaciones de valores predeterminados para la especificación del gestor de colas”](#) en la página 97.

Despliegue de un gestor de colas simple utilizando la IBM MQ Operator

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

Acerca de esta tarea

Esta tarea ofrece 3 opciones para desplegar un gestor de colas en OpenShift:

1. [Despliegue un gestor de colas con la consola de OpenShift.](#)
2. [Despliegue un gestor de colas con la CLI de OpenShift.](#)

3. Despliegue un gestor de colas con IBM Cloud Pak for Integration Platform UI.

Procedimiento

- **Opción 1: Desplegar un gestor de colas con la consola de OpenShift .**

- a) Despliegue un gestor de colas.
 - a. Inicie sesión en la consola de OpenShift con las credenciales de administrador del clúster de Red Hat OpenShift Container Platform .
 - b. Cambie **Proyecto** por el espacio de nombres donde ha instalado el IBM MQ Operator. Seleccione el espacio de nombres en la lista desplegable **Proyecto** .
 - c. En el panel de navegación, pulse **Operadores > Operadores instalados**.
 - d. En la lista del panel Operadores instalados, busque y pulse **IBM MQ**.
 - e. Pulse el separador **Gestor de colas** .
 - f. Pulse el botón **Crear QueueManager** . Se visualiza el panel de creación de instancia y ofrece dos métodos para configurar el recurso: la **vista Formulario** y la **vista YAML**. La **vista Formulario** está seleccionada de forma predeterminada.
- b) Configure el gestor de colas.

Paso 2 Opción 1: Configurar en la **vista Formulario**.

La **vista Formulario** abre un formulario que puede utilizar para ver o modificar la configuración de recursos.

- a. Junto a **Licencia**, pulse la flecha para expandir la sección de aceptación de licencia.
- b. Establezca **Aceptar licencia** en **true** si acepta el acuerdo de licencia.
- c. Pulse la flecha para abrir la lista desplegable y seleccione una licencia. 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”](#) en la [página 140](#). Debe aceptar la licencia para desplegar un gestor de colas.
- d. 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 .

Paso 2 Opción 2: Configurar en la **vista YAML**.

La **vista YAML** abre un editor que contiene un archivo YAML de ejemplo para un QueueManager. Actualice los valores del archivo siguiendo los pasos siguientes.

- a. Cambie `metadata.namespace` por el nombre del proyecto (espacio de nombres).
 - b. Cambie el valor de `spec.license.license` por la serie de licencia que coincida con sus requisitos. Consulte [“Referencia de licencia para mq.ibm.com/v1beta1”](#) en la [página 140](#) para obtener los detalles de la licencia.
 - c. Cambie `spec.license.accept` por `true` si acepta el acuerdo de licencia.
 - d. 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 .
- c) Verifique la creación del gestor de colas.

Puede verificar que ha creado un gestor de colas realizando los pasos siguientes:

 - a. Asegúrese de que está en el espacio de nombres en el que ha creado el IBM MQ Operator .
 - b. En la pantalla **Inicio** , pulse **Operadores > Operadores instalados** y, a continuación, seleccione el IBM MQ Operator instalado para el que ha creado el gestor de colas.
 - c. Pulse el separador **Gestor de colas** . La creación se completa cuando el estado de QueueManager es Running.

- **Opción 2: Desplegar un gestor de colas con la CLI de OpenShift .**

- 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: 9.4.0.0-r1
  license:
    accept: false
    license: L-BMSF-5YDSLRL
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
```

Importante: Si acepta el acuerdo de licencia, cambie `accept: false` por `accept: true`. Consulte [“Referencia de licencia para mq.ibm.com/v1beta1”](#) en la página 140 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 en IBM Cloud Pak for Integration. Para que el inicio de sesión único funcione, primero debe instalar otros componentes de IBM Cloud Pak for Integration. Consulte [“Instalación de IBM MQ Operator para su uso con CP4I”](#) en la página 40.

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: 9.4.0.0-r1
  license:
    accept: false
    license: L-EHXT-MQCRN9
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
```

Importante: Si acepta el acuerdo de licencia de MQ, cambie `accept: false` por `accept: true`. Consulte [“Referencia de licencia para mq.ibm.com/v1beta1”](#) en la página 140 para obtener detalles sobre la licencia.

b) Cree el objeto QueueManager .

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

c) Verifique la creación del gestor de colas.

Verifique que ha creado un gestor de colas realizando los pasos siguientes:

a. Valide el despliegue:

```
oc describe queuemanager Queue_Manager_Resource_Name
```

b. Compruebe el estado:

```
oc describe queuemanager quickstart
```

- **Opción 3: Despliegue un gestor de colas con IBM Cloud Pak for Integration Platform UI.**

Siga las instrucciones de [Despliegue de una instancia utilizando la IU de Platform](#).

Tareas relacionadas

[“Configuración de una ruta para conectarse a un gestor de colas desde fuera de un clúster de Red Hat OpenShift”](#) en la página 85

Necesita una ruta de Red Hat OpenShift para conectar una aplicación a un gestor de colas de IBM MQ desde fuera de un clúster de Red Hat OpenShift . Debe habilitar TLS en el gestor de colas y la aplicación cliente de IBM MQ , porque SNI sólo está disponible en el protocolo TLS cuando se utiliza un protocolo TLS 1.2 o superior. Red Hat OpenShift Container Platform Router utiliza SNI para direccionar solicitudes al gestor de colas IBM MQ .

[“Conexión con el IBM MQ Console desplegado en un clúster de Red Hat OpenShift”](#) en la página 130
Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

[“Ejemplos para configurar un gestor de colas”](#) en la página 67

Un gestor de colas se puede configurar ajustando el contenido del recurso personalizado QueueManager .

Ejemplos para configurar un gestor de colas

Un gestor de colas se puede configurar ajustando el contenido del recurso personalizado QueueManager .

Acerca de esta tarea

Utilice los ejemplos siguientes como ayuda para configurar un gestor de colas utilizando el archivo YAML QueueManager .

Procedimiento

- [“Ejemplo: Suministro de archivos MQSC e INI”](#) en la página 67
- [“Ejemplo: Configuración de un gestor de colas con autenticación TLS mutua”](#) en la página 70

Ejemplo: Suministro de archivos MQSC e INI

Este ejemplo crea un Kubernetes ConfigMap que contiene dos archivos MQSC y un archivo INI. A continuación, se despliega un gestor de colas que procesa estos archivos MQSC e INI.

Acerca de esta tarea

Los archivos [MQSC](#) e [INI](#) se pueden proporcionar cuando se despliega un gestor de colas. Los datos MQSC e INI deben estar definidos en uno o más Kubernetes [ConfigMaps](#) y [Secretos](#). Se deben crear en el espacio de nombres (proyecto) donde desplegará el gestor de colas.

Nota: Se debe utilizar un secreto Kubernetes cuando el archivo MQSC o INI contiene datos confidenciales.

Ejemplo

El ejemplo siguiente crea un Kubernetes ConfigMap que contiene dos archivos MQSC y un archivo INI. A continuación, se despliega un gestor de colas que procesa estos archivos MQSC e INI.

Ejemplo de ConfigMap : aplique el siguiente YAML en el clúster:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mqsc-ini-example
data:
  example1.mqsc: |
    DEFINE QLOCAL('DEV.QUEUE.1') REPLACE
    DEFINE QLOCAL('DEV.QUEUE.2') REPLACE
  example2.mqsc: |
    DEFINE QLOCAL('DEV.DEAD.LETTER.QUEUE') REPLACE
  example.ini: |
    Channels:
      MQIBindType=FASTPATH
```

Ejemplo de QueueManager : despliegue el gestor de colas con la configuración siguiente, utilizando la línea de mandatos o utilizando la consola web de Red Hat OpenShift Container Platform :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-qm
spec:
  version: 9.4.0.0-r1
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  web:
    enabled: true
  queueManager:
    name: "MQSCINI"
  mqsc:
    - configMap:
        name: mqsc-ini-example
        items:
          - example1.mqsc
          - example2.mqsc
  ini:
    - configMap:
        name: mqsc-ini-example
        items:
          - example.ini
  storage:
    queueManager:
      type: ephemeral
```

Importante: Si acepta el acuerdo de licencia de IBM MQ Advanced , cambie `accept: false` por `accept: true`. Consulte [Referencia de licencias para mq.ibm.com/v1beta1](#) para obtener detalles sobre la licencia.

Más información:

- Un gestor de colas se puede configurar para utilizar un único Kubernetes ConfigMap o Secret (como se muestra en este ejemplo) o varios ConfigMaps y Secrets.
- Puede elegir utilizar todos los datos MQSC e INI de un Kubernetes ConfigMap o Secreto (tal como se muestra en este ejemplo) o configurar cada gestor de colas para utilizar sólo un subconjunto de los archivos disponibles.
- Los archivos MQSC e INI se procesan en orden alfabético en función de su clave. Por lo tanto, `example1.mqsc` siempre se procesará antes de `example2.mqsc`, independientemente del orden en el que aparecen en la configuración del gestor de colas.
- Si varios archivos MQSC o INI tienen la misma clave, en varios Kubernetes ConfigMaps o Secretos, este conjunto de archivos se procesa basándose en el orden en el que se definen los archivos en la configuración del gestor de colas.
- Cuando se ejecuta un pod de gestor de colas, los cambios en Kubernetes ConfigMap no se seleccionan porque IBM MQ Operator no es consciente del cambio. Si realiza cambios en el ConfigMap, por ejemplo, cambios en los mandatos MQSC o en los archivos INI, debe reiniciar manualmente los gestores de colas para recoger estos cambios. Para gestores de colas de una sola instancia, suprima el pod para desencadenar el reinicio necesario. Para despliegues de alta disponibilidad nativos, reinicie primero los pods en espera suprimiéndolos. Cuando estén de nuevo en un estado de ejecución, suprima el pod activo para reiniciarlo. Este orden de reinicios garantiza un tiempo de inactividad mínimo para el gestor de colas.




Creación de una PKI autofirmada utilizando OpenSSL

IBM MQ le permite utilizar TLS mutuo para la autenticación, donde ambos extremos de una conexión proporcionan un certificado, y los detalles del certificado se utilizan para establecer una identidad con el gestor de colas. En este tema se muestra cómo crear un PKI (Public Key Infrastructure) de ejemplo utilizando la herramienta de línea de mandatos OpenSSL , creando dos certificados que se pueden utilizar en otros ejemplos.

Antes de empezar

Asegúrese de que la herramienta de línea de mandatos OpenSSL esté instalada.

Instale IBM MQ client y añada `samp/bin` y `bin` a su `PATH`. Necesita el mandato `runmqicred`, que se puede instalar como parte de IBM MQ client de la siguiente manera:

-   Para Windows y Linux: instale el cliente redistribuible de IBM MQ para el sistema operativo desde <https://ibm.biz/mq94redistclients>
-  Para Mac: Descargue y configure IBM MQ MacOS Toolkit: <https://developer.ibm.com/tutorials/mq-macos-dev/>

Acerca de esta tarea

Importante: Los ejemplos descritos aquí no son adecuados para un entorno de producción, y están pensados únicamente como ejemplos para ponerse en marcha rápidamente. La gestión de certificados es un asunto complejo para los usuarios avanzados. Para la producción, debe tener en cuenta cosas como la rotación, la revocación, la longitud de la clave, la recuperación tras desastre y mucho más.

Estos pasos se han probado utilizando OpenSSL 3.1.4.

Procedimiento

1. Crear una clave privada para utilizarla para la entidad emisora de certificados interna

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out ca.key
```

Se crea una clave privada para la entidad emisora de certificados interna en un archivo denominado `ca.key`. Este archivo debe mantenerse a salvo y en secreto-se utilizará para firmar certificados para la entidad emisora de certificados interna.

2. Emitir un certificado autofirmado para la entidad emisora de certificados interna

```
openssl req -x509 -new -nodes -key ca.key -sha512 -days 30 -subj "/CN=example-selfsigned-ca" -out ca.crt
```

`-days` especifica el número de días que el certificado de CA raíz será válido.

Se crea un certificado en un archivo denominado `ca.crt`. Este certificado contiene la información pública sobre la entidad emisora de certificados interna y se puede compartir libremente.

3. Crear una clave privada y un certificado para un gestor de colas

- a) Crear una clave privada y una solicitud de firma de certificado para un gestor de colas

```
openssl req -new -nodes -out example-qm.csr -newkey rsa:4096 -keyout example-qm.key -subj '/CN=example-qm'
```

Se crea una clave privada en un archivo denominado `example-qm.key`, y se crea una solicitud de firma de certificado en un archivo denominado `example-qm.csr`

- b) Firme la clave del gestor de colas con la entidad emisora de certificados interna

```
openssl x509 -req -in example-qm.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out example-qm.crt -days 7 -sha512
```

`-days` especifica el número de días que el certificado será válido.

Se crea un certificado firmado en un archivo denominado `example-qm.crt`

- c) Crear un secreto de Kubernetes con la clave y el certificado del gestor de colas

```
oc create secret generic example-qm-tls --type="kubernetes.io/tls" --from-file=tls.key=example-qm.key --from-file=tls.crt=example-qm.crt --from-file=ca.crt
```

Se crea un secreto de Kubernetes denominado `example-qm-tls`. Este secreto contiene la clave privada para el gestor de colas, el certificado público y el certificado de CA.

4. Crear una clave privada y un certificado para una aplicación

a) Crear una clave privada y una solicitud de firma de certificado para una aplicación

```
openssl req -new -nodes -out example-app1.csr -newkey rsa:4096 -keyout example-app1.key  
-subj '/CN=example-app1'
```

Se crea una clave privada en un archivo denominado *example-app1.key*, y se crea una solicitud de firma de certificado en un archivo denominado *example-app1.csr*

b) Firme la clave del gestor de colas con la entidad emisora de certificados interna

```
openssl x509 -req -in example-app1.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out  
example-app1.crt -days 7 -sha512
```

-days especifica el número de días que el certificado será válido.

Se crea un certificado firmado en un archivo denominado *example-app1.crt*

c) Crear un almacén de claves PKCS#12 con la clave y el certificado de la aplicación

IBM MQ utiliza una base de datos de claves, no archivos de claves individuales. El gestor de colas contenerizado creará la base de datos de claves para el gestor de colas a partir de un secreto, pero para las aplicaciones de clientes, debe crear manualmente la base de datos de claves.

```
openssl pkcs12 -export -in "example-app1.crt" -name "example-app1" -certfile "ca.crt"  
-inkey "example-app1.key" -out "example-app1.p12" -passout pass:PASSWORD
```

Donde *CONTRASEÑA* es una contraseña de su elección.

Se crea un almacén de claves en un archivo denominado *example-app1.p12*. La clave y el certificado de la aplicación se almacenan dentro, con una "etiqueta" o "nombre descriptivo" de "example-app1", así como el certificado de CA.

d) Si utiliza un Apple Mac arm64 , tendrá que configurar un archivo adicional que combine los certificados de aplicación y CA.

Por ejemplo:

```
cat example-app1.crt ca.crt > example-app1-chain.crt
```

Tareas relacionadas

[“Ejemplo: Configuración de un gestor de colas con autenticación TLS mutua”](#) en la página 70

Este ejemplo despliega un gestor de colas en OpenShift Container Platform utilizando IBM MQ Operator. Se utiliza TLS mutuo para la autenticación, para correlacionar desde un certificado TLS con una identidad en el gestor de colas.

[“Ejemplo: Configuración de HA nativa utilizando IBM MQ Operator”](#) en la página 76

Este ejemplo despliega un gestor de colas utilizando la característica de alta disponibilidad nativa en OpenShift Container Platform utilizando IBM MQ Operator. Se utiliza TLS mutuo para la autenticación, para correlacionar desde un certificado TLS con una identidad en el gestor de colas.

[“Configuración de un gestor de colas de varias instancias utilizando IBM MQ Operator”](#) en la página 82

Este ejemplo despliega un gestor de colas de varias instancias utilizando OpenShift Container Platform utilizando IBM MQ Operator. Se utiliza TLS mutuo para la autenticación, para correlacionar desde un certificado TLS con una identidad en el gestor de colas.

Ejemplo: Configuración de un gestor de colas con autenticación TLS mutua

Este ejemplo despliega un gestor de colas en OpenShift Container Platform utilizando IBM MQ Operator. Se utiliza TLS mutuo para la autenticación, para correlacionar desde un certificado TLS con una identidad en el gestor de colas.

Antes de empezar

Para completar este ejemplo, primero debe haber completado los siguientes requisitos previos:

- Cree un proyecto/espacio de nombres OpenShift Container Platform (OCP) para este ejemplo.
- En la línea de mandatos, inicie sesión en el clúster de OCP y cambie al espacio de nombres anterior.
- Asegúrese de que el IBM MQ Operator esté instalado y disponible en el espacio de nombres anterior.

Acerca de esta tarea

Este ejemplo proporciona un YAML de recurso personalizado que define un gestor de colas que se desplegará en OpenShift Container Platform. También detalla los pasos adicionales necesarios para desplegar el gestor de colas con TLS habilitado.

Procedimiento

1. Cree un par de certificados tal como se describe en [“Creación de una PKI autofirmada utilizando OpenSSL”](#) en la página 68.

2. Crear una correlación de configuración que contenga mandatos MQSC y un archivo INI

Cree un Kubernetes ConfigMap que contenga los mandatos MQSC para crear una cola nueva y un canal SVRCONN, y para añadir un registro de autenticación de canal que permita el acceso al canal.

Asegúrese de que está en el espacio de nombres que ha creado anteriormente (consulte [Antes de empezar](#)) y, a continuación, especifique el YAML siguiente en la consola web de OCP o utilizando la línea de mandatos.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-tls-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
    ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
    MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
    AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
```

El MQSC define un canal denominado *MTLS.SVRCONN* y una cola denominada *EXAMPLE.QUEUE*. El canal está configurado para permitir el acceso sólo a los clientes que presentan un certificado con un "nombre común" de *example-app1*. Este es el nombre común utilizado en uno de los certificados creados en el paso [“1”](#) en la [página 71](#). Las conexiones en este canal con este nombre común se correlacionan con un ID de usuario de *app1*, que está autorizado para conectarse al gestor de colas y para acceder a la cola de ejemplo. El archivo INI habilita una política de seguridad, lo que significa que el ID de usuario *app1* no es necesario que exista en un registro de usuarios externo; sólo existe como nombre en esta configuración.

3. Desplegar el gestor de colas

Cree un nuevo gestor de colas utilizando el siguiente recurso personalizado YAML. Asegúrese de que está en el espacio de nombres que ha creado antes de empezar esta tarea y, a continuación, especifique el YAML siguiente en la consola web de OCP o utilizando la línea de mandatos. Compruebe que se ha especificado la licencia correcta y acepte la licencia cambiando `false` a `true`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
```

```

license: L-EHXT-MQCRN9
use: Production
queueManager:
  name: EXAMPLEQM
mqsc:
  - configMap:
      name: example-tls-configmap
      items:
        - example-tls.mqsc
ini:
  - configMap:
      name: example-tls-configmap
      items:
        - example-tls.ini
storage:
  queueManager:
    type: ephemeral
version: 9.4.0.0-r1
pki:
  keys:
    - name: default
      secret:
        secretName: example-qm-tls
        items:
          - tls.key
          - tls.crt
          - ca.crt

```

Tenga en cuenta que el secreto *example-qm-tls* se ha creado en el paso “1” en la [página 71](#) y que el ConfigMap *example-tls-configmap* se ha creado en el paso “2” en la [página 71](#)

4. Confirmar que el gestor de colas se está ejecutando

El gestor de colas se está desplegando ahora. Confirme que está en estado Running antes de continuar. Por ejemplo:

```
oc get qmgr exampleqm
```

5. Probar la conexión con el gestor de colas

Para confirmar que el gestor de colas está configurado para la comunicación TLS mutua, siga los pasos de “[Prueba de una conexión TLS mutua con un gestor de colas desde el portátil](#)” en la [página 72](#).

Resultados




Enhorabuena, ha desplegado correctamente un gestor de colas con TLS habilitado y que utiliza los detalles proporcionados en el certificado TLS para autenticarse con el gestor de colas y proporcionar una identidad.

Prueba de una conexión TLS mutua con un gestor de colas desde el portátil

Después de haber creado un gestor de colas utilizando IBM MQ Operator, puede probar que funciona conectándose a él y colocando y obteniendo un mensaje. Esta tarea le lleva a través de cómo conectarse utilizando los programas de ejemplo de IBM MQ , ejecutándolos en una máquina fuera del clúster de Kubernetes , como por ejemplo el portátil.

Antes de empezar

Para completar este ejemplo, primero debe haber completado los siguientes requisitos previos:

- Instale el IBM MQ client. Necesita los mandatos **amqsputc** y **amqsgetc** , que se pueden instalar como parte de IBM MQ client de la siguiente manera:
 -   Para Windows y Linux: instale el cliente redistribuible de IBM MQ para el sistema operativo desde <https://ibm.biz/mq94redistclients>
 -  Para Mac: Descargue y configure IBM MQ MacOS Toolkit: <https://developer.ibm.com/tutorials/mq-macos-dev/>

- Asegúrese de que tiene los archivos de claves y certificados necesarios descargados en un directorio de la máquina y de que conoce la contraseña del almacén de claves. Por ejemplo, estos archivos se crean en [“Creación de una PKI autofirmada utilizando OpenSSL”](#) en la página 68:
 - `example-app1.p12`
 - `example-app1-chain.crt` (sólo si utiliza un arm64 Apple Mac)
- Despliegue un gestor de colas configurado con TLS en el clúster de OCP, por ejemplo, siguiendo los pasos de [“Ejemplo: Configuración de un gestor de colas con autenticación TLS mutua”](#) en la página 70

Acerca de esta tarea

Este ejemplo utiliza los programas de ejemplo IBM MQ que se ejecutan en una máquina fuera del clúster de Kubernetes como, por ejemplo, el portátil, para conectarse a un QueueManager configurado con TLS y para transferir y obtener mensajes.

Procedimiento

1. Confirme que el gestor de colas se está ejecutando

El gestor de colas se está desplegando ahora. Confirme que está en estado `Running` antes de continuar. Por ejemplo:

```
oc get qmgr exampleqm
```

2. Buscar el nombre de host del gestor de colas

Utilice el mandato siguiente para buscar el nombre de host completo del gestor de colas desde fuera del clúster OCP, utilizando la ruta que se crea automáticamente: `exampleqm-ibm-mq-qm`:

```
oc get route exampleqm-ibm-mq-qm --template="{{.spec.host}}"
```

3. Crear una tabla de definiciones de canal de cliente (CCDT) de IBM MQ

Cree un archivo denominado `ccdt.json` con el contenido siguiente:

```
{
  "channel": [
    {
      "name": "MTLS.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "hostname from previous step",
            "port": 443
          }
        ],
        "queueManager": "EXAMPLEQM"
      },
      "transmissionSecurity": {
        "cipherSpecification": "ANY_TLS13",
        "certificateLabel": "example-app1"
      },
      "type": "clientConnection"
    }
  ]
}
```

La conexión utiliza el puerto 443, porque es el puerto en el que escucha el direccionador de Red Hat OpenShift Container Platform. El tráfico se reenviará al gestor de colas en el puerto 1414.

Si ha utilizado un nombre de canal diferente, también tendrá que ajustarlo. Los ejemplos de TLS mutuo utilizan un canal denominado `MTLS.SVRCONN`

Para obtener más detalles, consulte [Configuración de una CCDT en formato JSON](#)

4. Crear un archivo INI de cliente para configurar los detalles de conexión

Cree un archivo denominado `mqclient.ini` en el directorio actual. **amqsputc** y **amqsgetc** leerán este archivo.

```
Channels:  
  ChannelDefinitionDirectory=.  
  ChannelDefinitionFile=ccdt.json  
SSL:  
  OutboundSNI=HOSTNAME  
  SSLKeyRepository=example-app1.p12  
  SSLKeyRepositoryPassword=password you used when creating the p12 file
```

Asegúrese de actualizar la contraseña de `SSLKeyRepository` a la contraseña que ha elegido al crear el archivo PKCS#12. Existen otras formas de establecer la contraseña del almacén de claves, incluido el uso de una contraseña cifrada. Para obtener más información, consulte [Suministro de la contraseña del repositorio de claves para un IBM MQ MQI client en AIX, Linux, and Windows](#)

Tenga en cuenta que Red Hat OpenShift Container Platform Router utiliza SNI para direccionar solicitudes al gestor de colas IBM MQ. El atributo `OutboundSNI=HOSTNAME` garantiza que el cliente de IBM MQ incluya la información necesaria para que el direccionador funcione con la ruta predeterminada configurada por IBM MQ Operator. Para obtener más información, consulte [“Configuración de una ruta para conectarse a un gestor de colas desde fuera de un clúster de Red Hat OpenShift”](#) en la página 85.

5. Si está utilizando un Apple Mac arm64, debe configurar una variable de entorno adicional.

```
export MQSSLTRUSTSTORE=example-app1-chain.crt
```

Este archivo contiene la cadena de certificados completa, incluidos los certificados de aplicación y CA.

6. Transferir mensajes a la cola

Ejecute el siguiente mandato:

```
/opt/mqm/samp/bin/amqsputc EXAMPLE.QUEUE EXAMPLEQM
```

Si la conexión con el gestor de colas es satisfactoria, se genera la siguiente respuesta:

```
target queue is EXAMPLE.QUEUE
```

Transfiera varios mensajes a la cola, especificando texto y, a continuación, pulsando **Intro** cada vez.

Para finalizar, pulse **Intro** dos veces.

7. Recuperar los mensajes de la cola

Ejecute el siguiente mandato:

```
/opt/mqm/samp/bin/amqsgetc EXAMPLE.QUEUE EXAMPLEQM
```

Los mensajes que ha añadido en el paso anterior se han consumido y son la salida. Después de unos segundos, el mandato sale.

Resultados

Enhorabuena, ha probado correctamente la conexión de un gestor de colas con TLS habilitado y ha mostrado que puede transferir y obtener mensajes de forma segura en el gestor de colas desde un cliente.

Ejemplo: personalización de anotaciones de servicio de licencia

IBM MQ Operator añade automáticamente anotaciones IBM License Service a los recursos desplegados. Estos son supervisados por IBM License Service, y se generan informes que corresponden a la titularidad necesaria.

Acerca de esta tarea

Las anotaciones añadidas por IBM MQ Operator son las que se esperan en situaciones estándar y se basan en los valores de licencia seleccionados durante el despliegue de un gestor de colas.

Ejemplo

Si **License** se establece en L-RJON-BZFQU2 (IBM Cloud Pak for Integration 2021.2.1), y **Use** se establece en NonProduction, se aplican las anotaciones siguientes:

- cloudpakId: c8b82d189e7545f0892db9ef2731b90d
- cloudpakName: IBM Cloud Pak for Integration
- Contenedores productCharged: qmgr
- Proporción de productCloudpak: '4:1'
- productID: 21dfe9a0f00f444f888756d835334909
- productName: IBM MQ Advanced para No producción
- productMetric: VIRTUAL_PROCESSOR_CORE
- productVersion: 9.2.3.0

En IBM Cloud Pak for Integration, los despliegues de IBM App Connect Enterprise incluyen una titularidad restringida para IBM MQ. En estas situaciones, es necesario alterar temporalmente estas anotaciones para asegurarse de que IBM License Service captura el uso correcto. Para ello, utilice el enfoque descrito en [“Adición de anotaciones y etiquetas personalizadas a los recursos del gestor de colas”](#) en la [página 96](#).

Por ejemplo, si IBM MQ se despliega bajo la titularidad de IBM App Connect Enterprise , utilice el enfoque que se muestra en el fragmento de código siguiente:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productMetric: FREE
```

Hay otras dos razones comunes por las que las anotaciones de licencia pueden requerir modificación:

1. IBM MQ Advanced se incluye en la titularidad de otro producto de IBM .
 - En esta situación, utilice el enfoque descrito anteriormente para IBM App Connect Enterprise.
2. IBM MQ se despliega bajo una licencia de IBM Cloud Pak for Integration .
 - Si tiene una licencia de IBM Cloud Pak for Integration , puede decidir desplegar un gestor de colas bajo la proporción IBM MQ o IBM MQ Advanced . Si realiza el despliegue bajo una proporción de IBM MQ , debe asegurarse de que no utiliza ninguna prestación avanzada como, por ejemplo, HA nativa o Advanced Message Security.
 - En esta situación, utilice las anotaciones siguientes para el uso de producción:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: c661609261d5471fb4ff8970a36bccea
    productCloudpakRatio: '4:1'
    productName: IBM MQ for Production
    productMetric: VIRTUAL_PROCESSOR_CORE
```

- Utilice las anotaciones siguientes para uso no de producción:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
```

```
productID: 151bec68564a4a47a14e6fa99266deff
productCloudpakRatio: '8:1'
productName: IBM MQ for Non-Production
productMetric: VIRTUAL_PROCESSOR_CORE
```

Configuración de la alta disponibilidad para gestores de colas utilizando la IBM MQ Operator

Acerca de esta tarea

Procedimiento

- [“HA nativa” en la página 21.](#)
- [“Ejemplo: Configuración de HA nativa utilizando IBM MQ Operator” en la página 76.](#)
- [“Configuración de un gestor de colas de varias instancias utilizando IBM MQ Operator” en la página 82.](#)

Configuración de HA nativa utilizando IBM MQ Operator

La HA nativa se configura utilizando la API de QueueManager y las opciones avanzadas están disponibles utilizando un archivo INI.

La alta disponibilidad nativa se configura utilizando `.spec.queueManager.availability` de la API de QueueManager, por ejemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: nativeha-example
spec:
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    availability:
      type: NativeHA
    version: 9.4.0.0-r1
```

El campo `.spec.queueManager.availability.type` debe establecerse en NativeHA.

En `.spec.queueManager.availability`, también puede configurar un secreto TLS y cifrados para utilizarlos entre instancias de gestor de colas al replicar. Esto se recomienda encarecidamente y hay disponible una guía paso a paso en [“Ejemplo: Configuración de HA nativa utilizando IBM MQ Operator” en la página 76.](#)

Tareas relacionadas

[“Ejemplo: Configuración de HA nativa utilizando IBM MQ Operator” en la página 76](#)

Este ejemplo despliega un gestor de colas utilizando la característica de alta disponibilidad nativa en OpenShift Container Platform utilizando IBM MQ Operator. Se utiliza TLS mutuo para la autenticación, para correlacionar desde un certificado TLS con una identidad en el gestor de colas.

Ejemplo: Configuración de HA nativa utilizando IBM MQ Operator

Este ejemplo despliega un gestor de colas utilizando la característica de alta disponibilidad nativa en OpenShift Container Platform utilizando IBM MQ Operator. Se utiliza TLS mutuo para la autenticación, para correlacionar desde un certificado TLS con una identidad en el gestor de colas.

Antes de empezar

Para completar este ejemplo, primero debe haber completado los siguientes requisitos previos:

- Cree un proyecto/espacio de nombres OpenShift Container Platform (OCP) para este ejemplo.
- En la línea de mandatos, inicie sesión en el clúster de OCP y cambie al espacio de nombres anterior.
- Asegúrese de que el IBM MQ Operator esté instalado y disponible en el espacio de nombres anterior.

Acerca de esta tarea

Este ejemplo proporciona un YAML de recurso personalizado que define un gestor de colas que se desplegará en OpenShift Container Platform. También detalla los pasos adicionales necesarios para desplegar el gestor de colas con TLS habilitado.

Procedimiento

1. Cree un par de certificados tal como se describe en [“Creación de una PKI autofirmada utilizando OpenSSL”](#) en la página 68.

2. Crear una correlación de configuración que contenga mandatos MQSC y un archivo INI

Cree un Kubernetes ConfigMap que contenga los mandatos MQSC para crear una cola nueva y un canal SVRCONN, y para añadir un registro de autenticación de canal que permita el acceso al canal.

Asegúrese de que está en el espacio de nombres que ha creado anteriormente (consulte [Antes de empezar](#)) y, a continuación, especifique el YAML siguiente en la consola web de OCP o utilizando la línea de mandatos.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-nativeha-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*)' USERSRC(NOACCESS)
    ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
    MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
    AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
```

El MQSC define un canal denominado *MTLS.SVRCONN* y una cola denominada *EXAMPLE.QUEUE*. El canal está configurado para permitir el acceso sólo a los clientes que presentan un certificado con un "nombre común" de *example-app1*. Este es el nombre común utilizado en uno de los certificados creados en el paso [“1”](#) en la [página 77](#). Las conexiones en este canal con este nombre común se correlacionan con un ID de usuario de *app1*, que está autorizado para conectarse al gestor de colas y para acceder a la cola de ejemplo. El archivo INI habilita una política de seguridad, lo que significa que el ID de usuario *app1* no es necesario que exista en un registro de usuarios externo; sólo existe como nombre en esta configuración.

3. Desplegar el gestor de colas

Cree un nuevo gestor de colas utilizando el siguiente recurso personalizado YAML. Asegúrese de que está en el espacio de nombres que ha creado antes de empezar esta tarea y, a continuación, especifique el YAML siguiente en la consola web de OCP o utilizando la línea de mandatos. Compruebe que se ha especificado la licencia correcta y acepte la licencia cambiando `false` a `true`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
```

```

license: L-EHXT-MQCRN9
use: Production
queueManager:
  name: EXAMPLEQM
  availability:
    type: NativeHA
  tls:
    secretName: example-qm-tls
mqsc:
- configMap:
  name: example-nativeha-configmap
  items:
  - example-tls.mqsc
ini:
- configMap:
  name: example-nativeha-configmap
  items:
  - example-tls.ini
storage:
  queueManager:
    type: persistent-claim
version: 9.4.0.0-r1
pki:
  keys:
  - name: default
    secret:
      secretName: example-qm-tls
      items:
      - tls.key
      - tls.crt
      - ca.crt

```

Tenga en cuenta que el secreto *example-qm-tls* se ha creado en el paso “1” en la página 77 y que el ConfigMap *example-nativeha-configmap* se ha creado en el paso “2” en la página 77

El tipo de disponibilidad se establece en *NativeHA* y se selecciona el almacenamiento persistente. Se utilizará la clase de almacenamiento predeterminada configurada en el clúster de Kubernetes. Si no tiene una clase de almacenamiento configurada como predeterminada, o si desea utilizar una clase de almacenamiento diferente, añada `defaultClass: storage_class_name` en `spec.queueManager.storage`.

Los tres pods de un gestor de colas de HA nativa replican datos a través de la red. Este enlace no está cifrado de forma predeterminada, pero este ejemplo utiliza el certificado del gestor de colas para cifrar el tráfico. Puede especificar un certificado diferente para seguridad adicional. El secreto TLS de HA nativa debe ser un secreto TLS de Kubernetes, que tiene una estructura determinada (por ejemplo, la clave privada debe denominarse *tls.key*).

4. Confirmar que el gestor de colas se está ejecutando

El gestor de colas se está desplegando ahora. Confirme que está en estado `Running` antes de continuar. Por ejemplo:

```
oc get qmgr exampleqm
```

5. Probar la conexión con el gestor de colas

Para confirmar que el gestor de colas está configurado y disponible, siga los pasos de [“Prueba de una conexión TLS mutua con un gestor de colas desde el portátil”](#) en la página 72.

6. Forzar que falle el pod activo

Para validar la recuperación automática del gestor de colas, simule una anomalía de pod:

a) Ver los pods activo y en espera

Ejecute el siguiente mandato:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

Tenga en cuenta que, en el campo **READY**, el pod activo devuelve el valor 1/1, mientras que los pods de réplica devuelven el valor 0/1.

b) Suprimir el pod activo

Ejecute el mandato siguiente, especificando el nombre completo del pod activo:

```
oc delete pod exampleqm-ibm-mq-value
```

c) Volver a ver el estado del pod

Ejecute el siguiente mandato:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

d) Ver el estado del gestor de colas

Ejecute el mandato siguiente, especificando el nombre completo de uno de los otros pods:

```
oc exec -t Pod -- dspmq -o nativeha -x -m EXAMPLEQM
```

Debería ver que el estado muestra que la instancia activa ha cambiado, por ejemplo:

```
QMNAME(EXAMPLEQM) ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

e) Vuelva a probar la conexión con el gestor de colas

Para confirmar que el gestor de colas se ha recuperado, siga los pasos de [“Prueba de una conexión TLS mutua con un gestor de colas desde el portátil”](#) en la página 72.

Resultados

Enhorabuena, ha desplegado correctamente un gestor de colas con alta disponibilidad nativa y autenticación TLS mutua, y ha verificado que se recupera automáticamente cuando falla el pod activo.

Visualización del estado de los gestores de colas de HA nativa para contenedores IBM MQ

Para los contenedores de IBM MQ , puede ver el estado de las instancias de HA nativa ejecutando el mandato **dspmq** dentro de uno de los Pods en ejecución.

Acerca de esta tarea

Puede utilizar el mandato **dspmq** en uno de los Pods en ejecución para ver el estado operativo de una instancia de gestor de colas. La información devuelta depende de si la instancia está activa o es una réplica. La información proporcionada por la instancia activa es definitiva, es posible que la información de los nodos de réplica esté obsoleta.

Puede realizar las acciones siguientes:

- Ver si la instancia del gestor de colas en el nodo actual está activa o es una réplica.
- Ver el estado operativo de HA nativa de la instancia en el nodo actual.
- Ver el estado operativo de las tres instancias en una configuración de HA nativa.

Los siguientes campos de estado se utilizan para notificar el estado de configuración de HA nativa:

ROLE

Especifica el rol actual de la instancia y es uno de Active, Replica o Unknown.

INSTANCIA

El nombre proporcionado para esta instancia del gestor de colas cuando se creó utilizando la opción **-lx** del mandato **crtmqm** .

INSYNC

Indica si la instancia puede tomar el control como instancia activa si es necesario.

QUORUM

Informa del estado de quórum con el formato *number_of_instances_in-sync/number_of_instances_configured*.

REPLADDR

La dirección de réplica de la instancia del gestor de colas.

CONNECTV

Indica si el nodo está conectado a la instancia activa.

BACKLOG

Indica el número de KB que la instancia está detrás.

CONNINST

Indica si la instancia con nombre está conectada a esta instancia.

ALTDATE

Indica la fecha en la que esta información se actualizó por última vez (en blanco si nunca se ha actualizado).

ALLTIME

Indica la hora a la que se actualizó por última vez esta información (en blanco si nunca se ha actualizado).

Procedimiento

- Busque los pods que forman parte del gestor de colas.

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- Ejecute `dspm` en uno de los pods

```
oc exec -t Pod dspm
```

```
oc rsh Pod
```

para un shell interactivo, donde puede ejecutar `dspm` directamente.

- Para determinar si una instancia de gestor de colas se está ejecutando como instancia activa o como réplica:

```
oc exec -t Pod dspm -o status -m QMgrName
```

Una instancia activa de un gestor de colas denominado BOB notificaría el estado siguiente:

```
QMNAME(BOB)          STATUS(Running)
```

Una instancia de réplica de un gestor de colas denominado BOB notificaría el estado siguiente:

```
QMNAME(BOB)          STATUS(Replica)
```

Una instancia inactiva notificaría el siguiente estado:

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- Para determinar el estado operativo de HA nativa de la instancia en el pod especificado:

```
oc exec -t Pod dspm -o nativeha -m QMgrName
```

La instancia activa de un gestor de colas denominado BOB puede notificar el estado siguiente:

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Una instancia de réplica de un gestor de colas denominado BOB puede notificar el estado siguiente:

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```


Una instancia inactiva de un gestor de colas denominado BOB puede notificar el siguiente estado:

```
QMNAME(BOB)                ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Para determinar el estado operativo de HA nativa de todas las instancias de la configuración de HA nativa:

```
oc exec -t Pod dspmq -o nativeha -x -m QMgrName
```

Si emite este mandato en el nodo que ejecuta la instancia activa del gestor de colas BOB, es posible que reciba el siguiente estado:

```
QMNAME(BOB)                ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

Si emite este mandato en un nodo que ejecuta una instancia de réplica del gestor de colas BOB, es posible que reciba el siguiente estado, que indica que una de las réplicas está retrasada:

```
QMNAME(BOB)                ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

Si emite este mandato en un nodo que ejecuta una instancia inactiva del gestor de colas BOB, es posible que reciba el siguiente estado:

```
QMNAME(BOB)                ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
```

Si emite el mandato cuando las instancias todavía están negociando qué está activo y cuáles son réplicas, recibirá el estado siguiente:

```
QMNAME(BOB)                STATUS(Negotiating)
```

Tareas relacionadas

[“Ejemplo: Configuración de HA nativa utilizando IBM MQ Operator” en la página 76](#)

Este ejemplo despliega un gestor de colas utilizando la característica de alta disponibilidad nativa en OpenShift Container Platform utilizando IBM MQ Operator. Se utiliza TLS mutuo para la autenticación, para correlacionar desde un certificado TLS con una identidad en el gestor de colas.

Referencia relacionada

[Mandato dspmq \(visualizar gestores de colas\)](#)

Ajuste avanzado para HA nativa

Valores avanzados para ajustar temporizaciones e intervalos. No es necesario utilizar estos valores a menos que se sepa que los valores predeterminados no coinciden con los requisitos del sistema.

Las opciones básicas para configurar HA nativa se manejan utilizando la API de `QueueManager`, que IBM MQ Operator utiliza para configurar los archivos INI del gestor de colas subyacente. Hay algunas opciones más avanzadas que sólo se pueden configurar utilizando un archivo INI, bajo la sección `NativeHALocalInstance`. Consulte también [“Ejemplo: Suministro de archivos MQSC e INI” en la página 67](#) para obtener más información sobre cómo configurar un archivo INI.

HeartbeatInterval

El intervalo de latidos define la frecuencia en milisegundos a la que una instancia activa de un gestor de colas de HA nativo envía una pulsación de red. El rango válido del valor del intervalo de pulsaciones es de 500 (0.5 segundos) a 60000 (1 minuto), un valor fuera de este rango hace que el gestor de colas no se pueda iniciar. Si se omite este atributo, se utiliza un valor predeterminado de 5000 (5 segundos). Cada instancia debe utilizar el mismo intervalo de pulsaciones.

HeartbeatTimeout

El tiempo de espera de latido define cuánto tiempo espera una instancia de réplica de un gestor de colas de HA nativo antes de decidir que la instancia activa no responde. El rango válido del valor de tiempo de espera de intervalo de pulsaciones es de 500 (0.5 segundos) a 120000 (2 minutos). El valor del tiempo de espera de pulsaciones debe ser mayor o igual que el intervalo de pulsaciones.

Un valor no válido hace que el gestor de colas no se inicie. Si se omite este atributo, una réplica espera 2 x HeartbeatInterval antes de iniciar el proceso para seleccionar una nueva instancia activa. Cada instancia debe utilizar el mismo tiempo de espera de latido.

RetryInterval

El intervalo de reintento define la frecuencia en milisegundos a la que un gestor de colas HA nativo debe reintentar un enlace de réplica anómalo. El rango válido del intervalo de reintento es de 500 (0.5 segundos) a 120000 (2 minutos). Si se omite este atributo, una réplica espera 2 x HeartbeatInterval antes de reintentar un enlace de réplica fallido.

Finalización de gestores de colas de HA nativa

Puede utilizar el mandato `endmqm` para finalizar un gestor de colas activo o de réplica que forme parte de un grupo HA nativo.

Procedimiento

- Para finalizar la instancia activa de un gestor de colas, consulte [Finalización de gestores de colas HA nativos](#) en la sección Configuración de esta documentación.

Configuración de un gestor de colas de varias instancias utilizando IBM MQ Operator

Este ejemplo despliega un gestor de colas de varias instancias utilizando OpenShift Container Platform utilizando IBM MQ Operator. Se utiliza TLS mutuo para la autenticación, para correlacionar desde un certificado TLS con una identidad en el gestor de colas.

Antes de empezar

Para completar este ejemplo, primero debe haber completado los siguientes requisitos previos:

- Cree un proyecto/espacio de nombres OpenShift Container Platform (OCP) para este ejemplo.
- En la línea de mandatos, inicie sesión en el clúster de OCP y cambie al espacio de nombres anterior.
- Asegúrese de que el IBM MQ Operator esté instalado y disponible en el espacio de nombres anterior.

Acerca de esta tarea

Este ejemplo proporciona un YAML de recurso personalizado que define un gestor de colas que se desplegará en OpenShift Container Platform. También detalla los pasos adicionales necesarios para desplegar el gestor de colas con TLS habilitado.

Procedimiento

1. Determinar una clase de almacenamiento adecuada

Se puede acceder al almacenamiento en un clúster de Kubernetes utilizando varias [Modalidades de acceso de volumen persistente](#). Un gestor de colas de varias instancias crea varios volúmenes persistentes: uno para cada gestor de colas y al menos un volumen compartido. El volumen

compartido para un gestor de colas de varias instancias debe utilizar una clase de almacenamiento `ReadWriteMany`. La clase de almacenamiento predeterminada en un clúster Kubernetes suele ser para una clase de almacenamiento `ReadWriteOnce` (almacenamiento en bloque). Por ejemplo, si utiliza Red Hat OpenShift Data Foundation, la clase de almacenamiento `ocs-storagecluster-cephfs` proporciona un sistema de archivos compartidos adecuado. La elección del sistema de archivos es muy importante, porque no todos los sistemas de archivos compartidos manejan el bloqueo de archivos de la misma manera. Consulte [Planificación del soporte del sistema de archivos en Multiplatforms y Declaración de prueba para sistemas de archivos del gestor de colas multiinstancia de IBM MQ](#).

2. Cree un par de certificados tal como se describe en [“Creación de una PKI autofirmada utilizando OpenSSL”](#) en la página 68.
3. Crear una correlación de configuración que contenga mandatos MQSC y un archivo INI
 Cree un Kubernetes ConfigMap que contenga los mandatos MQSC para crear una cola nueva y un canal SVRCONN, y para añadir un registro de autenticación de canal que permita el acceso al canal.
 Asegúrese de que está en el espacio de nombres que ha creado anteriormente (consulte [Antes de empezar](#)) y, a continuación, especifique el YAML siguiente en la consola web de OCP o utilizando la línea de mandatos.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-miqm-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*) USERSRC(NOACCESS)
    ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
    MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
    AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
```

El MQSC define un canal denominado `MTLS.SVRCONN` y una cola denominada `EXAMPLE.QUEUE`. El canal está configurado para permitir el acceso sólo a los clientes que presentan un certificado con un "nombre común" de `example-app1`. Este es el nombre común utilizado en uno de los certificados creados en el paso [“2”](#) en la [página 83](#). Las conexiones en este canal con este nombre común se correlacionan con un ID de usuario de `app1`, que está autorizado para conectarse al gestor de colas y para acceder a la cola de ejemplo. El archivo INI habilita una política de seguridad, lo que significa que el ID de usuario `app1` no es necesario que exista en un registro de usuarios externo; sólo existe como nombre en esta configuración.

4. Desplegar el gestor de colas
 Cree un nuevo gestor de colas utilizando el siguiente recurso personalizado YAML. Asegúrese de que está en el espacio de nombres que ha creado antes de empezar esta tarea y, a continuación, especifique el YAML siguiente en la consola web de OCP o utilizando la línea de mandatos. Compruebe que se ha especificado la licencia correcta y acepte la licencia cambiando `false` a `true`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: EXAMPLEQM
  availability:
```

```

type: MultiInstance
mqsc:
- configMap:
  name: example-miqm-configmap
  items:
  - example-tls.mqsc
ini:
- configMap:
  name: example-miqm-configmap
  items:
  - example-tls.ini
storage:
  defaultClass: STORAGE_CLASS
version: 9.4.0.0-r1
pki:
  keys:
  - name: default
    secret:
      secretName: example-qm-tls
      items:
      - tls.key
      - tls.crt
      - ca.crt

```

Cambie *STORAGE_CLASS* por la clase de almacenamiento que ha identificado en el paso “1” en la [página 82](#).

Tenga en cuenta que el secreto *example-qm-tls* se ha creado en el paso “2” en la [página 83](#) y que el ConfigMap *example-miqm-configmap* se ha creado en el paso “3” en la [página 83](#).

El tipo de disponibilidad se establece en *MultiInstance*, lo que hace que el almacenamiento persistente se seleccione automáticamente.

5. Confirmar que el gestor de colas se está ejecutando

El gestor de colas se está desplegando ahora. Confirme que está en estado *Running* antes de continuar. Por ejemplo:

```
oc get qmgr exampleqm
```

6. Probar la conexión con el gestor de colas

Para confirmar que el gestor de colas está configurado y disponible, siga los pasos de “[Prueba de una conexión TLS mutua con un gestor de colas desde el portátil](#)” en la [página 72](#).

7. Forzar que falle el pod activo

Para validar la recuperación automática del gestor de colas, simule una anomalía de pod:

a) Ver los pods activo y en espera

Ejecute el siguiente mandato:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

Tenga en cuenta que, en el campo **READY**, el pod activo devuelve el valor 1/1, mientras que el pod en espera devuelve el valor 0/1.

b) Suprimir el pod activo

Ejecute el mandato siguiente, especificando el nombre completo del pod activo:

```
oc delete pod exampleqm-ibm-mq-value
```

c) Volver a ver el estado del pod

Ejecute el siguiente mandato:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

d) Ver el estado del gestor de colas

Ejecute el mandato siguiente, especificando el nombre completo del otro pod:

```
oc exec -t Pod -- dspmq -x
```

Debería ver que el estado muestra que la instancia activa ha cambiado, por ejemplo:

```
QMNAME(EXAMPLEQM)                                STATUS(Running as standby)
INSTANCE(exampleqm-ibm-mq-1) MODE(Active)
INSTANCE(exampleqm-ibm-mq-0) MODE(Standby)
```

e) Vuelva a probar la conexión con el gestor de colas

Para confirmar que el gestor de colas se ha recuperado, siga los pasos de [“Prueba de una conexión TLS mutua con un gestor de colas desde el portátil”](#) en la [página 72](#).

Resultados

Enhorabuena, ha desplegado correctamente un gestor de colas de varias instancias con autenticación TLS mutua y ha verificado que se recupera automáticamente cuando falla el pod activo.

Configuración de una ruta para conectarse a un gestor de colas desde fuera de un clúster de Red Hat OpenShift

Necesita una ruta de Red Hat OpenShift para conectar una aplicación a un gestor de colas de IBM MQ desde fuera de un clúster de Red Hat OpenShift . Debe habilitar TLS en el gestor de colas y la aplicación cliente de IBM MQ , porque SNI sólo está disponible en el protocolo TLS cuando se utiliza un protocolo TLS 1.2 o superior. Red Hat OpenShift Container Platform Router utiliza SNI para direccionar solicitudes al gestor de colas IBM MQ .

Acerca de esta tarea

La configuración necesaria de la [Ruta de Red Hat OpenShift](#) depende del comportamiento de [Indicación de nombre de servidor \(SNI\)](#) de la aplicación cliente. IBM MQ da soporte a dos valores de cabecera SNI diferentes en función de la configuración y el tipo de cliente. Una cabecera SNI se establece en el nombre de host del destino del cliente o se establece de forma alternativa en el nombre de canal IBM MQ . Para obtener información sobre cómo IBM MQ correlaciona un nombre de canal con un nombre de host, consulte [Cómo IBM MQ proporciona varias prestaciones de certificados](#).

Si una cabecera SNI se establece en un nombre de canal IBM MQ o un nombre de host se controla utilizando el atributo **OutboundSNI** . Los valores posibles son `OutboundSNI=CHANNEL` (el valor predeterminado) o `OutboundSNI=HOSTNAME`. Para obtener más información, consulte [Stanza SSL del archivo de configuración de cliente](#). Tenga en cuenta que CHANNEL y HOSTNAME son los valores exactos que utiliza; no son nombres de variable que sustituya por un nombre de canal o nombre de host real.

Comportamientos de cliente con distintos valores de OutboundSNI

Si **OutboundSNI** se establece en HOSTNAME, los clientes siguientes establecen un nombre de host SNI siempre que se proporcione un nombre de host en el nombre de conexión:

- Clientes C
- Clientes .NET en modalidad no gestionada
- Clientes Java/JMS

Si **OutboundSNI** se establece en HOSTNAME y se utiliza una dirección IP en el nombre de conexión, los clientes siguientes envían una cabecera SNI en blanco:

- Clientes C
- Clientes .NET en modalidad no gestionada
- Clientes Java/JMS (que no pueden realizar una búsqueda DNS inversa del nombre de host)

Si **OutboundSNI** se establece en CHANNEL, o no se establece, en su lugar se utiliza un nombre de canal IBM MQ y siempre se envía, tanto si se utiliza un nombre de host como un nombre de conexión de dirección IP.

Los siguientes tipos de cliente no dan soporte al establecimiento de una cabecera SNI en un nombre de canal IBM MQ y, por lo tanto, siempre intenta establecer la cabecera SNI en un nombre de host independientemente del valor **OutboundSNI** :

- Clientes AMQP
- Clientes XR

El cliente IBM MQ gestionado .NET establece SERVERNAME en el nombre de host respectivo si la propiedad **OutboundSNI** se establece en HOSTNAME, lo que permite que un cliente IBM MQ gestionado .NET se conecte a un gestor de colas utilizando rutas de Red Hat OpenShift .

Si una aplicación cliente se conecta a un gestor de colas desplegado en un clúster de Red Hat OpenShift a través de IBM MQ Internet Pass-Thru (MQIPT), MQIPT se puede configurar para establecer SNI en el nombre de host utilizando la propiedad [SSLClientOutboundSNI](#) en la definición de ruta.

OutboundSNI, varios certificados y rutas de Red Hat OpenShift

IBM MQ utiliza la cabecera SNI para proporcionar varias funciones de certificados. Si una aplicación se está conectando a un canal IBM MQ que está configurado para utilizar un certificado diferente a través del campo CERTLABL, la aplicación debe conectarse con un valor de **OutboundSNI** de CHANNEL.

Si la configuración de ruta de Red Hat OpenShift requiere un HOSTNAME SNI, no podrá utilizar la funcionalidad de varios certificados de IBM MQ y no podrá establecer un valor CERTLABL en ningún objeto de canal de IBM MQ .

Si una aplicación con un valor **OutboundSNI** distinto de CHANNEL se conecta a un canal con una etiqueta de certificado configurada, la aplicación se rechaza con un MQRC_SSL_INITIALIZATION_ERROR y se imprime un mensaje AMQ9673 en los registros de errores del gestor de colas.

Para obtener más información sobre cómo IBM MQ proporciona varias funciones de certificados, consulte [Cómo IBM MQ proporciona varias funciones de certificados](#) .

Ejemplo

Las aplicaciones cliente que establecen SNI en el canal MQ requieren que se cree una nueva ruta Red Hat OpenShift para cada canal al que desea conectarse. También tiene que utilizar nombres de canal exclusivos en el clúster de Red Hat OpenShift Container Platform , para permitir el direccionamiento al gestor de colas correcto.

Es importante que los nombres de canal de MQ no terminen en minúsculas debido a la forma en que IBM MQ correlaciona los nombres de canal con las cabeceras SNI.

Para determinar el nombre de host necesario para cada una de las nuevas rutas de Red Hat OpenShift , debe correlacionar cada nombre de canal con una dirección SNI. Consulte [Cómo IBM MQ proporciona la prestación de varios certificados para obtener más información](#).

A continuación, debe crear una nueva ruta de Red Hat OpenShift para cada canal, aplicando lo siguiente `yaml` en el clúster:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: unique_name_for_the_route
  namespace: namespace_of_your_MQ_deployment
spec:
  host: SNI_address_mapping_for_the_channel
  to:
    kind: Service
    name: name_of_Kubernetes_Service_for_your_MQ_deployment (for example "queue_manager_name-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 "queue_manager_name-ibm-mq-qm")>
-n namespace of your MQ deployment -o jsonpath="{.spec.host}"
```

El puerto para la conexión de cliente debe establecerse en el puerto utilizado por el direccionador de Red Hat OpenShift Container Platform , normalmente 443.

Tareas relacionadas

“Conexión con el IBM MQ Console desplegado en un clúster de Red Hat OpenShift” en la página 130
Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

Integración de IBM MQ con el rastreo de IBM Instana

IBM Instana se puede utilizar para rastrear transacciones dentro de IBM Cloud Pak for Integration.

Antes de empezar

Este documento cubre el rastreo de IBM Instana , que es el proceso de rastrear mensajes a través de un sistema. No cubre la supervisión de IBM Instana , en la que se recuperan detalles sobre el estado de un gestor de colas de IBM MQ . Para obtener información sobre la supervisión de IBM MQ mediante IBM Instana , consulte [Supervisión de IBM MQ](#) . Para obtener instrucciones detalladas sobre la supervisión autenticada, consulte [“Configuración de la supervisión de IBM Instana autenticada con TLS”](#) en la página 89.

Nota:

- Esta característica solo está soportada en los operandos de IBM MQ versión 9.3.1.0-r2 o posterior.
- Puede ejecutar el rastreo de IBM Instana en versiones anteriores de IBM MQ Operator y del gestor de colas, pero no de forma nativa. Consulte [Configuración de IBM MQ Tracing](#) en la documentación de IBM Instana .

Antes de poder realizar el rastreo de IBM Instana con el operador de IBM MQ , debe desplegar un agente de IBM Instana backend y IBM Instana . De forma predeterminada, un gestor de colas de IBM MQ se comunica con un agente de IBM Instana desplegado en el mismo nodo que el pod del gestor de colas.

Acerca de esta tarea

La habilitación de la integración con IBM Instana hace que se instale una salida de API de IBM MQ en el gestor de colas. La salida de API envía datos de rastreo a los agentes de IBM Instana sobre los mensajes que fluyen a través del gestor de colas.

La salida de API añade cabeceras RFH2 a cada mensaje. Estas cabeceras contienen información de rastreo.

Los agentes de IBM Instana son responsables de enviar los datos de rastreo al programa de fondo IBM Instana .

Para obtener información sobre el despliegue de un programa de fondo de IBM Instana y agentes de IBM Instana , consulte [Habilitación de enlaces de supervisión de Instana en la interfaz de usuario de plataforma](#) en la documentación de IBM Instana .

Procedimiento

Despliegue estándar

- Despliegue un gestor de colas con el rastreo de IBM Instana habilitado.

De forma predeterminada, el rastreo de IBM Instana está inhabilitado.

Si utiliza IBM Cloud Pak for Integration Platform UI o la consola web de OpenShift :

1. Pulse **Telemetría > Rastreo > Instana**.
2. Establezca el conmutador **Habilitar rastreo de Instana** en true.

Si está desplegando a través de YAML, utilice el siguiente fragmento de código:

```
spec:
  telemetry:
    tracing:
      instana:
        enabled: true
```

Despliegue avanzado

- Comunicarse con el agente de IBM Instana a través de https.

De forma predeterminada, la salida IBM Instana para IBM MQ se comunica con el agente de IBM Instana a través de http. La dirección de host del agente se establece en la dirección IP del nodo en el que se ejecuta el gestor de colas. Esto coincide con la configuración descrita en [Habilitación de la supervisión de IBM Instana](#) en la documentación de IBM Instana , donde los agentes de IBM Instana se despliegan mediante el operador del agente de IBM Instana como un daemonset.

Actualmente, la comunicación entre la salida IBM Instana para IBM MQ y el agente IBM Instana da soporte a los protocolos http o https. Para utilizar https, primero debe configurarse el agente IBM Instana para utilizar el cifrado TLS. Consulte [Configuración del cifrado TLS para el punto final de agente](#) en la documentación de IBM Instana . A continuación, el protocolo se puede establecer en https como se indica a continuación:

Si está utilizando la consola web de OpenShift :

1. Pulse **Telemetría > Instana**.
2. Expanda la lista desplegable **Configuración avanzada** .
3. Establezca el **protocolo de comunicación del agente de Instana** en https.

Si está desplegando a través de YAML, utilice el siguiente fragmento de código:

```
spec:
  telemetry:
    instana:
      enabled: true
      protocol: https
```

- Establezca **agentHost**

Si los agentes de IBM Instana no se han desplegado como daemonset en el clúster de Openshift donde se ejecuta el gestor de colas, debe establecer el valor **agentHost** en el nombre de host o dirección IP donde se ejecuta el agente de IBM Instana . El valor **agentHost** no debe incluir un protocolo o puerto.

Si está utilizando la consola web de OpenShift :

1. Pulse **Telemetría > Instana**.
2. Expanda la lista desplegable **Configuración avanzada** .
3. Escriba el nombre de host en el recuadro de texto **Host de agente de Instana** .

Si está desplegando a través de YAML, utilice el siguiente fragmento de código:

```
spec:
  telemetry:
    instana:
      enabled: true
      agentHost: 9.9.9.9
```

Qué hacer a continuación

Consulte también [“Despliegue de un gestor de colas simple utilizando la IBM MQ Operator”](#) en la página 64.

Configuración de la supervisión de IBM Instana autenticada con TLS

Para poder supervisar un gestor de colas a través de un agente de IBM Instana , debe configurar tanto el agente como el gestor de colas.

Antes de empezar

La sección "Configuración" de "Supervisión de IBM MQ" de la documentación de IBM Instana proporciona información general sobre la configuración de supervisión de IBM Instana . Sin embargo, no incluye detalles sobre la configuración del gestor de colas.

Antes de poder realizar el rastreo de IBM Instana con el operador de IBM MQ , debe desplegar un agente de IBM Instana backend y IBM Instana . Para ello, consulte [Habilitación de la supervisión de IBM Instana en la interfaz de usuario de CP4I Platform](#) en la documentación de IBM Instana .

Procedimiento

1. [Generar certificados.](#)
2. [Configure los agentes de IBM Instana.](#)
3. [Configure el gestor de colas.](#)
4. [Verificar y depurar.](#)

Tareas relacionadas

“Integración de IBM MQ con el rastreo de IBM Instana” en la [página 87](#)

IBM Instana se puede utilizar para rastrear transacciones dentro de IBM Cloud Pak for Integration.

Generar un certificado y una clave para el agente de IBM Instana y el gestor de colas

Para la comunicación TLS entre el agente de IBM Instana y el gestor de colas, ambos deben tener un certificado y la clave privada correspondiente.

Antes de empezar

Esta es la primera de las cuatro tareas para [configurar la supervisión de IBM Instana autenticada con TLS](#).

Nota: Los valores utilizados en la generación de estos certificados son para fines de demostración. Al desplegar en un entorno de producción, asegúrese de que el asunto y la caducidad del certificado son adecuados.

Procedimiento

Gestor de colas de IBM MQ

Para comunicarse con el agente de IBM Instana a través de TLS, el gestor de colas debe tener un certificado y la clave privada correspondiente. Si ya los tiene, omite esta sección.

1. Genere un certificado y una clave privada para el gestor de colas.

Ejecute el siguiente mandato:

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

agente de IBM Instana

Para que el agente realice la comunicación TLS con el gestor de colas IBM MQ , el agente debe tener un certificado y la clave privada correspondiente. Si ya tiene una clave privada y un certificado en un almacén de claves JKS que desea utilizar, omite esta sección.

2. Genere un certificado y una clave privada para el agente de IBM Instana .

Ejecute el siguiente mandato:

```
openssl req \  
-newkey rsa:2048 -nodes -keyout application.key \  
-subj "/CN=instana-agent/OU=app team1" \  
-x509 -days 3650 -out application.crt
```

3. Almacene el certificado y la clave privada en un almacén de claves PKCS12 .

Ejecute el mandato siguiente, sustituyendo *su_contraseña* por la contraseña que desea utilizar para proteger el almacén de claves. Realice esta sustitución en todos los pasos posteriores.

```
openssl pkcs12 -export -out application.p12 -inkey application.key -in application.crt  
-passout pass:your_password
```

4. Convierta el almacén de claves PKCS12 en un almacén de claves JKS.

Ejecute el siguiente mandato:

```
keytool -importkeystore \  
-srckeystore application.p12 \  
-srcstoretype pkcs12 \  
-destkeystore application.jks \  
-deststoretype JKS \  
-srcstorepass your_password \  
-deststorepass your_password \  
-noprompt
```

5. Etiquete el certificado.

Ejecute el siguiente mandato:

```
keytool -changealias -alias "1" -destalias "instana" -keypass your_password -keystore  
application.jks -storepass your_password -noprompt
```

6. Importe el certificado del gestor de colas en el almacén de claves.

Ejecute el siguiente mandato:

```
keytool -importcert -file server.crt -keystore application.jks -storepass your_password  
-alias myca -noprompt
```

Qué hacer a continuación

Ahora está preparado para [configurar los agentes para la IBM Instana supervisión](#).

Supervisión de Instana: configuración de agentes

Monte el almacén de claves en los agentes de IBM Instana y, a continuación, configure la supervisión para un gestor de colas específico.

Antes de empezar

Esta tarea presupone que ha [generado un certificado y una clave para los agentes de IBM Instana y el gestor de colas](#).

Procedimiento

Montaje del almacén de claves en los agentes de IBM Instana

1. Cree un secreto a partir del almacén de claves JKS en el espacio de nombres del agente de IBM Instana .

Ejecute el mandato siguiente, sustituyendo *keystore_secret_name* por el nombre que desea utilizar. Realice esta sustitución en todos los pasos posteriores.

```
oc create secret generic keystore_secret_name --from-file=./application.jks -n instana-agent
```

2. En el espacio de nombres instana-agent, utilice el mandato `oc edit daemonset instana-agent` para editar el daemonset instana-agent para incluir el siguiente `volumeMount` adicional y el volumen:

```
volumeMounts:
- name: mq-key-jks-name
  subPath: application.jks
  mountPath: /opt/instana/agent/etc/application.jks
volumes:
- name: mq-key-jks-name
  secret:
    secretName: keystore_secret_name
```

Configuración de la supervisión para un gestor de colas específico

3. En el espacio de nombres instana-agent, utilice el mandato `oc edit configmap instana-agent` para editar el configmap instana-agent.
4. Añada la sección siguiente bajo `configuration.yaml`: `|`. Si ya ha definido esta sección, añada el nuevo gestor de colas a la lista.

```
com.instana.plugin.ibmmq:
  enabled: true
  poll_rate: 60
  queueManagers:
    QUEUE_MANAGER_NAME:
      channel: 'INSTANA.A.SVRCONN'
      keystorePassword: 'your_password'
      keystore: '/opt/instana/agent/etc/application.jks'
      cipherSuite: 'TLS_RSA_WITH_AES_256_CBC_SHA256'
```

donde

- `su_contraseña` es la contraseña del almacén de claves JKS
- `QUEUE_MANAGER_NAME` es el nombre del gestor de colas de IBM MQ subyacente que se va a desplegar, en lugar del nombre del operando del gestor de colas.

Nota: Si `QUEUE_MANAGER_NAME` no se establece en el nombre del gestor de colas subyacente y, en su lugar, se establece en el operando, la supervisión no funcionará. El nombre subyacente se define en `spec.queuemanager.name` para el operando del gestor de colas.

5. Suprima los pods instana-agent en el espacio de nombres instana-agent. Esto hace que se reinicien y que comiencen a supervisar con los nuevos valores.

Qué hacer a continuación

Ahora está preparado para [configurar el gestor de colas para la supervisión de IBM Instana](#).

Supervisión de Instana: configuración del gestor de colas

Configure un gestor de colas que utilice TLS para comunicarse con el agente de IBM Instana. La autenticación para esta conexión se realiza utilizando un [SSLPEERMAP](#).

Antes de empezar

Esta tarea presupone que ha [configurado los agentes para la IBM Instana supervisión](#).

Procedimiento

1. Configure el gestor de colas a través de MQSC e INI.

MQSC se utiliza para configurar un nuevo canal habilitado para TLS y, a continuación, configurar dicho canal para autenticar el agente de IBM Instana de conexión si tiene un certificado con los campos necesarios. En este caso, correlacionamos cualquier cliente de conexión con un certificado que contiene los campos `CN=instana-agent,OU=app_team1` con el usuario `app1`. A continuación, MQSC otorga permiso al usuario `app1` para realizar las operaciones necesarias para la supervisión de IBM Instana.

El archivo INI se utiliza para otorgar permisos a nuestro usuario externo app1.

El siguiente configmap contiene los valores necesarios de MQSC e INI. Despléguelo en el espacio de nombres del gestor de colas.

```
apiVersion: v1
data:
  channel.mqsc: |-
    DEFINE CHANNEL('INSTANA.A.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    ALTER QMGR CONNAUTH(' ')
    REFRESH SECURITY
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
  ACTION(REPLACE)
    SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=instana-agent,OU=app
team1') USERSRC(MAP) MCAUSER('app1')
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(ALL)
    SET AUTHREC PROFILE('SYSTEM.ADMIN.COMMAND.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
  AUTHADD(PUT,INQ,DSP,CHG)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(TOPIC) AUTHADD(DSP)
    SET AUTHREC PROFILE('*') PRINCIPAL('app1') OBJTYPE(TOPIC) AUTHADD(DSP)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP,CHG,GET)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(LISTENER) AUTHADD(DSP)
    SET AUTHREC PROFILE('AMQ.*') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP,CHG)
  REFRESH SECURITY TYPE(CONNAUTH)
  auth.ini: |-
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
kind: ConfigMap
metadata:
  namespace: your-queue-manager-namespace
  name: qmgr-monitoring-config
```

donde *su-espacio-nombres-gestor-colas* es el espacio de nombres en el que se desplegará el gestor de colas.

Nota: Si está supervisando colas definidas por el usuario, debe añadir líneas adicionales al mapa de configuración MQSC, otorgando permisos DSP, CHG y GET a dichas colas. Por ejemplo:

```
SET AUTHREC PROFILE('MYQUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP,CHG,GET).
```

Este ejemplo utiliza un mapa de configuración para los datos MQSC e INI, pero puede utilizar un secreto si las adiciones que realice son confidenciales. Para obtener información general sobre el despliegue con MQSC e INI, consulte [“Ejemplo: Suministro de archivos MQSC e INI” en la página 67](#).

2. Para que se realice una conexión TLS, el gestor de colas debe confiar en el certificado del agente de IBM Instana . Para conseguirlo, cree un secreto que contenga sólo el certificado del agente de IBM Instana :

```
oc create secret generic instana-certificate-secret --from-file=./application.crt -n your-queue-manager-namespace
```

3. El gestor de colas debe presentar su propio certificado para el reconocimiento TLS y requiere acceso a la clave privada asociada. Despliegue un secreto que contenga la clave y el certificado que ha creado anteriormente o que ya posee:

```
oc create secret tls qm-tls-secret --cert server.crt --key server.key -n your-queue-manager-namespace
```

Con el configmap y el secreto creados, está preparado para crear el propio gestor de colas.

4. Asegúrese de que el archivo YAML del gestor de colas no establezca la variable de entorno **MQSNOAUT** en el contenedor del gestor de colas.

De lo contrario, una vez habilitado, el mecanismo de autenticación no funcionará. La eliminación de la variable después del despliegue no hace que se vuelva a habilitar el mecanismo y que se vuelva a crear el gestor de colas.

5. Añada las secciones siguientes a la definición del gestor de colas, donde *MYQM* es el nombre del gestor de colas:

```
spec:
  queueManager:
    name: MYQM #(a)
    ini: #(b)
    - configMap:
      items:
        - auth.ini
      name: qmgr-monitoring-config
    mqsc: #(c)
    - configMap:
      items:
        - channel.mqsc
      name: qmgr-monitoring-config
  pki:
    keys: #(d)
    - name: default
      secret:
        items:
          - tls.key
          - tls.crt
        secretName: qm-tls-secret
    trust: #(e)
    - name: app
      secret:
        items:
          - application.crt
        secretName: instana-certificate-secret
```

Las secciones marcadas de la especificación se describen de la forma siguiente:

- Asegúrese de que ha asignado un nombre exclusivo al gestor de colas subyacente. Si el gestor de colas subyacente no tiene un nombre exclusivo, es posible que la supervisión no funcione según lo previsto. Este nombre debe coincidir con el nombre del mapa de configuración del agente de IBM Instana que se ha editado anteriormente.
 - La información de INI que se ha grabado en el configmap se añade al gestor de colas.
 - La información de MQSC que se ha escrito en el mapa de configuración se añade al gestor de colas.
 - El certificado del gestor de colas y la clave privada se añaden al almacén de claves del gestor de colas.
 - El certificado de agente de IBM Instana se añade al almacén de confianza del gestor de colas.
6. Opcional: Habilite el rastreo de IBM Instana en el gestor de colas supervisado.
- Si desea hacerlo, consulte [“Integración de IBM MQ con el rastreo de IBM Instana”](#) en la página 87.
7. Despliegue el gestor de colas.

Qué hacer a continuación

Ahora está preparado para [verificar y depurar la IBM Instana supervisión](#).

OpenShift Operator 2.2.0 CP4I **Supervisión de Instana: Verificación y depuración**

Para poder supervisar un gestor de colas a través de un agente de IBM Instana, debe configurar tanto el agente como el gestor de colas.

Antes de empezar

Esta tarea presupone que ha [configurado el gestor de colas para la IBM Instana supervisión](#).

Procedimiento

Verificando

- Para verificar que ha realizado correctamente el despliegue, consulte el gestor de colas en el panel de control de IBM Instana.

El gestor de colas debe estar visible en la sección de servicios de la página de la aplicación y también en la vista Infraestructura.

Depuración

Nota: Estos pasos de depuración presuponen un despliegue de OpenShift del agente de IBM Instana que se ejecuta como un daemonset.

Si no puede ver el gestor de colas en el panel de control de IBM Instana , es posible que haya configurado incorrectamente el gestor de colas. Utilice los pasos siguientes para investigar.

2. Identifique el nodo en el que se ejecuta el pod del gestor de colas activo.

Ejecute el mandato siguiente en el espacio de nombres del gestor de colas:

```
oc get pods -o wide -n your-queue-manager-namespace
```

3. Para determinar qué pod de agente de IBM Instana se ejecuta en el mismo nodo que el gestor de colas, ejecute el mismo mandato en el espacio de nombres instana-agent:

```
oc get pods -o wide -n instana-agent-namespace
```

4. Para ayudarle a comprender los problemas del lado del agente de IBM Instana , obtenga los registros del pod del agente de IBM Instana y busque entradas relacionadas con 'mq' o con el nombre del gestor de colas.

Ejecute el siguiente mandato:

```
oc logs instana-agent-pod -c instana-agent -n instana-agent
```

5. Compruebe los registros del gestor de colas.

Si el agente ha intentado conectarse al gestor de colas, los registros del gestor de colas deben indicar por qué la conexión no ha sido satisfactoria. Ejecute el siguiente mandato:

```
oc logs your-queue-manager-name -n your-queue-manager-namespace
```

Resultados

Ha completado las cuatro tareas para [configurar la supervisión de IBM Instana autenticada con TLS](#).

Creación de una imagen con archivos MQSC e INI personalizados, utilizando la CLI de Red Hat 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 Red Hat OpenShift para el registro autorizado de IBM en el proyecto Red Hat OpenShift , siga los pasos para [Crear el secreto de clave de titularidad](#).

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:9.4.0.0-r1
      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:9.4.0.0-r1"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

Tendrá que sustituir los dos lugares donde se menciona la base IBM MQ, para que apunte a la imagen base correcta para la versión y el arreglo que desea utilizar (consulte [“Historial de releases de IBM MQ Operator”](#) en la página 5 para obtener más detalles). 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 simple utilizando la IBM MQ Operator”](#) en la página 64, añadiendo la nueva imagen personalizada al YAML.

Puede añadir el siguiente fragmento de YAML a su QueueManager YAML normal, donde *my-namespace* es el Red Hat OpenShift proyecto/espacio de nombres 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 simple utilizando la IBM MQ Operator”](#) en la página 64

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

Adición de anotaciones y etiquetas personalizadas a los recursos del gestor de colas

Puede añadir anotaciones y etiquetas personalizadas a los metadatos de QueueManager .

Acerca de esta tarea

Las anotaciones y etiquetas personalizadas se añaden a todos los recursos excepto a las PVC. Si una anotación o etiqueta personalizada coincide con una clave existente, se utiliza el valor establecido por IBM MQ Operator .

Procedimiento

- Añada anotaciones personalizadas.

Para añadir anotaciones personalizadas a los recursos del gestor de colas, incluido el pod, añada las anotaciones bajo metadata. Por ejemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    annotationKey: "value"
```

- Añada etiquetas personalizadas.

Para añadir etiquetas personalizadas a los recursos del gestor de colas, incluido el pod, añada las etiquetas bajo metadata. Por ejemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  labels:
    labelKey: "value"
```

Inhabilitación de comprobaciones de webhook en tiempo de ejecución

Las comprobaciones de webhook de tiempo de ejecución garantizan que las clases de almacenamiento son viables para el gestor de colas. Inhabilítelos para mejorar el rendimiento, o porque no son válidos para el entorno.

Acerca de esta tarea

Las comprobaciones de webhook en tiempo de ejecución se realizan en la configuración del gestor de colas. Comprueban que las clases de almacenamiento son adecuadas para el tipo de gestor de colas seleccionado.

Puede optar por inhabilitar estas comprobaciones para reducir el tiempo empleado en la creación del gestor de colas, o porque las comprobaciones no son válidas para su entorno específico.

Nota: Después de inhabilitar las comprobaciones de webhook de tiempo de ejecución, se permiten los valores de clase de almacenamiento. Esto podría dar como resultado un gestor de colas roto.

Procedimiento

- Inhabilite las comprobaciones de webhook en tiempo de ejecución.

Añada la anotación siguiente bajo metadata. Por ejemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.cp4i/disable-webhook-runtime-checks" : "true"
```

Inhabilitación de las actualizaciones de valores predeterminados para la especificación del gestor de colas

El IBM MQ Operator actualiza los valores no especificados en la especificación del gestor de colas con sus valores predeterminados. Puede inhabilitar este comportamiento si desea evitar modificaciones en la especificación del gestor de colas. Los campos de estado del gestor de colas se siguen actualizando.

Procedimiento

- Inhabilite las actualizaciones de valores predeterminados del gestor de colas.

Añada la anotación siguiente bajo metadata. Por ejemplo:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.mq/write-defaults-spec" : "false"
```

Nota: Los ejemplos de inicio rápido tienen esta anotación aplicada de forma predeterminada.

Ejecución del contenedor IBM MQ con un sistema de archivos raíz de sólo lectura

Puede configurar el contenedor IBM MQ para que se ejecute con un sistema de archivos raíz de sólo lectura. Esto impide que los atacantes copien y ejecuten código malicioso en el contenedor.

Acerca de esta tarea

La habilitación del sistema de archivos raíz de sólo lectura hace que los archivos de contenedor sean inmutables. Es decir, en el sistema de archivos de contenedor, los archivos se pueden ver pero no se pueden modificar y no se pueden crear archivos nuevos. Los archivos sólo se pueden modificar o crear en un sistema de archivos montado.

Cuando se habilita un sistema de archivos raíz de sólo lectura, se crean dos volúmenes efímeros Reutilizables y Tmp y se montan en los directorios /run y /tmp respectivamente en el contenedor.

- El volumen `Reutilizable` contiene los archivos, almacenes de claves y otros archivos utilizados para configurar el gestor de colas.
- El volumen `Tmp` contiene archivos de diagnóstico, por ejemplo, los archivos RAS del gestor de colas.

Puesto que estos volúmenes son efímeros, los archivos de estos volúmenes se pierden al reiniciar el pod.

El tipo del volumen creado para los datos del gestor de colas depende del tipo de almacenamiento. De forma predeterminada, se monta un volumen persistente. O, si el tipo de almacenamiento es efímero, se monta un volumen efímero. Si el tamaño de los datos del volumen supera el valor especificado para la propiedad `sizeLimit`, Kubernetes puede expulsar el contenedor y crear uno nuevo.

Un sistema de archivos raíz de sólo lectura no está habilitado de forma predeterminada. Para habilitarlo, realice los pasos siguientes:

Procedimiento

1. Utilice la API `spec.securityContext` para habilitar el sistema de archivos raíz de sólo lectura.

Para el gestor de colas, establezca la propiedad `readOnlyRootFilesystem` en `“.spec.securityContext”` en la [página 154](#) en `true`.

IBM MQ Operator crea dos volúmenes efímeros, `Scratch` y `Tmp`.

2. Opcional: Establezca o cambie el tipo de almacenamiento de datos del gestor de colas.

De forma predeterminada, una reclamación de volumen persistente se monta en `/mnt/mqm`. O bien, si la propiedad `type` se establece en `efímero` en `“.spec.queueManager.storage.queueManager”` en la [página 152](#), se crea y se monta un volumen efímero.

3. Para cada volumen efímero, considere cuidadosamente por cuánto pueden crecer los datos. Establezca el valor de la propiedad `sizeLimit` en consecuencia, incluidas las unidades SI.

- Para el volumen efímero `Reutilizable`, establezca la propiedad `sizeLimit` en `“.spec.queueManager.storage.scratch”` en la [página 153](#). El valor predeterminado es `"100M"`.
- Para el volumen efímero `Tmp`, establezca la propiedad `sizeLimit` en `“.spec.queueManager.storage.tmp”` en la [página 153](#). El valor predeterminado es `"2Gi"`.
- Si el `type` del volumen del gestor de colas se establece en `efímero`, establezca la propiedad `sizeLimit` en `“.spec.queueManager.storage.queueManager”` en la [página 152](#). El valor predeterminado es `"2Gi"`.

Configuración de IBM MQ Console con un registro básico utilizando IBM MQ Operator

Para iniciar sesión en IBM MQ Console, puede proporcionar su propia configuración al gestor de colas.

Antes de empezar

Si está desplegando un gestor de colas con una licencia de IBM MQ Advanced for Developers, hay una configuración simple incorporada. Consulte [“Ejemplo de gestor de colas YAML que describe cómo especificar contraseñas para usuarios de admin y app”](#) en la [página 172](#). Si está desplegando un gestor de colas de licencias de IBM Cloud Pak for Integration, puede habilitar la integración con IBM Cloud Pak for Integration Keycloak para iniciar sesión en IBM MQ Console utilizando el inicio de sesión único. Consulte [“Conexión con el IBM MQ Console desplegado en un clúster de Red Hat OpenShift”](#) en la [página 130](#).

Procedimiento

1. **Cree una contraseña y cifrela utilizando `securityUtility`.**

Se utiliza un `ConfigMap` para almacenar las credenciales que utiliza para acceder al gestor de colas. Para mejorar la seguridad, codifique estas credenciales con el mandato `securityUtility`.

De forma alternativa, puede utilizar un secreto, que protege las credenciales en la capa Kubernetes . Sin embargo, las herramientas de supervisión o resolución de problemas pueden exponer el archivo subyacente de forma insegura.

2. Opcional: **Inicie sesión en la interfaz de línea de mandatos (CLI) de Red Hat OpenShift .**

Si utiliza la CLI de OpenShift , inicie sesión utilizando `oc login`.

De forma alternativa, puede utilizar la consola de OpenShift .

3. **Cree un ConfigMap con la configuración.**

Para obtener ayuda para crear la configuración XML, consulte [Seguridad de IBM MQ Console y REST API](#).

El ejemplo siguiente crea un usuario dentro del grupo MQWebAdminGroup. A los miembros del MQWebAdminGroup se les asigna el rol MQWebAdmin . En este ejemplo:

- **Debe** sustituir `USERNAME` y `PASSWORD` por sus propios valores. Tenga en cuenta que `USERNAME` se utiliza dos veces en el ejemplo.

Debe especificar `NAMESPACE` como el espacio de nombres en el que se ha desplegado IBM MQ Operator y en el que el gestor de colas estará, o ya está, desplegado.

a) Utilice la consola de OpenShift o la línea de mandatos para crear el siguiente ConfigMap:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: mqwebuserconfigmap
  namespace: NAMESPACE
data:
  mqwebuser.xml: |
    <?xml version="1.0" encoding="UTF-8"?>
    <server>
      <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>basicAuthenticationMQ-1.0</feature>
      </featureManager>
      <enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
          <security-role name="MQWebAdmin">
            <group name="MQWebAdminGroup" realm="defaultRealm"/>
          </security-role>
        </application-bnd>
      </enterpriseApplication>
      <basicRegistry id="basic" realm="defaultRealm">
        <user name="USERNAME" password="PASSWORD"/>
        <group name="MQWebAdminGroup">
          <member name="USERNAME"/>
        </group>
      </basicRegistry>
      <sslDefault sslRef="mqDefaultSSLConfig"/>
    </server>
```

b) Opcional: Si utiliza la línea de mandatos, aplique ConfigMap:

```
oc apply -f mqwebuserconfigmap.yaml
```

Para los pasos restantes, elija una de las opciones siguientes:

- Despliegue un nuevo gestor de colas con la configuración para acceder a la IBM MQ Console.
- Aplique la configuración que proporciona al IBM MQ Console acceso a un gestor de colas existente.

4. Opcional: **Despliegue un nuevo gestor de colas con la configuración para acceder a la IBM MQ Console.**

a) Cree un gestor de colas.

Establezca los proveedores de autenticación y autorización en `manual` y proporcione el ConfigMap `mqwebuserconfigmap` recién creado mediante una de las opciones siguientes:

- Opción 1: A través del gestor de colas YAML

Añada el código siguiente bajo la sección web del gestor de colas YAML:

```
...
web:
  enabled: true
  console:
    authentication:
      provider: manual
    authorization:
      provider: manual
  manualConfig:
    configMap:
      name: mqwebuserconfigmap
```

- Opción 2: A través de la vista Formulario de la consola de OpenShift :
 - i) En la consola de OpenShift , seleccione **Operadores > Operadores instalados**.
 - ii) Seleccione el despliegue del IBM MQ Operator.
 - iii) Seleccione **Gestor de colas** y pulse **Crear QueueManager**.
 - iv) Seleccione las opciones relevantes para el gestor de colas.
 - v) Seleccione **Web** y establezca **Habilitar servidor web** en true.
 - vi) Abra el recuadro de lista **Configuración avanzada** .
 - vii) En el recuadro de lista **Consola** , establezca **proveedor** para **Autenticación y Autorización** en manual.
 - viii) Abra el recuadro de lista **Configuración** .
 - ix) Abra el recuadro de lista **ConfigMap** y seleccione el ConfigMap mqwebuserconfigmap que se ha creado en el paso “3” en la página 99.
 - x) Pulse **Crear**.

Ahora puede acceder al IBM MQ Console del nuevo gestor de colas a través de las credenciales especificadas en el ConfigMap creado en el paso “3” en la página 99.

5. Opcional: **Aplicar configuración que habilita el IBM MQ Console para un gestor de colas existente.**

Edite el YAML del gestor de colas para el que está habilitando el IBM MQ Console:

- a. En la consola de OpenShift , seleccione **Operadores > Operadores instalados**.
- b. Seleccione el despliegue del IBM MQ Operator.
- c. Seleccione **Gestor de colas** y seleccione el nombre del gestor de colas.
- d. Seleccione **YAML**.
- e. Sustituya la sección web existente del YAML del gestor de colas por el código siguiente:

```
...
web:
  enabled: true
  console:
    authentication:
      provider: manual
    authorization:
      provider: manual
  manualConfig:
    configMap:
      name: mqwebuserconfigmap
```

- f. Pulse **Guardar**.

Ahora puede acceder al IBM MQ Console del gestor de colas existente a través de las credenciales especificadas en el ConfigMap creado en el paso “3” en la página 99.

Si el proveedor de almacenamiento da soporte a la expansión de volúmenes, utilice esta tarea para expandir un volumen persistente. En función del proveedor de almacenamiento, la expansión puede producirse en línea o fuera de línea.

Antes de empezar

La expansión de volumen satisfactoria se basa en el proveedor de almacenamiento para satisfacer la solicitud de expansión. Consulte la documentación de los proveedores de almacenamiento para determinar si se da soporte al redimensionamiento en línea y para obtener información sobre los procedimientos de redimensionamiento fuera de línea.

Si el proveedor de almacenamiento no puede satisfacer la solicitud de expansión, la reclamación de volumen persistente puede entrar en un estado con avisos o errores. Si la expansión falla, un administrador de OpenShift puede recuperar manualmente el estado de reclamación de volumen persistente y cancelar la expansión. Consulte [Recuperación de anomalía al expandir volúmenes](#) en la documentación de Red Hat OpenShift .

Acerca de esta tarea

Para ayudarle a gestionar el almacenamiento persistente, Kubernetes define dos recursos de API:

- Un PersistentVolume (PV), que es un fragmento de almacenamiento en el clúster que ha suministrado un administrador o que se ha suministrado dinámicamente utilizando clases de almacenamiento. Se puede suministrar de forma estática o dinámica.
- Una reclamación PersistentVolume(PVC), que es una solicitud de almacenamiento por parte de un usuario. También actúa como una comprobación de reclamación para el recurso.

Para obtener más información, consulte [Volúmenes persistentes](#) en la documentación de Kubernetes .



Aviso:

- Si la clase de almacenamiento utilizada para crear PVC de gestor de colas no da soporte al redimensionamiento en línea, se produce el redimensionamiento fuera de línea. Durante el redimensionamiento fuera de línea, es necesaria la intervención del usuario para completar la expansión del volumen, por lo que los gestores de colas experimentan un tiempo de inactividad.
- Para el redimensionamiento fuera de línea de volúmenes compartidos para [gestores de colas de varias instancias](#), los pods activo y en espera deben desactivarse al mismo tiempo al realizar la intervención del usuario.
- OpenShift no da soporte a la reducción del tamaño de las PVC. Si se intenta reducir el tamaño de los volúmenes persistentes, el gestor de colas pasará al estado 'Error'.
- Este procedimiento no se aplica a volúmenes efímeros.

Para expandir un PV utilizado por el contenedor de IBM MQ , realice los pasos siguientes.

Procedimiento

1. Preparar para expandir volúmenes

- a) Decida qué volúmenes desea expandir.
- b) Determine la clase o clases de almacenamiento que están utilizando los volúmenes.

Por ejemplo:

```
spec:
  queueManager:
    storage:
      persistedData:
        enabled: true
        type: persistent-claim
        class: ocs-storagecluster-cephfs (1)
      queueManager:
```

```
type: persistent-claim
recoveryLogs:
  enabled: true
  type: persistent-claim
defaultClass: ocs-storagecluster-ceph-rbd (2)
```

Notas:

- (1) Si el volumen define una clase de almacenamiento específica, las PVC de este tipo la utilizan.
- (2) Si se establece **defaultClass**, esta clase de almacenamiento se utiliza para todos los volúmenes sin una clase de almacenamiento específica. Si **defaultClass** no está establecido y un tipo de volumen no ha especificado una clase, se utiliza la clase de almacenamiento predeterminada para el clúster.

También puede confirmar la clase de almacenamiento en uso describiendo las PVC subyacentes. Por ejemplo:

```
oc describe pvc pvc-name
```

- c) Valide que la clase de almacenamiento da soporte a la expansión de volumen.

Una clase de almacenamiento puede tener la propiedad **.allowVolumeExpansion** definida:

- Si esta propiedad se establece en `true`, se da soporte a la expansión de volumen.
- Si esta propiedad se establece en `false`, o esta propiedad no está definida, la clase de almacenamiento no permite la expansión de volumen. En este caso, consulte la documentación del proveedor de almacenamiento para ver si esta característica se puede habilitar.

También puede describir una clase de almacenamiento para determinar si da soporte a la expansión de volumen. Por ejemplo:

```
oc describe sc storage-class-name
```

- d) Consulte la documentación del proveedor de almacenamiento para ver si se utiliza un procedimiento en línea o fuera de línea para la expansión de volúmenes.

Un procedimiento fuera de línea requiere que los pods del gestor de colas se reinicien manualmente, mientras que un procedimiento en línea no lo hace. Consulte la documentación del proveedor de almacenamiento para ver los procedimientos de redimensionamiento fuera de línea.

- e) Compruebe si el gestor de colas tiene una condición de estado con la razón 'StorageMismatch'.

Si el gestor de colas tiene esta condición de estado, los volúmenes listados en la condición se expanden si habilita la expansión de volumen. Si no desea que esto suceda, cambie los campos de tamaño asociados con cada tipo de volumen en la definición del gestor de colas para que coincidan con las PVC suministradas. La condición de estado se elimina cuando se realiza para todos los volúmenes no coincidentes.

2. Expandir volúmenes



Aviso:

- Si ha modificado anteriormente alguno de los campos de tamaño de volumen en la definición del gestor de colas, los volúmenes empiezan a expandirse cuando **.allowVolumeExpansion** se establece en `true` en la definición del gestor de colas.
- El proveedor de almacenamiento puede tener restricciones sobre el tamaño máximo de un volumen debido a las limitaciones del sistema de archivos o a la disponibilidad del hardware local. Para evitar anomalías, valide estas limitaciones en la documentación del proveedor de almacenamiento antes de expandir los volúmenes.
- Las reducciones en el tamaño de PVC no están soportadas por OpenShift. Si expande el tamaño de un volumen, no puede reducirlo. Si el intento de hacerlo falla, el IBM MQ Operator no puede devolver la PVC a su estado original.

Ejemplo de definición de gestor de colas que ilustra la expansión de volumen:

```
spec:
  queueManager:
    storage:
      allowVolumeExpansion: true (A)
      persistedData:
        enabled: true
        type: persistent-claim
        size: 3Gi (B)
      queueManager:
        type: persistent-claim
        size: 4Gi (B)
      recoveryLogs:
        enabled: true
        type: persistent-claim
        size: 3Gi (B)
```

- a) Para permitir la expansión de volúmenes para el gestor de colas, establezca el campo **.spec.queueManager.storage.allowVolumeExpansion** (A) en el gestor de colas en true.
 - b) Ahora puede aumentar los campos de tamaño (B) para cualquiera de los tipos de volumen habilitados. La aplicación de estos cambios iniciará la expansión de volumen.
3. **Valide que las PVC se hayan redimensionado.**

Notas:

- La expansión de volumen puede tardar algún tiempo. Si la validación no es satisfactoria la primera vez, considere esperar unos minutos y volver a validar.
 - La expansión de volumen sólo se completa sin la acción del usuario cuando se realiza un redimensionamiento en línea.
 - Algunos proveedores de almacenamiento redondean el tamaño de almacenamiento que ha solicitado. El volumen expandido debe tener el mismo o mayor tamaño que la solicitud.
- a) Compruebe si hay condiciones de estado en el gestor de colas. Consulte la tabla siguiente para obtener condiciones, explicaciones y acciones sugeridas.

<i>Tabla 1. Condiciones de estado para el almacenamiento.</i>		
CONDICIÓN	Mensaje	explicación
StorageMismatch	Storage sizes defined in the QueueManager resource do not match the capacity of one or more provisioned PVCs [pvc-list]. AllowVolumeExpansion is set to false in the QueueManager resource so the MQ Operator will not attempt to reconcile these differences.	La expansión de volumen no se produce porque .allowVolumeExpansion no se ha establecido en true en la definición del gestor de colas.

<i>Tabla 1. Condiciones de estado para el almacenamiento. (continuación)</i>		
CONDICIÓN	Mensaje	explicación
StorageExpansionPending	Volume expansion is pending for the following PVCs [pvc-list]	La expansión de volumen todavía está teniendo lugar. Si esta condición de estado persiste durante un periodo de tiempo prolongado, siga los pasos siguientes para recopilar más información porque podría estar teniendo lugar un redimensionamiento fuera de línea o un error de redimensionamiento.
Failed	Hay muchos posibles mensajes relacionados con el almacenamiento que pueden crear una condición de estado 'Failed'. Por ejemplo: 'MQ Queue Manager failed to deploy: persistentvolumeclaims "<pvc>" is forbidden: only dynamically provisioned pvc can be resized and the storageclass the provisions the pvc must support resize.'	Si el gestor de colas tiene condiciones de estado 'Failed' con texto que hace referencia al almacenamiento, consulte el mensaje dentro de la condición de estado. El mensaje de ejemplo que se proporciona aquí se debe a que se utiliza una clase de almacenamiento que no da soporte a la expansión.

- b) Para cada PVC que haya expandido, compruebe que la capacidad haya aumentado para que coincida o sea mayor que el valor especificado en la definición del gestor de colas.

Los gestores de colas HA pueden tener varias PVC de cada tipo. Para obtener la capacidad de una PVC, ejecute el mandato siguiente:

```
oc get pvc pvc-name -o template --template '{{.status.capacity.storage}}'
```

- c) Compruebe que la PVC no tenga condiciones de estado o sucesos que sugieran un cambio de tamaño fallido:

```
oc describe pvc pvc-name
```

- Es posible que la PVC tenga la condición de estado `FileSystemResizePending` con el mensaje 'Esperando a que el usuario (re-) inicie un pod para finalizar el redimensionamiento del sistema de archivos del volumen en el nodo'. Esta condición de estado se genera para los cambios de tamaño en línea y fuera de línea. Para un redimensionamiento en línea, esta condición de estado desaparece sin acción del usuario después de que se complete el redimensionamiento en línea.
- Si la PVC tiene una condición de suceso o estado que indica un cambio de tamaño fallido, consulte [Recuperación de error al expandir volúmenes](#) en la documentación de Red Hat OpenShift.

- d) Compruebe que los pods del gestor de colas no tengan condiciones de estado o sucesos que sugieran un cambio de tamaño fallido. Para despliegues de alta disponibilidad, compruebe cada réplica.

```
oc describe pod queue-manager-pod-name
```


- Si el pod tiene una condición de suceso o estado que indica un cambio de tamaño fallido, consulte [Recuperación de anomalía al expandir volúmenes](#) en la documentación de Red Hat OpenShift . El texto del error puede ayudarle a resolver el problema, o evitar que se produzca el mismo problema si intenta redimensionar de nuevo después de la recuperación.

4. Reiniciar pods al cambiar el tamaño fuera de línea

Si el proveedor de almacenamiento utiliza un procedimiento de redimensionamiento fuera de línea al expandir volúmenes, para que la expansión de volumen se complete, es necesario reiniciar los pods del gestor de colas que montan los volúmenes que se están redimensionando.

Para gestores de colas de varias instancias, los registros de recuperación y los volúmenes de datos persistentes se comparten entre los pods activo y en espera. Para que se complete el redimensionamiento de estos volúmenes, descienda ambos pods al mismo tiempo.

Consulte la documentación del proveedor de almacenamiento para conocer su procedimiento de redimensionamiento fuera de línea.

Detención de un gestor de colas (mq.ibm.com/stop)

Detenga un gestor de colas añadiendo una anotación a la definición del gestor de colas.

Acerca de esta tarea

Los gestores de colas creados por el operador IBM MQ tienen un `StatefulSet` asociado. Este `StatefulSet` declara el número de Pods que se van a desplegar para un tipo de disponibilidad de gestor de colas determinado a través del campo `.replicas`. Toma el valor de 1 (Instancia única), 2 (Multiinstancia) o 3 (NativeHA).

Nota: Cambiar manualmente el valor en el campo `.replicas` impide que el gestor de colas funcione correctamente.

En algunos casos, es posible que desee detener el gestor de colas para que el `StatefulSet` tenga un recuento de réplicas de 0 y no se despliegue ningún Pods . Algunos ejemplos de cuándo es posible que desee hacerlo incluyen durante el mantenimiento o un procedimiento de copia de seguridad.

Nota: Puesto que no hay ningún gestor de colas Pods desplegado cuando se detiene el gestor de colas, usted y sus aplicaciones no podrán acceder al gestor de colas hasta que se vuelva a iniciar.

Procedimiento

- Para detener el gestor de colas, añada la anotación siguiente a la definición del gestor de colas en la sección `.metadata.annotations`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: my-qm
  annotations:
    "mq.ibm.com/stop" : "true"
```

- Para reiniciar el gestor de colas y devolverlo a su número correcto de réplicas, elimine la anotación del gestor de colas o establezca su valor en `'false'`.

Despliegue y configuración de gestores de colas utilizando Helm

Puede desplegar y configurar un gestor de colas en Kubernetes utilizando el diagrama Helm de ejemplo.

Acerca de esta tarea

Si no está utilizando Red Hat OpenShift Container Platform, no se da soporte a IBM MQ Operator . Puede utilizar el diagrama Helm de ejemplo para desplegar en otros tipos de clústeres de Kubernetes .

Procedimiento

- Para obtener información sobre cómo utilizar Helm para desplegar su propia imagen de contenedor de IBM MQ , consulte [Ejemplo IBM MQ Helm chart](#)

Referencia relacionada

“Soporte para IBM MQ en contenedores” en la página 8

No todas las características de IBM MQ están disponibles y soportadas de la misma forma en contenedores.

OpenShift

CP4I-SC2

CD

Migración a IBM MQ Operator

Este conjunto de temas describe los pasos clave para migrar un gestor de colas IBM MQ existente a un entorno de contenedor utilizando IBM MQ Operator en Red Hat OpenShift Container Platform.

Acerca de esta tarea

Los clientes que despliegan IBM MQ en Red Hat OpenShift se pueden separar en los siguientes escenarios:

1. Creación de un nuevo despliegue de IBM MQ en Red Hat OpenShift para nuevas aplicaciones.
2. Ampliación de una red IBM MQ en Red Hat OpenShift para aplicaciones nuevas en Red Hat OpenShift.
3. Mover un despliegue de IBM MQ a Red Hat OpenShift para continuar dando soporte a las aplicaciones existentes.

Sólo para el escenario 3 es necesario migrar la configuración de IBM MQ . Los otros escenarios se consideran nuevos despliegues.

Este conjunto de temas se centra en el escenario 3 y describe los pasos clave para migrar un gestor de colas IBM MQ existente a un entorno de contenedor utilizando IBM MQ Operator. Debido a la flexibilidad y al uso extensivo de IBM MQ, hay varios pasos opcionales. Cada uno de ellos incluye una sección "¿Necesito hacer esto?". La verificación de su necesidad debería ahorrarle tiempo durante la migración.

También debe tener en cuenta qué datos migrar:

1. Migre IBM MQ con la misma configuración pero sin mensajes en cola existentes.
2. Migre IBM MQ con la misma configuración y los mismos mensajes existentes.

Una migración de versión a versión típica puede utilizar cualquiera de los métodos. En un gestor de colas IBM MQ típico en el punto de migración, hay pocos mensajes almacenados en colas, lo que hace que la opción 1 sea adecuada para muchos casos. En el caso de la migración a una plataforma de contenedor es aún más común utilizar la opción 1, para reducir la complejidad de la migración y permitir un despliegue verde azul. Por lo tanto, las instrucciones se centran en este escenario.

El objetivo de este escenario es crear un gestor de colas en el entorno de contenedor que coincida con la definición del gestor de colas existente. Esto permite que las aplicaciones conectadas a la red existentes simplemente se reconfiguren para que apunten al nuevo gestor de colas, sin cambiar ninguna otra configuración o lógica de aplicación.

A lo largo de esta migración, generará varios archivos de configuración que se aplicarán al nuevo gestor de colas. Para simplificar la gestión de estos archivos, debe crear un directorio y generarlos en ese directorio.

Procedimiento

1. [“Comprobación de que las funciones necesarias están disponibles” en la página 107](#)
2. [“Extracción de la configuración del gestor de colas” en la página 107](#)
3. Opcional: [“Opcional: Extracción y adquisición de las claves y certificados del gestor de colas” en la página 108](#)
4. Opcional: [“Opcional: Configuración de LDAP” en la página 110](#)

5. [Opcional: “Opcional: Cambio de las direcciones IP y los nombres de host en la configuración de IBM MQ” en la página 118](#)
6. [“Actualización de la configuración del gestor de colas para un entorno de contenedor” en la página 119](#)
7. [“Selección de la arquitectura HA de destino para IBM MQ que se ejecuta en contenedores” en la página 122](#)
8. [“Creación de los recursos para el gestor de colas” en la página 123](#)
9. [“Creación del nuevo gestor de colas en Red Hat OpenShift” en la página 124](#)
10. [“Verificación del nuevo despliegue de contenedor” en la página 128](#)

Comprobación de que las funciones necesarias están disponibles

IBM MQ Operator no incluye todas las características disponibles en IBM MQ Advanced, y debe verificar que estas características no son necesarias. Otras características están parcialmente soportadas y se pueden volver a configurar para que coincidan con lo que está disponible en el contenedor.

Antes de empezar

Este es el primer paso en [“Migración a IBM MQ Operator” en la página 106](#).

Procedimiento

1. Verifique que la imagen de contenedor de destino incluye todas las funciones necesarias.
Para obtener la información más reciente, consulte [“Cómo utilizar IBM MQ en contenedores” en la página 8](#).
2. El IBM MQ Operator tiene un único puerto de tráfico de IBM MQ , conocido como escucha. Si tiene varios escuchas, simplifíquelo para utilizar un único escucha en el contenedor. Debido a que este no es un escenario común, esta modificación no se documenta en detalle.
3. Si se utilizan salidas de IBM MQ , migralas al contenedor mediante capas en los binarios de salida de IBM MQ . Se trata de un escenario de migración avanzada y, por lo tanto, no se incluye aquí.
Para obtener un esquema de los pasos, consulte [“Creación de una imagen con archivos MQSC e INI personalizados, utilizando la CLI de Red Hat OpenShift” en la página 94](#).
4. Si el sistema IBM MQ incluye alta disponibilidad, revise las opciones disponibles.
Consulte [“Planificación de la alta disponibilidad para IBM MQ en contenedores” en la página 19](#).

Qué hacer a continuación

Ahora está preparado para [extraer la configuración del gestor de colas](#).

Extracción de la configuración del gestor de colas

La mayoría de la configuración es portable entre gestores de colas. Por ejemplo, las cosas con las que interactúan las aplicaciones, como las definiciones de colas, temas y canales. Utilice esta tarea para extraer la configuración del gestor de colas IBM MQ existente.

Antes de empezar

Esta tarea presupone que ha [comprobado que las funciones necesarias están disponibles](#).

Procedimiento

1. Inicie sesión en la máquina con la instalación de IBM MQ existente.
2. Realice una copia de seguridad de la configuración.

Ejecute el siguiente mandato:

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

Notas de uso para este mandato:

- Este mandato almacena la copia de seguridad en el directorio tmp . Puede almacenar la copia de seguridad en otra ubicación, pero este escenario presupone el directorio tmp para los mandatos posteriores.
- Sustituya *QMGR_NAME* por el nombre del gestor de colas del entorno. Si no está seguro del valor, ejecute el mandato **dspmq** para ver los gestores de colas disponibles en la máquina. A continuación se muestra una salida de mandato **dspmq** de ejemplo para un gestor de colas denominado qm1:

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

El mandato **dspmq** requiere que se inicie el gestor de colas IBM MQ ; de lo contrario, recibirá el error siguiente:

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

Si es necesario, inicie el gestor de colas ejecutando el mandato siguiente:

```
strmqm QMGR_NAME
```

Qué hacer a continuación

Ahora está preparado para [extraer y adquirir las claves y certificados del gestor de colas](#).

Opcional: Extracción y adquisición de las claves y certificados del gestor de colas

IBM MQ se puede configurar para cifrar el tráfico de red en el gestor de colas con TLS. Utilice esta tarea para verificar que el gestor de colas está utilizando TLS, para extraer claves y certificados y para configurar TLS en el gestor de colas migrado.

Antes de empezar

Esta tarea presupone que ha [extraído la configuración del gestor de colas](#).

Acerca de esta tarea

¿Necesito hacer esto?

IBM MQ se puede configurar para cifrar el tráfico en el gestor de colas. Este cifrado se completa utilizando un repositorio de claves que está configurado en el gestor de colas. A continuación, los canales IBM MQ habilitan la comunicación TLS. Si no está seguro de si la comunicación TLS está configurada en el entorno, ejecute el mandato siguiente para verificar:

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

Si no se encuentran resultados, no se está utilizando TLS. Sin embargo, esto no significa que TLS no deba configurarse en el gestor de colas migrado. Hay varias razones por las que es posible que desee cambiar este comportamiento:

- El enfoque de seguridad en el entorno de Red Hat OpenShift debe mejorarse en comparación con el entorno anterior.
- Si necesita acceder al gestor de colas migrado desde fuera del entorno de Red Hat OpenShift , es necesario que TLS pase a través de la ruta de Red Hat OpenShift .

Nota: Los certificados de gestor de colas con el mismo nombre distinguido (DN) de sujeto que el certificado de emisor (CA) no están soportados. Un certificado debe tener un nombre distinguido de asunto exclusivo. El producto comprueba que los DN no son iguales.

Procedimiento

1. Extraiga los certificados de confianza del almacén existente.

Si TLS está actualmente en uso en el gestor de colas, es posible que el gestor de colas tenga almacenados varios certificados de confianza. Es necesario extraerlos y copiarlos en el nuevo gestor de colas. Realice uno de los siguientes pasos opcionales:

- Para agilizar la extracción de los certificados, ejecute el script siguiente en el sistema local:

```
#!/bin/bash
keyr=$(grep SSLKEYR $1)
if [ -n "${keyr}" ]; then
  keyrlocation=$(sed -n "s/^\.*'\(.*\)'.*\$/\1/ p" <<< ${keyr})
  mapfile -t runmqakmResult < <(runmqakm -cert -list -db ${keyrlocation}.kdb -stashed)
  cert=1
  for i in "${runmqakmResult[@]:2}"
  do
    certlabel=$(echo ${i:2} | xargs)
    echo Extracting certificate $certlabel to $cert.cert
    runmqakm -cert -extract -db ${keyrlocation}.kdb -label "$certlabel" -target $
  {cert}.cert -stashed
    cert=$((cert+1))
  done
fi
```

Al ejecutar el script, especifique la ubicación de la copia de seguridad de IBM MQ como argumento y se extraerán los certificados. Por ejemplo, si el script se denomina `extractCert.sh` y la copia de seguridad de IBM MQ se encuentra en `/tmp/backup.mqsc`, ejecute el mandato siguiente:

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

- De forma alternativa, ejecute los mandatos siguientes en el orden que se muestra:

a. Identifique la ubicación del repositorio de claves TLS del gestor de colas:

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

Salida de ejempl:

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

donde se encuentra el almacén de claves en `/run/runmqserver/tls/key.kdb`

b. Basándose en esta información de ubicación, consulte el almacén de claves para determinar los certificados almacenados:

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

Salida de ejempl:

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

c. Extraiga cada uno de los certificados listados. Para ello, ejecute el mandato siguiente:

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

En los ejemplos mostrados anteriormente, esto equivale a los mandatos siguientes:

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

2. Adquirir una nueva clave y certificado para el gestor de colas

Para configurar TLS en el gestor de colas migrado, genere una nueva clave y certificado. A continuación, se utiliza durante el despliegue. En muchas organizaciones, esto significa ponerse en contacto con el equipo de seguridad para solicitar una clave y un certificado. En algunas organizaciones, esta opción no está disponible y se utilizan certificados autofirmados.

El ejemplo siguiente genera un certificado autofirmado en el que la caducidad se establece en 10 años:

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

Se crean dos archivos nuevos:

- qmgr.key es la clave privada para el gestor de colas
- qmgr.crt es el certificado público

Qué hacer a continuación

Ahora está preparado para [configurar LDAP](#).

OpenShift

CP4I-SC2

CD

Opcional: Configuración de LDAP

El IBM MQ Operator se puede configurar para utilizar varios enfoques de seguridad diferentes. Normalmente, LDAP es el más eficaz para un despliegue de empresa, y LDAP se utiliza para este escenario de migración.

Antes de empezar

Esta tarea presupone que ha [extraído y adquirido las claves y certificados del gestor de colas](#).

Acerca de esta tarea

¿Necesito hacer esto?

Si ya está utilizando LDAP para la autenticación y autorización, no es necesario realizar ningún cambio.

Si no está seguro de si se está utilizando LDAP, ejecute el mandato siguiente:

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

Salida de ejemplo:

```
DEFINE AUTHINFO('USE.LDAP') +
  AUTHTYPE(IDPWLDAP) +
  ADOPTCTX(YES) +
  CONNAME('ldap-service.ldap(389)') +
  CHCKCLNT(REQUIRED) +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  * LDAPPWD('*****') +
  SHORTUSR('uid') +
```

```
GRPFIELD('cn') +
USRFIELD('uid') +
AUTHORMD(SEARCHGRP) +
* ALTDATA(2020-11-26) +
* ALTTIME(15.44.38) +
REPLACE
```

Hay dos atributos en la salida que son de interés particular:

AUTHTYPE

Si tiene el valor IDPWLDAP, está utilizando LDAP para la autenticación.

Si el valor está en blanco, u otro valor, LDAP no está configurado. En este caso, compruebe el atributo AUTHORMD para ver si los usuarios de LDAP se están utilizando para la autorización.

AUTHORMD

Si tiene el valor OS, no está utilizando LDAP para la autorización.

Para modificar la autorización y la autenticación para utilizar LDAP, realice las tareas siguientes:

Procedimiento

1. Actualice la copia de seguridad de IBM MQ para el servidor LDAP.
2. Actualice la copia de seguridad de IBM MQ para la información de autorización LDAP.

OpenShift CP4I-SC2 CD Parte 1 de LDAP: Actualización de la copia de seguridad de IBM MQ para el servidor LDAP

Una descripción completa de cómo configurar LDAP está fuera del ámbito de este escenario. Este tema proporciona un resumen del proceso, un ejemplo y referencias a más información.

Antes de empezar

Esta tarea presupone que ha extraído y adquirido las claves y certificados del gestor de colas.

Acerca de esta tarea

¿Necesito hacer esto?

Si ya está utilizando LDAP para la autenticación y autorización, no es necesario realizar ningún cambio. Si no está seguro de si se está utilizando LDAP, consulte [“Opcional: Configuración de LDAP”](#) en la página 110.

Hay dos partes para configurar el servidor LDAP:

1. [Definir una configuración LDAP](#).
2. [Asocie la configuración LDAP con la definición del gestor de colas](#).

Más información para ayudarle con esta configuración:

- [Visión general del repositorio de usuarios](#)
- [Guía de referencia para el mandato AUTHINFO](#)

Procedimiento

1. Defina una configuración LDAP.

Edite el archivo backup.mqsc para definir un nuevo objeto **AUTHINFO** para el sistema LDAP. Por ejemplo:

```
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
```

```

SECCOMM(NO) +
USRFIELD('uid') +
SHORTUSR('uid') +
BASEDNU('ou=people,dc=ibm,dc=com') +
AUTHORMD(SEARCHGRP) +
BASEDNG('ou=groups,dc=ibm,dc=com') +
GRPFIELD('cn') +
CLASSGRP('groupOfUniqueNames') +
FINDGRP('uniqueMember')
REPLACE

```

donde

- **CONNNAME** es el nombre de host y el puerto correspondientes al servidor LDAP. Si existen varias direcciones para la resiliencia, estas se pueden configurar utilizando una lista separada por comas.
- **LDAPUSER** es el nombre distinguido correspondiente al usuario que IBM MQ utiliza al conectarse a LDAP para consultar registros de usuario.
- **LDAPPWD** es la contraseña que corresponde al usuario **LDAPUSER**.
- **SECCOM** especifica si la comunicación con el servidor LDAP debe utilizar TLS. Valores posibles:
 - YES: se utiliza TLS y el servidor de IBM MQ presenta un certificado.
 - ANON: TLS se utiliza sin que el servidor de IBM MQ presente un certificado.
 - NO: TLS no se utiliza durante la conexión.
- **USRFIELD** especifica el campo en el registro LDAP con el que se compara el nombre de usuario presentado.
- **SHORTUSR** es un campo dentro del registro LDAP que no supera los 12 caracteres de longitud. El valor de este campo será la identidad certificada si la autenticación es satisfactoria.
- **BASEDNU** es el DN base que se debe utilizar para buscar LDAP.
- **BASEDNG** es el DN base para grupos dentro de LDAP.
- **AUTHORMD** define el mecanismo utilizado para resolver la pertenencia a grupos para el usuario. Hay cuatro opciones:
 - SO: Consultar el sistema operativo para los grupos asociados con el nombre abreviado.
 - SEARCHGRP: busque el usuario autenticado en las entradas de grupo en LDAP.
 - SEARCHUSR: buscar información de pertenencia a grupos en el registro de usuario autenticado.
 - SRCHGRPSN: busque en las entradas de grupo en LDAP el nombre de usuario abreviado de los usuarios autenticados (definido por el campo SHORTUSR).
- **GRPFIELD** es el atributo dentro del registro de grupo LDAP que corresponde a un nombre simple. Si se especifica, se puede utilizar para definir registros de autorización.
- **CLASSUSR** es la clase de objeto LDAP que corresponde a un usuario.
- **CLASSGRP** es la clase de objeto LDAP que corresponde a un grupo.
- **FINDGRP** es el atributo dentro del registro LDAP que corresponde a la pertenencia a grupos.

La nueva entrada se puede colocar en cualquier lugar del archivo, sin embargo, puede resultarle útil tener nuevas entradas al principio del archivo:


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

2. Asocie la configuración LDAP con la definición del gestor de colas.

Debe asociar la configuración LDAP con la definición del gestor de colas. Inmediatamente debajo de la entrada DEFINE AUTHINFO hay una entrada ALTER QMGR . Modifique la entrada CONNAUTH para que se corresponda con el nombre AUTHINFO recién creado. Por ejemplo, en el ejemplo anterior se ha definido AUTHINFO(USE.LDAP) , lo que significa que el nombre es USE.LDAP. Por lo tanto, cambie CONNAUTH('SYSTEM.DEFAULT.AUTHINFO.IDPWOS') por CONNAUTH('USE.LDAP'):

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

Para que el conmutador a LDAP se produzca inmediatamente, llame a un mandato REFRESH SECURITY añadiendo una línea inmediatamente después del mandato ALTER QMGR :

```

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

```

Qué hacer a continuación

Ahora está preparado para actualizar la copia de seguridad de IBM MQ para la información de autorización LDAP.

Parte 2 de LDAP: Actualización de la copia de seguridad de IBM MQ para la información de autorización de LDAP

IBM MQ proporciona reglas de autorización detalladas que controlan el acceso a los objetos de IBM MQ . Si ha cambiado la autenticación y autorización a LDAP, es posible que las reglas de autorización no sean válidas y requieran actualización.

Antes de empezar

Esta tarea presupone que ha [actualizado la copia de seguridad para el servidor LDAP](#).

Acerca de esta tarea

¿Necesito hacer esto?

Si ya está utilizando LDAP para la autenticación y autorización, no es necesario realizar ningún cambio. Si no está seguro de si se está utilizando LDAP, consulte [“Opcional: Configuración de LDAP”](#) en la página 110.

Hay dos partes para actualizar la información de autorización LDAP:

1. [Elimine todas las autorizaciones existentes del archivo](#).
2. [Definir nueva información de autorización para LDAP](#).

Procedimiento

1. Elimine todas las autorizaciones existentes del archivo.

En el archivo de copia de seguridad, cerca del final del archivo, debería ver varias entradas que empiezan por SET AUTHREC:


```

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

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

```

Busque las entradas existentes y suprimalas. El enfoque más sencillo es eliminar todas las reglas de SET AUTHREC existentes y, a continuación, crear nuevas entradas basadas en las entradas LDAP.

2. Definir nueva información de autorización para LDAP

En función de la configuración del gestor de colas y del número de recursos y grupos, esto puede ser una actividad que consume mucho tiempo o sencilla. En el ejemplo siguiente se presupone que el gestor de colas sólo tiene una única cola denominada Q1y que desea permitir que el grupo LDAP apps tenga acceso.

```

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

```

El primer mandato AUTHREC añade permiso para acceder al gestor de colas y el segundo proporciona acceso a la cola. Si es necesario acceder a una segunda cola, se necesita un tercer mandato AUTHREC , a menos que haya decidido utilizar comodines para proporcionar un acceso más genérico.

Este es otro ejemplo. Si un grupo de administradores (denominado admins) necesita acceso completo al gestor de colas, añada los mandatos siguientes:

```

SET AUTHREC PROFILE('*') OBJTYPE(Queue) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Topic) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Channel) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(CLNTCONN) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(AUTHINFO) GROUP('admins') AUTHADD(ALL)

```

```
SET AUTHREC PROFILE('*') OBJTYPE(LISTENER) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(NAMELIST) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(PROCESS) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(SERVICE) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(QMGR) GROUP('admins') AUTHADD(ALL)
```

Qué hacer a continuación

Ahora está preparado para cambiar las direcciones IP y los nombres de host en la configuración de IBM MQ.

Opcional: Cambio de las direcciones IP y los nombres de host en la configuración de IBM MQ

La configuración de IBM MQ puede tener direcciones IP y nombres de host especificados. En algunas situaciones estas pueden permanecer, mientras que en otras situaciones es necesario actualizarlas.

Antes de empezar

Esta tarea presupone que ha configurado LDAP.

Acerca de esta tarea

¿Necesito hacer esto?

En primer lugar, determine si tiene direcciones IP o nombres de host especificados, aparte de la configuración LDAP definida en la sección anterior. Para ello, ejecute el siguiente mandato:

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

Salida de ejempl:

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

En este ejemplo, la búsqueda devuelve tres resultados. Un resultado corresponde a la configuración LDAP definida anteriormente. Esto se puede ignorar, porque el nombre de host del servidor LDAP sigue siendo el mismo. Los otros dos resultados son entradas de conexión vacías, por lo que también se pueden ignorar. Si no tiene ninguna entrada adicional, puede omitir el resto de este tema.

Procedimiento

1. Comprender las entradas devueltas.

IBM MQ puede incluir direcciones IP, nombres de host y puertos en muchos aspectos de la configuración. Podemos clasificarlos en dos categorías:

- a. **Ubicación de este gestor de colas:** información de ubicación que este gestor de colas utiliza o publica, que otros gestores de colas o aplicaciones de una red IBM MQ pueden utilizar para la conectividad.

- b. **Ubicación de dependencias de gestor de colas:** las ubicaciones de otros gestores de colas o sistemas que este gestor de colas necesita conocer.

Puesto que este escenario sólo se centra en los cambios en esta configuración del gestor de colas, sólo manejamos las actualizaciones de configuración para la categoría (a). Sin embargo, si otros gestores de colas o aplicaciones hacen referencia a esta ubicación de gestor de colas, es posible que sea necesario actualizar sus configuraciones para que coincidan con la nueva ubicación de este gestor de colas.

Hay dos objetos clave que pueden contener información que se debe actualizar:

- Escuchas: representan la dirección de red en la que escucha IBM MQ .
 - Canal CLUSTER RECEIVER: si el gestor de colas forma parte de un clúster de IBM MQ , este objeto existe. Especifica la dirección de red a la que se pueden conectar otros gestores de colas.
2. En la salida original del mandato `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` , identifique si hay algún canal CLUSTER RECEIVER definido. Si es así, actualice las direcciones IP.

Para identificar si hay algún canal CLUSTER RECEIVER definido, busque las entradas con CHLTYPE (CLUSRCVR) en la salida original:

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

Si existen entradas, actualice CONNAME con la ruta de IBM MQ Red Hat OpenShift . Este valor se basa en el entorno de Red Hat OpenShift y utiliza una sintaxis predecible:

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

Por ejemplo, si el despliegue del gestor de colas se denomina qm1 en el espacio de nombres cp4i y `openshift_app_route_hostname` es `apps.callumj.icp4i.com` , el URL de ruta es el siguiente:

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

El número de puerto para la ruta suele ser 443. A menos que el administrador de Red Hat OpenShift le indique lo contrario, este es normalmente el valor correcto. Utilizando esta información, actualice los campos CONNAME . Por ejemplo:

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

En la salida original del mandato `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` , verifique si existen entradas para LOCLADDR o IPADDRV. Si lo hacen, supármalos. No son relevantes en un entorno de contenedor.

Qué hacer a continuación

Ahora está preparado para [actualizar la configuración del gestor de colas para un entorno de contenedor](#).

Actualización de la configuración del gestor de colas para un entorno de contenedor

Cuando se ejecuta en un contenedor, el contenedor define determinados aspectos de configuración y puede entrar en conflicto con la configuración exportada.

Antes de empezar

Esta tarea presupone que ha [cambiado la configuración de IBM MQ de direcciones IP y nombres de host](#).

Acerca de esta tarea

El contenedor define los siguientes aspectos de configuración:

- Las definiciones de escucha (que corresponden a los puertos expuestos).
- La ubicación de cualquier almacén TLS potencial.

Por lo tanto, debe actualizar la configuración exportada:

1. Elimine las definiciones de escucha.
2. Defina la ubicación del repositorio de claves TLS.

Procedimiento

1. Elimine las definiciones de escucha.

En la configuración de copia de seguridad, busque DEFINE LISTENER. Debe estar entre las definiciones AUTHINFO y SERVICE . Resalte el área y suprimala.


```

*backup.mqsc
** ALTDATE(2020-11-20) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.CRLLDAP') +
  AUTHTYPE(CRLLDAP) +
  CONNAME(' ') +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.LU62') +
  TRPTYPE(LU62) +
  CONTROL(MANUAL) +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.NETBIOS') +
  TRPTYPE(NETBIOS) +
  CONTROL(MANUAL) +
  LOCLNAME(' ') +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.SPX') +
  TRPTYPE(SPX) +
  CONTROL(MANUAL) +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.TCP') +
  TRPTYPE(TCP) +
  CONTROL(MANUAL) +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE SERVICE('SYSTEM.AMQP.SERVICE') +
  CONTROL(QMGR) +
  SERVTYPE(SERVER) +
  STARTCMD('+MQ_INSTALL_PATH+\bin\amqp.bat') +
  STARTARG('start -m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\.'
  STOPCMD('+MQ_INSTALL_PATH+\bin64\endmqsd.exe') +

```

2. Defina la ubicación del repositorio de claves TLS.

La copia de seguridad del gestor de colas contiene la configuración TLS para el entorno original. Esto es diferente del entorno de contenedor y, por lo tanto, se necesitan un par de actualizaciones:

- Cambie la entrada **CERTLABL** por default
- Cambie la ubicación del repositorio de claves TLS (**SSLKEYR**) por: /run/runmqserver/tls/key

Para buscar la ubicación del atributo **SSLKEYR** en el archivo, busque **SSLKEYR**. Normalmente, sólo se encuentra una entrada. Si se encuentran varias entradas, compruebe que está editando el objeto **QMGR** tal como se muestra en la siguiente ilustración:

```

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

```

Qué hacer a continuación

Ahora está preparado para seleccionar la arquitectura de destino para IBM MQ que se ejecuta en contenedores.

OpenShift CP4I-SC2 CD Selección de la arquitectura HA de destino para IBM MQ que se ejecuta en contenedores

Elija entre una única instancia (un único pod de Kubernetes), varias instancias (dos pods) y HA nativa (un pod de réplica activo y dos pods de réplica en espera) para cumplir los requisitos de alta disponibilidad.

Antes de empezar

Esta tarea presupone que ha [actualizado la configuración del gestor de colas para un entorno de contenedor](#).

Acerca de esta tarea

IBM MQ Operator proporciona tres opciones de alta disponibilidad:

- **Instancia única:** se inicia un contenedor único (Pod) y es responsabilidad de Red Hat OpenShift reiniciarse en caso de anomalía. Debido a las características de un conjunto con estado dentro de Kubernetes, hay varias situaciones en las que esta migración tras error puede tardar un periodo de tiempo prolongado o puede requerir que se complete una acción administrativa.
- **Varias instancias:** se inician dos contenedores (cada uno en un Pod independiente), uno en modalidad activa y otro en espera. Esta topología permite una migración tras error mucho más rápida. Requiere un sistema de archivos Read Write Many que cumpla los requisitos de IBM MQ .
- **HA nativa:** tres contenedores (cada uno en un pod independiente), cada uno con una instancia del gestor de colas. Una instancia es el gestor de colas activo, procesando mensajes y grabando en su registro de recuperación. Siempre que se graba el registro de recuperación, el gestor de colas activo envía los datos a las otras dos instancias, conocidas como réplicas. Si el pod que ejecuta el gestor de colas activo falla, una de las instancias de réplica del gestor de colas toma el control del rol activo y tiene datos actuales con los que operar.

En esta tarea sólo elige la arquitectura HA de destino. Los pasos para configurar la arquitectura elegida se describen en una tarea posterior en este escenario ([“Creación del nuevo gestor de colas en Red Hat OpenShift”](#) en la página 124).

Procedimiento

1. Revise las tres opciones.

Para obtener una descripción completa de estas opciones, consulte [“Planificación de la alta disponibilidad para IBM MQ en contenedores”](#) en la página 19.

2. Seleccione la arquitectura HA de destino.

Si no está seguro de qué opción elegir, empiece con la opción **Instancia única** y verifique si cumple los requisitos de alta disponibilidad.

Qué hacer a continuación

Ahora está preparado para [crear los recursos del gestor de colas](#).

Creación de los recursos para el gestor de colas

Importe la configuración de IBM MQ y los certificados y claves TLS en el entorno de Red Hat OpenShift .

Antes de empezar

Esta tarea presupone que ha [seleccionado la arquitectura de destino para IBM MQ que se ejecuta en contenedores](#).

Acerca de esta tarea

En las secciones anteriores ha extraído, actualizado y definido dos recursos:

- Configuración de IBM MQ
- Claves y certificados TLS

Debe importar estos recursos en el entorno de Red Hat OpenShift antes de desplegar el gestor de colas.

Procedimiento

1. Importe la configuración de IBM MQ en Red Hat OpenShift.

En las instrucciones siguientes se presupone que tiene la configuración de IBM MQ en el directorio actual, en un archivo denominado `backup.mqsc`. De lo contrario, debe personalizar el nombre de archivo en función del entorno.

- a) Inicie sesión en el clúster utilizando `oc login`.
- b) Cargue la configuración de IBM MQ en un configmap.

Ejecute el siguiente mandato:

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

- c) Verifique que el archivo se ha cargado correctamente.

Ejecute el siguiente mandato:

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

2. Importar los recursos TLS de IBM MQ

Tal como se describe en [“Opcional: Extracción y adquisición de las claves y certificados del gestor de colas”](#) en la [página 108](#), es posible que sea necesario TLS para el despliegue del gestor de colas. Si es así, ya debería tener un número de archivos que terminen en `.crt` y `.key`. Debe añadirlos a los secretos de Kubernetes para que el gestor de colas haga referencia a ellos en el momento del despliegue.

Por ejemplo, si tenía una clave y un certificado para el gestor de colas, se podría llamar a ellos:

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

Para importar estos archivos, ejecute el mandato siguiente:

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

Kubernetes proporciona este útil programa de utilidad cuando se importa una clave pública y privada coincidente. Si tiene certificados adicionales para añadir, por ejemplo en el almacén de confianza del gestor de colas, ejecute el mandato siguiente:

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

Por ejemplo, si los archivos que se van a importar son `trust1.crt`, `trust2.crt` y `trust3.crt`, el mandato es el siguiente:

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

Qué hacer a continuación

Ahora está preparado para [crear el nuevo gestor de colas en Red Hat OpenShift](#).

Creación del nuevo gestor de colas en Red Hat OpenShift

Despliegue una sola instancia o un gestor de colas de varias instancias en Red Hat OpenShift.

Antes de empezar

Esta tarea presupone que ha creado los recursos del gestor de colasy que ha instalado IBM MQ Operator en Red Hat OpenShift.

Acerca de esta tarea

Tal como se describe en “Selección de la arquitectura HA de destino para IBM MQ que se ejecuta en contenedores” en la página 122, hay tres topologías de despliegue posibles. Por lo tanto, este tema proporciona tres plantillas diferentes:

- Plantilla 1: Desplegar un gestor de colas de una sola instancia.
- Plantilla 2: Desplegar un gestor de colas de varias instancias.
- Plantilla 3: Desplegar un gestor de colas HA nativo.

Importante: Solo complete una de las tres plantillas, basándose en la topología que prefiera.

Procedimiento

- **Plantilla 1: Desplegar un gestor de colas de una sola instancia.**

El gestor de colas migrado se despliega en Red Hat OpenShift utilizando un archivo YAML. A continuación se muestra un ejemplo, basado en los nombres utilizados en los temas anteriores:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
spec:
  version: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLRL
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
          items:
            - tls.key
            - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
  mqsc:
    - configMap:
        name: my-mqsc-migrated
        items:
          - backup.mqsc
```

En función de los pasos que haya realizado, es posible que sea necesario personalizar el YAML anterior. Para ayudarle con esto, aquí está una explicación de este YAML:

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

Define el objeto, tipo y nombre de Kubernetes . El único campo que requiere personalización es el campo name .

```
spec:
  version: 9.4.0.0-r1
  license:
    accept: true
```

```
license: L-BMSF-5YDSLRL
use: "Production"
```

Esto corresponde a la información de versión y licencia para el despliegue. Si necesita personalizar esto, utilice la información proporcionada en [“Referencia de licencia para mq.ibm.com/v1beta1”](#) en la [página 140](#).

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

Para que el gestor de colas se configure para utilizar TLS, debe hacer referencia a los certificados y claves relevantes. El campo `secretName` hace referencia al secreto Kubernetes creado en la sección [Importar los recursos de IBM MQ TLS](#) y la lista de elementos (`tls.key` y `tls.crt`) son los nombres estándar que Kubernetes asigna cuando se utiliza la sintaxis `oc create secret tls`. Si tiene certificados adicionales para añadir al almacén de confianza, estos se pueden añadir de forma similar, pero los elementos son los nombres de archivo correspondientes utilizados durante la importación. Por ejemplo, se puede utilizar el código siguiente para crear los certificados de almacén de confianza:

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

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

Importante: Si TLS no es necesario, suprima la sección TLS del YAML.

```
web:
  enabled: true
```

Esto habilita la consola web para el despliegue

```
queueManager:
  name: QM1
```

Define el nombre del gestor de colas como QM1. El gestor de colas se personaliza en función de sus requisitos, por ejemplo, cuál era el nombre original del gestor de colas.

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

El código anterior extrae la configuración del gestor de colas que se ha importado en la sección [Importar la configuración de IBM MQ](#). Si ha utilizado nombres diferentes, debe modificar `my-mqsc-migrated` y `backup.mqsc`.

Tenga en cuenta que el YAML de ejemplo presupone que la clase de almacenamiento predeterminada para el entorno Red Hat OpenShift está definida como una clase de almacenamiento RWX o RWO. Si no se ha definido un valor predeterminado en el entorno, debe especificar la clase de almacenamiento que se va a utilizar. Puede hacerlo ampliando el YAML como se indica a continuación:

```
queueManager:
  name: QM1
```

```
storage:
  defaultClass: my_storage_class
  queueManager:
    type: persistent-claim
```

Añada el texto resaltado, con el atributo de clase personalizado para que coincida con su entorno. Para descubrir los nombres de clase de almacenamiento dentro del entorno, ejecute el mandato siguiente:

```
oc get storageclass
```

A continuación se muestra una salida de ejemplo devuelta por este mandato:

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

El código siguiente muestra cómo hacer referencia a la configuración de IBM MQ que se ha importado en la sección [Importar la configuración de IBM MQ](#) . Si ha utilizado nombres diferentes, debe modificar `my-mqsc-migrated` y `backup.mqsc`.

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

Ha desplegado el gestor de colas de instancia única. Esto completa la plantilla. Ahora está preparado para [verificar el nuevo despliegue de contenedor](#).

- **Plantilla 2: Desplegar un gestor de colas de varias instancias.**

El gestor de colas migrado se despliega en Red Hat OpenShift utilizando un archivo YAML. El ejemplo siguiente se basa en los nombres utilizados en las secciones anteriores.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1mi
spec:
  version: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLRL
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
          items:
            - tls.key
            - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    availability: MultiInstance
  storage:
    defaultClass: aws-efs
    persistedData:
      enabled: true
    queueManager:
      enabled: true
    recoveryLogs:
      enabled: true
  mqsc:
    - configMap:
        name: my-mqsc-migrated
      items:
        - backup.mqsc
```


Aquí hay una explicación de este YAML. La mayoría de la configuración sigue el mismo enfoque que [desplegar un gestor de colas de instancia única](#), por lo tanto, aquí solo se explican los aspectos de disponibilidad y almacenamiento del gestor de colas.

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

Especifica el nombre del gestor de colas como QM1 y establece el despliegue en MultiInstance en lugar de la instancia única predeterminada.

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

Un gestor de colas de varias instancias de IBM MQ depende del almacenamiento RWX. De forma predeterminada, un gestor de colas se despliega en modalidad de instancia única y, por lo tanto, se necesitan opciones de almacenamiento adicionales al cambiar a la modalidad de varias instancias. En el ejemplo de YAML anterior, se definen tres volúmenes persistentes de almacenamiento y una clase de volumen persistente. Esta clase de volumen persistente debe ser una clase de almacenamiento RWX. Si no está seguro de los nombres de clase de almacenamiento en el entorno, puede ejecutar el mandato siguiente para descubrirlos:

```
oc get storageclass
```

A continuación se muestra una salida de ejemplo devuelta por este mandato:

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

El código siguiente muestra cómo hacer referencia a la configuración de IBM MQ que se ha importado en la sección [Importar la configuración de IBM MQ](#) . Si ha utilizado nombres diferentes, debe modificar `my-mqsc-migrated` y `backup.mqsc`.

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

Ha desplegado el gestor de colas de varias instancias. Esto completa la plantilla. Ahora está preparado para [verificar el nuevo despliegue de contenedor](#).

- **Plantilla 3: Desplegar un gestor de colas de HA nativa.**

Para ver un ejemplo de creación de un gestor de colas de HA nativa, consulte [“Ejemplo: Configuración de HA nativa utilizando IBM MQ Operator”](#) en la página 76.

Verificación del nuevo despliegue de contenedor

Ahora que IBM MQ se ha desplegado en Red Hat OpenShift, puede verificar el entorno utilizando los ejemplos de IBM MQ .

Antes de empezar

Esta tarea presupone que ha [creado el nuevo gestor de colas en Red Hat OpenShift](#).

Importante: Esta tarea presupone que TLS no está habilitado en el gestor de colas.

Acerca de esta tarea

En esta tarea se ejecutan los ejemplos de IBM MQ desde dentro del contenedor del gestor de colas migrado. Sin embargo, es posible que prefiera utilizar sus propias aplicaciones que se ejecutan desde otro entorno.

Necesita la siguiente información:

- Nombre de usuario LDAP
- Contraseña LDAP
- IBM MQ Nombre de canal
- Nombre de cola

Este código de ejemplo utiliza los valores siguientes. Tenga en cuenta que los valores serán diferentes.

- Nombre de usuario de LDAP: mqapp
- Contraseña LDAP: mqapp
- IBM MQ Nombre de canal: DEV.APP.SVRCONN
- Nombre de cola: Q1

Procedimiento

1. Ejecute en el contenedor IBM MQ en ejecución.

Utilice el mandato siguiente:

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

donde `qm1-ibm-mq-0` es el pod que hemos desplegado en [“Creación del nuevo gestor de colas en Red Hat OpenShift”](#) en la [página 124](#). Si ha llamado al despliegue algo diferente, personalice este valor.

2. Envíe un mensaje.

Ejecute los mandatos siguientes:

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

Se le solicitará una contraseña y, a continuación, podrá enviar un mensaje.

3. Verifique que el mensaje se ha recibido correctamente.

Ejecute el ejemplo GET:

```
./amqsgetc Q1 QM1
```

Resultados

Ha completado la [“Migración a IBM MQ Operator”](#) en la [página 106](#).

Qué hacer a continuación

Utilice la información siguiente para ayudarle con escenarios de migración más complejos:

Migración de mensajes en cola

Para migrar mensajes en cola existentes, siga las instrucciones del tema siguiente para exportar e importar mensajes después de que el nuevo gestor de colas esté en su lugar: [Utilización del programa de utilidad dmpmqmsg entre dos sistemas](#).

Conexión a IBM MQ desde fuera del entorno de Red Hat OpenShift

El gestor de colas desplegado puede estar expuesto a clientes y gestores de colas de IBM MQ fuera del entorno de Red Hat OpenShift . El proceso depende de la versión de IBM MQ que se conecta al entorno de Red Hat OpenShift . Consulte [“Configuración de una ruta para conectarse a un gestor de colas desde fuera de un clúster de Red Hat OpenShift”](#) en la página 85.

Operación de IBM MQ en contenedores

Si necesita trabajar o interactuar con gestores de colas de IBM MQ que se ejecutan en contenedores, consulte los temas siguientes para obtener más información.

Procedimiento

- [“Funcionamiento de IBM MQ utilizando la IBM MQ Operator”](#) en la página 130.
- [“Visualización del estado de los gestores de colas HA nativa”](#) en la página 137.
- [“Finalización manual de instancias de gestor de colas de HA nativa”](#) en la página 140.

Funcionamiento de IBM MQ utilizando la IBM MQ Operator

Procedimiento

- [“Conexión con el IBM MQ Console desplegado en un clúster de Red Hat OpenShift”](#) en la página 130.
- [“Supervisión cuando se utiliza IBM MQ Operator”](#) en la página 131.
- [“Copia de seguridad y restauración de la configuración del gestor de colas utilizando la CLI de Red Hat OpenShift”](#) en la página 137.

Conexión con el IBM MQ Console desplegado en un clúster de Red Hat 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

El URL IBM MQ Console se puede encontrar en la página de detalles de QueueManager en la consola web de Red Hat OpenShift o en IBM Cloud Pak for Integration Platform UI. De forma alternativa, se puede encontrar en la CLI de Red Hat OpenShift ejecutando el mandato siguiente:

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

Si está utilizando una licencia de IBM Cloud Pak for Integration , el IBM MQ Console utiliza Keycloak para la gestión de identidades y accesos. Consulte [Gestión de identidad y acceso](#) en la documentación de IBM Cloud Pak for Integration .

Si está utilizando una licencia de IBM MQ , el IBM MQ Console no está preconfigurado y debe configurarlo usted mismo. Si desea más información, consulte [Configuración de usuarios y roles](#). Para ver un ejemplo, consulte [“Configuración de IBM MQ Console con un registro básico utilizando IBM MQ Operator”](#) en la página 98

Tareas relacionadas

[“Configuración de una ruta para conectarse a un gestor de colas desde fuera de un clúster de Red Hat OpenShift”](#) en la página 85

Necesita una ruta de Red Hat OpenShift para conectar una aplicación a un gestor de colas de IBM MQ desde fuera de un clúster de Red Hat OpenShift . Debe habilitar TLS en el gestor de colas y la aplicación cliente de IBM MQ , porque SNI sólo está disponible en el protocolo TLS cuando se utiliza un protocolo

TLS 1.2 o superior. Red Hat OpenShift Container Platform Router utiliza SNI para direccionar solicitudes al gestor de colas IBM MQ .

OpenShift CP4I **Concesión de permisos para IBM MQ Console**

Los permisos para IBM MQ Console se gestionan de forma diferente en función del uso de licencia.

Acerca de esta tarea

- Si está utilizando una licencia de IBM Cloud Pak for Integration , el IBM MQ Console utiliza Keycloak para la gestión de identidades y accesos.
 - Consulte [Gestión de identidad y acceso](#) en la documentación de IBM Cloud Pak for Integration .
 - Si ha configurado previamente usuarios con IAM en versiones anteriores de IBM MQ Operator, consulte [Migración de usuarios de IAM a Keycloak](#).
- Si está utilizando una licencia de IBM MQ , el IBM MQ Console no está preconfigurado y debe configurarlo usted mismo.
 - Para obtener más información sobre usuarios y roles, consulte [Configuración de usuarios y roles](#).
 - Para ver un ejemplo sencillo, consulte “[Configuración de IBM MQ Console con un registro básico utilizando IBM MQ Operator](#)” en la página 98.
 - De forma alternativa, puede instalar el operador IBM Cloud Pak for Integration para configurar Keycloak como se ha descrito anteriormente.

OpenShift CP4I **Supervisión cuando se utiliza IBM MQ Operator**

Los gestores de colas gestionados por IBM MQ Operator pueden producir métricas compatibles con Prometheus.

Puede ver estas métricas utilizando la [pila de supervisión de Red Hat OpenShift Container Platform \(OCP\)](#). Abra el separador **Métricas** en OCPy, a continuación, pulse **Observar > Métricas**. Las métricas del gestor de colas están habilitadas de forma predeterminada, pero se pueden inhabilitar estableciendo `.spec.metrics.enabled` en `false`.

Prometheus es una base de datos de series temporales y un motor de evaluación de reglas para métricas. Los contenedores de IBM MQ exponen un punto final de métricas que puede consultar Prometheus. Las métricas se generan a partir de los temas del sistema MQ para la supervisión y el rastreo de actividad.

OpenShift Container Platform incluye una pila de supervisión preconfigurada, preinstalada y de actualización automática que utiliza un servidor Prometheus . La pila de supervisión de OpenShift Container Platform debe configurarse para supervisar proyectos definidos por el usuario. Para obtener más información, consulte [Habilitación de la supervisión para proyectos definidos por el usuario](#). El IBM MQ Operator crea un `ServiceMonitor` cuando crea un `QueueManager` con las métricas habilitadas, que el operador Prometheus puede descubrir.

OpenShift CP4I **Métricas publicadas cuando se utiliza IBM MQ Operator**

Los contenedores del gestor de colas pueden publicar métricas compatibles con Red Hat OpenShift Monitoring.

Métrica	Tipo	Descripción
ibmmq_qmgr_commit_total	counter	Recuento de Commit
ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage	gauge	Carga de CPU - promedio de quince minutos

Métrica	Tipo	Descripción
ibmmq_qmgr_cpu_load_five_minute_average_percentage	gauge	Carga de CPU - promedio de cinco minutos
ibmmq_qmgr_cpu_load_one_minute_average_percentage	gauge	Carga de CPU - promedio de un minuto
ibmmq_qmgr_destructive_get_bytes_total	counter	Total de intervalos de obtención destructiva - recuento de bytes
ibmmq_qmgr_destructive_get_total	counter	Total de intervalos de obtención destructiva - recuento
ibmmq_qmgr_durable_subscription_alter_total	counter	Alterar recuento de suscripciones duraderas
ibmmq_qmgr_durable_subscription_create_total	counter	Crear recuento de suscripciones duraderas
ibmmq_qmgr_durable_subscription_delete_total	counter	Suprimir recuento de suscripciones duraderas
ibmmq_qmgr_durable_subscription_resume_total	counter	Reanudar recuento de suscripciones duraderas
ibmmq_qmgr_errors_file_system_free_space_percentage	gauge	Sistema de archivos de errores MQ - espacio libre
ibmmq_qmgr_errors_file_system_in_use_bytes	gauge	Sistema de archivos de errores MQ - bytes en uso
ibmmq_qmgr_expired_message_total	counter	Recuento de mensajes caducados
ibmmq_qmgr_failed_browse_total	counter	Recuento de exámenes anómalo
ibmmq_qmgr_failed_mqcb_total	counter	Recuento de MQCB anómalo
ibmmq_qmgr_failed_mqclose_total	counter	Recuento de MQCLOSE anómalo
ibmmq_qmgr_failed_mqconn_mqconnx_total	counter	Recuento de MQCONN/MQCONNx anómalo
ibmmq_qmgr_failed_mqget_total	counter	MQGET anómalo - recuento
ibmmq_qmgr_failed_mqinq_total	counter	Recuento de MQINQ anómalo

Métrica	Tipo	Descripción
ibmmq_qmgr_failed_mqopen_total	counter	Recuento de MQOPEN anómalo
ibmmq_qmgr_failed_mqput1_total	counter	Recuento de MQPUT1 anómalo
ibmmq_qmgr_failed_mqput_total	counter	Recuento de MQPUT anómalo
ibmmq_qmgr_failed_mqset_total	counter	Recuento de MQSET anómalo
ibmmq_qmgr_failed_mqsubrq_total	counter	Recuento de MQSUBRQ anómalo
ibmmq_qmgr_failed_subscription_create_alter_resume_total	counter	Crear/alterar/reanudar recuento de suscripciones anómalo
ibmmq_qmgr_failed_subscription_delete_total	counter	Recuento de anomalías de supresión de suscripción
ibmmq_qmgr_failed_topic_mqput_mqput1_total	counter	Recuento de MQPUT/MQPUT1 del tema anómalo
ibmmq_qmgr_fdc_files	gauge	Recuento de archivos de MQ FDC
ibmmq_qmgr_log_file_system_in_use_bytes	gauge	Sistema de archivos de registro - bytes en uso
ibmmq_qmgr_log_file_system_max_bytes	gauge	Sistema de archivos de registro - bytes máx
ibmmq_qmgr_log_in_use_bytes	gauge	Registro - bytes en uso
ibmmq_qmgr_log_logical_written_bytes_total	counter	Registro - bytes lógicos escritos
ibmmq_qmgr_log_max_bytes	gauge	Registro – máx bytes
ibmmq_qmgr_log_occupied_by_reusable_extents_bytes	gauge	Registro - bytes ocupados por extensiones reutilizables
ibmmq_qmgr_log_physical_written_bytes_total	counter	Registro - bytes físicos escritos
ibmmq_qmgr_log_primary_space_in_use_percentage	gauge	Registro - espacio primario actual en uso

Métrica	Tipo	Descripción
ibmmq_qmgr_log_required_for_media_recovery_bytes	gauge	Registro - bytes necesarios para la recuperación de soportes
ibmmq_qmgr_log_workload_primary_space_utilization_percentage	gauge	Registro - utilización de espacio primario de carga de trabajo
ibmmq_qmgr_log_write_latency_seconds	gauge	Registro - latencia de escritura
ibmmq_qmgr_log_write_size_bytes	gauge	Registro - tamaño de escritura
ibmmq_qmgr_mqcb_total	counter	Recuento de MQCB
ibmmq_qmgr_mqclose_total	counter	Recuento de MQCLOSE
ibmmq_qmgr_mqconn_mqconnx_total	counter	Recuento de MQCONN/MQCONNX
ibmmq_qmgr_mqctl_total	counter	Recuento de MQCTL
ibmmq_qmgr_mqdisc_total	counter	Recuento de MQDISC
ibmmq_qmgr_mqinq_total	counter	Recuento de MQINQ
ibmmq_qmgr_mqopen_total	counter	Recuento de MQOPEN
ibmmq_qmgr_mqput_mqput1_bytes_total	counter	Total de intervalos de recuento de bytes de MQPUT/MQPUT1
ibmmq_qmgr_mqput_mqput1_total	counter	Total de intervalos de recuento de MQPUT/MQPUT1
ibmmq_qmgr_mqset_total	counter	Recuento de MQSET
ibmmq_qmgr_mqstat_total	counter	Recuento de MQSTAT
ibmmq_qmgr_mqsubrq_total	counter	Recuento de MQSUBRQ
ibmmq_qmgr_non_durable_subscription_create_total	counter	Crear recuento de suscripciones no duraderas
ibmmq_qmgr_non_durable_subscription_delete_total	counter	Suprimir recuento de suscripciones no duraderas
ibmmq_qmgr_non_persistent_message_browse_bytes_total	counter	Examen de mensajes no persistentes - recuento de bytes

Métrica	Tipo	Descripción
ibmmq_qmgr_non_persistent_message_browse_total	counter	Examen de mensajes no persistentes - recuento
ibmmq_qmgr_non_persistent_message_destructive_get_total	counter	Obtención destructiva de mensajes no persistentes - recuento
ibmmq_qmgr_non_persistent_message_get_bytes_total	counter	Se han obtenido mensajes no persistentes - recuento de bytes
ibmmq_qmgr_non_persistent_message_mqput1_total	counter	Recuento de MQPUT1 de mensajes no persistentes
ibmmq_qmgr_non_persistent_message_mqput_total	counter	Recuento de MQPUT de mensajes no persistentes
ibmmq_qmgr_non_persistent_message_put_bytes_total	counter	Mensajes no persistentes de transferencia - recuento de bytes
ibmmq_qmgr_non_persistent_topic_mqput1_total	counter	No persistentes - recuento de MQPUT/MQPUT1 del tema
ibmmq_qmgr_persistent_message_browse_bytes_total	counter	Examen de mensajes persistentes - recuento de bytes
ibmmq_qmgr_persistent_message_browse_total	counter	Examen de mensajes persistentes - recuento
ibmmq_qmgr_persistent_message_destructive_get_total	counter	Obtención destructiva de mensajes persistentes - recuento
ibmmq_qmgr_persistent_message_get_bytes_total	counter	Se han obtenido mensajes persistentes - recuento de bytes
ibmmq_qmgr_persistent_message_mqput1_total	counter	Recuento de MQPUT1 de mensajes persistentes
ibmmq_qmgr_persistent_message_mqput_total	counter	Recuento de MQPUT de mensajes persistentes
ibmmq_qmgr_persistent_message_put_bytes_total	counter	Mensajes persistentes de transferencia - recuento de bytes
ibmmq_qmgr_persistent_topic_mqput1_total	counter	Persistentes - recuento de MQPUT/MQPUT1 del tema

Métrica	Tipo	Descripción
ibmmq_qmgr_published_to_subscribers_bytes_total	counter	Publicado en suscriptores - recuento de bytes
ibmmq_qmgr_published_to_subscribers_message_total	counter	Publicado en suscriptores - recuento de mensajes
ibmmq_qmgr_purged_queue_total	counter	Recuento de colas depuradas
ibmmq_qmgr_queue_manager_file_system_free_space_percentage	gauge	Sistema de archivos del gestor de colas - espacio libre
ibmmq_qmgr_queue_manager_file_system_in_use_bytes	gauge	Sistema de archivos del gestor de colas - bytes en uso
ibmmq_qmgr_ram_free_percentage	gauge	Porcentaje libre de RAM
ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes	gauge	Total de bytes de RAM - estimación para el gestor de colas
ibmmq_qmgr_rollback_total	counter	Recuento de retrotracciones
ibmmq_qmgr_system_cpu_time_estimate_for_queue_manager_percentage	gauge	Tiempo de CPU del sistema – estimación de porcentaje para el gestor de colas
ibmmq_qmgr_system_cpu_time_percentage	gauge	Porcentaje de tiempo de CPU del sistema
ibmmq_qmgr_topic_mqput_mqput1_total	counter	Total de intervalos de MQPUT/MQPUT1 del tema
ibmmq_qmgr_topic_put_bytes_total	counter	Total de intervalos de transferencia de bytes del tema
ibmmq_qmgr_trace_file_system_free_space_percentage	gauge	Sistema de archivos de rastreo MQ - espacio libre
ibmmq_qmgr_trace_file_system_in_use_bytes	gauge	Sistema de archivos de rastreo MQ - bytes en uso
ibmmq_qmgr_user_cpu_time_estimate_for_queue_manager_percentage	gauge	Tiempo de CPU de usuario – estimación de porcentaje para el gestor de colas
ibmmq_qmgr_user_cpu_time_percentage	gauge	Porcentaje de tiempo de CPU del usuario

Información relacionada

Métricas publicadas en los temas del sistema

Copia de seguridad y restauración de la configuración del gestor de colas utilizando la CLI de Red Hat 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 utilizando **oc login**.

Procedimiento

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

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

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

Por ejemplo, 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
```

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

dmpmqcfg genera la configuración de MQSC del gestor de colas.

```
oc exec -it pod_name -- dmpmqcfg > 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, 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
```

- 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
```

Visualización del estado de los gestores de colas HA nativa

Para contenedores creados de forma personalizada, puede ver el estado de las instancias de HA nativa utilizando el mandato **dspmq**.

Acerca de esta tarea

Puede utilizar el mandato **dspmqr** para ver el estado operativo de una instancia de gestor de colas en un nodo. La información devuelta depende de si la instancia está activa o es una réplica. La información proporcionada por la instancia activa es definitiva, es posible que la información de los nodos de réplica esté obsoleta.

Puede realizar las acciones siguientes:

- Ver si la instancia del gestor de colas en el nodo actual está activa o es una réplica.
- Ver el estado operativo de HA nativa de la instancia en el nodo actual.
- Ver el estado operativo de las tres instancias en una configuración de HA nativa.

Los siguientes campos de estado se utilizan para notificar el estado de configuración de HA nativa:

ROLE

Especifica el rol actual de la instancia y es uno de Active, Replica o Unknown.

INSTANCIA

El nombre proporcionado para esta instancia del gestor de colas cuando se creó utilizando la opción **-lr** del mandato **crtmqmr**.

INSYNC

Indica si la instancia puede tomar el control como instancia activa si es necesario.

QUORUM

Informa del estado de quórum con el formato *number_of_instances_in-sync/number_of_instances_configured*.

REPLADDR

La dirección de réplica de la instancia del gestor de colas.

CONNECTV

Indica si el nodo está conectado a la instancia activa.

BACKLOG

Indica el número de KB que la instancia está detrás.

CONNINST

Indica si la instancia con nombre está conectada a esta instancia.

ALTDAT

Indica la fecha en la que esta información se actualizó por última vez (en blanco si nunca se ha actualizado).

ALTTIME

Indica la hora a la que se actualizó por última vez esta información (en blanco si nunca se ha actualizado).

Procedimiento

- Para determinar si una instancia de gestor de colas se está ejecutando como instancia activa o como réplica:

```
dspmqr -o status -m QMgrName
```

Una instancia activa de un gestor de colas denominado BOB notificaría el estado siguiente:

```
QMNAME(BOB)          STATUS(Running)
```

Una instancia de réplica de un gestor de colas denominado BOB notificaría el estado siguiente:

```
QMNAME(BOB)          STATUS(Replica)
```

Una instancia inactiva notificaría el siguiente estado:

```
QMNAME(BOB)                STATUS(Ended Immediately)
```

- Para determinar el estado operativo de HA nativa de la instancia en el nodo actual:

```
dspmqr -o nativeha -m QMgrName
```

La instancia activa de un gestor de colas denominado BOB puede notificar el estado siguiente:

```
QMNAME(BOB)                ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Una instancia de réplica de un gestor de colas denominado BOB puede notificar el estado siguiente:

```
QMNAME(BOB)                ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Una instancia inactiva de un gestor de colas denominado BOB puede notificar el siguiente estado:

```
QMNAME(BOB)                ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Para determinar el estado operativo de HA nativa de todas las instancias de la configuración de HA nativa:

```
dspmqr -o nativeha -x -m QMgrName
```

Si emite este mandato en el nodo que ejecuta la instancia activa del gestor de colas BOB, es posible que reciba el siguiente estado:

```
QMNAME(BOB)                ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTD(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTD(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTD(2022-01-12) ALTTIME(12.03.44)
```

Si emite este mandato en un nodo que ejecuta una instancia de réplica del gestor de colas BOB, es posible que reciba el siguiente estado, que indica que una de las réplicas está retrasada:

```
QMNAME(BOB)                ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTD(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTD(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTD(2022-01-12) ALTTIME(12.03.44)
```

Si emite este mandato en un nodo que ejecuta una instancia inactiva del gestor de colas BOB, es posible que reciba el siguiente estado:

```
QMNAME(BOB)                ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTD( ) ALTTIME( )
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTD( ) ALTTIME( )
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTD( ) ALTTIME( )
```

Si emite el mandato cuando las instancias todavía están negociando qué está activo y cuáles son réplicas, recibirá el estado siguiente:

```
QMNAME(BOB)                STATUS(Negotiating)
```

Referencia relacionada

[Mandato dspmqr \(visualizar gestores de colas\)](#)

MQ Adv. Finalización manual de instancias de gestor de colas de HA nativa

Puede utilizar el mandato `endmqm` para finalizar un gestor de colas activo o de réplica que forme parte de un grupo HA nativo.

Procedimiento

- Para finalizar la instancia activa de un gestor de colas, consulte [Finalización de gestores de colas HA nativos](#) en la sección Configuración de esta documentación.

OpenShift CP4I Información de referencia para IBM MQ en contenedores

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

OpenShift CP4I Referencia de API para IBM MQ Operator

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

OpenShift CP4I Referencia de API para [mq.ibm.com/v1beta1](#)

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

OpenShift CP4I-SC2 CP4I CD Referencia de licencia para [mq.ibm.com/v1beta1](#)

Versiones de licencia actuales

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	Versiones de IBM MQ aplicables
L-JTPV-KYG8TF	Production o NonProduction	IBM Cloud Pak for Integration 16.1.0	9.4.0
L-BMSF-5YDSLRL	Production o NonProduction	IBM Cloud Pak for Integration Edición limitada 16.1.0	9.4.0
L-EHXT-MQCRN9	Production	IBM MQ Advanced 9.4	9.4.0
L-CLXQ-ADXTK3	Development	IBM MQ Advanced for Developers (sin garantía) 9.4	9.4.0

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

Versiones de licencia más antiguas

Consulte [Versiones de licencia más antiguas](#) en la documentación de IBM MQ 9.3 .

QueueManager

Un QueueManager es un servidor IBM MQ que proporciona servicios de cola y publicación/suscripción a aplicaciones. IBM MQ : <https://ibm.biz/BdPZqj>. Referencia de licencia: <https://ibm.biz/BdPZfq>.

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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources .
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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-classes .
metadata	
spec Especificación de QueueManager	El estado deseado de QueueManager.
status Estado de QueueManager	El estado observado de QueueManager.

.especificación

El estado deseado de QueueManager.

Aparece en:

- [“QueueManager” en la página 141](#)

Campo	Descripción
affinity	Reglas de afinidad Kubernetes estándar. Para obtener más información, consulte https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core .
annotations Anotaciones	El campo de anotaciones sirve como paso a través para las anotaciones de pod. Los usuarios pueden añadir cualquier anotación a este campo y hacer que se aplique al pod. Las anotaciones aquí sobrescriben las anotaciones predeterminadas si se proporcionan. Requiere MQ Operator 1.3.0 o superior.
Matriz imagePullSecrets LocalObjectReference	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 https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod .
labels Etiquetas	El campo de etiquetas sirve como paso a través para las etiquetas de pod. Los usuarios pueden añadir cualquier etiqueta a este campo y hacer que se aplique al pod. Las etiquetas aquí sobrescriben las etiquetas predeterminadas si se proporcionan. Requiere MQ Operator 1.3.0 o superior.

Campo	Descripción
<code>license</code> Licencia	Valores que controlan la aceptación de la licencia y qué métricas de licencia utilizar.
<code>pki</code> 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).
<code>queueManager</code> Configuración de QueueManager	Valores para el contenedor del gestor de colas y el gestor de colas subyacente.
<code>securityContext</code> SecurityContext	Valores de seguridad para añadir al <code>securityContext</code> del pod del gestor de colas.
<code>telemetry</code> Telemetría	Valores para la configuración de Open Telemetry. Requiere MQ Operator 2.2.0 o superior.
<code>template</code> 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.
<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. El valor predeterminado es 30 segundos.
<code>tracing</code> TracingConfig	Valores para rastrear la integración con el panel de control de operaciones de Cloud Pak for Integration.
<code>Serie version</code>	Valor que controla la versión de MQ que se utilizará (necesario). Por ejemplo: <code>9.1.5.0-r2</code> 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.
<code>web</code> Configuración de WebServer	Valores para el servidor web de MQ.

.spec.annotations

El campo de anotaciones sirve como paso a través para las anotaciones de pod. Los usuarios pueden añadir cualquier anotación a este campo y hacer que se aplique al pod. Las anotaciones aquí sobrescriben las anotaciones predeterminadas si se proporcionan. Requiere MQ Operator 1.3.0 o superior.

Aparece en:

- [“especificación” en la página 141](#)

.spec.imagePullSecrets

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

Aparece en:

- [“especificación” en la página 141](#)

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

.spec.labels

El campo de etiquetas sirve como paso a través para las etiquetas de pod. Los usuarios pueden añadir cualquier etiqueta a este campo y hacer que se aplique al pod. Las etiquetas aquí sobrescriben las etiquetas predeterminadas si se proporcionan. Requiere MQ Operator 1.3.0 o superior.

Aparece en:

- [“especificación” en la página 141](#)

.spec.license

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

Aparece en:

- [“especificación” en la página 141](#)

Campo	Descripción
accept boolean	Si acepta o no la licencia asociada con este software (obligatorio).
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 https://ibm.biz/BdPZfq para ver los valores válidos.
Serie metric	Valor que especifica qué métrica de licencia se debe utilizar. Por ejemplo, ProcessorValueUnit, VirtualProcessorCore o ManagedVirtualServer. El valor predeterminado es ProcessorValueUnit cuando se utiliza una licencia de MQ y VirtualProcessorCore cuando se utiliza una licencia de Cloud Pak for Integration .
Serie use	Valor que controla cómo se utilizará el software, donde la licencia admite varios usos. Consulte https://ibm.biz/BdPZfq para ver los valores válidos.

.spec.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” en la página 141](#)

Campo	Descripción
Matriz keys PKISource	Claves privadas para añadir al repositorio de claves del gestor de colas.
Matriz trust PKISource	Certificados para añadir al repositorio de claves del gestor de colas.

.spec.pki.keys

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

Aparece en:

- [“.spec.pki” en la página 143](#)

Campo	Descripción
Serie name	El nombre se utiliza como etiqueta para la clave o el certificado. Debe ser una serie alfanumérica en minúsculas.
secret <u>Secreto</u>	Proporcione una clave utilizando un secreto de Kubernetes .

.spec.pki.keys.secret

Proporcione una clave utilizando un secreto de Kubernetes .

Aparece en:

- [“.spec.pki.keys” en la página 143](#)

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 .

.spec.pki.trust

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

Aparece en:

- [“.spec.pki” en la página 143](#)

Campo	Descripción
Serie name	El nombre se utiliza como etiqueta para la clave o el certificado. Debe ser una serie alfanumérica en minúsculas.
secret <u>Secreto</u>	Proporcione una clave utilizando un secreto de Kubernetes .

.spec.pki.trust.secret

Proporcione una clave utilizando un secreto de Kubernetes .

Aparece en:

- [“.spec.pki.trust” en la página 144](#)

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 .

.spec.queueManager

Valores para el contenedor del gestor de colas y el gestor de colas subyacente.

Aparece en:

- [“.especificación” en la página 141](#)

Campo	Descripción
availability Disponibilidad	Valores de disponibilidad para el gestor de colas, como por ejemplo si se debe utilizar o no un par activo-en espera o una alta disponibilidad nativa.
debug boolean	Indica si se deben registrar o no los mensajes de depuración del código específico del contenedor en el registro del contenedor. El valor predeterminado es false.
Serie image	La imagen de contenedor que se utilizará.
Serie imagePullPolicy	Valor que controla cuándo el kubelet intenta extraer la imagen especificada. El valor predeterminado es IfNotPresent.
Matriz ini INISource	Valores para proporcionar INI para el gestor de colas. Requiere MQ Operator 1.1.0 o superior.
livenessProbe QueueManagerLivenessProbe	Valores que controlan el análisis de actividad.
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. El valor predeterminado es Basic.
metrics Métricas de QueueManager	Valores para métricas de estilo Prometheus.
Matriz mqsc MQSCSource	Valores para proporcionar MQSC para el gestor de colas. Requiere MQ Operator 1.1.0 o superior.
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).
readinessProbe QueueManagerReadinessProbe	Valores que controlan el sondeo de preparación.
recoveryLogs RecoveryLogs	Valores para los registros de recuperación de MQ . Requiere MQ Operator 2.4.0 o superior.
resources Recursos	Valores que controlan los requisitos de recursos.
route Ruta	Valores para la ruta del gestor de colas. Requiere MQ Operator 1.4.0 o superior.
startupProbe StartupProbe	Valores que controlan el sondeo de inicio. Sólo se aplica a los despliegues MultiInstance y NativeHA . Requiere MQ Operator 1.5.0 o superior.
storage Almacenamiento de QueueManager	Valores de almacenamiento para controlar el uso del gestor de colas de volúmenes persistentes y clases de almacenamiento.

.spec.queueManager.availability

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

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
<code>tls Tls</code>	Valores de TLS opcionales para configurar la comunicación segura entre réplicas de NativeHA . Requiere MQ Operator 1.5.0 o superior.
<code>Serie type</code>	El tipo de disponibilidad a utilizar. Utilice <code>SingleInstance</code> para un único pod, que Kubernetes reiniciará automáticamente (en algunos casos). Utilice <code>MultiInstance</code> para un par de pods, uno de los cuales es el gestor de colas de <code>active</code> y el otro está en espera. Utilice <code>NativeHA</code> para la réplica de alta disponibilidad nativa (requiere MQ Operator 1.5.0 o superior). El valor predeterminado es <code>SingleInstance</code> . Consulte http://ibm.biz/BdqAQa para obtener más detalles.
<code>Serie updateStrategy</code>	La estrategia de actualización que se utilizará para los gestores de colas <code>MultiInstance</code> y <code>NativeHA</code> . Utilice <code>RollingUpdate</code> para habilitar las actualizaciones continuas automáticas siempre que cambie la configuración del gestor de colas. Utilice <code>OnDelete</code> para inhabilitar las actualizaciones continuas automáticas. Los cambios del gestor de colas sólo se aplicarán cuando se supriman los pods (incluidas las supresiones de pods desencadenadas por factores externos). El valor predeterminado es <code>RollingUpdate</code> . Requiere MQ Operator 1.6.0 o superior.

.spec.queueManager.availability.tls

Valores de TLS opcionales para configurar la comunicación segura entre réplicas de NativeHA . Requiere MQ Operator 1.5.0 o superior.

Aparece en:

- [“.spec.queueManager.availability” en la página 145](#)

Campo	Descripción
<code>Serie cipherSpec</code>	El nombre de la CipherSpec para NativeHA TLS.
<code>Serie secretName</code>	El nombre del secreto de Kubernetes .

.spec.queueManager.ini

Origen de los archivos de configuración de INI.

Aparece en:

- [“.spec.queueManager” en la página 144](#)

Campo	Descripción
<code>configMap ConfigMapINISource</code>	ConfigMap representa un Kubernetes ConfigMap que contiene información de INI.
<code>secret SecretINISource</code>	El secreto representa un secreto de Kubernetes que contiene información de INI.

.spec.queueManager.ini.configMap

ConfigMap representa un Kubernetes ConfigMap que contiene información de INI.

Aparece en:

- [“.spec.queueManager.ini” en la página 146](#)

Campo	Descripción
items matriz	Claves dentro del origen de Kubernetes que se debe aplicar.
Serie name	El nombre del origen de Kubernetes .

.spec.queueManager.ini.secret

El secreto representa un secreto de Kubernetes que contiene información de INI.

Aparece en:

- [“.spec.queueManager.ini”](#) en la página 146

Campo	Descripción
items matriz	Claves dentro del origen de Kubernetes que se debe aplicar.
Serie name	El nombre del origen de Kubernetes .

.spec.queueManager.livenessProbe

Valores que controlan el análisis de actividad.

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
failureThreshold entero	Número mínimo de anomalías consecutivas para que se considere que el sondeo ha fallado después de haber sido satisfactorio. Toma de forma predeterminada 1.
initialDelaySeconds entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicie el sondeo. El valor predeterminado es 90 segundos para SingleInstance. El valor predeterminado es 0 segundos para despliegues de MultiInstance y NativeHA . Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds entero	Con qué frecuencia (en segundos) se realizará el análisis. El valor predeterminado es 10 segundos.
successThreshold entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio después de haber fallado. Toma de forma predeterminada 1.
timeoutSeconds entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. El valor predeterminado es 5 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.metrics

Valores para métricas de estilo Prometheus.

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
enabled boolean	Indica si se debe habilitar o no un punto final para métricas compatibles con Prometheus. El valor predeterminado es true.

.spec.queueManager.mqsc

Origen de los archivos de configuración MQSC.

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
configMap ConfigMapMQSCSource	ConfigMap representa un Kubernetes ConfigMap que contiene información de MQSC.
secret SecretMQSCSource	El secreto representa un secreto de Kubernetes que contiene información de MQSC.

.spec.queueManager.mqsc.configMap

ConfigMap representa un Kubernetes ConfigMap que contiene información de MQSC.

Aparece en:

- [“.spec.queueManager.mqsc”](#) en la página 148

Campo	Descripción
items matriz	Claves dentro del origen de Kubernetes que se debe aplicar.
Serie name	El nombre del origen de Kubernetes .

.spec.queueManager.mqsc.secret

El secreto representa un secreto de Kubernetes que contiene información de MQSC.

Aparece en:

- [“.spec.queueManager.mqsc”](#) en la página 148

Campo	Descripción
items matriz	Claves dentro del origen de Kubernetes que se debe aplicar.
Serie name	El nombre del origen de Kubernetes .

.spec.queueManager.readinessProbe

Valores que controlan el sondeo de preparación.

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
failureThreshold entero	Número mínimo de anomalías consecutivas para que se considere que el sondeo ha fallado después de haber sido satisfactorio. Toma de forma predeterminada 1.
initialDelaySeconds entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicie el sondeo. El valor predeterminado es 10 segundos para SingleInstance. El valor predeterminado es 0 para despliegues de MultiInstance y NativeHA . Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Campo	Descripción
periodSeconds entero	Con qué frecuencia (en segundos) se realizará el análisis. El valor predeterminado es 5 segundos.
successThreshold entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio después de haber fallado. Toma de forma predeterminada 1.
timeoutSeconds entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. El valor predeterminado es 3 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.recoveryLogs

Valores para los registros de recuperación de MQ . Requiere MQ Operator 2.4.0 o superior.

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
logFilePages entero	Los datos del registro de recuperación se conservan en una serie de archivos. El tamaño del archivo de registro se especifica en unidades de páginas de 4 KB.

.spec.queueManager.resources

Valores que controlan los requisitos de recursos.

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
limits Límites	Valores de CPU y memoria.
requests Solicitudes	Valores de CPU y memoria.

.spec.queueManager.resources.limits

Valores de CPU y memoria.

Aparece en:

- [“.spec.queueManager.resources”](#) en la página 149

Campo	Descripción
cpu	
memory	

.spec.queueManager.resources.requests

Valores de CPU y memoria.

Aparece en:

- [“.spec.queueManager.resources”](#) en la página 149

Campo	Descripción
cpu	
memory	

.spec.queueManager.route

Valores para la ruta del gestor de colas. Requiere MQ Operator 1.4.0 o superior.

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
enabled boolean	Indica si se debe habilitar o no la ruta. El valor predeterminado es true.

.spec.queueManager.startupProbe

Valores que controlan el sondeo de inicio. Sólo se aplica a los despliegues MultiInstance y NativeHA . Requiere MQ Operator 1.5.0 o superior.

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
failureThreshold entero	Número mínimo de anomalías consecutivas para que el analizador se considere anómalo. El valor predeterminado es 24.
initialDelaySeconds entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicie el sondeo. El valor predeterminado es 0 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds entero	Con qué frecuencia (en segundos) se realizará el análisis. El valor predeterminado es 5 segundos.
successThreshold entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio. Toma de forma predeterminada 1.
timeoutSeconds entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. El valor predeterminado es 5 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.storage

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

Aparece en:

- [“.spec.queueManager”](#) en la página 144

Campo	Descripción
allowVolumeExpansion boolean	Indica si se debe permitir o no la expansión de volúmenes.

Campo	Descripción
Serie defaultClass	Clase de almacenamiento que se aplica a todos los volúmenes persistentes de este gestor de colas de forma predeterminada. Los volúmenes persistentes específicos pueden definir su propia clase de almacenamiento que alterará temporalmente este valor de clase de almacenamiento predeterminado. Si type of availability es SingleInstance o NativeHA, la clase de almacenamiento puede ser de tipo ReadWriteOnce o ReadWriteMany. Si type of availability es MultiInstance, la clase de almacenamiento debe ser del tipo ReadWriteMany.
defaultDeleteClaim boolean	Indica si se deben suprimir o no todos los volúmenes cuando se suprime el gestor de colas. Los volúmenes persistentes específicos pueden definir su propio valor para deleteClaim que alterará temporalmente este valor de reclamación defaultDelete. El valor predeterminado es false.
persistedData QueueManagerOptionalVolume	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.
queueManager 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.
recoveryLogs QueueManagerOptionalVolume	Detalles de volumen persistente para los registros de recuperación de MQ . Necesario cuando se utiliza el gestor de colas de varias instancias.
scratch Reutilizable	Valores del volumen efímero reutilizable del gestor de colas. Este volumen se montará como la carpeta '/run' en el contenedor. Sólo es aplicable si el sistema de archivos raíz está establecido en sólo lectura. Requiere MQ Operator 3.0.0 o superior.
tmp Tmp	Valores para el volumen efímero Tmp del gestor de colas. Este volumen se montará en el contenedor como la carpeta '/tmp'. Los archivos de datos de diagnóstico, como el archivo zip generado por el mandato runmqras, se crearán en este volumen. Sólo es aplicable si el sistema de archivos raíz está establecido en sólo lectura. Requiere MQ Operator 3.0.0 o superior.

.spec.queueManager.storage.persistedData

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.

Aparece en:

- [“.spec.queueManager.storage” en la página 150](#)

Campo	Descripción
Serie class	Clase de almacenamiento a utilizar para este volumen. Sólo es válido si type es persistent-claim. Si type of availability es SingleInstance o NativeHA, la clase de almacenamiento puede ser de tipo ReadWriteOnce o ReadWriteMany. Si type of availability es MultiInstance, la clase de almacenamiento debe ser del tipo ReadWriteMany.
deleteClaim boolean	Indica si este volumen se debe suprimir o no cuando se suprime el gestor de colas.
enabled boolean	Si este volumen se debe habilitar o no como un volumen independiente, o si se debe colocar en el volumen queueManager predeterminado. El valor predeterminado es false.

Campo	Descripción
Serie size	Tamaño del PersistentVolume que se debe pasar a Kubernetes, incluidas las unidades SI. Sólo es válido si type es persistent-claim. Por ejemplo, 2Gi. El valor predeterminado es 2Gi.
Serie sizeLimit	Límite de tamaño cuando se utiliza un volumen ephemeral . 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 ephemeral y el sistema de archivos raíz está establecido en sólo lectura. Requiere MQ Operator 3.0.0 o superior.
Serie type	Tipo de volumen a utilizar. Elija ephemeral para utilizar almacenamiento no persistente o persistent-claim para utilizar un volumen persistente. El valor predeterminado es persistent-claim.

.spec.queueManager.storage.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:

- [“.spec.queueManager.storage”](#) en la página 150

Campo	Descripción
Serie class	Clase de almacenamiento a utilizar para este volumen. Sólo es válido si type es persistent-claim. Si type of availability es SingleInstance o NativeHA, la clase de almacenamiento puede ser de tipo ReadWriteOnce o ReadWriteMany. Si type of availability es MultiInstance, la clase de almacenamiento debe ser del tipo ReadWriteMany.
deleteClaim boolean	Indica si este volumen se debe suprimir o no cuando se suprime el gestor de colas.
Serie size	Tamaño del PersistentVolume que se debe pasar a Kubernetes, incluidas las unidades SI. Sólo es válido si type es persistent-claim. Por ejemplo, 2Gi. El valor predeterminado es 2Gi.
Serie sizeLimit	Límite de tamaño cuando se utiliza un volumen ephemeral . 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 ephemeral y el sistema de archivos raíz está establecido en sólo lectura. Requiere MQ Operator 3.0.0 o superior.
Serie type	Tipo de volumen a utilizar. Elija ephemeral para utilizar almacenamiento no persistente o persistent-claim para utilizar un volumen persistente. El valor predeterminado es persistent-claim.

.spec.queueManager.storage.recoveryLogs

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

Aparece en:

- [“.spec.queueManager.storage”](#) en la página 150

Campo	Descripción
Serie class	Clase de almacenamiento a utilizar para este volumen. Sólo es válido si type es persistent-claim. Si type of availability es SingleInstance o NativeHA, la clase de almacenamiento puede ser de tipo ReadWriteOnce o ReadWriteMany. Si type of availability es MultiInstance, la clase de almacenamiento debe ser del tipo ReadWriteMany.
deleteClaim boolean	Indica si este volumen se debe suprimir o no cuando se suprime el gestor de colas.
enabled boolean	Si este volumen se debe habilitar o no como un volumen independiente, o si se debe colocar en el volumen queueManager predeterminado. El valor predeterminado es false.
Serie size	Tamaño del PersistentVolume que se debe pasar a Kubernetes, incluidas las unidades SI. Sólo es válido si type es persistent-claim. Por ejemplo, 2Gi. El valor predeterminado es 2Gi.
Serie sizeLimit	Límite de tamaño cuando se utiliza un volumen ephemeral . 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 ephemeral y el sistema de archivos raíz está establecido en sólo lectura. Requiere MQ Operator 3.0.0 o superior.
Serie type	Tipo de volumen a utilizar. Elija ephemeral para utilizar almacenamiento no persistente o persistent-claim para utilizar un volumen persistente. El valor predeterminado es persistent-claim.

.spec.queueManager.storage.scratch

Valores del volumen efímero reutilizable del gestor de colas. Este volumen se montará como la carpeta '/run' en el contenedor. Sólo es aplicable si el sistema de archivos raíz está establecido en sólo lectura. Requiere MQ Operator 3.0.0 o superior.

Aparece en:

- [“.spec.queueManager.storage” en la página 150](#)

Campo	Descripción
Serie sizeLimit	Límite de tamaño del volumen efímero, incluidas las unidades SI. Por ejemplo, 2Gi. Sólo es válido cuando el sistema de archivos raíz está establecido en sólo lectura. Requiere MQ Operator 3.0.0 o superior.

.spec.queueManager.storage.tmp

Valores para el volumen efímero Tmp del gestor de colas. Este volumen se montará en el contenedor como la carpeta '/tmp'. Los archivos de datos de diagnóstico, como el archivo zip generado por el mandato runmqras, se crearán en este volumen. Sólo es aplicable si el sistema de archivos raíz está establecido en sólo lectura. Requiere MQ Operator 3.0.0 o superior.

Aparece en:

- [“.spec.queueManager.storage” en la página 150](#)

Campo	Descripción
Serie sizeLimit	Límite de tamaño del volumen efímero, incluidas las unidades SI. Por ejemplo, 2Gi. Sólo es válido cuando el sistema de archivos raíz está establecido en sólo lectura. Requiere MQ Operator 3.0.0 o superior.

.spec.securityContext

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

Aparece en:

- [“especificación” en la página 141](#)

Campo	Descripción
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. El establecimiento de este valor en true afecta a qué objeto de restricciones de contexto de seguridad (SCC) 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. El valor predeterminado es false. Para obtener más información, consulte https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html .
readOnlyRootFilesystem boolean	Indica si se deben habilitar o no los valores del sistema de archivos raíz de sólo lectura para el gestor de colas. El valor predeterminado es false. Requiere MQ Operator 3.0.0 o superior.
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.

.spec.telemetry

Valores para la configuración de Open Telemetry. Requiere MQ Operator 2.2.0 o superior.

Aparece en:

- [“especificación” en la página 141](#)

Campo	Descripción
tracing Rastreo	Valores para el rastreo de Open Telemetry.

.spec.telemetry.tracing

Valores para el rastreo de Open Telemetry.

Aparece en:

- [“.spec.telemetry” en la página 154](#)

Campo	Descripción
instana Instana	Valores para el rastreo de Instana.

.spec.telemetry.tracing.instana

Valores para el rastreo de Instana.

Aparece en:

- [“.spec.telemetry.tracing”](#) en la página 154

Campo	Descripción
Serie agentHost	El nombre de host del agente de Instana al que enviar datos de rastreo. Esto no debe incluir un protocolo.
enabled boolean	Indica si se debe habilitar o no el rastreo de Instana. El valor predeterminado es false.
Serie protocol	El protocolo que se utilizará en la comunicación con el agente de Instana. http y https están soportados.

.spec.template

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”](#) en la página 141

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

.spec.tracing

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

Aparece en:

- [“.especificación”](#) en la página 141

Campo	Descripción
agent TracingAgent	Solo en Cloud Pak for Integration , puede configurar valores para el agente de rastreo opcional.
collector TracingCollector	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. El valor predeterminado es false.
Serie namespace	Espacio de nombres donde está instalado el panel de control de operaciones de Cloud Pak for Integration .

.spec.tracing.agent

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

Aparece en:

- [“.spec.tracing” en la página 155](#)

Campo	Descripción
Serie <code>image</code>	La imagen de contenedor que se utilizará.
Serie <code>imagePullPolicy</code>	Valor que controla cuándo el kubelet intenta extraer la imagen especificada. El valor predeterminado es <code>IfNotPresent</code> .
<code>livenessProbe</code> <code>TracingProbe</code>	Valores que controlan el análisis de actividad.
<code>readinessProbe</code> <code>TracingProbe</code>	Valores que controlan el sondeo de preparación.

.spec.tracing.agent.livenessProbe

Valores que controlan el análisis de actividad.

Aparece en:

- [“.spec.tracing.agent” en la página 155](#)

Campo	Descripción
<code>failureThreshold</code> entero	Número mínimo de anomalías consecutivas para que se considere que el sondeo ha fallado después de haber sido satisfactorio. Toma de forma predeterminada 1.
<code>initialDelaySeconds</code> entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicien los sondeos de actividad. El valor predeterminado es 10 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
<code>periodSeconds</code> entero	Con qué frecuencia (en segundos) se realizará el análisis. El valor predeterminado es 10 segundos.
<code>successThreshold</code> entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio después de haber fallado. Toma de forma predeterminada 1.
<code>timeoutSeconds</code> entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. El valor predeterminado es 2 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.agent.readinessProbe

Valores que controlan el sondeo de preparación.

Aparece en:

- [“.spec.tracing.agent” en la página 155](#)

Campo	Descripción
<code>failureThreshold</code> entero	Número mínimo de anomalías consecutivas para que se considere que el sondeo ha fallado después de haber sido satisfactorio. Toma de forma predeterminada 1.

Campo	Descripción
initialDelaySeconds entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicien los sondeos de actividad. El valor predeterminado es 10 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds entero	Con qué frecuencia (en segundos) se realizará el análisis. El valor predeterminado es 10 segundos.
successThreshold entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio después de haber fallado. Toma de forma predeterminada 1.
timeoutSeconds entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. El valor predeterminado es 2 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.collector

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

Aparece en:

- [“.spec.tracing”](#) en la página 155

Campo	Descripción
Serie image	La imagen de contenedor que se utilizará.
Serie imagePullPolicy	Valor que controla cuándo el kubelet intenta extraer la imagen especificada. El valor predeterminado es IfNotPresent.
livenessProbe TracingProbe	Valores que controlan el análisis de actividad.
readinessProbe TracingProbe	Valores que controlan el sondeo de preparación.

.spec.tracing.collector.livenessProbe

Valores que controlan el análisis de actividad.

Aparece en:

- [“.spec.tracing.collector”](#) en la página 157

Campo	Descripción
failureThreshold entero	Número mínimo de anomalías consecutivas para que se considere que el sondeo ha fallado después de haber sido satisfactorio. Toma de forma predeterminada 1.
initialDelaySeconds entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicien los sondeos de actividad. El valor predeterminado es 10 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds entero	Con qué frecuencia (en segundos) se realizará el análisis. El valor predeterminado es 10 segundos.
successThreshold entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio después de haber fallado. Toma de forma predeterminada 1.

Campo	Descripción
timeoutSeconds entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. El valor predeterminado es 2 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.collector.readinessProbe

Valores que controlan el sondeo de preparación.

Aparece en:

- [“spec.tracing.collector” en la página 157](#)

Campo	Descripción
failureThreshold entero	Número mínimo de anomalías consecutivas para que se considere que el sondeo ha fallado después de haber sido satisfactorio. Toma de forma predeterminada 1.
initialDelaySeconds entero	Número de segundos después de que se haya iniciado el contenedor antes de que se inicien los sondeos de actividad. El valor predeterminado es 10 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds entero	Con qué frecuencia (en segundos) se realizará el análisis. El valor predeterminado es 10 segundos.
successThreshold entero	Número mínimo de éxitos consecutivos para que el sondeo se considere satisfactorio después de haber fallado. Toma de forma predeterminada 1.
timeoutSeconds entero	Número de segundos después de los cuales se agota el tiempo de espera del análisis. El valor predeterminado es 2 segundos. Más información: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.web

Valores para el servidor web de MQ .

Aparece en:

- [“especificación” en la página 141](#)

Campo	Descripción
console Consola	Valores para la consola web de MQ . Requiere MQ Operator 3.0.0 o superior.
enabled boolean	Indica si se debe habilitar o no el servidor web. El valor predeterminado es false.
manualConfig ManualConfig	Valores para proporcionar la configuración XML del servidor web. Requiere MQ Operator 3.0.0 o superior.

.spec.web.console

Valores para la consola web de MQ . Requiere MQ Operator 3.0.0 o superior.

Aparece en:

- [“spec.web” en la página 158](#)

Campo	Descripción
authentication <u>Autenticación</u>	Valores de autenticación para la consola web de MQ . Requiere MQ Operator 3.0.0 o superior.
authorization <u>Autorización</u>	Valores de autorización para la consola web de MQ . Requiere MQ Operator 3.0.0 o superior.

.spec.web.console.authentication

Valores de autenticación para la consola web de MQ . Requiere MQ Operator 3.0.0 o superior.

Aparece en:

- [“.spec.web.console”](#) en la página 158

Campo	Descripción
Serie provider	El proveedor de autenticación que se debe utilizar para la consola web de MQ . Utilice <code>integration-keycloak</code> para utilizar el inicio de sesión único con la interfaz de usuario de Cloud Pak for Integration Platform (Keycloak). El valor predeterminado es <code>integration-keycloak</code> si utiliza una licencia de Cloud Pak for Integration o <code>manual</code> si utiliza una licencia de MQ . Utilice <code>manual</code> si desea proporcionar su propia configuración.

.spec.web.console.authorization

Valores de autorización para la consola web de MQ . Requiere MQ Operator 3.0.0 o superior.

Aparece en:

- [“.spec.web.console”](#) en la página 158

Campo	Descripción
Serie provider	El proveedor de autorización que se debe utilizar para la consola web de MQ . Utilice <code>integration-keycloak</code> para utilizar roles proporcionados por Cloud Pak for Integration Keycloak. Utilice <code>manual</code> si desea proporcionar su propia configuración. El valor predeterminado es <code>integration-keycloak</code> si utiliza una licencia de Cloud Pak for Integration o <code>manual</code> si utiliza una licencia de MQ .

.spec.web.manualConfig

Valores para proporcionar la configuración XML del servidor web. Requiere MQ Operator 3.0.0 o superior.

Aparece en:

- [“.spec.web”](#) en la página 158

Campo	Descripción
configMap <u>ConfigMap</u>	ConfigMap representa un Kubernetes ConfigMap que contiene la configuración XML del servidor web.
secret <u>Secreto</u>	El secreto representa un secreto de Kubernetes que contiene la configuración XML del servidor web. El uso de un secreto protege las credenciales en la capa de Kubernetes , pero es posible que las herramientas de supervisión o resolución de problemas puedan exponer el archivo subyacente de forma insegura. Para mejorar la seguridad, codifique las credenciales utilizando "securityUtility".

.spec.web.manualConfig.configMap

ConfigMap representa un Kubernetes ConfigMap que contiene la configuración XML del servidor web.

Aparece en:

- [“.spec.web.manualConfig” en la página 159](#)

Campo	Descripción
Serie name	El nombre del origen de Kubernetes .

.spec.web.manualConfig.secret

El secreto representa un secreto de Kubernetes que contiene la configuración XML del servidor web.

El uso de un secreto protege las credenciales en la capa de Kubernetes , pero es posible que las herramientas de supervisión o resolución de problemas puedan exponer el archivo subyacente de forma insegura. Para mejorar la seguridad, codifique las credenciales utilizando "securityUtility".

Aparece en:

- [“.spec.web.manualConfig” en la página 159](#)

Campo	Descripción
Serie name	El nombre del origen de Kubernetes .

.estado

El estado observado de QueueManager.

Aparece en:

- [“QueueManager” en la página 141](#)

Campo	Descripción
Serie adminUiUrl	URL para la interfaz de usuario de administración.
availability Disponibilidad	Estado de disponibilidad del gestor de colas.
Matriz conditions QueueManagerStatusCondi tion	Las condiciones representan las últimas observaciones disponibles del estado del gestor de colas.
Matriz endpoints QueueManagerStatusEndpoi nt	Información sobre los puntos finales que este gestor de colas está exponiendo, como por ejemplo puntos finales de interfaz de usuario o API.
metadata Metadatos	Los metadatos representan información adicional para el gestor de colas, incluido el estado de Integración-Keycloak .
Serie name	Nombre del gestor de colas.
Serie phase	Fase del estado del gestor de colas.
versions QueueManagerStatusVersion	Versión de MQ que se está utilizando y otras versiones disponibles en IBM Entitled Registry.

.status.availability

Estado de disponibilidad del gestor de colas.

Aparece en:

- [“.estado” en la página 160](#)

Campo	Descripción
initialQuorumEstablished boolean	Si se ha establecido o no un quórum inicial para NativeHA.

.status.conditions

QueueManagerStatusCondition define las condiciones del gestor de colas.

Aparece en:

- [“.estado” en la página 160](#)

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

.status.endpoints

QueueManagerStatusEndpoint define los puntos finales para QueueManager.

Aparece en:

- [“.estado” en la página 160](#)

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.

.status.metadata

Los metadatos representan información adicional para el gestor de colas, incluido el estado de Integración-Keycloak .

Aparece en:

- [“.estado” en la página 160](#)

Campo	Descripción
integrationKeycloak <u>IntegrationKeycloak</u>	QueueManagerStatusIntegrationKeycloak define el estado de Integración-Keycloak para QueueManager.

.status.metadata.integrationKeycloak

QueueManagerStatusIntegrationKeycloak define el estado de Integración-Keycloak para QueueManager.

Aparece en:

- [“.status.metadata”](#) en la página 161

Campo	Descripción
Serie <code>clientName</code>	

.status.versions

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

Aparece en:

- [“.estado”](#) en la página 160

Campo	Descripción
<code>available</code> QueueManagerStatusVersion Disponible	Otras versiones de MQ disponibles en IBM Entitled Registry.
Serie <code>reconciled</code>	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.

.status.versions.available

Otras versiones de MQ disponibles en IBM Entitled Registry.

Aparece en:

- [“.status.versions”](#) en la página 162

Campo	Descripción
<code>channels</code> matriz	Canales que están disponibles para actualizar automáticamente la versión de MQ .
Matriz <code>versions</code> Versions	Versiones específicas de MQ que están disponibles.

.status.versions.available.versions

`QueueManagerStatusVersion` define una versión de MQ.

Aparece en:

- [“.status.versions.available”](#) en la página 162

Campo	Descripción
Matriz <code>licenses</code> Licencias	Licencias aplicables para esta versión de QueueManager.
Serie <code>name</code>	Versión <code>name</code> para esta versión de QueueManager. Son valores válidos para el campo <code>spec.version</code> .

.status.versions.available.versions.licenses

`QueueManagerStatusLicense` define una licencia.

Aparece en:

- [“.status.versions.available.versions”](#) en la página 162

Campo	Descripción
Serie <code>displayName</code>	Nombre de visualización de la licencia.

Campo	Descripción
Serie link	Enlace al contenido de la licencia.
matchesCurrentType boolean	Si la licencia coincide o no con el tipo de licencia utilizado actualmente.
Serie name	Nombre de la licencia.

  **Condiciones de estado para QueueManager (mq.ibm.com/v1beta1)**

Los campos **status.conditions** se actualizan para reflejar la condición del recurso QueueManager . En general, las condiciones describen situaciones anormales. Un gestor de colas en un estado preparado y en buen estado no tiene condiciones **Error** o **Pending** . Es posible que tenga algunas condiciones **Warning** de advertencia.

Se definen las condiciones siguientes para un recurso QueueManager :

Tabla 2. Condiciones de estado del gestor de colas

Componente	Tipo de condición	Código de razón	Aviso de mensaje
QueueManager ³	Bloqueado	OperatorDependenc y	Para instalar, esta instancia requiere que [IBM Cloud Pak for Integration] configure Keycloak . Esta instancia permanecerá en estado [Pendiente] hasta que se informe de Keycloak como [KeycloakReady] en el recurso Cp4iServicesBinding para este QueueManager.
			Para realizar la instalación, esta instancia requiere el operador [IBM IAM]. Esta instancia permanecerá en estado [Bloqueado] hasta que [IBM Cloud Pak foundational services] instale el operador.
	Pendiente	Creando	El gestor de colas de MQ se está desplegando
	Pendiente	OidcPending	El gestor de colas de MQ está esperando el registro de cliente OIDC
	Pendiente	Detenido	El gestor de colas de MQ se ha detenido porque la anotación 'mq.ibm.com/stop' está presente y se ha establecido en 'true' en la definición de QueueManager . Cuando se detiene el recuento de réplicas de QueueManager StatefulSet se establece en cero, eliminando todos los pods del gestor de colas de MQ .
	Error	Fallido	El gestor de colas de MQ no se ha podido desplegar
	Aviso	UnsupportedVersion	Un operando ha sido instalado por un operador que no está soportado en OCP versión < versión_ ocp > . Este operando no está soportado.
	Aviso	Soporte de CP4I-LTS	Se ha instalado un CP4I-LTS operando < mq_ versión > pero está siendo gestionado por un operador que no cumple los requisitos para la duración de soporte ampliada. Este operando no cumple los requisitos para la duración de soporte ampliada.
Aviso	Soporte de CP4I-LTS	Se ha instalado un operando de CP4I-LTS < mq_ versión > pero la versión del gestor de colas no cumple los requisitos para la duración de soporte ampliada. Este operando no cumple los requisitos para la duración de soporte ampliada.	

³ Las condiciones Creating y Failed supervisan el progreso global de despliegue del gestor de colas. Si está utilizando una licencia de IBM Cloud Pak for Integration y la consola web está habilitada, la condición OidcPending registra el estado del gestor de colas mientras espera a que el registro de cliente OIDC se complete con IAM.

Tabla 2. Condiciones de estado del gestor de colas (continuación)

Componente	Tipo de condición	Código de razón	Aviso de mensaje
Pod ⁴	Pendiente	PodPending	Se está desplegando el pod para el gestor de colas de MQ
	Error	PodFailed	Se está desplegando el pod para el gestor de colas de MQ
Almacenamiento ⁵	Pendiente	StoragePending	Se está suministrando almacenamiento para el gestor de colas MQ
	Aviso	StorageEphemeral	Utilización del almacenamiento efímero para un gestor de colas MQ de producción
	Aviso	StorageExpansionPendiente	La expansión de volumen está pendiente para las siguientes PVC [< lista de pvcs>]
	Aviso	StorageMismatch	Los tamaños de almacenamiento definidos en el recurso QueueManager no coinciden con la capacidad de una o más PVC suministradas [< lista de pvcs>]. AllowVolumeExpansion se establece en false en el recurso QueueManager por lo que el operador de MQ no intentará conciliar estas diferencias.
	Error	StorageFailed	El almacenamiento para el gestor de colas de MQ no se ha podido suministrar

Linux Anotaciones de licencia al crear su propia imagen de contenedor de IBM MQ

Las anotaciones de licencia le permiten realizar un seguimiento del uso según los límites definidos en el contenedor, en lugar de hacerlo según la máquina subyacente. Los clientes se configuran para desplegar el contenedor con anotaciones específicas que el IBM License Service utiliza para realizar el seguimiento del uso.

Al desplegar una imagen de contenedor de IBM MQ autocompilada, existen dos enfoques comunes para la licencia:

- Licencia de toda la máquina que ejecuta el contenedor.
- Licenciar el contenedor basándose en los límites asociados.

⁴ Las condiciones de pod supervisan el estado de los pods durante el despliegue de un gestor de colas. Si ve alguna condición PodFailed, la condición general del gestor de colas también se establecerá en Failed.

⁵ Las condiciones de almacenamiento supervisan el progreso (condiciónStoragePending) de las solicitudes para crear volúmenes para el almacenamiento persistente y notifican errores de enlace y otras anomalías. Las condiciones de almacenamiento también supervisan el progreso de las expansiones de volumen y alertan de discrepancias entre los tamaños de almacenamiento definidos en la definición del gestor de colas y el tamaño de las PVC desplegadas. Si se produce algún error durante el suministro de almacenamiento, la condición StorageFailed se añade a la lista de condiciones y la condición general del gestor de colas se establece en Failed.

Ambas opciones están disponibles para los clientes, y se pueden encontrar más detalles en la [página de licencias de contenedor de IBM en Passport Advantage](#).

Si el contenedor de IBM MQ se va a licenciar basándose en los límites del contenedor, es necesario instalar IBM License Service para realizar un seguimiento del uso. Puede encontrar más información sobre los entornos soportados y las instrucciones de instalación en la [página `ibm-licensing-operator` en GitHub](#).

El IBM License Service se instala en el clúster de Kubernetes donde se despliega el contenedor de IBM MQ y se utilizan las anotaciones de pod para realizar un seguimiento del uso. Por lo tanto, los clientes deben desplegar el pod con anotaciones específicas que el IBM License Service utiliza. En función de la titularidad y las prestaciones desplegadas en el contenedor, utilice una o varias de las anotaciones siguientes.

Nota: Muchas de las anotaciones contienen una o ambas de las líneas siguientes:

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

Debe editar estas líneas antes de utilizar la anotación:

- Para `productChargedContainers`, debe elegir "All" o sustituir el nombre real del contenedor.
- Para `productMetric`, debe elegir uno de los valores ofrecidos.

Anotaciones para utilizar con una titularidad de producto de IBM MQ

Si tiene una titularidad de producto de IBM MQ, seleccione la anotación siguiente que coincida con la titularidad que ha adquirido y que desea utilizar.

- ["IBM MQ" en la página 168](#)
- ["IBM MQ Avanzado" en la página 168](#)
- ["IBM MQ para entorno de no producción" en la página 168](#)
- ["IBM MQ Advanced para entornos no productivos" en la página 168](#)
- ["IBM MQ Advanced para desarrolladores" en la página 168](#)

Las anotaciones IBM MQ para utilizar con configuraciones de alta disponibilidad de varias instancias de IBM MQ son las siguientes. Consulte también el tema ["Selección de las anotaciones correctas para configuraciones de alta disponibilidad"](#) en la página 167.

- ["IBM MQ Contenedor de varias instancias" en la página 168](#)
- ["IBM MQ Advanced Container Multi Instance" en la página 169](#)
- ["IBM MQ Container Multi Instance para Entorno No Productivo" en la página 169](#)
- ["IBM MQ Advanced Container Multi Instance for Non-Production Environment" en la página 169](#)

Anotaciones para utilizar con la titularidad del producto CP4I

Si tiene titularidad de IBM Cloud Pak for Integration (CP4I), seleccione la anotación siguiente que coincida con la titularidad que ha adquirido y que desea utilizar.

- ["IBM MQ con titularidad de CP4I" en la página 169](#)
- ["IBM MQ Avanzado con titularidad de CP4I" en la página 169](#)
- ["IBM MQ for Non-Production Environment con titularidad de CP4I" en la página 169](#)
- ["IBM MQ Advanced for Non-Production Environment con titularidad de CP4I" en la página 169](#)

Las anotaciones CP4I para utilizar con configuraciones de alta disponibilidad de varias instancias de IBM MQ son las siguientes. Consulte también el tema ["Selección de las anotaciones correctas para configuraciones de alta disponibilidad"](#) en la página 167.

- ["IBM MQ Contenedor de varias instancias con titularidad de CP4I" en la página 170](#)

- [“Titularidad de IBM MQ Advanced Container Multi Instance con CP4I” en la página 170](#)
- [“Titularidad de IBM MQ Container Multi Instance for Non-Production Environment con CP4I” en la página 170](#)
- [“IBM MQ Advanced Container Multi Instance for Non-Production Environment con titularidad de CP4I” en la página 170](#)

Selección de las anotaciones correctas para configuraciones de alta disponibilidad

IBM MQ Varias instancias

Cuando despliegue un par de gestores de colas en una configuración de alta disponibilidad de varias instancias de IBM MQ , debe utilizar la misma anotación en ambas instancias. Se debe seleccionar una de las anotaciones siguientes, en función de la titularidad adquirida:

- Titularidad autónoma de IBM MQ o IBM MQ Advanced
 - [“IBM MQ Contenedor de varias instancias” en la página 168](#)
 - [“IBM MQ Advanced Container Multi Instance” en la página 169](#)
 - [“IBM MQ Container Multi Instance para Entorno No Productivo” en la página 169](#)
 - [“IBM MQ Advanced Container Multi Instance for Non-Production Environment” en la página 169](#)
- IBM Cloud Pak for Integration titularidad
 - [“IBM MQ Contenedor de varias instancias con titularidad de CP4I” en la página 170](#)
 - [“Titularidad de IBM MQ Advanced Container Multi Instance con CP4I” en la página 170](#)
 - [“Titularidad de IBM MQ Container Multi Instance for Non-Production Environment con CP4I” en la página 170](#)
 - [“IBM MQ Advanced Container Multi Instance for Non-Production Environment con titularidad de CP4I” en la página 170](#)

Cuando se utiliza con la titularidad de IBM Cloud Pak for Integration , las proporciones de titularidad en las anotaciones garantizan que se registra el consumo de titularidad correcto. Cuando se utiliza con titularidades IBM MQ o IBM MQ Advanced autónomas, las anotaciones notificadas en el License Service para cada instancia deben correlacionarse con los componentes de titularidad de IBM MQ de la forma siguiente:

- IBM MQ Advanced container Varias instancias
 - 1 x IBM MQ Advanced **y** 1 x IBM MQ Advanced Réplica de alta disponibilidad **o**
 - 2 x IBM MQ Advanced⁶
- IBM MQ Advanced container Multi Instance para Entorno No Productivo
 - 1 x IBM MQ Advanced **y** 1 x IBM MQ Advanced Réplica de alta disponibilidad **o**
 - 2 x IBM MQ Advanced para entorno no productivo)⁶
- IBM MQ Contenedor de varias instancias
 - 1 x IBM MQ **y** 1 x IBM MQ Réplica de alta disponibilidad **o**
 - 2 x IBM MQ⁶
- IBM MQ Container Multi Instance para Entorno No Productivo
 - 1 x IBM MQ **y** 1 x IBM MQ Réplica de alta disponibilidad **o**
 - 2 x IBM MQ para entorno no productivo) ⁶

IBM MQ HA nativa

⁶ Esta opción de titularidad es subóptima y solo se debe utilizar si no hay disponible ninguna titularidad de la parte de Réplica de alta disponibilidad relevante.

Si está desplegando tres gestores de colas en un quórum de HA nativa, sólo la instancia activa consume titularidad. Todas las instancias deben tener la misma anotación. Se debe seleccionar una de las opciones siguientes, en función de la titularidad adquirida:

- Titularidad autónoma de IBM MQ o IBM MQ Advanced
 - [“IBM MQ Avanzado” en la página 168](#)
 - [“IBM MQ Advanced para entornos no productivos” en la página 168](#)
- IBM Cloud Pak for Integration titularidad
 - [“IBM MQ Avanzado con titularidad de CP4I” en la página 169](#)
 - [“IBM MQ Advanced for Non-Production Environment con titularidad de CP4I” en la página 169](#)

Anotaciones

El resto de este tema detalla el contenido de cada anotación.

IBM MQ

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

IBM MQ Avanzado

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

IBM MQ para entorno de no producción

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

IBM MQ Advanced para entornos no productivos

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

IBM MQ Advanced para desarrolladores

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

IBM MQ Contenedor de varias instancias

```
productID: "2dea73b866b648b6b4abe2a85eb76964"  
productName: "IBM MQ Container Multi Instance"
```



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

IBM MQ Advanced Container Multi Instance

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

IBM MQ Container Multi Instance para Entorno No Productivo

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

IBM MQ Advanced Container Multi Instance for Non-Production Environment

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

IBM MQ con titularidad de CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"  
cloudpakName: "IBM Cloud Pak for Integration"  
productID: "c661609261d5471fb4ff8970a36bccea"  
productName: "IBM MQ"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productCloudpakRatio: "4:1"
```

IBM MQ Avanzado con titularidad de CP4I

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

IBM MQ for Non-Production Environment con titularidad de CP4I

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

IBM MQ Advanced for Non-Production Environment con titularidad de CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

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

IBM MQ Contenedor de varias instancias con titularidad de CP4I

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

Titularidad de IBM MQ Advanced Container Multi Instance con CP4I

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

Titularidad de IBM MQ Container Multi Instance for Non-Production Environment con CP4I

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

IBM MQ Advanced Container Multi Instance for Non-Production Environment con titularidad de CP4I

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

IBM MQ Advanced for Developers imagen de contenedor

Hay disponible una imagen de contenedor precompilada para IBM MQ Advanced for Developers. Esta imagen está disponible en IBM Container Registry. Esta imagen es adecuada para utilizarla con Docker, Podman, Kubernetes otros entornos de contenedor.

Imágenes disponibles

Las imágenes de IBM MQ se almacenan en IBM Container Registry:

- IBM MQ Advanced for Developers 9.4.0.0: icr.io/ibm-messaging/mq:9.4.0.0-r1

Consulta rápida

- Licencia:
 - Referencia de licencias para mq.ibm.com/v1beta1 y [Apache License 2.0](https://www.apache.org/licenses/LICENSE-2.0). Tenga en cuenta que la licencia de IBM MQ Advanced for Developers no permite una distribución adicional y que los términos restringen el uso a una máquina de desarrollador.
- Dónde archivar problemas:
 - [GitHub](https://github.com)
- Disponible para las siguientes arquitecturas de CPU:
 - amd64
 - s390x
 - ppc64le

Utilización

Ejecute [IBM MQ Advanced for Developers](#) en un contenedor.

Consulte la [documentación de uso](#) para obtener detalles sobre cómo ejecutar un contenedor.

Para poder utilizar la imagen, debe aceptar los términos de la licencia de IBM MQ estableciendo la variable de entorno **LICENSE**.

Variables de entorno soportadas

LANG

Establezca el idioma en el que desea que se imprima la licencia.

LICENSE

Establezca `accept` para aceptar las condiciones de licencia de IBM MQ Advanced for Developers.



Establezca `view` para ver las condiciones de licencia.

MQ_ADMIN_PASSWORD

Especifique la contraseña del usuario administrador.

Debe tener al menos 8 caracteres de longitud.

No hay ninguna contraseña predeterminada para el usuario administrativo.

  A partir de IBM MQ 9.4.0, esta variable ya no se proporciona. [El YAML de ejemplo de este tema](#) muestra cómo puede crear esta variable usted mismo y protegerla con un secreto.



MQ_APP_PASSWORD

Especifique la contraseña del usuario de la aplicación.

Si se establece, esto hace que el canal **DEV.APP.SVRCONN** pase a estar protegido y solo permita conexiones que suministren un ID de usuario y una contraseña válidos.

Debe tener al menos 8 caracteres de longitud.

No hay ninguna contraseña predeterminada para el usuario de la aplicación.

  A partir de IBM MQ 9.4.0, esta variable ya no se proporciona. [El YAML de ejemplo de este tema](#) muestra cómo puede crear esta variable usted mismo y protegerla con un secreto.

MQ_DEV

Establezca `false` para detener la creación de los objetos predeterminados.

MQ_MÉTRICAS_HABILITADAS

Establezca `true` para generar métricas de Prometheus para el gestor de colas.

MQ_LOGGING_CONSOLE_SOURCE

Especifique una lista separada por comas de orígenes para los registros que se duplican en la ubicación **stdout** del contenedor.

Los valores válidos son `qmgr`, `web` y `mqsc`.

El valor predeterminado es `qmgr`, `web`.

El valor opcional es `mqsc`. Esta opción se puede utilizar para reflejar el contenido de `autocfgmqsc.LOG` en el registro de contenedor.

MQ_LOGGING_CONSOLE_FORMAT

Cambie el formato de los registros que se imprimen en la ubicación **stdout** del contenedor.

Establezca `basic` para utilizar un formato legible simple. Éste es el valor predeterminado.

Establezca `json` para utilizar el formato JSON (un objeto JSON en cada línea).

MQ_LOGGING_CONSOLE_EXCLUDE_ID

Especifique una lista separada por comas de los ID de mensaje para los mensajes de registro que se excluyen.

Los mensajes de registro siguen apareciendo en el archivo de registro en el disco, pero no se imprimen en la ubicación **stdout** del contenedor.

El valor predeterminado es `AMQ5041I`, `AMQ5052I`, `AMQ5051I`, `AMQ5037I`, `AMQ5975I`.

mq_qmgr_name

Establezca el nombre con el que desea que se cree el gestor de colas.

Para obtener más información sobre la configuración de desarrollador predeterminada soportada por la imagen IBM MQ Advanced for Developers, consulte la [documentación de configuración de desarrollador predeterminada](#).

Ejemplo de gestor de colas YAML que describe cómo especificar contraseñas para usuarios de `admin` y `app`

Para los usuarios de los ID de usuario `admin` y `app`, debe proporcionar contraseñas al desplegar un gestor de colas utilizando la licencia de `Development`. A continuación se muestra un YAML de gestor de colas de ejemplo que muestra cómo hacerlo con IBM MQ Operator.

El mandato siguiente crea un secreto que contiene contraseñas para los usuarios de `admin` y `app`.

```
oc create secret generic my-mq-dev-passwords --from-literal=dev-admin-password=passw0rd --from-literal=dev-app-password=passw0rd
```

El siguiente YAML utiliza estas contraseñas al desplegar un gestor de colas.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm-dev
spec:
  license:
    accept: false
    license: L-CLXQ-ADXTK3
    use: Development
  web:
    enabled: true
  template:
    pod:
      containers:
        - env:
            - name: MQ_DEV
              value: "true"
            - name: MQ_CONNAUTH_USE_HTTP
              value: "true"
            - name: MQ_ADMIN_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-admin-password
            - name: MQ_APP_PASSWORD
```

```
valueFrom:
  secretKeyRef:
    name: my-mq-dev-passwords
    key: dev-app-password
name: qmgr
queueManager:
  storage:
    queueManager:
      type: persistent-claim
    name: QUICKSTART
version: 9.4.0.0-r1
```

Resolución de problemas de IBM MQ en contenedores

Si tiene problemas con la ejecución de IBM MQ en un contenedor, puede utilizar las técnicas que se describen aquí para ayudarle a diagnosticar y resolver los problemas.

Procedimiento

- [“Resolución de problemas de reinicios no planificados de IBM MQ en contenedores”](#) en la página 173.
- [“Resolución de problemas con IBM MQ Operator”](#) en la página 174.

OpenShift CP4I Kubernetes Resolución de problemas de reinicios no planificados de IBM MQ en contenedores

En la mayoría de los sistemas de gestión de contenedores, como Red Hat OpenShift Container Platform y Kubernetes, los contenedores se reinician habitualmente. No es normal que un contenedor sea longevo. En este tema se explica el ciclo de vida del contenedor, cómo puede investigar un reinicio y las razones detrás de un reinicio de contenedor no planificado.

Si no ha visto ningún problema con el despliegue de IBM MQ y éste continúa ejecutándose según lo esperado, es probable que la solución funcione según lo previsto. Puede ver un mensaje de registro como este en el registro de contenedor:

```
Signal received: terminated
```

Esto significa que la señal SIGTERM se ha enviado al contenedor MQ , pidiéndole que termine. Los contenedores Linux tienen la responsabilidad de responder a las señales POSIX , que son mensajes estandarizados enviados a un programa para desencadenar el comportamiento.

Cuando el contenedor IBM MQ recibe una señal SIGTERM, emite un mandato `endmqm -w -r -tp` para detener el gestor de colas. Una vez que el gestor de colas se haya detenido, el contenedor se detendrá. Si el gestor de colas tarda mucho tiempo en detenerse, es posible que se envíe una señal SIGKILL, que terminará inmediatamente los procesos de Linux . La cantidad de tiempo entre un SIGTERM y un SIGKILL se conoce como "periodo de gracia de terminación" en Kubernetes, y se puede configurar en el recurso QueueManager (si está utilizando el IBM MQ Operator), o directamente en el recurso Pod. El valor predeterminado es 30 segundos, de los cuales un segundo está reservado para que el contenedor se cierre y el resto se proporciona a IBM MQ. Por ejemplo, en el caso predeterminado, se emite un `endmqm -w -tp 29` , que indica al gestor de colas que tiene 29 segundos para concluir.

Razones del desalojo del pod

La señal SIGTERM la utiliza Kubernetes (y, por lo tanto, Red Hat OpenShift Container Platform) para terminar correctamente un pod. Consulte [Terminación de pods](#) en la documentación de Kubernetes . Kubernetes utiliza los términos ["interrupción de pod"](#) y ["desalojo"](#) para el proceso mediante el cual los pods en nodos se terminan de forma voluntaria o involuntaria. Hay muchas razones por las que un pod puede ser desalojado, incluyendo:

- **Terminación por kubelet.** Esto puede ser por varias razones, incluyendo las siguientes:
 - El pod se puede terminar porque el nodo se está cerrando (quizás como parte de una actualización de clúster continua)

- El pod se puede terminar debido a la "presión" del nodo (donde kubelet termina de forma proactiva los pods para reclamar recursos en un nodo). El administrador del clúster de Kubernetes puede configurar umbrales de desalojo que pueden variar entre clústeres.
- El pod se puede terminar porque el pod ha fallado en su análisis de actividad. Se puede configurar un análisis de actividad en Kubernetes para comprobar que un pod sigue en buen estado. El IBM MQ Operator configura un sondeo de actividad del gestor de colas que llama al mandato **dspmqr** para comprobar un estado de ejecución válido. Si el gestor de colas no está en buen estado, o si la ejecución del propio analizador tarda demasiado tiempo, el kubelet considerará que se ha producido un error. Los umbrales para el número de anomalías a tolerar son configurables, ya sea en el recurso QueueManager (si está utilizando IBM MQ Operator), o directamente en el recurso Pod.
- **Prevención por parte del planificador de Kubernetes.** Esto puede suceder si el planificador de Kubernetes tiene que ejecutar un pod de prioridad más alta
- **Nodo corrupto.** Un nodo puede estar "corrupto" y los pods que no toleran la corrupción se desalojan. Los administradores de Kubernetes utilizan las marcas para "repeler" los pods de nodos específicos. Por ejemplo, para decir que los pods de IBM MQ ya no deben ejecutarse en nodos que tienen hardware especial que ahora está reservado para otras cargas de trabajo.
- **Solicitar a través de la API de desalojo.** Esto lo puede llamar un administrador para desalojar Pods
- **Recogida de basura de pod.** Esto puede suceder si el nodo se queda fuera de servicio o se elimina a través de la API de Kubernetes .

Determinar por qué se ha desalojado un pod de gestor de colas

Las fuentes potenciales de información para ayudar a entender por qué un Pod fue desalojado incluyen:

- **Sucesos de clúster.** Por ejemplo, [Visualización de información de sucesos del sistema en un clúster de OpenShift Container Platform](#) .
- **Sucesos de auditoría de clúster.** Consulte [Visualización de registros de auditoría en Red Hat OpenShift Container Platform](#).
- **Nodos bajo presión.** Busque nodos bajo presión de CPU, red o memoria. Puede verlo en el estado Nodo. Tenga en cuenta que para el momento en que llegue a mirar, es posible que el nodo ya no esté bajo presión.
- **Red Hat OpenShift Container Platform Monitoring** u otras métricas de supervisión pueden mostrar cosas como problemas de latencia de disco. Una métrica útil de Prometheus es [ibmmq_qmgr_log_write_latency_seconds](#). Esta información procede de los temas de estadísticas de MQ .

Información relacionada

[Documentación de Kubernetes sobre planificación, prevención y desalojo](#)

OpenShift

CP4I

Resolución de problemas con IBM MQ Operator

Si tiene problemas con IBM MQ Operator, utilice las técnicas descritas para ayudarle a diagnosticar y resolver los mismos.

Procedimiento

- [“Recopilación de información de resolución de problemas para los gestores de colas desplegados con IBM MQ Operator” en la página 174](#)
- [“Resolución de problemas: obtención de acceso a los datos del gestor de colas” en la página 176](#)

OpenShift

CP4I

Recopilación de información de resolución de problemas para los gestores de colas desplegados con IBM MQ Operator

Recopilación de información de resolución de problemas que se debe proporcionar al soporte de IBM al generar un nuevo caso de soporte.

Procedimiento

1. Recopilar información de proveedor de nube.

Este es el proveedor de nube que aloja el clúster de Red Hat OpenShift (por ejemplo, IBM Cloud).

2. Recopilar información de arquitectura.

La arquitectura del clúster de Red Hat OpenShift es una de las siguientes:

- Linux for x86-64
- Linux on Power Systems (ppc64le)
- Linux for IBM Z

3. Recopile información de despliegue de IBM MQ .

a) Inicie sesión en el clúster de Red Hat OpenShift , utilizando un shell de bash/zsh .

b) Defina las variables de entorno siguientes:

```
export QM=QueueManager_name
export QM_NAMESPACE=QueueManager_namespace
export MQ_OPERATOR_NAMESPACE=mq_operator_namespace
```

Donde *QueueManager_name* es el nombre del recurso *QueueManager* , *QueueManager_namespace* es el espacio de nombres donde se despliega y *mq_operator_namespace* es el espacio de nombres donde se despliega IBM MQ Operator . Puede ser el mismo que el espacio de nombres *QueueManager* .

- c) Ejecute los mandatos siguientes y proporcione todos los archivos de salida resultantes al soporte de IBM .

```
# OCP / Kubernetes: Version
oc version -o yaml > ocversion.yaml

# QueueManager: YAML
oc get qmgr $QM -n $QM_NAMESPACE -o yaml > "queue-manager-$QM.yaml"

# MQ Queue Manager: Pods
oc get pods -n $QM_NAMESPACE -o wide --selector "app.kubernetes.io/instance=$QM" > "qm-pods-$QM.txt"

# MQ Queue Manager: Pod YAML
oc get pods -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-pods-$QM.yaml"

# MQ Queue Manager: Pod Logs
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc logs -n $QM_NAMESPACE --previous "$p" > "qm-logs-previous-$p.txt"; oc logs -n $QM_NAMESPACE $p > "qm-logs-$p.txt";done

# MQ Queue Manager: Describe Pods
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc describe pod $p -n $QM_NAMESPACE > "qm-pod-describe-$p.txt"; done

# MQ Web UI: Console Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/installations/Installation1/servers/mqweb/logs/console.log" "web-$p-console.log"; done

# MQ Web UI: Messages Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/installations/Installation1/servers/mqweb/logs/messages.log" "web-$p-messages.log"; done

# MQ Queue Manager: routes defined by operator
oc get routes -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-routes-$QM.yaml"

# MQ Queue Manager: routes to QM
oc get routes -n $QM_NAMESPACE -o yaml --field-selector "spec.to.name=$QM-ibm-mq" > "qm-routes2-$QM.yaml"

# MQ Queue Manager: stateful set
oc get statefulset -n $QM_NAMESPACE -o yaml ${QM}-ibm-mq > "qm-statefulset-$QM.yaml"
```

```

# MQ Queue Manager: revisions of the stateful set
oc get controllerrevisions.apps -o yaml -n $QM_NAMESPACE --selector "app.kubernetes.io/instance=$QM" > "qm-statefulset-revisions-$QM.yaml"

# MQ Queue Manager: Pod events
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do oc get -o custom-columns="LAST SEEN:.lastTimestamp,TYPE:.type,REASON:.reason,KIND:.involvedObject.kind,NAME:.involvedObject.name,MESSAGE:.message" event -n $QM_NAMESPACE --field-selector involvedObject.name="$p" > "qm-pod-events-$p.txt"; done

# MQ Queue Manager: StatefulSet events
oc get events -n $QM_NAMESPACE -o custom-columns="LAST SEEN:.lastTimestamp,TYPE:.type,REASON:.reason,KIND:.involvedObject.kind,NAME:.involvedObject.name,MESSAGE:.message" --field-selector involvedObject.name="$QM-ibm-mq" > "qm-statefulset-events-$QM.txt"

# MQ Queue Manager: services
oc get services -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-services-$QM.yaml"

# MQ Queue Manager: PVCs
oc get pvc -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-pvcs-$QM.yaml"

# MQ Operator: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-mq\\|NAME" > mq-operator-csv.txt

# Cloud Pak Foundational Services: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-common-service-operator\\|NAME" > common-services-csv.txt

# Cloud Pak for Integration: Version (if applicable)
oc get csv -n $QM_NAMESPACE | grep "^ibm-integration-platform-navigator\\|NAME" > cp4i-csv.txt

# Output from runmqras (this may take a while to execute)
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/instance=$QM" | cut -d ' ' -f 1); do timestamp=$(TZ=UTC date +"%Y%m%d_%H%M%S"); oc exec -n $QM_NAMESPACE $p -- runmqras -workdirectory "/tmp/runmqras_${timestamp}" -section logger,mqweb,nativeha,trace; oc cp -n $QM_NAMESPACE --retries=10 "$p:tmp/runmqras_${timestamp}/" .; done

# MQ Operator: Pod Log
oc logs -n $MQ_OPERATOR_NAMESPACE $(oc get pods -n $MQ_OPERATOR_NAMESPACE --no-headers --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/managed-by=olm | cut -d ' ' -f 1) > mq-operator-log.txt

```

Nota:

La mayoría de estos mandatos requieren acceso al espacio de nombres donde se despliega el gestor de colas. Sin embargo, la recopilación del registro de IBM MQ Operator puede requerir adicionalmente acceso de **administrador de clúster** si el IBM MQ Operator está instalado **con ámbito de clúster**.

Tareas relacionadas

[Recopilación de información de resolución de problemas para el soporte de IBM](#)

Resolución de problemas: obtención de acceso a los datos del gestor de colas

Utilice la herramienta de inspector de PVC para obtener acceso a los archivos de una PVC de gestor de colas donde no se puede establecer un shell remoto en el pod del gestor de colas. Esto puede deberse a que el pod está en un estado **Error** o **CrashLoopBackOff** . Esta herramienta está diseñada para su uso con gestores de colas desplegados por IBM MQ Operator.

Antes de empezar

Para utilizar la herramienta de inspector de PVC, debe tener acceso al espacio de nombres del gestor de colas.

Acerca de esta tarea

Como ayuda para la resolución de problemas, puede acceder a los datos almacenados en las reclamaciones de volúmenes persistentes (PVC) asociadas a un gestor de colas determinado. Para ello, utilice una herramienta para montar las PVC en un conjunto de pods de inspector. A continuación, puede obtener un shell remoto en cualquiera de los pods de inspector para leer los archivos.

En función del tipo de despliegue, se crean entre uno y tres pods de inspector. Los volúmenes específicos de un pod determinado de un gestor de colas Native-HA o de varias instancias están disponibles en el pod de inspector de PVC asociado. Los volúmenes compartidos están disponibles en todos los inspectores. El nombre del pod de inspector contiene el nombre del pod de gestor de colas asociado.

Procedimiento

1. Descargue la herramienta de inspector PVC de MQ .

La herramienta está disponible aquí: <https://github.com/ibm-messaging/mq-pvc-tool>.

2. Asegúrese de que ha iniciado sesión en el clúster.
3. Busque el nombre del gestor de colas y el espacio de nombres en el que se ejecuta el gestor de colas.
4. Ejecute la herramienta de inspector en el gestor de colas.

- a) Ejecute el mandato siguiente, especificando el nombre del gestor de colas y su nombre de espacio de nombres.

```
./pvc-tool.sh queue_manager_name queue_manager_namespace_name
```

- b) Una vez completada la herramienta, ejecute el mandato siguiente para ver los pods de inspector que se están creando.

```
oc get pods
```

5. Vea los archivos montados en el pod de inspector.

- a) Cada pod de inspector de PVC está asociado con un pod de gestor de colas, por lo que puede haber varios pods de inspector. Acceda a uno de estos pods ejecutando el mandato siguiente:

```
oc ish pvc-inspector-pod-name
```

Se coloca en el directorio que contiene los directorios PVC montados.

- b) Liste los directorios PVC ejecutando el mandato siguiente:

```
ls
```

- c) Consulte una lista de las PVC ejecutando el mandato siguiente fuera de la sesión de shell remota:

```
oc get pvc
```

- d) Limpie los pods creados por la herramienta, ejecutando el mandato siguiente:

```
oc delete pods -l tool=mq-pvc-inspector
```


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 que actualmente pueden adquirirse en su zona. Las referencias a programas, productos o servicios de IBM no pretenden establecer ni implicar que sólo puedan utilizarse dichos 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:

Intellectual Property Licensing
Legal and Intellectual Property Law
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 realizar mejoras y/o cambios en los productos y/o programas descritos en esta publicación en cualquier momento sin previo aviso.

Las referencias hechas en esta publicación a sitios web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen de modo alguno un aval de esos sitios web. Los materiales de estos sitios web no forman parte de los materiales para este producto IBM, por lo que la utilización de dichos sitios web es a 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, es posible que algunas mediciones se hayan estimado mediante 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. Todas las preguntas sobre las prestaciones de productos que no son 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 DERECHOS DE AUTOR:

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 IBM 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. 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 Estados Unidos y/o 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 (<https://www.eclipse.org/>).

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



Número Pieza:

(1P) P/N: