

9.2

IBM MQ en contenedores

IBM

Nota

Antes de utilizar esta información y el producto al que se refiere, lea la información en [“Avisos” en la página 163](#).

Esta edición se aplica a la versión 9 release 2 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
Planificación de IBM MQ en contenedores.....	5
Cómo utilizar IBM MQ en contenedores.....	5
Soporte para IBM MQ Operator.....	6
Dependencias para IBM MQ Operator.....	10
Permisos con ámbito de clúster necesarios para IBM MQ Operator.....	10
Consideraciones sobre el almacenamiento para IBM MQ Operator.....	11
Soporte para crear sus propias imágenes de contenedor de gestor de colas de IBM MQ.....	13
Alta disponibilidad para IBM MQ en contenedores.....	16
Recuperación tras desastre para IBM MQ en contenedores.....	18
Autenticación de usuario y autorización para IBM MQ en contenedores.....	19
Planificación de la escalabilidad y el rendimiento para IBM MQ en contenedores.....	19
Utilización de IBM MQ en IBM Cloud Pak for Integration y Red Hat OpenShift.....	20
Historial de releases de IBM MQ Operator.....	20
Migración de IBM MQ a IBM Cloud Pak for Integration.....	36
Instalación y desinstalación de IBM MQ Operator en Red Hat OpenShift.....	61
Actualización de IBM MQ Operator y gestores de colas.....	73
Despliegue y configuración de gestores de colas utilizando IBM MQ Operator.....	81
Funcionamiento de IBM MQ utilizando la IBM MQ Operator.....	117
Resolución de problemas con IBM MQ Operator.....	127
Referencia de API para IBM MQ Operator.....	128
Creación de su propio contenedor de IBM MQ y código de despliegue.....	150
Planificación de una imagen propia de gestor de colas de IBM MQ utilizando un contenedor.....	151
Creación de una imagen de contenedor de gestor de colas de IBM MQ de ejemplo.....	151
Ejecución de aplicaciones de enlaces locales en contenedores separados.....	154
Creación del grupo HA nativo si crea sus propios contenedores.....	156
Avisos.....	163
Información acerca de las interfaces de programación.....	164
Marcas registradas.....	165

Multi 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.

Multi 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 Red Hat OpenShift Container Platform](#)).

Procedimiento

- [“Cómo utilizar IBM MQ en contenedores”](#) en la página 5.
- [“Soporte para IBM MQ Operator”](#) en la página 6.
- [“Soporte para crear sus propias imágenes de contenedor de gestor de colas de IBM MQ”](#) en la página 13.
- [“Consideraciones sobre el almacenamiento para IBM MQ Operator”](#) en la página 11.
- [“Alta disponibilidad para IBM MQ en contenedores”](#) en la página 16.
- [“Recuperación tras desastre para IBM MQ en contenedores”](#) en la página 18.
- [“Autenticación de usuario y autorización para IBM MQ en contenedores”](#) en la página 19.

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 añade un nuevo recurso personalizado QueueManager a Red Hat OpenShift Container Platform. 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 158

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

- Utilizar las API REST para la administración o la mensajería.
- Utilizar cualquiera de los siguientes componentes de MQ:
 - Managed File Transfer Agentes y sus recursos. Sin embargo, puede utilizar IBM MQ Operator para proporcionar uno o varios gestores de colas de Coordinación, Mandato o Agente.
 - AMQP
 - IBM MQ Bridge to Salesforce
 - IBM MQ Bridge to blockchain (no soportado en contenedores)
 - IBM MQ Telemetry Transport (MQTT).
- Personalice las opciones utilizadas con **crtmqm**, **strmqm** y **endmqm**, como la configuración de páginas de archivo de registro. La mayoría de las opciones se pueden configurar utilizando un archivo INI.

Tenga en cuenta que IBM MQ Operator y los contenedores están evolucionando rápidamente y, por lo tanto, no están soportados en los releases de Long Term Support .

IBM MQ Operator incluye imágenes de contenedor precompiladas, así como código de despliegue para la ejecución en Red Hat OpenShift Container Platform. IBM MQ Operator se puede utilizar para desplegar la imagen de contenedor de IBM MQ proporcionada, o una imagen de contenedor en capas encima de ella, pero no se puede utilizar para desplegar imágenes de contenedor de MQ creadas de forma personalizada.

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 [“Creación de su propio contenedor de IBM MQ y código de despliegue”](#) en la página 150.

Conceptos relacionados

[“Soporte para IBM MQ Operator”](#) en la página 6

El IBM MQ Operator sólo está soportado cuando se despliega en Red Hat OpenShift Container Platform.

[“Soporte para crear sus propias imágenes de contenedor de gestor de colas de IBM MQ”](#) en la página 13

IBM MQ proporciona código para crear un contenedor de gestor de colas de IBM MQ en GitHub. Esto se basa en el proceso que IBM utiliza para crear su propio contenedor soportado, y puede utilizar este repositorio GitHub para simplificar y acelerar la creación de sus propias imágenes de contenedor.

Soporte para IBM MQ Operator

El IBM MQ Operator sólo está soportado cuando se despliega en Red Hat OpenShift Container Platform.

IBM MQ Operator utiliza imágenes basadas en releases de IBM MQ Continuous Delivery (CD), aunque hay disponible un release Extended Update Support (EUS) con IBM Cloud Pak for Integration. Las versiones de CD están soportadas durante un máximo de un año, o para dos versiones de CD, la que sea más larga. Los releases de Long Term Support de IBM MQ no están disponibles a través de IBM MQ Operator. IBM Cloud Pak for Integration 2020.4.1 es un release de soporte de actualización ampliado (EUS), que está soportado durante 18 meses, si utiliza una versión de IBM MQ marcada como -eus. De lo contrario, IBM MQ 9.2 se considera un release de Continuous Delivery con IBM MQ Operator.

IBM MQ Operator utiliza imágenes de contenedor que proporcionan una instalación de IBM MQ en un Red Hat Universal Base Image (UBI), que incluye bibliotecas clave de Linux® y programas de utilidad utilizados por IBM MQ. El UBI está soportado por Red Hat cuando se ejecuta en Red Hat OpenShift.

IBM MQ Operator está soportado en las arquitecturas amd64y s390x (z/Linux).

Conceptos relacionados

“Soporte para crear sus propias imágenes de contenedor de gestor de colas de IBM MQ” en la página 13
 IBM MQ proporciona código para crear un contenedor de gestor de colas de IBM MQ en GitHub. Esto se basa en el proceso que IBM utiliza para crear su propio contenedor soportado, y puede utilizar este repositorio GitHub para simplificar y acelerar la creación de sus propias imágenes de contenedor.

OpenShift EUS CP4I CD Soporte de versiones del IBM MQ Operator

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

- “Versiones de IBM MQ disponibles” en la página 7
- “Versiones de Red Hat OpenShift Container Platform compatibles” en la página 8
- “Versiones de IBM Cloud Pak for Integration” en la página 8
- “Versiones de IBM MQ disponibles en operadores más antiguos” en la página 8
- “Versiones compatibles de Red Hat OpenShift Container Platform para operadores más antiguos” en la página 9

Versiones de IBM MQ disponibles

Canal de operador	Versión del operador	Versiones de IBM MQ							
		9.1.5	CD de 9.2.0	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.6	1.6	⚠	⚠	→	⚠	●	●		
v1.7	1.7	⚠	⚠	→	⚠	●	●	●	
v1.8	1.8	⚠	⚠	→	⚠	⚠	●	●	●

Clave:

- Soporte de Continuous Delivery disponible
- Extended Update Support disponible
- Sólo está disponible durante la migración desde el operando Extended Update Support a un operando Continuous Delivery .
- ⚠ En desuso. A medida que los releases de IBM MQ dejan de tener soporte, pueden seguir siendo configurables en el operador, pero ya no son elegibles para el soporte y se pueden eliminar en futuros releases.

Consulte “Historial de releases de IBM MQ Operator” en la página 20 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 Red Hat OpenShift Container Platform ¹				
		4.6	4.7 ²	4.8	4.9	4.10
v1.6	1.6	●	●	●	●	●
v1.7	1.7	●	●	●	●	●
v1.8	1.8	●	●	●	●	●

Clave:



Soporte de Continuous Delivery disponible



Extended Update Support disponible

Versiones de IBM Cloud Pak for Integration

IBM MQ Operator 1.8.x está soportado para su uso como parte de IBM Cloud Pak for Integration versión 2021.4.1, o independientemente.

IBM MQ Operator 1.7.x está soportado para su uso como parte de IBM Cloud Pak for Integration versión 2021.4.1, o de forma independiente.

IBM MQ Operator 1.6.x está soportado para su uso como parte de IBM Cloud Pak for Integration versión 2021.2.1, 2021.3.1o independientemente.

IBM MQ Operator 1.5.x ya no está soportado.

IBM MQ Operator 1.4.x ya no está soportado.

IBM MQ Operator 1.3.x ya no está soportado.

IBM MQ Operator 1.2.x ya no está soportado.

Los IBM MQ Operators 1.1.x y 1.0.x ya no están soportados.

Versiones de IBM MQ disponibles en operadores más antiguos

La tabla siguiente se aplica a las versiones de IBM MQ Operator que ahora han alcanzado el "final de vida".

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

¹ Las versiones de Red Hat OpenShift Container Platform están sujetas a sus propias fechas de soporte. Consulte [Política de ciclo de vida de Red Hat OpenShift Container Platform](#) para obtener más información.

² IBM MQ Operator depende de IBM Cloud Pak foundational services. Si desea utilizar Red Hat OpenShift Container Platform 4.7, primero debe actualizar la versión de IBM Cloud Pak foundational services.

Canal de operador	Versión del operador	Versiones de IBM MQ							
		9.1.5	CD de 9.2.0	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.5	1.5	⚠	⚠	→	⚠	⚠			

Clave:

→

Sólo está disponible durante la migración desde el operando Extended Update Support a un operando Continuous Delivery .

⚠

En desuso. A medida que los releases de IBM MQ salen del soporte, pueden seguir siendo configurables en IBM MQ Operator, pero ya no son elegibles para el soporte.

Consulte “Historial de releases de IBM MQ Operator” en la página 20 para obtener detalles completos de cada versión, incluidas características detalladas, cambios y arreglos en cada versión.

Versiones compatibles de Red Hat OpenShift Container Platform para operadores más antiguos

La tabla siguiente se aplica a las versiones de IBM MQ Operator que ahora han alcanzado el "final de vida".

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

Clave:

→

Sólo está disponible durante la migración desde el operando Extended Update Support a un operando Continuous Delivery .

⚠

La versión de IBM MQ Operator ha alcanzado el "final del ciclo de vida", pero anteriormente estaba disponible en esta versión de Red Hat OpenShift Container Platform

³ Las versiones de Red Hat OpenShift Container Platform están sujetas a sus propias fechas de soporte. Consulte [Política de ciclo de vida de Red Hat OpenShift Container Platform](#) para obtener más información.

⁴ Red Hat OpenShift Container Platform 4.4 ha alcanzado el "final del ciclo de vida". Consulte [Política de ciclo de vida de Red Hat OpenShift Container Platform](#) para obtener más información.

⁵ Red Hat OpenShift Container Platform 4.5 ha alcanzado el "final del ciclo de vida". Consulte [Política de ciclo de vida de Red Hat OpenShift Container Platform](#) para obtener más información.

⁶ IBM MQ Operator depende de IBM Cloud Pak foundational services. Si desea utilizar Red Hat OpenShift Container Platform 4.7, primero debe actualizar la versión de IBM Cloud Pak foundational services.

El IBM MQ Operator depende del operador de IBM Cloud Pak foundational services , que también instala el operador de IBM Operand Deployment Lifecycle Manager (ODLM). Estos operadores se instalarán automáticamente al instalar el IBM MQ Operator. Estos operadores dependientes tienen una pequeña ocupación de CPU y memoria, y se utilizan para desplegar recursos adicionales en algunas circunstancias.

Al crear un `QueueManager`, el IBM MQ Operator creará un `OperandRequest` para servicios adicionales que necesite. El operador ODLM cumplirá el `OperandRequest` e instalará y creará una instancia de los servicios necesarios, si es necesario. Los servicios necesarios se determinan basándose en el acuerdo de licencia aceptado al desplegar el gestor de colas y en qué componentes del gestor de colas se solicitan.

- Si elige una licencia de IBM MQ Advanced o IBM MQ Advanced for Developers , no se solicitarán servicios adicionales. Por ejemplo, en el caso siguiente, no se utilizan los IBM Cloud Pak foundational services :

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

- Si elige una licencia de IBM Cloud Pak for Integration y elige habilitar el servidor web, el IBM MQ Operator también creará una instancia del operador de IBM Identity and Access Management (IAM), para habilitar el inicio de sesión único. El operador de IAM ya estará disponible si ha instalado el operador de IBM Cloud Pak for Integration . Por ejemplo:

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

Sin embargo, si inhabilita el servidor web, no se solicita ningún IBM Cloud Pak foundational services . Por ejemplo:

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

Las versiones anteriores de IBM MQ Operator siempre solicitaban la instalación de IBM Licensing Operator (y sus dependencias), para realizar un seguimiento del uso de licencia. A partir de IBM MQ Operator 1.5 , el servicio de licencias no se solicita y es necesario solicitarlo por separado.

IBM MQ Operator requiere 1 núcleo de CPU y 1 GB de memoria. Para obtener un desglose detallado de los requisitos de hardware y software para los operadores dependientes, consulte [Requisitos de hardware y recomendaciones para servicios básicos](#).

Puede elegir la cantidad de CPU y memoria utilizada por los gestores de colas. Consulte [“.spec.queueManager.resources”](#) en la [página 138](#) para obtener más información.

Referencia relacionada

[“Referencia de licencia para mq.ibm.com/v1beta1”](#) en la [página 128](#)

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: **create, get, update**
- 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**

OpenShift CP4I Kubernetes Consideraciones sobre el 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 114). Si esto no se completa, será necesario configurar IBM MQ cada vez que se reinicie el contenedor.

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

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

Almacén persistente

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

- `spec.queueManager.availability` controla la modalidad de disponibilidad. Si utiliza `SingleInstance` sólo necesita almacenamiento `ReadWriteOnce` , mientras que `MultiInstance` requiere una clase de almacenamiento que admita `ReadWriteMany` con las características de bloqueo de archivos correctas. IBM MQ proporciona una [declaración de soporte](#) y una [declaración de prueba](#). La modalidad de disponibilidad también influye en el diseño del volumen persistente. Para obtener más información, consulte [“Alta disponibilidad para IBM MQ en contenedores”](#) en la página 16
- `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: [99]
```

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

Nota: No necesita sistemas de archivos compartidos si utiliza HA nativa (consulte [“Alta disponibilidad para IBM MQ en contenedores”](#) en la página 16). En concreto, no debe utilizar NFSv3.

Linux Soporte para crear sus propias imágenes de contenedor de gestor de colas de IBM MQ

IBM MQ proporciona código para crear un contenedor de gestor de colas de IBM MQ en GitHub. Esto se basa en el proceso que IBM utiliza para crear su propio contenedor soportado, y puede utilizar este repositorio GitHub para simplificar y acelerar la creación de sus propias imágenes de contenedor.

El código se proporciona en el repositorio GitHub de mq-container aquí: <https://github.com/ibm-messaging/mq-container>. Esto se proporciona bajo una licencia de Apache 2.0, con el soporte proporcionado por la comunidad.

El repositorio no utiliza los paquetes rpm estándar de Linux; utiliza el paquete comprimido para despliegues de contenedor. La ventaja de este enfoque es que puede ejecutar en entornos de contenedor más seguros sin necesidad de permisos escalados. Sin embargo, esto afecta a las opciones de seguridad disponibles, porque IBM MQ utiliza tradicionalmente permisos escalados para la autenticación basada en el sistema operativo. Para un despliegue de contenedor, el uso de la autenticación basada en el sistema operativo normalmente no es una buena práctica; en su lugar, puede utilizar la autenticación TLS o LDAP mutua. Con IBM MQ Advanced for Developers, también puede utilizar la autenticación basada en archivos, lo que permite a los usuarios empezar rápidamente.

El gestor de colas de datos replicados (RDQM) no está soportado en un entorno de contenedor. Puede obtener prestaciones similares a RDQM utilizando [“HA nativa”](#) en la página 94.

Conceptos relacionados

[“Soporte para IBM MQ Operator”](#) en la página 6

El IBM MQ Operator sólo está soportado cuando se despliega en Red Hat OpenShift Container Platform.

[IBM MQ](#)

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.

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. Basándose en la titularidad y las prestaciones desplegadas en el contenedor, utiliza una o varias de las anotaciones siguientes:

- [“Contenedor de IBM MQ Advanced” en la página 14](#)
- [“Contenedor de réplica de IBM MQ Advanced High Availability” en la página 14](#)
- [“Contenedor base de IBM MQ” en la página 14](#)
- [“Contenedor de réplica de IBM MQ Base High Availability” en la página 14](#)
- [“Contenedor de IBM MQ Advanced for Developers” en la página 14](#)
- [“IBM MQ Contenedor avanzado con una titularidad de CP4I \(Producción\)” en la página 15](#)
- [“IBM MQ Contenedor de réplica de alta disponibilidad avanzado con una titularidad de CP4I \(Producción\)” en la página 15](#)
- [“IBM MQ Contenedor avanzado con una titularidad CP4I \(No producción\)” en la página 15](#)
- [“Contenedor de réplica de IBM MQ Advanced High Availability con una titularidad de CP4I \(no de producción\)” en la página 15](#)
- [“IBM MQ Base con una titularidad de CP4I \(Producción\)” en la página 15](#)
- [“IBM MQ Réplica de alta disponibilidad base con una titularidad de CP4I \(Producción\)” en la página 15](#)
- [“IBM MQ Base con una titularidad de CP4I \(Non-Production\)” en la página 16](#)
- [“IBM MQ Réplica de alta disponibilidad base con una titularidad de CP4I \(No producción\)” en la página 16](#)

Contenedor de IBM MQ Advanced

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

Contenedor de réplica de IBM MQ Advanced High Availability

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

Contenedor base de IBM MQ

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

Contenedor de réplica de IBM MQ Base High Availability

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

Contenedor de IBM MQ Advanced for Developers

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

IBM MQ Contenedor avanzado con una titularidad de CP4I (Producción)

```
productName: "IBM MQ Advanced with CP4I License"  
productID: "208423bb063c43288328b1d788745b0c"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "2:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Contenedor de réplica de alta disponibilidad avanzado con una titularidad de CP4I (Producción)

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

IBM MQ Contenedor avanzado con una titularidad CP4I (No producción)

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

Contenedor de réplica de IBM MQ Advanced High Availability con una titularidad de CP4I (no de producción)

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

IBM MQ Base con una titularidad de CP4I (Producción)

```
productName: "IBM MQ with CP4I License"  
productID: "c661609261d5471fb4ff8970a36bccea"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "4:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Réplica de alta disponibilidad base con una titularidad de CP4I (Producción)

```
productName: "IBM MQ High Availability Replica with CP4I License"  
productID: "2a2a8e0511c849969d2f286670ea125e"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "20:1"
```

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

IBM MQ Base con una titularidad de CP4I (Non-Production)

```
productName: "IBM MQ with CP4I License Non-Production"
productID: "151bec68564a4a7a14e6fa99266deff"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "8:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Réplica de alta disponibilidad base con una titularidad de CP4I (No producción)

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

Alta disponibilidad para IBM MQ en contenedores

Hay tres opciones para la alta disponibilidad con IBM MQ Operator: **Gestor de colas HA nativo** (que tiene una réplica activa y dos réplicas en espera), **Gestor de colas multiinstancia** (que es un par activo-en espera, utilizando 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

Los gestores de colas HA nativos están disponibles en IBM Cloud Pak for Integration 2021.2.1, utilizando IBM MQ Operator 1.6 o superior, con IBM MQ 9.2.3 o superior.

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 una con su propio conjunto de volúmenes persistentes de Kubernetes . Los requisitos de IBM MQ para los 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 94](#)

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.

Tareas relacionadas

[“Configuración de la alta disponibilidad para gestores de colas utilizando la IBM MQ Operator” en la página 94](#)

Referencia relacionada

[Configuraciones de alta disponibilidad](#)

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](#)

Autenticación de usuario y autorización para IBM MQ en contenedores

IBM MQ se puede configurar para utilizar usuarios y grupos LDAP. De forma alternativa, puede utilizar usuarios y grupos del sistema operativo local dentro de la imagen de contenedor. IBM MQ Operator no permite al usuario de usuarios y grupos del sistema operativo, debido a problemas de seguridad.

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, 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. Al utilizar IBM MQ en IBM Cloud Pak for Integration y Red Hat OpenShift, debe configurar el gestor de colas para utilizar LDAP para la autenticación y autorización de usuarios. Para obtener información sobre cómo configurar IBM MQ para hacer esto, consulte [Autenticación de conexiones: Repositorios de usuario y Autorización LDAP](#)

Multi 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. En Red Hat OpenShift Container Platform, hay un límite predeterminado de 4096 procesos por contenedor (1024 procesos hasta OpenShift 4.11). 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.

Utilización de IBM MQ en IBM Cloud Pak for Integration y Red Hat OpenShift

IBM MQ Operator despliega y gestiona IBM MQ como parte de IBM Cloud Pak for Integration, o autónomo en Red Hat OpenShift Container Platform

Procedimiento

- [“Historial de releases de IBM MQ Operator”](#) en la página 20.
- [“Migración de IBM MQ a IBM Cloud Pak for Integration”](#) en la página 36.
- [“Instalación y desinstalación de IBM MQ Operator en Red Hat OpenShift”](#) en la página 61.
- [“Actualización de IBM MQ Operator y gestores de colas”](#) en la página 73.
- [“Despliegue y configuración de gestores de colas utilizando IBM MQ Operator”](#) en la página 81.
- [“Funcionamiento de IBM MQ utilizando la IBM MQ Operator”](#) en la página 117.
- [“Referencia de API para IBM MQ Operator”](#) en la página 128.

Historial de releases de IBM MQ Operator

IBM MQ Operator

IBM MQ Operator 1.8.2



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

Canal de operador

v1.8

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, [9.2.0.5-r3-eus](#), 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, [9.2.5.0-r3](#)

Versiones de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 y superior

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 y superior (canalv3)

Novedades

- Actualización de solo seguridad basada en [IBM MQ Operator 1.8.0](#).
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

IBM MQ Operator 1.8.1



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

Canal de operador

v1.8

Valores permitidos para `.spec.version`

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

Versiones de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 y superior

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 y superior (canalv3)

Novedades

- Actualización de solo seguridad basada en [IBM MQ Operator 1.8.0](#).
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

IBM MQ Operator 1.8.0



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

Canal de operador

v1.8

Valores permitidos para `.spec.version`

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

Versiones de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 y superior

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 y superior (canalv3)

Novedades

- Añade condiciones de estado para versiones de IBM MQ en desuso.

Novedades

- Imágenes movidas de Docker Hub a IBM Container Registry.

- Los clientes con reglas de cortafuegos pueden tener que ajustarlas para acceder a las imágenes en IBM Container Registry.
- Los clientes Airgap experimentan un reinicio de nodo al actualizar a IBM MQ Operator 1.8.0.
- Versiones en desuso: IBM MQ 9.1.5, 9.2.0 CD, 9.2.1, 9.2.2. Es posible que estas versiones no estén reconciliadas por futuras versiones de IBM MQ Operator.
- Cambios en la lógica de licencia: los clientes que actualizan a IBM MQ 9.2.5 sólo pueden utilizar las licencias especificadas para trabajar con IBM MQ 9.2.5. Consulte [“Referencia de licencia para mq.ibm.com/v1beta1”](#) en la página 128
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

IBM MQ Operator 1.7.0



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.4.1

Canal de operador

v1.7

Valores permitidos para .spec.version

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

Versiones de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 y superior

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.8 y superior (canalv3)

Novedades

- Añade IBM MQ 9.2.4 como release de entrega continua

IBM MQ Operator 1.6.0



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.2.1

Canal de operador

v1.6

Valores permitidos para .spec.version

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

Versiones de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 y superior

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.7 y superior (canalv3)

Novedades

- Añade IBM MQ 9.2.3 como release de entrega continua (amd64 solo para IBM Cloud Pak for Integration 2021.2.1; amd64 o s390x cuando se utiliza una licencia de IBM MQ)
- Nuevo tipo de disponibilidad para gestores de colas: [HA nativa](#). Disponible para uso de producción, como parte de IBM Cloud Pak for Integration 2021.2.1.

Novedades

- IBM MQ Operator 1.6 y posteriores utilizan IBM Container Registry en lugar de Docker Hub. Esto significa que necesita utilizar un CatalogSource de [icr.io](#). Consulte [“Instalación y desinstalación de IBM MQ Operator en Red Hat OpenShift”](#) en la página 61.

- La actualización continua de HA nativa ya no espera a que una réplica esté sincronizada antes de pasar a la siguiente réplica.
- Soluciona el problema con la afinidad de HA nativa en OCP 4.7 y superior.
- Soluciona el problema al utilizar certificados firmados por CA con HA nativa.

IBM MQ Operator 1.5.0



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2021.1.1

Canal de operador

v1.5

Valores permitidos para `.spec.version`

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

Versiónes de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 y superior

Versiónes de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.7 y superior (canalv3)

Novedades

- Añade IBM MQ 9.2.2 como release de entrega continua (amd64 solo para IBM Cloud Pak for Integration 2021.1.1; amd64 o s390x cuando se utiliza una licencia de IBM MQ)
- Nuevo tipo de disponibilidad para gestores de colas: **HA nativa**. Disponible sólo para fines de evaluación, como parte de IBM Cloud Pak for Integration 2021.1.1.
- Integración con métricas de Red Hat OpenShift Container Platform Cluster Monitoring for Prometheus, proporcionando un recurso `ServiceMonitor`

Novedades

- IBM Licensing Operator ya no se crea de forma predeterminada al crear un gestor de colas
- Las actualizaciones de los gestores de colas de varias instancias se manejan ahora en un orden continuo. Como parte de este cambio, se ha introducido un sondeo de inicio de Kubernetes que afecta a los valores utilizados al configurar el sondeo de actividad. El sondeo de inicio se inicia inmediatamente y, a continuación, espera a que el gestor de colas se inicie correctamente. Si el sondeo de inicio pasa en cualquier momento dentro de este periodo de espera, se inician los sondeos de actividad y preparación. Anteriormente, si tenía un gestor de colas que tardaba en iniciarse, es posible que haya aumentado el valor de `initialDelaySeconds` en el sondeo de actividad. Si lo ha hecho, ahora debería revertir `initialDelaySeconds` al valor anterior.
- El `CustomResourceDefinition` se actualiza de `apiextensions.k8s.io/v1beta1` a `apiextensions.k8s.io/v1`

Problemas conocidos y limitaciones

- Requiere IBM Cloud Pak foundational services 3.7, que contiene un cambio incompatible en el componente Identity and Access Management (IAM). Si tiene algún gestor de colas que utilice una licencia de IBM Cloud Pak for Integration, después de esta actualización, será necesario reiniciar el gestor de colas para acceder a la consola web y también verá [otros errores](#) al iniciar sesión en la consola web. Puede corregir estos errores actualizando al valor más reciente de `.spec.version` para la versión de IBM MQ elegida, una vez completada la actualización del operador.
- La actualización continua no se inicia automáticamente si está actualizando la versión de MQ. Debe suprimir manualmente los pods.

IBM MQ Operator 1.4.0



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1 (IBM MQ Operator 1.4.0 es un release CD y no es apto para el soporte de actualización ampliado)

Canal de operador

v1.4

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, [9.2.1.0-r1](#)

Versiones de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.6 y superior

Novedades

- Añade IBM MQ 9.2.1 como release de entrega continua
- Ahora puede impedir la creación de la ruta del gestor de colas predeterminada estableciendo `.spec.queueManager.route.enabled` en `false`

Problemas conocidos y limitaciones

- Al actualizar un `QueueManager` con un tipo de disponibilidad de `MultiInstance`, ambos Pods se suprimirán inmediatamente. Los dos deben ser reiniciados rápidamente por Red Hat OpenShift Container Platform.

IBM MQ Operator 1.3.8 (EUS)



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal de operador

v1.3-eus

Valores permitidos para `.spec.version`

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

Versiones de Red Hat OpenShift Container Platform

Solo Red Hat OpenShift Container Platform 4.6

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canalstable-v1)

Novedades

- Añade una nueva versión de operando [9.2.0.6-r3-eus](#).
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

IBM MQ Operator 1.3.7 (EUS)



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal de operador

v1.3-eus

Valores permitidos para `.spec.version`

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

Versiones de Red Hat OpenShift Container Platform

Solo Red Hat OpenShift Container Platform 4.6

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canalstable-v1)

Novedades

- Añade una nueva versión de operando [9.2.0.6-r2-eus](#).
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

IBM MQ Operator 1.3.6 (EUS)



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal de operador

v1.3-eus

Valores permitidos para `.spec.version`

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

Versiones de Red Hat OpenShift Container Platform

Solo Red Hat OpenShift Container Platform 4.6

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canalstable-v1)

Novedades

- Añade una nueva versión de operando [9.2.0.6-r1-eus](#).
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

IBM MQ Operator 1.3.5 (EUS)



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal de operador

v1.3-eus

Valores permitidos para `.spec.version`

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

Versiones de Red Hat OpenShift Container Platform

Solo Red Hat OpenShift Container Platform 4.6

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canalstable-v1)

Novedades

- Añade una nueva versión de operando [9.2.0.5-r3-eus](#).
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

IBM MQ Operator 1.3.4 (EUS)



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal de operador

v1.3-eus

Valores permitidos para `.spec.version`

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

Versiones de Red Hat OpenShift Container Platform

Solo Red Hat OpenShift Container Platform 4.6

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canalstable-v1)

Novedades

- Añade una nueva versión de operando [9.2.0.5-r2-eus](#)
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

IBM MQ Operator 1.3.3 (EUS)



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal de operador

v1.3-eus

Valores permitidos para .spec.version

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

Versiones de Red Hat OpenShift Container Platform

Solo Red Hat OpenShift Container Platform 4.6

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canalstable-v1)

Novedades

- Añade una nueva versión de operando [9.2.0.5-r1-eus](#)
- Las vulnerabilidades que se tratan se detallan en este [Boletín de seguridad](#).

IBM MQ Operator 1.3.2 (EUS)



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal de operador

v1.3-eus

Valores permitidos para .spec.version

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

Versiones de Red Hat OpenShift Container Platform

Solo Red Hat OpenShift Container Platform 4.6

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canalstable-v1)

Novedades

- Añade una nueva versión de operando [9.2.0.4-r1-eus](#)

IBM MQ Operator 1.3.1 (EUS)



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal de operador

v1.3-eus

Valores permitidos para .spec.version

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

Versiones de Red Hat OpenShift Container Platform

Solo Red Hat OpenShift Container Platform 4.6

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canalstable-v1)

Novedades

- Añade una nueva versión de operando [9.2.0.2-r1-eus](#)

IBM MQ Operator 1.3.0 (EUS)



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.4.1

Canal de operador

v1.3-eus

Valores permitidos para .spec.version

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

Versiones de Red Hat OpenShift Container Platform

Solo Red Hat OpenShift Container Platform 4.6

Versiones de IBM Cloud Pak foundational services

IBM Cloud Pak foundational services 3.6 (canalstable-v1)

Novedades

- El soporte de actualización ampliado (EUS) se ofrece para los campos de `.spec.version` que terminan en `-eus`, cuando se utiliza una licencia de IBM Cloud Pak for Integration
- Añade una nueva forma de establecer etiquetas y anotaciones en el recurso `QueueManager` utilizando `.spec.labels` y `.spec.annotations`

Novedades

- Mejora el manejo de errores al intentar cambiar de una sola instancia a varias instancias
- Mejoras en la forma en que se representan las propiedades `QueueManager` en IBM Cloud Pak for Integration Platform Navigatory en la "Vista de formulario" de la consola web de Red Hat OpenShift Container Platform
- Arregla la métrica de licencia predeterminada cuando se utiliza una licencia de IBM Cloud Pak for Integration, para que sea `VirtualProcessorCore`
- Arregla el separador **Recursos** para `QueueManager` en la consola web de Red Hat OpenShift Container Platform, que ahora muestra correctamente los recursos gestionados por IBM MQ Operator para ese gestor de colas

Problemas conocidos y limitaciones

- Al actualizar un `QueueManager` con un tipo de disponibilidad de `MultiInstance`, ambos Pods se suprimirán inmediatamente. Los dos deben ser reiniciados rápidamente por Red Hat OpenShift Container Platform.

IBM MQ Operator 1.2.0



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.3.1

Canal de operador

v1.2

Valores permitidos para .spec.version

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

Versiones de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.4 y posteriores

Novedades

- Añade soporte para z/Linux
- Añade condiciones de estado más detalladas al recurso `QueueManager`. Para obtener más información, consulte [“Condiciones de estado para QueueManager \(mq.ibm.com/v1beta1\)”](https://mq.ibm.com/v1beta1) en la [página 148](#)
- Añade comprobaciones de tiempo de ejecución adicionales para evitar el uso de clases de almacenamiento no válidas. Para obtener más información, consulte [“Inhabilitación de comprobaciones de webhook en tiempo de ejecución”](#) en la [página 116](#)
- Simplifica la experiencia de los gestores de colas de varias instancias: ahora se puede elegir con una sola propiedad (`.spec.queueManager.availability.type`) en el recurso `QueueManager`
- Simplifica la elección de una clase de almacenamiento no predeterminada, introduciendo la propiedad `.spec.queueManager.storage.defaultClass` en el recurso `QueueManager`

Novedades

- Mejoras en la forma en que se representan las propiedades `QueueManager` en IBM Cloud Pak for Integration Platform Navigatory en la "Vista de formulario" de la consola web de Red Hat OpenShift Container Platform
- Si hay disponible una versión actualizada del gestor de colas, ahora se marcará en IBM Cloud Pak for Integration Platform Navigator

IBM MQ Operator 1.1.0



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.2.1

Canal de operador

v1.1

Valores permitidos para `.spec.version`

9.1.5.0-r2, 9.2.0.0-r1

Versiones de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.4 y posteriores

Novedades

- Añade IBM MQ Advanced 9.2.0 como release de entrega continua
- Añade una característica para especificar información de INI y MQSC en un ConfigMap o Secret
- Habilita el navegador de esquemas cuando se utiliza la consola web de Red Hat OpenShift Container Platform

Novedades

- Soluciona el problema con la política de red, afectando a Red Hat OpenShift en IBM Cloud
- Mejoras en el gancho web de validación, para evitar combinaciones no válidas de valores en recursos de `QueueManager`

IBM MQ Operator 1.0.0



Versión de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 2020.2.1

Canal de operador

v1.0

Valores permitidos para `.spec.version`

[9.1.5.0-r2](#)

Versiones de Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform 4.4 y posteriores

Novedades

- Versión inicial del operador, introducción a la API de `mq.ibm.com/v1beta1`

Imágenes de contenedor del gestor de colas para su uso con IBM MQ Operator

9.2.5.0-r3



Versión de operador necesaria

[1.8.2](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- [Novedades en IBM MQ 9.2.5](#)

Novedades

- [Qué ha cambiado en IBM MQ 9.2.5](#)
- Basado en [Red Hat Universal Base Image 8.6-751](#)

9.2.5.0-r2



Versión de operador necesaria

[1.8.1](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- [Novedades en IBM MQ 9.2.5](#)

Novedades

- [Qué ha cambiado en IBM MQ 9.2.5](#)
- Basado en [Red Hat Universal Base Image 8.5-240.1648458092](#)

9.2.5.0-r1



Versión de operador necesaria

[1.8.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r1](#)
- [icr.io/ibm-messaging/mq:9.2.5.0-r1](#)

Novedades

- [Novedades en IBM MQ 9.2.5](#)

Novedades

- [Qué ha cambiado en IBM MQ 9.2.5](#)
- La opción Gestores de colas remotos no válida ahora se ha eliminado de IBM MQ Console
- Basado en [Red Hat Universal Base Image 8.5-240](#)

9.2.4.0-r1



Versión de operador necesaria

[1.7.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- [Novedades en IBM MQ 9.2.4](#)

Novedades

- [Qué ha cambiado en IBM MQ 9.2.4](#)
- Basado en [Red Hat Universal Base Image 8.5-204](#)

9.2.3.0-r1



Versión de operador necesaria

[1.6.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.3.0-r1](#) (soloamd64)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.3.0-r1](#)
- [docker.io/ibmcom/mq:9.2.3.0-r1](#)

Novedades

- [Novedades en IBM MQ 9.2.3](#)

- Soporte para MQ [HA nativa](#) para uso de producción, cuando se utiliza con una licencia de IBM Cloud Pak for Integration . Tenga en cuenta que los gestores de colas que utilizan HA nativa bajo una licencia de evaluación con IBM MQ 9.2.2 no se pueden actualizar a 9.2.3. El periodo de evaluación ha finalizado.

Novedades

- [Qué ha cambiado en IBM MQ 9.2.3](#)
- Basado en [Red Hat Universal Base Image 8.4-205](#)

9.2.2.0-r1



Versión de operador necesaria

[1.5.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.2.0-r1](#) (soloamd64)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.2.0-r1](#)
- [docker.io/ibmcom/mq:9.2.2.0-r1](#)

Novedades

- [Novedades en IBM MQ 9.2.2](#)
- Soporte para MQ [Native HA](#) para fines de evaluación, cuando se utiliza con una licencia de IBM Cloud Pak for Integration

Novedades

- [Qué ha cambiado en IBM MQ 9.2.2](#)
- Se ha solucionado un problema que provocaba FDC al concluir un gestor de colas de IBM MQ Advanced for Developers
- Basado en [Red Hat Universal Base Image 8.3-291](#)

9.2.1.0-r2



Versión de operador necesaria

[1.5.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- Soluciona el problema con el inicio de sesión único con IBM Cloud Pak foundational services 3.7 y superior.
- Basado en [Red Hat Universal Base Image 8.3-291](#)

9.2.1.0-r1



Versión de operador necesaria

[1.4.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.1.0-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.2.1.0-r1`
- `docker.io/ibmcom/mq:9.2.1.0-r1`

Novedades

- [Novedades en IBM MQ 9.2.1](#)
- La información de conexión para la ruta predeterminada está disponible en la consola web de MQ

Novedades

- [Qué ha cambiado en IBM MQ 9.2.1](#)
- Basado en [Red Hat Universal Base Image 8.3-230](#)

9.2.0.6-r3-eus



Versión de operador necesaria

[1.3.8](#) y fixpacks futuros

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- Incluye IBM MQ 9.2.0 Fix Pack 6. Para obtener más información, consulte [Lista de arreglos para IBM MQ Versión 9.2 LTS](#).
- Se basa en [Red Hat Universal Base Image 8.6-941](#).

9.2.0.6-r2-eus



Versión de operador necesaria

[1.3.7](#) y fixpacks futuros

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- Incluye IBM MQ 9.2.0 Fix Pack 6. Para obtener más información, consulte [Lista de arreglos para IBM MQ Versión 9.2 LTS](#).
- Se basa en [Red Hat Universal Base Image 8.6-902](#).

9.2.0.6-r1-eus



Versión de operador necesaria

[1.3.6](#) y fixpacks futuros

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- Incluye IBM MQ 9.2.0 Fix Pack 6. Para obtener más información, consulte [Lista de arreglos para IBM MQ Versión 9.2 LTS](#).
- Se basa en [Red Hat Universal Base Image 8.6-854](#).

9.2.0.5-r3-eus



Versión de operador necesaria

[1.3.5](#) y fixpacks futuros

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- Incluye IBM MQ 9.2.0 Fix Pack 5. Para obtener más información, consulte [Cambios en IBM MQ 9.2.0 Fix Pack 5](#) y [Lista de arreglos para IBM MQ Versión 9.2 LTS](#).
- Basado en [Red Hat Universal Base Image 8.6-751.1655117800](#).

9.2.0.5-r2-eus



Versión de operador necesaria

[1.3.4](#) y fixpacks futuros

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- Incluye IBM MQ 9.2.0 Fix Pack 5. Para obtener más información, consulte [Novedades en IBM MQ 9.2.0 Fix Pack 5](#) y la lista de arreglos de [para IBM MQ Versión 9.2 LTS](#)
- Basado en [Red Hat Universal Base Image 8.6-751](#)

9.2.0.5-r1-eus



Versión de operador necesaria

[1.3.3](#) y fixpacks futuros

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- Incluye IBM MQ 9.2.0 Fix Pack 5. Para obtener más información, consulte [Novedades en IBM MQ 9.2.0 Fix Pack 5](#) y la lista de arreglos de para IBM MQ Versión 9.2 LTS
- Basado en [Red Hat Universal Base Image 8.5-240.1648458092](#)

9.2.0.4-r1-eus



Versión de operador necesaria

[1.3.2](#) y fixpacks futuros

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- Incluye IBM MQ 9.2.0 Fix Pack 4. Para obtener más información, consulte [Novedades en IBM MQ 9.2.0 Fix Pack 4](#) y la lista de arreglos de para IBM MQ Versión 9.2 LTS
- Basado en [Red Hat Universal Base Image 8.5-204](#)

9.2.0.2-r2-eus



Versión de operador necesaria

[1.6.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- Soluciona el problema con el inicio de sesión único con IBM Cloud Pak foundational services 3.7 y superior, que sólo es necesario cuando se migra de un release EUS a un release CD.
- Basado en [Red Hat Universal Base Image 8.4-200.1622548483](#)

9.2.0.2-r1-eus



Versión de operador necesaria

[1.3.1](#) y fixpacks futuros ; [1.6.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

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

Novedades

- La integración del panel de control de operaciones utiliza el agente de rastreo y el recopilador versión 1.0.8
- Incluye IBM MQ 9.2.0 Fix Pack 2. Para obtener más información, consulte [Novedades en IBM MQ 9.2.0 Fix Pack 2](#) y la lista de arreglos de para IBM MQ Versión 9.2 LTS
- Basado en [Red Hat Universal Base Image 8.4-200.1622548483](#)

9.2.0.1-r1-eus



Versión de operador necesaria

[1.3.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.1-r1-eus](#)

Novedades

- Sólo está disponible cuando se utiliza una licencia de IBM Cloud Pak for Integration
- El soporte de actualización ampliado (EUS) está disponible cuando se utiliza IBM MQ Operator 1.3.x e IBM Common Services 3.6, en Red Hat OpenShift Container Platform 4.6

Novedades

- Incluye IBM MQ 9.2.0 Fix Pack 1. Para obtener más información, consulte [Novedades en IBM MQ 9.2.0 Fix Pack 1](#) y la lista de arreglos de [para IBM MQ Versión 9.2 LTS](#)
- Basado en [Red Hat Universal Base Image 8.3-201](#)
- Soluciona el problema con el análisis de actividad (`chkmqhealthy`) y el análisis de preparación (`chkmqready`) cuando se ejecuta en `SecurityContextConstraints` que permiten el escalamiento de privilegios.

9.2.0.0-r3



Versión de operador necesaria

[1.5.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r3](#)
- [docker.io/ibmcom/mq:9.2.0.0-r3](#)

Novedades

- Basado en [Red Hat Universal Base Image 8.3-291](#)

9.2.0.0-r2



Versión de operador necesaria

[1.2.0](#) o superior

Arquitecturas soportadas

amd64, s390x

Imágenes

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r2](#)
- [docker.io/ibmcom/mq:9.2.0.0-r2](#)

Novedades

- Ahora disponible en z/Linux

Novedades

- Basado en [Red Hat Universal Base Image 8.2-349](#)

9.2.0.0-r1



Versión de operador necesaria

[1.1.0](#) o superior

Arquitecturas soportadas

amd64

Imágenes

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

Novedades

- [Novedades en IBM MQ 9.2.0](#)

Novedades

- [Qué ha cambiado en IBM MQ 9.2.0](#)
- Utiliza el argumento `-ic` en `crtmqm` para aplicar automáticamente archivos MQSC. Sustituye el uso anterior de mandatos `runmqsc`
- Basado en [Red Hat Universal Base Image 8.2-301.1593113563](#)

9.1.5.0-r2



Versión de operador necesaria

[1.0.0](#) o superior

Arquitecturas soportadas

amd64

Imágenes

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

Novedades

- Basado en [Red Hat Universal Base Image 8.2-267](#)

Migración de IBM MQ a IBM Cloud Pak for Integration

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 IBM Cloud Pak for Integration.

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 37](#)
2. [“Extracción de la configuración del gestor de colas” en la página 38](#)
3. Opcional: [“Opcional: Extracción y adquisición de las claves y certificados del gestor de colas” en la página 39](#)
4. Opcional: [“Opcional: Configuración de LDAP” en la página 41](#)
5. Opcional: [“Opcional: Cambio de las direcciones IP y los nombres de host en la configuración de IBM MQ” en la página 49](#)
6. [“Actualización de la configuración del gestor de colas para un entorno de contenedor” en la página 50](#)
7. [“Selección de la arquitectura HA de destino para IBM MQ que se ejecuta en contenedores” en la página 53](#)
8. [“Creación de los recursos para el gestor de colas” en la página 54](#)
9. [“Creación del nuevo gestor de colas en Red Hat OpenShift” en la página 55](#)
10. [“Verificación del nuevo despliegue de contenedor” en la página 59](#)

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 de IBM MQ a IBM Cloud Pak for Integration”](#) en la página 36.

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 5.
2. El IBM MQ Operator tiene un único puerto de tráfico de IBM MQ , conocido como escucha. Si tiene varios escuchas, simplifíque esto 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 114.
4. Si el sistema IBM MQ incluye alta disponibilidad, revise las opciones disponibles.
Consulte [“Alta disponibilidad para IBM MQ en contenedores”](#) en la página 16.

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 mandato siguiente:

```
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 **dspmqr** para ver los gestores de colas disponibles en la máquina. A continuación se muestra una salida de mandato **dspmqr** de ejemplo para un gestor de colas denominado qm1:

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

El mandato **dspm** 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 utilizando TLS para cifrar el tráfico en el gestor de colas. Utilice esta tarea para verificar si 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 configurado en el gestor de colas. A continuación, los canales IBM MQ habilitan la comunicación TLS. Si no está seguro de si está configurado 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 .

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 runmqckmResult <<(runmqckm -cert -list -db $keyrlocation}.kdb -stashed)
```

```

cert=1
for i in "${runmqckmResult[@]:1}"
do
    certlabel=$(echo ${i} | xargs)
    echo Extracting certificate $certlabel to $cert.cert
    runmqckm -cert -extract -db ${keyrlocation}.kdb -label "$certlabel" -target $
{cert}.cert -stashed
    cert=${cert+1}
done
fi

```

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 almacén TLS:

```
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:

```
runmqckm -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,0=cert-manager

```

- c. Extraiga cada uno de los certificados listados. Para ello, ejecute el mandato siguiente:

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

En los ejemplos mostrados anteriormente, esto equivale a lo siguiente:

```

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

```

2. 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 EUS V 9.2.1 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 EUS V9.2.1 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 41](#).

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.

Edita 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

- **CONNAME** 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) +
* COMMANDO(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: dmpmqcfc -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDTE(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.

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 41.

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.

OpenShift EUS V9.2.1 CD 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 ejemplo:

```
*****
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, suprimalos. 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 suprímala.

```

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

```

2. Defina 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 > EUS > V 9.2.1 > 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) y varias instancias (dos pods) 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 dos 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 .

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 55).

Procedimiento

1. Revise las dos opciones.

Para obtener una descripción completa de estas dos opciones, consulte “Alta disponibilidad para IBM MQ en contenedores” en la página 16.

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 mandato siguiente:

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

- c) Verifique que el archivo se ha cargado correctamente.

Ejecute el mandato siguiente:

```
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 39](#), 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 colas](#) y 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 53, hay dos topologías de despliegue posibles. Por lo tanto, este tema proporciona dos plantillas diferentes:

- [Desplegar un gestor de colas de instancia única.](#)
- [Desplegar un gestor de colas de varias instancias.](#)

Importante: Solo complete una de las dos plantillas, basándose en la topología que prefiera.

Procedimiento

- Despliegue 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: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
          items:
            - tls.key
            - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
  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: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    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 128](#).

```
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.

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

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-mjsc-migrated` y `backup.mjsc`.

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-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 instancia única. Esto completa la plantilla. Ahora está preparado para [verificar el nuevo despliegue de contenedor](#).

- Despliegue 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: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    availability: MultiInstance
  storage:
    defaultClass: aws-efs
    persistedData:
      enabled: true
    queueManager:
      enabled: true
    recoveryLogs:
      enabled: true
  mqsc:
    - configMap:
        name: my-mqsc-migrated
      items:
        - backup.mqsc
```

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](#).

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 siguiente mandato:

```
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 55](#). 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 MQSERVER=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 de IBM MQ a IBM Cloud Pak for Integration”](#) en la [página 36](#).

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 110.

Instalación y desinstalación de IBM MQ Operator en Red Hat OpenShift

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

Procedimiento

- [“Dependencias para IBM MQ Operator”](#) en la página 10.
- [“Permisos con ámbito de clúster necesarios para IBM MQ Operator”](#) en la página 10.
- [“Instalación de IBM MQ Operator utilizando la consola web de Red Hat OpenShift”](#) en la página 61.
- [“Instalación de IBM MQ Operator utilizando la CLI de Red Hat OpenShift”](#) en la página 63.
- [“Instalación de IBM MQ Operator en un entorno aislado”](#) en la página 67.

Tareas relacionadas

[“Desinstalación de IBM MQ Operator utilizando la consola web de Red Hat OpenShift”](#) en la página 63
Puede utilizar la consola web de Red Hat OpenShift para desinstalar el IBM MQ Operator de Red Hat OpenShift.

[“Desinstalación de IBM MQ Operator utilizando la CLI de Red Hat OpenShift”](#) en la página 66
Puede utilizar la CLI de Red Hat OpenShift para desinstalar el IBM MQ Operator de Red Hat OpenShift. Existen diferencias en el proceso de desinstalación, en función de si el IBM MQ Operator está instalado en un único espacio de nombres, o está instalado y disponible para todos los espacios de nombres del clúster.

Instalación de IBM MQ Operator utilizando la consola web de Red Hat OpenShift

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

Antes de empezar

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

Procedimiento

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

Nota:

Este paso se aplica a los releases de IBM MQ Operator 1.5 y anteriores. El paso añade un catálogo de Common Services independiente. Para releases posteriores del operador, los Common Services se incluyen en el catálogo de IBM .

- a) Pulse el icono más en la parte superior derecha de la pantalla. Verá el recuadro de diálogo **Importar YAML**.
- b) Pegue la siguiente definición de recurso en el recuadro de diálogo.

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

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

c) Pulse **Crear**.

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

a) Pulse el icono más en la parte superior derecha de la pantalla. Verá el recuadro de diálogo **Importar YAML**.

b) 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
```

c) Pulse **Crear**.

3. 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.

a) En el panel de navegación, pulse **Inicio > Proyectos**.

Se visualiza la página Proyectos.

b) Pulse **Crear proyecto**. Se visualiza un área Crear proyecto.

c) Especifique los detalles del espacio de nombres que está creando. Por ejemplo, puede especificar "ibm-mq" como nombre.

d) Pulse **Crear**. Se crea el espacio de nombres para el IBM MQ Operator .

4. Instale el IBM MQ Operator.

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

Se visualiza la página OperatorHub.

b) En el campo **Todos los elementos** , especifique "IBM MQ".

Se visualiza la entrada de catálogo IBM MQ .

c) Seleccione **IBM MQ**.

Se visualiza la ventana IBM MQ .

d) Pulse **Instalar**.

Verá la página Crear suscripción de operador.

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

f) Establezca la modalidad de instalación en el espacio de nombres específico que ha creado 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.

g) Pulse **Suscribir**.

Verá IBM MQ en la página Operadores instalados.

- h) Compruebe el estado del operador en la página Operadores instalados, el estado cambiará a Satisfactorio cuando se complete la instalación.

Qué hacer a continuación

[“Preparación del proyecto Red Hat OpenShift para IBM MQ utilizando la consola web de Red Hat OpenShift” en la página 82](#)

Desinstalación de IBM MQ Operator utilizando la consola web de Red Hat OpenShift

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

Antes de empezar

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

Si el IBM MQ Operator se instala en todos los proyectos/espacios de nombres del clúster, repita los pasos del 1 al 5 del procedimiento siguiente para cada proyecto en el que desee suprimir gestores de colas.

Procedimiento

1. Seleccione **Operadores > Operadores instalados**.
2. En la lista desplegable **Proyecto** , seleccione un proyecto.
3. Pulse el operador **IBM MQ** .
4. Pulse el separador **Gestores de colas** para ver los gestores de colas gestionados por este IBM MQ Operator.
5. 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.

6. Opcional: Si procede, repita los pasos del 1 al 5 para cada proyecto en el que desee suprimir gestores de colas.
7. Vuelva a **Operadores > Operadores instalados**.
8. Junto al operador **IBM MQ** , pulse el menú de tres puntos y seleccione **Desinstalar operador**.
9. Si está utilizando Red Hat OpenShift Container Platform 4.7, es posible que tenga que suprimir manualmente el gancho web de validación desde la línea de mandatos:

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

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

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

Antes de empezar

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

Procedimiento

1. 

Opcional: Cree un **CatalogSource** para los operadores de IBM Common Services .

Nota:

Este paso se aplica a los releases de IBM MQ Operator 1.5 y anteriores. El paso añade un catálogo de Common Services independiente. Para releases posteriores del operador, los Common Services se incluyen en el catálogo de IBM .

- a) Cree un archivo YAML que defina el recurso **CatalogSource** .

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

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

- b) Aplique **CatalogSource** al servidor.

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

2. Cree un **CatalogSource** para los operadores de IBM

- a) Crear un archivo YAML que defina el recurso **CatalogSource**

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

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

- b) Aplique **CatalogSource** al servidor.

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

3. 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.

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

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

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

5. Inspeccione el IBM MQ Operator para verificar sus InstallModes soportados y los canales disponibles

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

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

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

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

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

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

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

- b) Crear el objeto **OperatorGroup**

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

7. Cree un archivo YAML de objeto **Subscription** para suscribir un espacio de nombres al IBM MQ Operator

- a) Revise [“Soporte de versiones del IBM MQ Operator”](#) en la página 7 para determinar qué canal de operador elegir.
- b) Cree un archivo denominado "mq-sub.yaml" con el contenido siguiente, pero cambiando **channel** para que coincida con el canal de la versión del IBM MQ Operator que desea instalar.

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

Para el uso de **AllNamespaces InstallMode**, especifique **openshift-operators** en el espacio de nombres. De lo contrario, especifique el espacio de nombres único relevante para el uso de **SingleNamespace InstallMode**. Tenga en cuenta que sólo debe cambiar el campo **namespace**, dejando el campo **sourceNamespace** tal cual.

- c) Crear el objeto **Subscription**

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

8. Comprobar el estado del operador

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

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

Qué hacer a continuación

[“Preparación del proyecto Red Hat OpenShift para IBM MQ utilizando la CLI de Red Hat OpenShift”](#) en la página 82

OpenShift

Puede utilizar la CLI de Red Hat OpenShift para desinstalar el IBM MQ Operator de Red Hat OpenShift. Existen diferencias en el proceso de desinstalación, en función de si el IBM MQ Operator está instalado en un único espacio de nombres, o está instalado y disponible para todos los espacios de nombres del clúster.

Antes de empezar

Inicie sesión en el clúster de Red Hat OpenShift utilizando `oc login`.

Procedimiento

- 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 correcto:

```
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) Opcional: Si nada más está utilizando servicios comunes, es posible que desee desinstalar el operador de servicios comunes y suprimir el grupo de operadores:

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

- b. Vea el grupo de operadores:

```
oc get operatorgroup
```

- c. Suprima el grupo de operadores:

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

- 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) Suprima las suscripciones:

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

- g) Si utiliza Red Hat OpenShift Container Platform 4.7, es posible que tenga que suprimir manualmente el gancho web de validación:

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

- h) Opcional: Si nada más está utilizando servicios comunes, es posible que desee desinstalar el operador de servicios comunes:

Siga las instrucciones de [Desinstalación de servicios comunes](#) en la documentación del producto IBM Cloud Pak foundational services .

Instalación de IBM MQ Operator en un entorno aislado

Esta guía de aprendizaje le guía en la instalación de IBM MQ Operator en un clúster de Red Hat OpenShift que no tiene conectividad a Internet. Puede instalar IBM MQ Operator en un entorno aislado utilizando un dispositivo de almacenamiento portátil o utilizando una máquina bastión.

Instalación de IBM MQ Operator en un entorno aislado utilizando un dispositivo de almacenamiento portátil

Para ver los pasos para completar la instalación, consulte [Duplicación de imágenes con un dispositivo de almacenamiento portátil](#) en la documentación de IBM Cloud Pak for Integration . Si está instalando sólo IBM MQ, sustituya todas las apariciones de las siguientes variables de entorno por los valores que se proporcionan aquí:

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

donde *número_versión* es la versión del caso que desea utilizar para realizar la instalación aislada. Para obtener una lista de las versiones de caso disponibles, consulte <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>. Revise [“Soporte de versiones del IBM MQ Operator”](#) en la [página 7](#) para determinar qué canal de operador elegir.

Instalación de IBM MQ Operator en un entorno aislado utilizando una máquina bastión

1. [“Requisitos previos” en la página 68](#)
2. [“Preparar un registro de Docker” en la página 68](#)
3. [“Preparar un host de bastion” en la página 69](#)
4. [“Crear variables de entorno para el instalador y el inventario de imágenes” en la página 70](#)
5. [“Descargar el instalador de IBM MQ y el inventario de imágenes” en la página 70](#)
6. [“Iniciar sesión en el clúster de Red Hat OpenShift Container Platform como administrador del clúster” en la página 70](#)
7. [“Crear un espacio de nombres de Kubernetes para IBM MQ Operator” en la página 70](#)
8. [“Duplicar las imágenes y configurar el clúster” en la página 70](#)
9. [“Instale el IBM MQ Operator.” en la página 72](#)
10. [“Desplegar gestor de colas de IBM MQ” en la página 73](#)

Requisitos previos

1. Se debe instalar un clúster de Red Hat OpenShift Container Platform . Para ver las versiones de Red Hat OpenShift Container Platform soportadas, consulte [“Soporte de versiones del IBM MQ Operator” en la página 7](#).
2. Debe estar disponible un registro de Docker . Para obtener más información, consulte [“Preparar un registro de Docker” en la página 68](#).
3. Debe haber configurado un servidor de bastion. Para obtener más información, consulte [“Preparar un host de bastion” en la página 69](#).

Preparar un registro de Docker

Se utiliza un registro local de Docker para almacenar todas las imágenes en el entorno local. Debe crear este registro y debe asegurarse de que cumple los requisitos siguientes:

- Soporta [Docker Manifest V2, Esquema 2](#).
- Da soporte a imágenes de varias arquitecturas.
- Es accesible desde el servidor de bastion y desde los nodos de clúster de Red Hat OpenShift Container Platform.
- Tiene el nombre de usuario y la contraseña de un usuario que puede grabar en el registro de destino desde el host de bastion.
- Tiene el nombre de usuario y la contraseña de un usuario que puede leer el registro de destino que se encuentra en los nodos de clúster de Red Hat OpenShift.
- Debe permitir separadores de vía de acceso en el nombre de la imagen.

Después de crear el registro de Docker , debe configurar el registro:

1. Crear espacios de nombres de registro
 - `ibmcom`: espacio de nombres para almacenar todas las imágenes del espacio de nombres `dockerhub.io/ibmcom`.

El espacio de nombres `ibmcom` es para todas las imágenes de IBM que están disponibles públicamente y no requieren credenciales para la extracción.

- `cp` -Espacio de nombres para almacenar las imágenes de IBM desde el repositorio de `cp.icr.io/cp`.

El espacio de nombres `cp` es para las imágenes del registro autorizado de IBM que requieren una clave de titularidad de producto y credenciales para extraer. Para obtener la clave de titularidad, inicie sesión en [MyIBM Container Software Library](#) con el ID y la contraseña de IBM asociados con

el software autorizado. En la sección **Claves de titularidad** , seleccione **Copiar clave** para copiar la clave de titularidad en el portapapeles y, a continuación, guárdela para utilizarla en los pasos siguientes.

- `opencloudio`: espacio de nombres para almacenar las imágenes de `quay.io/opencloudio`.

El espacio de nombres `opencloudio` es para seleccionar IBM imágenes de componente de código abierto que están disponibles en quay.io. Las imágenes de IBM Cloud Pak foundational services se alojan en `opencloudio`.

2. Compruebe que cada espacio de nombres cumpla los requisitos siguientes:

- Da soporte a la creación automática de repositorio.
- Tiene credenciales de un usuario que puede grabar y crear repositorios. El host de bastion utiliza estas credenciales.
- Tiene credenciales de un usuario que puede leer todos los repositorios. El clúster de Red Hat OpenShift Container Platform utiliza estas credenciales.

Preparar un host de bastion

Prepare un host bastión que pueda acceder al clúster de Red Hat OpenShift Container Platform , al registro local de Docker e Internet. El host de bastión debe estar en una plataforma Linux for x86-64 con cualquier sistema operativo al que den soporte las CLI de IBM Cloud Pak y de Red Hat OpenShift Container Platform.

Realice estos pasos en el nodo de bastion:

1. Instale OpenSSL versión 1.1.1.1 o superior.

2. Instale Docker o Podman en el nodo bastión.

- Para instalar Docker, ejecute estos mandatos:

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

- Para instalar Podman, consulte [Instrucciones de instalación de Podman](#)

3. Instale skopeo versión 1.x.x en el nodo bastión. Para instalar skopeo, ejecute estos mandatos:

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

4. Instale la CLI de IBM Cloud Pak. Instale la última versión del archivo binario para la plataforma. Para obtener más información, consulte [cloud-pak-cli](#).

a. Descargue el archivo binario.

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

Por ejemplo:

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

b. Extraiga el archivo binario.

```
tar -xvf binary-file-name
```

c. Ejecute los mandatos siguientes para modificar y mover el archivo

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

d. Confirme que se ha instalado `cloudctl`:

```
cloudctl --help
```

5. Instale la herramienta de CLI `oc` de Red Hat OpenShift Container Platform .

Para obtener más información, consulte [Herramientas de CLI de Red Hat OpenShift Container Platform](#)

6. Cree un directorio para dar servicio al almacén fuera de línea.

El siguiente es un directorio de ejemplo. Este ejemplo se utiliza en los pasos siguientes.

```
mkdir $HOME/offline
```

Nota: Este almacén fuera de línea debe ser persistente para evitar transferir datos más de una vez. La persistencia también le ayuda a ejecutar el proceso de duplicación varias veces o según una planificación.

Crear variables de entorno para el instalador y el inventario de imágenes

Cree las variables de entorno siguientes con el nombre de imagen del instalador y el inventario de imágenes:

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

donde *número_versión* es la versión del caso que desea utilizar para realizar la instalación aislada. Para obtener una lista de las versiones de caso disponibles, consulte <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>. Revise el soporte de versión de [para el IBM MQ Operator](#) para determinar qué canal de operador elegir.

Descargar el instalador de IBM MQ y el inventario de imágenes

Descargue el instalador de `ibm-mq` y el inventario de imágenes en el host bastión:

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

Iniciar sesión en el clúster de Red Hat OpenShift Container Platform como administrador del clúster

A continuación se muestra un mandato de ejemplo para iniciar sesión en el clúster de Red Hat OpenShift Container Platform :

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

Crear un espacio de nombres de Kubernetes para IBM MQ Operator

Cree una variable de entorno con un espacio de nombres para instalar el IBM MQ Operator, a continuación, cree el espacio de nombres:

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

Duplicar las imágenes y configurar el clúster

Realice estos pasos para duplicar las imágenes y configurar el clúster:

Nota: No utilice el signo de tilde entre comillas dobles en ningún mandato. Por ejemplo, no utilice args "--registry registry --user registry_userid --pass registry_password --inputDir ~/offline". La tilde no se expande y los mandatos podrían fallar.

1. Almacenar credenciales de autenticación para todos los registros de Docker de origen.

Todos los IBM Cloud Platform Common Services, la imagen de IBM MQ Operator y la imagen de IBM MQ Advanced Developer se almacenan en registros públicos que no requieren autenticación. Sin embargo, IBM MQ Advanced Server (no desarrollador), otros productos y componentes de terceros requieren uno o más registros autenticados. Los siguientes registros requieren autenticación:

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

Para obtener más información sobre estos registros, consulte [Crear espacios de nombres de registro](#).

Para configurar el siguiente mandato para configurar todos los registros que requieren autenticación, debe ejecutar el siguiente mandato. Ejecute el mandato por separado para cada registro con este formato:

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

El mandato almacena las credenciales de registro y las guarda en memoria caché, en un archivo en el sistema de archivos que se encuentra en la ubicación `$HOME/.airgap/secrets`.

2. Cree variables de entorno con la información de conexión del registro local de Docker .

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

Nota: El registro de Docker utiliza puertos estándar como, por ejemplo, 80 o 443. Si el registro de Docker utiliza un puerto no estándar, especifique el puerto utilizando la sintaxis `host:port`. Por ejemplo:

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

3. Configure un secreto de autenticación para el registro local de Docker .

Nota: es necesario realizar este paso solo una vez.

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

El mandato almacena las credenciales de registro y las guarda en memoria caché, en un archivo en el sistema de archivos que se encuentra en la ubicación `$HOME/.airgap/secrets`.

4. Configure un secreto de extracción de imágenes global y **ImageContentSourcePolicy**.

a. Compruebe si es necesario reiniciar el nodo.

- En Red Hat OpenShift Container Platform versión 4.4 y posteriores, y en una nueva instalación de IBM MQ Operator utilizando airgap, este paso reinicia todos los nodos de clúster. Es posible que los recursos del clúster no estén disponibles hasta que se aplique el nuevo secreto de extracción.

- En IBM MQ Operator 1.8, CASE se actualiza para incluir un origen de duplicación adicional para las imágenes. Por lo tanto, cuando actualiza desde versiones anteriores de IBM MQ Operator a la versión 1.8 o superior, se desencadena un reinicio de nodo.
- Para comprobar si este paso necesita un reinicio de nodo, añada la opción `--dry-run` al código de este paso. Esto genera el último **ImageContentSourcePolicy** y lo muestra en la ventana de consola (**stdout**). Si este **ImageContentSourcePolicy** difiere del clúster configurado **ImageContentSourcePolicy**, se produce un reinicio.

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

- Para configurar el secreto de extracción de imágenes global y **ImageContentSourcePolicy**, ejecute el código para este paso sin la opción `--dry-run` :

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

- Verifique que se ha creado el recurso **ImageContentSourcePolicy**.

```
oc get imageContentSourcePolicy
```

- Opcional: si utiliza un registro no seguro, debe añadir el registro local a la lista **insecureRegistries** del clúster.

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

- Verifique el estado del nodo del clúster.

```
oc get nodes
```

Tras la aplicación de **imageContentsourcePolicy** y el secreto de extracción de imagen global, puede ver el estado del nodo como **Ready** (Preparado), **Scheduling** (Planificando) o **Disabled** (Inhabilitado). Espere hasta que todos los nodos muestren un estado de **Ready** (Preparado).

- Duplique las imágenes en el registro local.

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

Instale el IBM MQ Operator.

1. Inicie sesión en la consola web del clúster de Red Hat OpenShift .
2. Cree el origen de catálogo. Utilice el mismo terminal que ejecutó los pasos anteriores.

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


3. Verifique que se ha creado el **CatalogSource** para el operador del instalador de Common Services .

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

4. Instale IBM MQ Operator utilizando OLM.

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

Se muestra la página **OperatorHub** .

b. En el campo **Todos los elementos**, escriba IBM MQ.

Se visualiza la entrada de catálogo IBM MQ .

c. Seleccione **IBM MQ**.

Se visualiza la ventana **IBM MQ** .

d. Pulse **Instalar**.

Se muestra la página **Crear suscripción del operador**.

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

f. Establezca **Modalidad de instalación** en el espacio de nombres específico que ha creado o en el ámbito de todo el clúster.

g. Pulse **Suscribir**.

IBM MQ se añade a la página **Operadores instalados** .

h. Compruebe el estado del operador en la página **Operadores instalados** . El estado cambia a **Succeeded** cuando se completa la instalación.

Desplegar gestor de colas de IBM MQ

Para crear un nuevo gestor de colas bajo el operador instalado, consulte [“Despliegue y configuración de gestores de colas utilizando IBM MQ Operator”](#) en la [página 81](#).

Tareas relacionadas

[“Preparación para actualizar el IBM MQ Operator o el gestor de colas en un entorno aislado”](#) en la [página 73](#)

En un clúster de Red Hat OpenShift que no tiene conectividad a Internet, hay pasos preparatorios que debe realizar antes de actualizar el IBM MQ Operator.

Actualización de IBM MQ Operator y gestores de colas

La actualización de IBM MQ Operator le permitirá actualizar a continuación los gestores de colas.

Procedimiento

- [“Actualización de IBM MQ Operator utilizando la consola web de Red Hat OpenShift”](#) en la [página 77](#).
- [“Actualización de IBM MQ Operator utilizando la CLI de Red Hat OpenShift”](#) en la [página 78](#).
- [“Actualización de un gestor de colas de IBM MQ utilizando la consola web de Red Hat OpenShift”](#) en la [página 79](#).
- [“Actualización de un gestor de colas de IBM MQ utilizando la CLI de Red Hat OpenShift”](#) en la [página 80](#).

Preparación para actualizar el IBM MQ Operator o el gestor de colas en un entorno aislado

En un clúster de Red Hat OpenShift que no tiene conectividad a Internet, hay pasos preparatorios que debe realizar antes de actualizar el IBM MQ Operator.

Antes de empezar

En este tema se presupone que ya ha configurado un registro de imágenes local en el que se duplican las imágenes de IBM Cloud Pak for Integration publicadas anteriormente.

Acerca de esta tarea

Para poder actualizar el IBM MQ Operator o el gestor de colas en un entorno aislado, debe duplicar las imágenes de IBM Cloud Pak for Integration más recientes.

Tenga en cuenta que los cuatro primeros pasos de esta tarea son los mismos que los pasos que realiza cuando [“Instalación de IBM MQ Operator en un entorno aislado”](#) en la página 67.

Procedimiento

1. Cree variables de entorno para el instalador y el inventario de imágenes.

Cree las variables de entorno siguientes con el nombre de imagen del instalador y el inventario de imágenes:

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

donde *número versión* es la versión del caso que desea utilizar para realizar la instalación aislada. Para obtener una lista de las versiones de caso disponibles, consulte <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>. Revise el soporte de versión de [para el IBM MQ Operator](#) para determinar qué canal de operador elegir.

2. Descargue el instalador de IBM MQ y el inventario de imágenes.

Descargue el instalador de `ibm-mq` y el inventario de imágenes en el host bastión:

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

3. Inicie sesión en el clúster de Red Hat OpenShift Container Platform como administrador del clúster.

A continuación se muestra un mandato de ejemplo para iniciar sesión en el clúster de Red Hat OpenShift Container Platform :

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

4. Duplique las imágenes y configure el clúster.

Realice estos pasos para duplicar las imágenes y configurar el clúster:

Nota: No utilice el signo de tilde entre comillas dobles en ningún mandato. Por ejemplo, no utilice `args "--registry registry --user registry_userid --pass registry_password --inputDir ~/offline"`. La tilde no se expande y los mandatos podrían fallar.

- a. Almacenar credenciales de autenticación para todos los registros de Docker de origen.

Todos los IBM Cloud Platform Common Services, la imagen de IBM MQ Operator y la imagen de IBM MQ Advanced Developer se almacenan en registros públicos que no requieren autenticación. Sin embargo, IBM MQ Advanced Server (no desarrollador), otros productos y componentes de terceros requieren uno o más registros autenticados. Los siguientes registros requieren autenticación:

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

Para obtener más información sobre estos registros, consulte [Crear espacios de nombres de registro](#).

Para configurar el siguiente mandato para configurar todos los registros que requieren autenticación, debe ejecutar el siguiente mandato. Ejecute el mandato por separado para cada registro con este formato:

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

El mandato almacena las credenciales de registro y las guarda en memoria caché, en un archivo en el sistema de archivos que se encuentra en la ubicación `$HOME/.airgap/secrets`.

b. Cree variables de entorno con la información de conexión del registro local de Docker .

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

Nota: El registro de Docker utiliza puertos estándar como, por ejemplo, 80 o 443. Si el registro de Docker utiliza un puerto no estándar, especifique el puerto utilizando la sintaxis `host:port`. Por ejemplo:

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

c. Configure un secreto de autenticación para el registro local de Docker .

Nota: es necesario realizar este paso solo una vez.

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

El mandato almacena las credenciales de registro y las guarda en memoria caché, en un archivo en el sistema de archivos que se encuentra en la ubicación `$HOME/.airgap/secrets`.

d. Configure un secreto de extracción de imágenes global y **ImageContentSourcePolicy**.

i) Compruebe si es necesario reiniciar el nodo.

- En Red Hat OpenShift Container Platform versión 4.4 y posteriores, y en una nueva instalación de IBM MQ Operator utilizando airgap, este paso reinicia todos los nodos de clúster. Es posible que los recursos del clúster no estén disponibles hasta que se aplique el nuevo secreto de extracción.
- En IBM MQ Operator 1.8, CASE se actualiza para incluir un origen de duplicación adicional para las imágenes. Por lo tanto, cuando actualiza desde versiones anteriores de IBM MQ Operator a la versión 1.8 o superior, se desencadena un reinicio de nodo.
- Para comprobar si este paso necesita un reinicio de nodo, añada la opción `--dry-run` al código de este paso. Esto genera el último **ImageContentSourcePolicy** y lo muestra en la ventana de consola (**stdout**). Si este **ImageContentSourcePolicy** difiere del clúster configurado **ImageContentSourcePolicy**, se produce un reinicio.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-cluster-airgap \  
--namespace ${NAMESPACE} \  
--args "--dry-run"
```

```
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $
${LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline --dryRun"
```

- ii) Para configurar el secreto de extracción de imágenes global y **ImageContentSourcePolicy**, ejecute el código para este paso sin la opción `--dry-run` :

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

- e. Verifique que se ha creado el recurso **ImageContentSourcePolicy**.

```
oc get imageContentSourcePolicy
```

- f. Opcional: si utiliza un registro no seguro, debe añadir el registro local a la lista **insecureRegistries** del clúster.

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

- g. Verifique el estado del nodo del clúster.

```
oc get nodes
```

Tras la aplicación de **imageContentsourcePolicy** y el secreto de extracción de imagen global, puede ver el estado del nodo como **Ready** (Preparado), **Scheduling** (Planificando) o **Disabled** (Inhabilitado). Espere hasta que todos los nodos muestren un estado de **Ready** (Preparado).

- h. Duplique las imágenes en el registro local.

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

5. Actualice el origen de catálogo.

Utilice el mismo terminal que ejecutó los pasos anteriores.

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

Qué hacer a continuación

Ahora está preparado para actualizar el IBM MQ Operator y el gestor de colas, completando una de las tareas siguientes:

- [“Actualización de IBM MQ Operator utilizando la consola web de Red Hat OpenShift” en la página 77](#)
- [“Actualización de IBM MQ Operator utilizando la CLI de Red Hat OpenShift” en la página 78](#)
- [“Actualización de un gestor de colas de IBM MQ utilizando la consola web de Red Hat OpenShift” en la página 79](#)
- [“Actualización de un gestor de colas de IBM MQ utilizando la CLI de Red Hat OpenShift” en la página 80](#)
- [“Actualización de un gestor de colas de IBM MQ en Red Hat OpenShift utilizando el Navigator de plataforma” en la página 81](#)

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

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

Antes de empezar

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

Para poder actualizar el IBM MQ Operator en un entorno aislado, debe duplicar las imágenes de IBM Cloud Pak for Integration más recientes. Consulte [Preparación para actualizar el IBM MQ Operator o el gestor de colas en un entorno aislado](#).

Procedimiento

1. Revise [“Soporte de versiones del IBM MQ Operator”](#) en la [página 7](#) para determinar a qué canal de operador se debe actualizar.
2. Opcional: Si está actualizando desde una versión de IBM MQ Operator anterior a 1.5 a IBM MQ Operator 1.5 o posterior, primero debe actualizar la versión de IBM Cloud Pak foundational services.
Para obtener más información, consulte [“Actualización de IBM Cloud Pak foundational services utilizando la consola web de Red Hat OpenShift”](#) en la [página 77](#).
3. Actualice el IBM MQ Operator. Las nuevas versiones mayores o 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) 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 7](#).

Qué hacer a continuación


Si ha actualizado a IBM Cloud Pak foundational services 3.7, los gestores de colas que utilicen una licencia de IBM Cloud Pak for Integration deberán actualizarse o reiniciarse. Para obtener más información sobre cómo hacerlo, consulte [“Actualización de un gestor de colas de IBM MQ utilizando la consola web de Red Hat OpenShift”](#) en la [página 79](#).

Actualización de IBM Cloud Pak foundational services utilizando la consola web de Red Hat OpenShift


Si está actualizando desde una versión de IBM MQ Operator anterior a 1.5 a IBM MQ Operator 1.5 o posterior, primero debe actualizar la versión de IBM Cloud Pak foundational services.

Antes de empezar

Nota: Sólo necesita completar esta tarea si está actualizando desde una versión de IBM MQ Operator anterior a 1.5 a IBM MQ Operator 1.5 o posterior.

 Si tiene algún gestor de colas que utilice una licencia de IBM Cloud Pak for Integration , después de esta actualización, será necesario reiniciar el gestor de colas para acceder a la consola web y también verá [otros errores](#) al iniciar sesión en la consola web. Puede corregir estos errores actualizando

al valor más reciente de `.spec.version` para la versión de IBM MQ elegida, una vez completada la actualización del operador.

 Si tiene gestores de colas existentes y utiliza el panel de control de operaciones de IBM Cloud Pak for Integration, consulte [“Despliegue o actualización de IBM MQ 9.2.2 o 9.2.3 con integración de Operations Dashboard en IBM Cloud Pak for Integration 2021.4”](#) en la página 113 antes de actualizar.

Procedimiento

1. Inicie sesión en la consola web del clúster de Red Hat OpenShift .
2. En el panel de navegación, pulse **Operadores > Operadores instalados**.
Se visualizan todos los operadores instalados en el proyecto especificado.
3. Seleccione el operador de **IBM Cloud Pak foundational services**. Tenga en cuenta que antes de la versión 3.7, esto se denominaba operador **IBM Common Services**
4. Vaya al separador **Suscripción**.
5. Pulse el **Canal**.
Se visualiza la ventana **Cambiar canal de actualización de suscripción** .
6. Seleccione el canal **v3 y**, a continuación, pulse **Guardar**.
El operador de IBM Cloud Pak foundational services actualiza a la versión más reciente disponible para el nuevo canal. Consulte [“Soporte de versiones del IBM MQ Operator”](#) en la página 7.

Qué hacer a continuación

Ahora está preparado para [actualizar el IBM MQ Operator](#).

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

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

Para poder actualizar el IBM MQ Operator en un entorno aislado, debe duplicar las imágenes de IBM Cloud Pak for Integration más recientes. Consulte [Preparación para actualizar el IBM MQ Operator o el gestor de colas en un entorno aislado](#).

Procedimiento

1. Revise [“Soporte de versiones del IBM MQ Operator”](#) en la página 7 para determinar a qué canal de operador se debe actualizar.
2. Opcional: Si está actualizando desde una versión de IBM MQ Operator anterior a 1.5 a IBM MQ Operator 1.5 o posterior, primero debe actualizar la versión de IBM Cloud Pak foundational services.
Para obtener más información, consulte [“Actualización de IBM Cloud Pak foundational services utilizando la CLI de Red Hat OpenShift”](#) en la página 79.
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) 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
```

Qué hacer a continuación


Si ha actualizado a IBM Cloud Pak foundational services 3.7, los gestores de colas que utilicen una licencia de IBM Cloud Pak for Integration deberán actualizarse o reiniciarse. Para obtener más información sobre cómo hacerlo, consulte [“Actualización de un gestor de colas de IBM MQ utilizando la CLI de Red Hat OpenShift”](#) en la página 80.


Actualización de IBM Cloud Pak foundational services utilizando la CLI de Red Hat OpenShift

Si está actualizando desde una versión de IBM MQ Operator anterior a 1.5 a IBM MQ Operator 1.5 o posterior, primero debe actualizar la versión de IBM Cloud Pak foundational services.

Antes de empezar

Nota: Sólo necesita completar esta tarea si está actualizando desde una versión de IBM MQ Operator anterior a 1.5 a IBM MQ Operator 1.5 o posterior.

 Si tiene algún gestor de colas que utilice una licencia de IBM Cloud Pak for Integration , después de esta actualización, será necesario reiniciar el gestor de colas para acceder a la consola web y también verá otros errores al iniciar sesión en la consola web. Puede corregir estos errores actualizando al valor más reciente de `.spec.version` para la versión de IBM MQ elegida, una vez completada la actualización del operador.

 Si tiene gestores de colas existentes y utiliza el panel de control de operaciones de IBM Cloud Pak for Integration , consulte [“Despliegue o actualización de IBM MQ 9.2.2 o 9.2.3 con integración de Operations Dashboard en IBM Cloud Pak for Integration 2021.4”](#) en la página 113 antes de actualizar.

Procedimiento

1. Inicie sesión en el clúster con **cloudctl login** (en IBM Cloud Pak for Integration) o **oc login**.
2. Asegúrese de que el canal de actualización de v3 IBM Cloud Pak foundational services esté disponible.

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

3. Aplique un parche a Subscription para pasar al canal de actualización deseado: v3

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

Qué hacer a continuación

Ahora está preparado para [actualizar el IBM MQ Operator](#).

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

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

- 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 y gestores de colas”](#) en la página 73.

Para poder actualizar el gestor de colas en un entorno aislado, debe duplicar las imágenes de IBM Cloud Pak for Integration más recientes. Consulte [Preparación para actualizar el IBM MQ Operator o el gestor de colas en un entorno aislado](#).

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 [“Soporte de versiones del IBM MQ Operator”](#) en la [página 7](#) para obtener una correlación de canales con versiones de IBM MQ Operator y versiones 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

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 73](#).

Para poder actualizar el gestor de colas en un entorno aislado, debe duplicar las imágenes de IBM Cloud Pak for Integration más recientes. Consulte [Preparación para actualizar el IBM MQ Operator o el gestor de colas en un entorno aislado](#).

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 7](#) para obtener una correlación de canales con versiones de IBM MQ Operator y versiones de gestor de colas de IBM MQ.

Utilice el siguiente mandato:

```
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 el Navigator de plataforma

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 Navigator.

Antes de empezar

- Inicie sesión en IBM Cloud Pak for Integration Platform Navigator 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 73.

Para poder actualizar el gestor de colas en un entorno aislado, debe duplicar las imágenes de IBM Cloud Pak for Integration más recientes. Consulte [Preparación para actualizar el IBM MQ Operator o el gestor de colas en un entorno aislado](#).

Procedimiento

1. En la página de inicio de IBM Cloud Pak for Integration Platform Navigator , 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.

Despliegue y configuración de gestores de colas utilizando IBM MQ Operator

IBM MQ 9.1.5 y posteriores se despliegan en Red Hat OpenShift utilizando IBM MQ Operator.

Acerca de esta tarea

Procedimiento

- [“Preparación del proyecto Red Hat OpenShift para IBM MQ”](#) en la página 81.
- [“Despliegue de un gestor de colas en un clúster de Red Hat OpenShift Container Platform”](#) en la página 83.

Preparación del proyecto Red Hat OpenShift para IBM MQ

Prepare el clúster de Red Hat OpenShift Container Platform para que esté preparado para desplegar un gestor de colas.

Procedimiento

- [“Preparación del proyecto Red Hat OpenShift para IBM MQ utilizando la consola web de Red Hat OpenShift”](#) en la página 82.
- [“Preparación del proyecto Red Hat OpenShift para IBM MQ utilizando la CLI de Red Hat OpenShift”](#) en la página 82.

Tareas relacionadas

[“Despliegue de un gestor de colas en un clúster de Red Hat OpenShift Container Platform”](#) en la página 83

Utilice el recurso personalizado QueueManager para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform.

Preparación del proyecto Red Hat OpenShift para IBM MQ utilizando la consola web de Red Hat OpenShift

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

Antes de empezar

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

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

Acerca de esta tarea

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

Procedimiento

1. Obtenga la clave de titularidad asignada a su ID.
 - a) Inicie sesión en [MyIBM Container Software Library](#) con el ID de IBM y la contraseña asociados al software autorizado.
 - b) En la sección **Entidades de titularidad**, seleccione **Copiar clave** para copiar la clave de titularidad en el portapapeles.
2. Cree un secreto que contenga la clave de titularidad, en el proyecto donde desea desplegar el gestor de colas.
 - a) En el panel de navegación, pulse **Cargas de trabajo > Secreto**.
Se visualiza la página Secretos.
 - b) En el desplegable **Proyecto** , seleccione el proyecto en el que desea instalar IBM MQ
 - c) Pulse el botón **Crear** y seleccione **Secreto de extracción de imágenes**
 - d) En el campo **Nombre** , especifique `ibm-entitlement-key`
 - e) En el campo **Dirección de servidor de registro** , especifique `cp.icr.io`
 - f) En el campo **Nombre de usuario** , especifique `cp`
 - g) En el campo **Contraseña** , especifique la clave de titularidad que ha copiado en el paso anterior
 - h) En el campo **Correo electrónico** , especifique el ID de IBM asociado con el software autorizado

Qué hacer a continuación

[“Despliegue de un gestor de colas utilizando la consola web de Red Hat OpenShift”](#) en la página 85

Preparación del proyecto Red Hat OpenShift para IBM MQ utilizando la CLI de Red Hat OpenShift

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

Antes de empezar

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

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

Acerca de esta tarea

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

Procedimiento

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

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

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




Qué hacer a continuación

[“Despliegue de un gestor de colas utilizando la CLI de Red Hat OpenShift” en la página 86](#)

Despliegue de un gestor de colas en un clúster de Red Hat OpenShift Container Platform

Utilice el recurso personalizado QueueManager para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform.

Procedimiento

-  [“Despliegue de un gestor de colas utilizando la IBM Cloud Pak for Integration Platform Navigator” en la página 84.](#)
-  [“Despliegue de un gestor de colas utilizando la consola web de Red Hat OpenShift” en la página 85.](#)
-  [“Despliegue de un gestor de colas utilizando la CLI de Red Hat OpenShift” en la página 86.](#)

Tareas relacionadas

[“Ejemplos para configurar un gestor de colas” en la página 87](#)

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

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

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

Antes de empezar

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

Si es la primera vez que despliega un gestor de colas en este proyecto de Red Hat OpenShift , siga los pasos para [“Preparación del proyecto Red Hat OpenShift para IBM MQ”](#) en la página 81.

Procedimiento

1. Despliegue un gestor de colas.

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

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

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

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

- d) En la sección **Detalles** , compruebe o actualice el campo **Nombre** y especifique el **Espacio de nombres** en el que desea crear la instancia del gestor de colas.
- e) Si acepta el acuerdo de licencia de IBM Cloud Pak for Integration , cambie **Aceptación de licencia** a **Activado**.
- f) En la sección **Gestor de colas** , compruebe o actualice el **Nombre** del gestor de colas subyacente. En versiones anteriores de IBM Cloud Pak for Integration Platform Navigator, utilice la sección **Configuración del gestor de colas** .

De forma predeterminada, el nombre del gestor de colas utilizado por las aplicaciones cliente de IBM MQ será el mismo que el nombre del QueueManager , pero con los caracteres no válidos (como los guiones) eliminados.

- g) Pulse **Crear**

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

2. Comprobar que el gestor de colas se está ejecutando

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

Tareas relacionadas

[“Configuración de una ruta para conectarse a un gestor de colas desde fuera de un clúster de Red Hat OpenShift”](#) en la página 110

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 117](#)
Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

Despliegue de un gestor de colas utilizando la consola web de Red Hat OpenShift

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

Antes de empezar

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

Si es la primera vez que despliega un gestor de colas en este proyecto de Red Hat OpenShift , siga los pasos para [“Preparación del proyecto Red Hat OpenShift para IBM MQ” en la página 81.](#)

Procedimiento

1. Despliegue un gestor de colas.

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

a) En la consola web de Red Hat OpenShift , en el panel de navegación, pulse **Operadores > Operadores instalados**

b) Pulse **IBM MQ**.

c) Pulse el separador **Gestor de colas** .

d) Pulse el botón **Crear QueueManager** .

Se visualiza un editor YAML, que contiene el YAML de ejemplo para un recurso QueueManager .

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

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

IBM MQ está disponible bajo varias licencias diferentes. Para obtener más información sobre las licencias válidas, consulte [“Referencia de licencia para mq.ibm.com/v1beta1” en la página 128.](#) Debe aceptar la licencia para desplegar un gestor de colas.

f) Pulse **Crear**

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

2. Comprobar que el gestor de colas se está ejecutando

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

Tareas relacionadas

[“Configuración de una ruta para conectarse a un gestor de colas desde fuera de un clúster de Red Hat OpenShift” en la página 110](#)

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 117](#)
Cómo conectarse al IBM MQ Console de un gestor de colas que se ha desplegado en un clúster de Red Hat OpenShift Container Platform .

OpenShift

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

Antes de empezar

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

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

Si es la primera vez que despliega un gestor de colas en este proyecto de Red Hat OpenShift, siga los pasos para [“Preparación del proyecto Red Hat OpenShift para IBM MQ”](#) en la [página 81](#).

Procedimiento

1. Despliegue un gestor de colas.

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

a) Crear un archivo YAML de QueueManager

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"
```

Importante: Si acepta el acuerdo de licencia de IBM Cloud Pak for Integration, cambie `accept: false` por `accept: true`. Consulte [“Referencia de licencia para mq.ibm.com/v1beta1”](#) en la [página 128](#) para obtener detalles sobre la licencia.

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

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart
spec:
  version: 9.2.5.0-r3
```

```

license:
  accept: false
  license: L-APIG-BZDDDY
web:
  enabled: true
queueManager:
  name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
template:
  pod:
    containers:
      - name: qmgr
        env:
          - name: MQSNOAUT
            value: "yes"

```

Importante: 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 128](#) para obtener detalles sobre la licencia.

b) Crear el objeto QueueManager

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

2. Comprobar que el gestor de colas se está ejecutando

Puede validar el despliegue ejecutando

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

y, a continuación, comprobando el estado.

Por ejemplo, ejecute

```
oc describe queuemanager quickstart
```

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

Tareas relacionadas

[“Configuración de una ruta para conectarse a un gestor de colas desde fuera de un clúster de Red Hat OpenShift”](#) en la [página 110](#)

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 117](#)
 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

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 88](#)
- [“Ejemplo: configuración de TLS”](#) en la [página 89](#)

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.

El suministro de MQSC e INI de esta forma requiere IBM MQ Operator 1.1 o superior.

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 siguiente configuración, utilizando la línea de mandatos o utilizando IBM Cloud Pak for Integration Platform Navigator:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-cp4i
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "MQSCINI"
    mqsc:
      - configMap:
          name: mqsc-ini-example
          items:
            - example1.mqsc
            - example2.mqsc
    ini:
      - configMap:
          name: mqsc-ini-example
          items:
            - example.ini
  storage:
    queueManager:
      type: ephemeral
```

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

Información adicional:

- Un gestor de colas se puede configurar para utilizar un único Kubernetes ConfigMap o Secret (como se muestra en este ejemplo) o varios Kubernetes 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.



Ejemplo: configuración de TLS

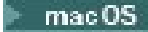
Este ejemplo despliega un gestor de colas en Red Hat OpenShift Container Platform utilizando IBM MQ Operator. La comunicación TLS unidireccional se configura entre un cliente de ejemplo y el gestor de colas. El ejemplo muestra una configuración satisfactoria colocando y obteniendo mensajes.

Antes de empezar

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

- Instale IBM MQ clienty añada `samp/bin` y `bin` a su `PATH`. Necesita las aplicaciones **runmqakm**, **amqspuic** y **amqsgetc** , que se pueden instalar como parte de IBM MQ client como se indica a continuación:

–   Para Windows y Linux: instale el cliente redistribuible de IBM MQ para el sistema operativo desde <https://ibm.biz/mq92redistclients>

–  Para Mac: Descargue y configure IBM MQ MacOS Toolkit: <https://developer.ibm.com/tutorials/mq-macos-dev/>

- Instale la herramienta OpenSSL para el sistema operativo.
- Cree un proyecto/espacio de nombres Red Hat 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 Red Hat OpenShift Container Platform. También detalla los pasos adicionales necesarios para desplegar el gestor de colas con TLS habilitado. Al finalizar, la colocación y obtención de mensajes valida que el gestor de colas se haya configurado con TLS.

Crear una clave privada TLS y certificados para el servidor IBM MQ

Los siguientes ejemplos de código muestran cómo crear un certificado autofirmado para el gestor de colas y cómo añadir el certificado a una base de datos de claves para que actúe como almacén de confianza para el cliente. Si ya tiene una clave privada y un certificado, puede utilizarlos en su lugar.

Tenga en cuenta que los certificados autofirmados solo se deben utilizar con fines de desarrollo.

Crear una clave privada autofirmada y un certificado público en el directorio actual

Ejecute el mandato siguiente:

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

Añadir la clave pública de servidor a una base de datos de claves de cliente

La base de datos de claves se utiliza como almacén de confianza para la aplicación cliente.

Cree la base de datos de claves de cliente:

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

Añada la clave pública generada anteriormente a la base de datos de claves de cliente:

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii  
-stashed
```

Configurar certificados TLS para el despliegue del gestor de colas

Para que el gestor de colas pueda hacer referencia y aplicar la clave y el certificado, cree un secreto TLS de Kubernetes, haciendo referencia a los archivos creados anteriormente. Cuando lo haga, asegúrese de que está en el espacio de nombres que ha creado antes de empezar esta tarea.

```
oc create secret tls example-tls-secret --key="tls.key" --cert="tls.crt"
```

Crear una correlación de configuración que contenga mandatos MQSC

Cree un mapa de configuración de Kubernetes 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 bloqueando sólo a los usuarios llamados *nobody*.

Tenga en cuenta que este enfoque sólo debe utilizarse con fines de desarrollo.

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 interfaz de usuario de OCP o utilizando la línea de mandatos.

```
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: example-tls-configmap  
data:  
  tls.mqsc: |  
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE  
    DEFINE CHANNEL(SECUREQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)  
    SSLCIPH('ANY_TLS12_OR_HIGHER')  
    SET CHLAUTH(SECUREQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

Crear la ruta OCP necesaria

Asegúrese de que está en el espacio de nombres que ha creado antes de empezar esta tarea y, a continuación, especifique el siguiente YAML en la interfaz de usuario de OCP o utilizando la línea de mandatos.

```
apiVersion: route.openshift.io/v1  
kind: Route  
metadata:  
  name: example-tls-route  
spec:  
  host: secureqmchl.chl.mq.ibm.com  
  to:  
    kind: Service  
    name: secureqm-ibm-mq  
  port:  
    targetPort: 1414  
  tls:  
    termination: passthrough
```

Tenga en cuenta que Red Hat OpenShift Container Platform Router utiliza SNI para direccionar solicitudes al gestor de colas IBM MQ. Si cambia el nombre de canal especificado en MQSC en la correlación de configuración creada anteriormente, también tendrá que cambiar el campo de host aquí y en el archivo CCDT creado más tarde. 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 110](#).

Desplegar el gestor de colas

Importante: En este ejemplo utilizamos la variable *MQSNOAUT* para inhabilitar la autorización en el gestor de colas, lo que nos permite centrarnos en los pasos necesarios para conectar un cliente utilizando TLS. Esto no se recomienda en un despliegue de producción de IBM MQ, porque hace que las aplicaciones que se conectan tengan poderes administrativos completos, sin ningún mecanismo para reducir los permisos para aplicaciones individuales.

Cree un nuevo gestor de colas utilizando el siguiente recurso personalizado YAML. Tenga en cuenta que hace referencia a la correlación de configuración y al secreto creados anteriormente, así como a la variable *MQSNOAUT*.

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 interfaz de usuario de OCP, utilizando la línea de mandatos o utilizando IBM Cloud Pak for Integration Platform Navigator. Compruebe que se ha especificado la licencia correcta y acepte la licencia cambiando *false* por *true*.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: secureqm
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    name: SECUREQM
  mqsc:
    - configMap:
        name: example-tls-configmap
        items:
          - tls.mqsc
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
          name: qmgr
  version: 9.2.5.0-r3
  web:
    enabled: true
  pki:
    keys:
      - name: example
        secret:
          secretName: example-tls-secret
          items:
            - tls.key
            - tls.crt
```

Confirmar 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 secureqm
```

Probar la conexión con el gestor de colas

Para confirmar que el gestor de colas está configurado para la comunicación TLS unidireccional, utilice las aplicaciones de ejemplo **amqspu~~t~~c** y **amqsget~~c~~** :

Buscar el nombre de host del gestor de colas

Utilice el mandato siguiente para buscar el nombre de host completo del gestor de colas para la ruta `secureqm-ibm-mq-qm`:

```
oc get routes secureqm-ibm-mq-qm
```

Especifique los detalles del gestor de colas

Cree un archivo CCDT .JSON que especifique los detalles del gestor de colas. Sustituya el valor de host por el nombre de host del paso anterior.

```
{
  "channel":
  [
    {
      "name": "SECUREQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<hostname from previous step>",
            "port": 443
          }
        ],
        "queueManager": "SECUREQM"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

Exportar variables de entorno

Exporte las variables de entorno siguientes, de la forma adecuada para su sistema operativo. Estas variables serán leídas por **amqsputc** y **amqsgetc**.

Actualice la vía de acceso a los archivos del sistema:

```
export MQCCDTURL='<full path to file>/CCDT.JSON'
export MQSSLKEYR='<full path to file>/clientkey'
```

Transferir mensajes a la cola

Ejecute el mandato siguiente:

```
amqsputc EXAMPLE.QUEUE SECUREQM
```

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.

Recuperar los mensajes de la cola

Ejecute el mandato siguiente:

```
amqsgetc EXAMPLE.QUEUE SECUREQM
```

Los mensajes que ha añadido en el paso anterior se han consumido y son la salida.

Después de unos segundos, el mandato sale.

Enhorabuena, ha desplegado correctamente un gestor de colas con TLS habilitado y ha mostrado que puede colocar 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 116](#).

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
```

OpenShift > CP4I Configuración de la alta disponibilidad para gestores de colas utilizando la IBM MQ Operator

Acerca de esta tarea

Procedimiento

- [V 9.2.3](#)
[“HA nativa” en la página 94.](#)
- [V 9.2.3](#)
[“Ejemplo: Configuración de un gestor de colas de HA nativa” en la página 96.](#)
- [“Ejemplo: Configuración de un gestor de colas de varias instancias” en la página 105.](#)

CP4I > V 9.2.3 > CD **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.

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

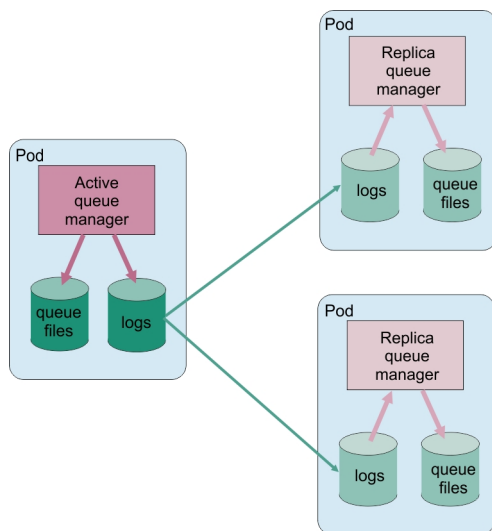


Figura 1. Ejemplo de configuración de HA nativa

CP4I V 9.2.3 CD 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-RJON-C7QG3S
    use: Production
  queueManager:
    availability:
      type: NativeHA
    version: 9.2.5.0-r3
```

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

La alta disponibilidad nativa está disponible en IBM MQ 9.2.3 o superior.

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 un gestor de colas de HA nativa”](#) en la página 96.

Referencia relacionada

“Ejemplo: Configuración de un gestor de colas de HA nativa” en la página 96



Este ejemplo muestra cómo desplegar un gestor de colas utilizando la característica de alta disponibilidad nativa en Red Hat OpenShift Container Platform (OCP) utilizando IBM MQ Operator.

 *Ejemplo: Configuración de un gestor de colas de HA nativa*

Este ejemplo muestra cómo desplegar un gestor de colas utilizando la característica de alta disponibilidad nativa en Red Hat OpenShift Container Platform (OCP) utilizando IBM MQ Operator.

Antes de empezar

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

- Instale IBM MQ client y añada los directorios `sample/bin` y `bin` instalados a su `PATH`. El cliente proporciona las aplicaciones `runmqakm`, `amqsputc` y `amqsgetc` necesarias para este ejemplo. Instale IBM MQ client como se indica a continuación:
 -  Para Windows y Linux: instale el cliente redistribuible de IBM MQ para el sistema operativo desde <https://ibm.biz/mq92redistclients>
 -  Para Mac: descargue y configure IBM MQ MacOS Toolkit. Consulte <https://ibm.biz/mqdevmacclient>
- Instale la herramienta OpenSSL para el sistema operativo. Necesita esto para generar un certificado autofirmado para el gestor de colas, si todavía no tiene una clave privada y un certificado.
- Cree un proyecto/espacio de nombres Red Hat OpenShift Container Platform (OCP) para este ejemplo y siga los pasos de la tarea [“Preparación del proyecto Red Hat OpenShift para IBM MQ”](#) en la página 81
- En la línea de mandatos, inicie sesión en el clúster de OCP y cambie al espacio de nombres que acaba de crear.
- Asegúrese de que el IBM MQ Operator esté instalado y disponible en el espacio de nombres.
- Configure una clase de almacenamiento predeterminada en el OCP, que utilizará el gestor de colas. Si desea completar esta guía de aprendizaje sin establecer una clase de almacenamiento predeterminada, consulte [Nota 2: Utilización de una clase de almacenamiento no predeterminada](#).

Acerca de esta tarea

Los gestores de colas HA nativos implican un activo y dos pods de Kubernetes de réplica. Estos se ejecutan como parte de un conjunto con estado de Kubernetes con exactamente tres réplicas y un conjunto de volúmenes persistentes de Kubernetes. Para obtener más información sobre los gestores de colas HA nativos, consulte [“Alta disponibilidad para IBM MQ en contenedores”](#) en la página 16.

El ejemplo proporciona un YAML de recurso personalizado que define un gestor de colas HA nativo que utiliza almacenamiento persistente y está configurado con TLS. Después de desplegar el gestor de colas en el OCP, simule la anomalía del pod del gestor de colas activo. Puede ver que se produce una recuperación automática y probar que se ha realizado correctamente colocando y obteniendo mensajes después de la anomalía.

Ejemplo

Crear una clave privada TLS y certificados para el servidor MQ

Puede crear un certificado autofirmado para el gestor de colas y añadir el certificado a una base de datos de claves para que actúe como almacén de confianza para el cliente. Si ya tiene una

clave privada y un certificado, puede utilizarlos en su lugar. Tenga en cuenta que solo debe utilizar certificados autofirmados para fines de desarrollo.

Para crear una clave privada autofirmada y un certificado público en el directorio actual, ejecute el mandato siguiente:

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

Crear una clave privada TLS y certificados para uso interno por parte de la HA nativa

Los tres pods de un gestor de colas de HA nativa replican datos a través de la red. Puede crear un certificado autofirmado para utilizarlo al replicar internamente. Tenga en cuenta que solo debe utilizar certificados autofirmados para fines de desarrollo.

Para crear una clave privada autofirmada y un certificado público en el directorio actual, ejecute el mandato siguiente:

```
openssl req -newkey rsa:2048 -nodes -keyout nativeha.key -subj "/CN=localhost" -x509 -days 3650 -out nativeha.crt
```

Añadir la clave pública del gestor de colas a una base de datos de claves de cliente

Una base de datos de claves de cliente se utiliza como almacén de confianza para la aplicación cliente.

Cree la base de datos de claves de cliente:

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

Añada la clave pública generada anteriormente a la base de datos de claves de cliente:

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

Crear un secreto que contenga certificados TLS para el despliegue del gestor de colas

Para que el gestor de colas pueda hacer referencia y aplicar la clave y el certificado, cree un secreto TLS de Kubernetes , haciendo referencia a los archivos creados anteriormente. Cuando lo haga, asegúrese de que está en el espacio de nombres que ha creado antes de empezar esta tarea.

```
oc create secret tls example-ha-secret --key="tls.key" --cert="tls.crt"
```

Crear un secreto que contenga la clave y el certificado TLS de HA nativa interna

Para que el gestor de colas pueda hacer referencia y aplicar la clave y el certificado, cree un secreto TLS de Kubernetes , haciendo referencia a los archivos creados anteriormente. Cuando lo haga, asegúrese de que está en el espacio de nombres que ha creado antes de empezar esta tarea.

```
oc create secret tls example-ha-secret-internal --key="nativeha.key" --cert="nativeha.crt"
```

Crear una correlación de configuración que contenga mandatos MQSC

Cree una correlación de configuración Kubernetes 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 bloqueando sólo a los usuarios llamados *nobody*.

Tenga en cuenta que este enfoque sólo debe utilizarse con fines de desarrollo.

Asegúrese de que está en el espacio de nombres que ha creado anteriormente (consulte “[Antes de empezar](#)” en la [página 96](#)) y, a continuación, especifique el YAML siguiente en la interfaz de usuario de OCP o utilizando la línea de mandatos:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-mi-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') DEFPSIST(YES) REPLACE
    DEFINE CHANNEL(HAQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
```

```
SSLCIPH('ANY_TLS12_OR_HIGHER')
SET CHLAUTH(HAQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

Configurar direccionamiento

Si utiliza un IBM MQ client o un kit de herramientas en IBM MQ 9.2.1 o posterior, puede configurar el direccionamiento al gestor de colas utilizando un archivo de configuración del gestor de colas (un archivo INI). En el archivo, establezca la variable *OutboundSNI* para direccionar basándose en el nombre de host en lugar del nombre de canal.

Cree un archivo en el directorio en el que está ejecutando los mandatos, denominado `mqclient.ini`, que contenga exactamente el texto siguiente:

```
SSL:
  OutboundSNI=HOSTNAME
```

No cambie ningún valor en este archivo INI. Por ejemplo, la serie `HOSTNAME` no debe cambiarse.

Para obtener más detalles, consulte [Stanza SSL del archivo de configuración de cliente](#).

Si utiliza un IBM MQ client o un kit de herramientas anterior a IBM MQ 9.2.1, debe crear una ruta OCP en lugar del archivo de configuración anterior. Siga los pasos de la [Nota 1: Creación de una ruta](#).

Desplegar el gestor de colas

Importante: En este ejemplo utilizamos la variable *MQSNOAUT* para inhabilitar la autorización en el gestor de colas, lo que nos permite centrarnos en los pasos necesarios para conectar un cliente utilizando TLS. Esto no se recomienda en un despliegue de producción de IBM MQ, porque hace que las aplicaciones que se conectan tengan poderes administrativos completos, sin ningún mecanismo para reducir los permisos para aplicaciones individuales.

Copie y actualice el siguiente YAML.

- Asegúrese de que se ha especificado la licencia correcta. Consulte [Referencia de licencias para mq.ibm.com/v1beta1](#). En IBM Cloud Pak for Integration 2021.1.1, la licencia debe ser la licencia de evaluación `L-RJON-BYRMYW`
- Acepte la licencia cambiando `false` a `true`.

YAML de recurso personalizado del gestor de colas:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: nativeha-example
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    name: HAEXAMPLE
    availability:
      type: NativeHA
    tls:
      secretName: example-ha-secret-internal
  mqsc:
    - configMap:
        name: example-mi-configmap
        items:
          - tls.mqsc
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
          name: qmgr
  version: 9.2.5.0-r3
  pki:
    keys:
      - name: example
        secret:
          secretName: example-ha-secret
```

```
items:
- tls.key
- tls.crt
```

Asegúrese de que está en el espacio de nombres creado anteriormente, despliegue el YAML actualizado, utilizando la consola web de Red Hat OpenShift Container Platform , la línea de mandatos o utilizando IBM Cloud Pak for Integration Platform Navigator.

Hay un breve retardo mientras el sistema configura el gestor de colas HA nativa, después del cual el gestor de colas debe estar disponible para su uso.

Validación

En esta sección validamos que el gestor de colas se comporta como se esperaba.

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 nativeha-example
```

Probar la conexión con el gestor de colas

Para confirmar que el gestor de colas está configurado para la comunicación TLS unidireccional, utilice las aplicaciones de ejemplo **amqsputc** y **amqsgetc** :

Buscar el nombre de host del gestor de colas

Para buscar el nombre de host del gestor de colas para la ruta `nativeha-example-ibm-mq-qm`, ejecute el mandato siguiente. El nombre de host se devuelve en el campo `HOST` .

```
oc get routes nativeha-example-ibm-mq-qm
```

Especifique los detalles del gestor de colas

Cree un archivo CCDT .JSON que especifique los detalles del gestor de colas. Sustituya el valor de host por el nombre de host devuelto por el paso anterior.

```
{
  "channel":
  [
    {
      "name": "HAQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<host from previous step>",
            "port": 443
          }
        ],
        "queueManager": "HAEXAMPLE"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

Exportar variables de entorno

Exporte las variables de entorno siguientes, de la forma adecuada para su sistema operativo. Estas variables serán leídas por **amqsputc** y **amqsgetc**.

Actualice la vía de acceso a los archivos del sistema:

```
export MQCCDTURL='<full_path_to_file>/CCDT.JSON'
export MQSSLKEYR='<full_path_to_file>/clientkey'
```

Transferir mensajes a la cola

Ejecute el mandato siguiente:

```
amqsputc EXAMPLE.QUEUE HAEXAMPLE
```

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.

Recuperar los mensajes de la cola

Ejecute el mandato siguiente:

```
amqsgetc EXAMPLE.QUEUE HAEXAMPLE
```

Los mensajes que ha añadido en el paso anterior se han consumido y son la salida.

Después de unos segundos, el mandato sale.

Forzar que falle el pod activo

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

Ver los pods activo y en espera

Ejecute el mandato siguiente:

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

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.

Suprimir el pod activo

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

```
oc delete pod nativeha-example-ibm-mq-<value>
```

Volver a ver el estado del pod

Ejecute el mandato siguiente:

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

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 HAEXAMPLE
```

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

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

Colocar y obtener mensajes de nuevo

Después de que el pod en espera se convierta en el pod activo (es decir, después de que el valor del campo **READY** se convierta en 1/1), utilice de nuevo los mandatos siguientes, como se ha descrito anteriormente, para transferir mensajes al gestor de colas y, a continuación, recuperar mensajes del gestor de colas:

```
amqsputc EXAMPLE.QUEUE HAEXAMPLE
```

```
amqsgetc EXAMPLE.QUEUE HAEXAMPLE
```

Enhorabuena, ha desplegado correctamente un gestor de colas HA nativo y ha demostrado que se puede recuperar automáticamente de una anomalía de pod.

Información adicional

Nota 1: Creación de una ruta

Si utiliza un IBM MQ client o un kit de herramientas anterior a IBM MQ 9.2.1, debe crear una ruta.

Para crear la ruta, asegúrese de que está en el espacio de nombres que ha creado anteriormente (consulte [“Antes de empezar” en la página 96](#)) y, a continuación, especifique el YAML siguiente en la consola web de Red Hat OpenShift Container Platform o utilizando la línea de mandatos:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: example-mi-route
spec:
  host: hamqchl.chl.mq.ibm.com
  to:
    kind: Service
    name: nativeha-example-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Tenga en cuenta que Red Hat OpenShift Container Platform Router utiliza SNI para direccionar solicitudes al gestor de colas IBM MQ . Si cambia el nombre de canal especificado en el [mapa de configuración que contiene los mandatos MQSC](#), también debe cambiar el campo de host aquí y en el archivo CCDT .JSON que especifica los detalles del gestor de colas. 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 110](#).

Nota 2: Utilización de una clase de almacenamiento no predeterminada


Este ejemplo espera que se haya configurado una clase de almacenamiento predeterminada en Red Hat OpenShift Container Platform, por lo que no se necesita información de almacenamiento en el [recurso personalizado del gestor de colas YAML](#). 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`.

El nombre de clase de almacenamiento debe coincidir exactamente con el nombre de una clase de almacenamiento que ya existe. Es decir, debe coincidir con el nombre devuelto por el mandato `oc get storageclass`. También debe dar soporte a `ReadWriteMany`. Para obtener más información, consulte [“Consideraciones sobre el almacenamiento para IBM MQ Operator” en la página 11](#).

Tareas relacionadas

[“Visualización del estado de los gestores de colas de HA nativa para contenedores certificados de IBM MQ” en la página 101](#)

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

 [Visualización del estado de los gestores de colas de HA nativa para contenedores certificados de IBM MQ](#)

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

Acerca de esta tarea

Importante:

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, Replicao Unknown.

INSTANCIA

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

CONNACTV

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).

ALTTIME

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 dspmq en uno de los pods

```
oc exec -t Pod dspmq
```

```
oc rsh Pod
```

para un shell interactivo, donde puede ejecutar dspmq directamente.

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

```
oc exec -t Pod dspmq -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 dspmq -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) 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)
```

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) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

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) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

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 dspmq \(visualizar gestores de colas\)](#)

“Ejemplo: Configuración de un gestor de colas de HA nativa” en la página 96

Este ejemplo muestra cómo desplegar un gestor de colas utilizando la característica de alta disponibilidad nativa en Red Hat OpenShift Container Platform (OCP) utilizando IBM MQ Operator.

CP4I **V 9.2.3** **CD** *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 88 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 $2 \times \text{HeartbeatInterval}$ 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 $2 \times \text{HeartbeatInterval}$ antes de reintentar un enlace de réplica fallido.

CP4I *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.

CP4I **V 9.2.2** **CD** *Evaluación de la característica HA nativa en IBM Cloud Pak for Integration 2021.1.1*

El periodo de evaluación de HA nativa de IBM Cloud Pak for Integration 2021.1.1 ha finalizado. Utilice la característica HA nativa actualizada que está disponible en IBM Cloud Pak for Integration 2021.2.1, utilizando IBM MQ Operator 1.6 o superior con IBM MQ 9.2.3 o superior.

Tareas relacionadas

“Visualización del estado de los gestores de colas de HA nativa para contenedores certificados de IBM MQ” en la [página 101](#)

Para los contenedores certificados 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.

Referencia relacionada

“Ejemplo: Configuración de un gestor de colas de HA nativa” en la [página 96](#)




Este ejemplo muestra cómo desplegar un gestor de colas utilizando la característica de alta disponibilidad nativa en Red Hat OpenShift Container Platform (OCP) utilizando IBM MQ Operator.

Ejemplo: Configuración de un gestor de colas de varias instancias

Este ejemplo muestra cómo desplegar un gestor de colas de varias instancias en Red Hat OpenShift Container Platform (OCP) utilizando IBM MQ Operator. En este ejemplo, también configura la comunicación TLS unidireccional entre un cliente de ejemplo y el gestor de colas. El ejemplo muestra una configuración satisfactoria colocando y obteniendo mensajes antes y después de un error de pod simulado.

Antes de empezar

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

- Instale IBM MQ client y añada los directorios `sample/bin` y `bin` instalados a su `PATH`. El cliente proporciona las aplicaciones **runmqakm**, **amqsputc** y **amqsgetc** necesarias para este ejemplo. Instale IBM MQ client como se indica a continuación:
 -   Para Windows y Linux: instale el cliente redistribuible de IBM MQ para el sistema operativo desde <https://ibm.biz/mq92redistclients>
 -  Para Mac: descargue y configure IBM MQ MacOS Toolkit. Consulte <https://developer.ibm.com/tutorials/mq-macos-dev/>.
- Instale la herramienta OpenSSL para el sistema operativo. Necesita esto para generar un certificado autofirmado para el gestor de colas, si todavía no tiene una clave privada y un certificado.
- Cree un proyecto/espacio de nombres Red Hat 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.
- Configure una clase de almacenamiento predeterminada en el OCP, que utilizará el gestor de colas. Si desea completar esta guía de aprendizaje sin establecer una clase de almacenamiento predeterminada, consulte [Nota 2: Utilización de una clase de almacenamiento no predeterminada](#).

Acerca de esta tarea

Los gestores de colas de varias instancias implican un pod de Kubernetes activo y en espera. Estos se ejecutan como parte de un conjunto con estado de Kubernetes con exactamente dos réplicas y un conjunto de volúmenes persistentes de Kubernetes . Para obtener más información sobre los gestores de colas de varias instancias, consulte [“Alta disponibilidad para IBM MQ en contenedores”](#) en la [página 16](#).

El ejemplo proporciona un YAML de recurso personalizado que define un gestor de colas de varias instancias con almacenamiento persistente y configurado con TLS. Después de desplegar el gestor de colas en el OCP, simule la anomalía del pod del gestor de colas activo. Puede ver que se produce una recuperación automática y probar que se ha realizado correctamente colocando y obteniendo mensajes después de la anomalía.

Ejemplo

Crear una clave privada TLS y certificados para el servidor MQ

Esta sección documenta cómo crear un certificado autofirmado para el gestor de colas y cómo añadir el certificado a una base de datos de claves para que actúe como almacén de confianza para el cliente. Si ya tiene una clave privada y un certificado, puede utilizarlos en su lugar. Tenga en cuenta que solo debe utilizar certificados autofirmados para fines de desarrollo.

Para crear una clave privada autofirmada y un certificado público en el directorio actual, ejecute el mandato siguiente:

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

Añadir la clave pública del gestor de colas a una base de datos de claves de cliente

Una base de datos de claves de cliente se utiliza como almacén de confianza para la aplicación cliente.

Cree la base de datos de claves de cliente:

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

Añada la clave pública generada anteriormente a la base de datos de claves de cliente:

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

Crear un secreto que contenga certificados TLS para el despliegue del gestor de colas

Para que el gestor de colas pueda hacer referencia y aplicar la clave y el certificado, cree un secreto TLS de Kubernetes, haciendo referencia a los archivos creados anteriormente. Cuando lo haga, asegúrese de que está en el espacio de nombres que ha creado antes de empezar esta tarea.

```
oc create secret tls example-mi-secret --key="tls.key" --cert="tls.crt"
```

Crear una correlación de configuración que contenga mandatos MQSC

Cree una correlación de configuración Kubernetes 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 bloqueando sólo a los usuarios llamados *nobody*.

Tenga en cuenta que este enfoque sólo debe utilizarse con fines de desarrollo.

Asegúrese de que está en el espacio de nombres que ha creado anteriormente (consulte [“Antes de empezar”](#) en la página 105) y, a continuación, especifique el YAML siguiente en la interfaz de usuario de OCP o utilizando la línea de mandatos:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-mi-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') DEFPSIST(YES) REPLACE
    DEFINE CHANNEL(MIQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    SET CHLAUTH(MIQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

Configurar direccionamiento

Si utiliza un IBM MQ client o un kit de herramientas en IBM MQ 9.2.1 o posterior, puede configurar el direccionamiento al gestor de colas utilizando un archivo de configuración del gestor de colas (un archivo INI). En el archivo, establezca la variable *OutboundSNI* para direccionar basándose en el nombre de host en lugar del nombre de canal.

Cree un archivo en el directorio en el que está ejecutando los mandatos, denominado `mqclient.ini`, que contiene el texto siguiente:

```
##* Module Name: mqclient.ini                                     *#
```

```

##* Type      : IBM MQ MQI client configuration file      *##
## Function   : Define the configuration of a client      *##
##*          *##
##*****#
##* Notes    :                                           *##
##* 1) This file defines the configuration of a client   *##
##*          *##
##*****#
SSL:
  OutboundSNI=HOSTNAME

```

Nota: No cambie ningún valor en esta página. Por ejemplo, la serie HOSTNAME debe dejarse tal cual. Para obtener más detalles, consulte [Stanza SSL del archivo de configuración de cliente](#).

Si utiliza un IBM MQ client o un kit de herramientas anterior a IBM MQ 9.2.1, debe crear una ruta OCP en lugar del archivo de configuración anterior. Siga los pasos de la [Nota 1: Creación de una ruta](#).

Desplegar el gestor de colas

Importante: En este ejemplo utilizamos la variable `MQSNOAUT` para inhabilitar la autorización en el gestor de colas, lo que nos permite centrarnos en los pasos necesarios para conectar un cliente utilizando TLS. Esto no se recomienda en un despliegue de producción de IBM MQ, porque hace que las aplicaciones que se conectan tengan poderes administrativos completos, sin ningún mecanismo para reducir los permisos para aplicaciones individuales.

Copie y actualice el siguiente YAML.

- Asegúrese de que se ha especificado la licencia correcta. Consulte [Referencia de licencias para mq.ibm.com/v1beta1](#).
- Acepte la licencia cambiando `false` a `true`.
- Si utiliza IBM Cloud File Storage, consulte [Nota 3: Utilización de IBM Cloud File Storage](#)

YAML de recurso personalizado del gestor de colas:

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: miexample
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  queueManager:
    name: MIEXAMPLE
    availability:
      type: MultiInstance
  mqsc:
    - configMap:
        name: example-mi-configmap
        items:
          - tls.mqsc
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
            name: qmgr
  version: 9.2.5.0-r3
  web:
    enabled: true
  pki:
    keys:
      - name: example
        secret:
          secretName: example-mi-secret
          items:
            - tls.key
            - tls.crt

```

Asegúrese de que está en el espacio de nombres que ha creado anteriormente, despliegue el YAML actualizado en la interfaz de usuario de OCP, utilizando la línea de mandatos o utilizando IBM Cloud Pak for Integration Platform Navigator.

Validación

Después de un breve retardo, el gestor de colas de varias instancias debe estar configurado y disponible para su uso. En esta sección validamos que el gestor de colas se comporta como se esperaba.

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 miexample
```

Probar la conexión con el gestor de colas

Para confirmar que el gestor de colas está configurado para la comunicación TLS unidireccional, utilice las aplicaciones de ejemplo **amqsputc** y **amqsgetc** :

Buscar el nombre de host del gestor de colas

Para buscar el nombre de host del gestor de colas para la ruta `miexample-ibm-mq-qm`, ejecute el mandato siguiente. El nombre de host se devuelve en el campo `HOST` .

```
oc get routes miexample-ibm-mq-qm
```

Especifique los detalles del gestor de colas

Cree un archivo CCDT .JSON que especifique los detalles del gestor de colas. Sustituya el valor de host por el nombre de host devuelto por el paso anterior.

```
{
  "channel":
  [
    {
      "name": "MIQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<host from previous step>",
            "port": 443
          }
        ],
        "queueManager": "MIEXAMPLE"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

Exportar variables de entorno

Exporte las variables de entorno siguientes, de la forma adecuada para su sistema operativo. Estas variables serán leídas por **amqsputc** y **amqsgetc**.

Actualice la vía de acceso a los archivos del sistema:

```
export MQCCDTURL='<full_path_to_file>/CCDT.JSON'
export MQSSLKEYR='<full_path_to_file>/clientkey'
```

Transferir mensajes a la cola

Ejecute el mandato siguiente:

```
amqsputc EXAMPLE.QUEUE MIEXAMPLE
```

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.

Recuperar los mensajes de la cola

Ejecute el mandato siguiente:

```
amqsgetc EXAMPLE.QUEUE MIEXAMPLE
```

Los mensajes que ha añadido en el paso anterior se han consumido y son la salida.

Después de unos segundos, el mandato sale.

Forzar que falle el pod activo

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

Ver los pods activo y en espera

Ejecute el mandato siguiente:

```
oc get pods
```

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.

Suprimir el pod activo

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

```
oc delete pod miexample-ibm-mq-<value>
```

Volver a ver el estado del pod

Ejecute el mandato siguiente:

```
oc get pods
```

Ver el registro de pod en espera

Ejecute el mandato siguiente, especificando el nombre completo del pod en espera:

```
oc logs miexample-ibm-mq-<value>
```

Debería ver el siguiente mensaje:

```
IBM MQ queue manager 'MIEXAMPLE' becoming the active instance.
```

Colocar y obtener mensajes de nuevo

Después de que el pod en espera se convierta en el pod activo (es decir, después de que el valor del campo **READY** se convierta en 1/1), utilice de nuevo los mandatos siguientes, como se ha descrito anteriormente, para transferir mensajes al gestor de colas y, a continuación, recuperar mensajes del gestor de colas:

```
amqsputc EXAMPLE.QUEUE MIEXAMPLE
```

```
amqsgetc EXAMPLE.QUEUE MIEXAMPLE
```

Enhorabuena, ha desplegado correctamente un gestor de colas de varias instancias y ha demostrado que se puede recuperar automáticamente de una anomalía de pod.

Información adicional

Nota 1: Creación de una ruta

Si está utilizando un IBM MQ client o un kit de herramientas anterior a IBM MQ 9.2.1, debe crear una ruta OCP.

Para crear la ruta, asegúrese de que está en el espacio de nombres que ha creado anteriormente (consulte [“Antes de empezar”](#) en la página 105) y, a continuación, especifique el YAML siguiente en la interfaz de usuario de OCP o utilizando la línea de mandatos:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: example-mi-route
spec:
  host: miqmchl.chl.mq.ibm.com
  to:
    kind: Service
    name: miexample-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Tenga en cuenta que Red Hat OpenShift Container Platform Router utiliza SNI para direccionar solicitudes al gestor de colas IBM MQ . Si cambia el nombre de canal especificado en el [mapa de configuración que contiene los mandatos MQSC](#), también debe cambiar el campo de host aquí y en el [archivo CCDT .JSON que especifica los detalles del gestor de colas](#). 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 110.

Nota 2: Utilización de una clase de almacenamiento no predeterminada

Este ejemplo espera que se haya configurado una clase de almacenamiento predeterminada en OCP, por lo que no se necesita información de almacenamiento en el [recurso personalizado del gestor de colas YAML](#). 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`.

El nombre de clase de almacenamiento debe coincidir exactamente con el nombre de una clase de almacenamiento que exista en el sistema OCP. Es decir, debe coincidir con el nombre devuelto por el mandato `oc get storageclass`. También debe dar soporte a `ReadWriteMany`. Para obtener más información, consulte [“Consideraciones sobre el almacenamiento para IBM MQ Operator”](#) en la página 11.

Nota 3: Utilización de IBM Cloud File Storage

En algunas situaciones, por ejemplo, cuando se utiliza IBM Cloud File Storage, también es necesario especificar el campo `securityGroups` en el [YAML de recurso personalizado del gestor de colas](#). Por ejemplo, añadiendo el siguiente campo hijo directamente bajo `spec` :

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

Para obtener más información, consulte [“Consideraciones sobre el almacenamiento para IBM MQ Operator”](#) en la página 11.

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



Atención: Este documento se aplica a las versiones 9.2.1 Continuous Delivery y posteriores de los clientes IBM MQ . Si el cliente utiliza la versión 9.2.0 Long Term Support o anterior, consulte la IBM

MQ 9.1 página de documentación [Conexión a un gestor de colas desplegado en un clúster de Red Hat OpenShift](#).

V 9.2.1 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](#).

V 9.2.1 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.

V 9.2.1

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
- Clientes .NET en modalidad gestionada (antes de IBM MQ 9.2.0 Fix Pack 4 para Long Term Support y antes de IBM MQ 9.2.3 para Continuous Delivery.)

V 9.2.0.4 **V 9.2.3**

A partir de IBM MQ 9.2.0 Fix Pack 4 para Long Term Support y IBM MQ 9.2.3 para Continuous Delivery, el cliente IBM MQ gestionado .NET se ha actualizado para establecer SERVERNAME en el nombre de host respectivo si la propiedad **OutboundSNI** se establece en HOSTNAME, lo que permite a un cliente IBM MQ gestionado .NET conectarse a un gestor de colas utilizando rutas de Red Hat OpenShift . Tenga en cuenta que, en IBM MQ 9.2.0 Fix Pack 4, la propiedad **OutboundSNI** sólo se añade y se soporta desde el archivo mqclient.ini ; no puede establecer la propiedad desde la aplicación .NET.

V 9.2.5

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: <provide a unique name for the Route>
  namespace: <the namespace of your MQ deployment>
spec:
  host: <SNI address mapping for the channel>
  to:
    kind: Service
    name: <the name of the Kubernetes Service for your MQ deployment (for example "<Queue Manager Name>-ibm-mq")>
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

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 117

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

CP4I Integración con el panel de control de operaciones de IBM Cloud Pak for Integration

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

Acerca de esta tarea

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

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

Tenga en cuenta también que para las versiones de IBM MQ Operator anteriores a 1.5, cuando el rastreo está habilitado, el agente de rastreo y las imágenes de recopilador desplegadas junto con el gestor de colas siempre eran las últimas versiones disponibles, lo que puede introducir una incompatibilidad si no está utilizando la última versión de IBM Cloud Pak for Integration.

Procedimiento

1. Desplegar un gestor de colas con el rastreo habilitado

De forma predeterminada, la característica de rastreo está inhabilitada.

Si está desplegando utilizando IBM Cloud Pak for Integration Platform Navigator, puede habilitar el rastreo durante el despliegue, estableciendo **Habilitar rastreo en Activado** y estableciendo el **Espacio de nombres de rastreo** en el espacio de nombres donde está instalado el panel de control de operaciones. Para obtener más información sobre el despliegue de un gestor de colas, consulte [“Despliegue de un gestor de colas utilizando la IBM Cloud Pak for Integration Platform Navigator” en la página 84](#)

Si está realizando el despliegue utilizando la CLI de Red Hat OpenShift o la consola web de Red Hat OpenShift, puede habilitar el rastreo con el siguiente fragmento de código YAML:

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

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

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

2. Si es la primera vez que se ha desplegado un gestor de colas con integración de panel de control de operaciones en este espacio de nombres, habrá que [Registrarse](#) en el panel de control de operaciones.

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

CP4I CD Despliegue o actualización de IBM MQ 9.2.2 o 9.2.3 con integración de Operations Dashboard en IBM Cloud Pak for Integration 2021.4

Cada versión de IBM MQ está asociada con una versión específica del agente del panel de control de operaciones y los componentes de recopilador, que se despliegan junto con un gestor de colas. IBM Cloud Pak for Integration 2021.4.1 introduce un cambio que hace que los componentes de agente y recopilador más antiguos no funcionen con el panel de control de operaciones. Para arreglarlo, debe

alterar temporalmente la versión del agente del panel de control de operaciones y las imágenes de recopilador que utilice, cuando utilice IBM MQ 9.2.2 o 9.2.3.

Despliegue de un nuevo gestor de colas IBM MQ 9.2.2 o 9.2.3

Al utilizar IBM Cloud Pak for Integration 2021.4.1 con IBM MQ 9.2.2 o 9.2.3, debe alterar temporalmente el agente del panel de control de operaciones y las imágenes de recopilador a las versiones de 2.4 en el archivo `YAML` de `QueueManager` . Por ejemplo:

```
spec:
  tracing:
    agent:
      image: cp.icr.io/cp/icp4i/od/icp4i-od-agent@sha256:27a211f0f78eff765d1f9520e0f9841f902600bb556827477b206e209cb44d20
    collector:
      image: cp.icr.io/cp/icp4i/od/icp4i-od-collector@sha256:dc70b1341b23dc72642ce68809811f9db0e8a0c46bda2508e8eb3d4035e04f4b
```

Si no lo hace, el pod `QueueManager` se atascará en el estado `Pending` . Al actualizar a IBM MQ 9.2.4, puede eliminar estas alteraciones temporales.

Actualización a IBM Cloud Pak for Integration 2021.4.1

Nota: Si mantiene el gestor de colas IBM MQ 9.2.2 o 9.2.3 , no complete el paso 3.

1. Actualice `QueueManager` para alterar temporalmente las imágenes de agente y recopilador, tal como se ha descrito anteriormente.
2. Actualice los operadores de IBM Cloud Pak for Integration , incluidos el panel de control de operaciones y el operador de IBM MQ , tal como se describe en [“Actualización de IBM MQ Operator y gestores de colas”](#) en la página 73.
3. (opcional) Para actualizar a IBM MQ 9.2.4 o posterior, actualice `QueueManager` para utilizar `.spec.version` para la versión de IBM MQy, a continuación, elimine la alteración temporal de las imágenes de agente y recopilador.

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 [“Preparación del proyecto Red Hat OpenShift para IBM MQ”](#) en la página 81.

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:9.2.5.0-r3
      RUN printf "DEFINE QLOCAL(foo) REPLACE\n" > /etc/mqm/my.mqsc \
        && printf "Channels:\n\tMQIBindType=FASTPATH\n" > /etc/mqm/my.ini
      LABEL summary "My custom MQ image"
  strategy:
    type: Docker
    dockerStrategy:
      from:
        kind: "DockerImage"
        name: "cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

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 20 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 en un clúster de Red Hat OpenShift Container Platform](#)” en la página 83, 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 en un clúster de Red Hat OpenShift Container Platform](#)” en la página 83

Utilice el recurso personalizado QueueManager para desplegar un gestor de colas en un clúster de Red Hat OpenShift Container Platform.

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.

El soporte para comprobaciones de tiempo de ejecución se ha introducido en IBM MQ Operator 1.2.

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"
```

Funcionamiento de IBM MQ utilizando la IBM MQ Operator

Acerca de esta tarea

Procedimiento

- [“Preparación del proyecto Red Hat OpenShift para IBM MQ”](#) en la página 81.
- [“Despliegue de un gestor de colas en un clúster de Red Hat OpenShift Container Platform”](#) en la página 83.

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 Navigator. De forma alternativa, se puede encontrar en la CLI de Red Hat OpenShift ejecutando el mandato siguiente:

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

Si está utilizando una licencia de IBM Cloud Pak for Integration , la consola web de IBM MQ se configura para utilizar IBM Cloud Pak Identity and Access Manager (IAM). Es posible que el administrador del clúster ya haya configurado el componente de IAM. Sin embargo, si es la primera vez que se utiliza IAM en el clúster de Red Hat OpenShift , tendrá que recuperar la contraseña de administrador inicial. Para obtener más información, consulte [Obtención de la contraseña de administrador inicial](#).

Si está utilizando una licencia de IBM MQ , la consola web de MQ no está preconfigurada y debe configurarla usted mismo. Para obtener más información, consulte [Configuración de usuarios y roles](#).

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 110

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 .

Otorgar permisos para IBM MQ Console utilizando IBM Cloud Pak IAM

Los permisos para IBM MQ Console se gestionan a través del IBM Cloud Pak Administration Hub, y no el IBM Cloud Pak for Integration Platform Navigator. IBM MQ no utiliza los permisos de "Automatización" proporcionados por IBM Cloud Pak for Integration, sino que utiliza los permisos básicos habilitados por IBM Cloud Pak Identity and Access Manager (IAM).

Procedimiento

1. Abra la consola de administración de IBM Cloud Pak.

En IBM Cloud Pak for Integration Platform UI, pulse el conmutador Cloud Pak (icono de 9 puntos) en la esquina superior derecha de la barra de herramientas y, a continuación, pulse el panel **IBM Cloud Pak** .

2. En el menú de navegación de la esquina superior izquierda, seleccione **Identidad y accesos**, a continuación, seleccione **Equipos e ID de servicios**.

3. Cree un equipo y, a continuación, añádale usuarios.

- a) Seleccione **Crear equipo**.

- b) Especifique un nombre de equipo y, a continuación, seleccione el dominio de seguridad para los usuarios que desea gestionar.

- c) Buscar usuarios

Estos usuarios ya deben existir en el proveedor de identidad.

- d) Cuando encuentre cada usuario, dele un rol. Debe ser "Administrador" o "Administrador de clústeres", para administrar IBM MQ utilizando IBM MQ Console.

4. Añada cada usuario a un espacio de nombres.

- a) Seleccione el equipo para editarlo.

- b) Seleccione **Recursos > Gestionar recursos**.

- c) Seleccione los espacios de nombres que desea que administre este equipo. Estos pueden ser cualquier espacio de nombres con un gestor de colas.

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.

Red Hat OpenShift Container Platform incluye una pila de supervisión preconfigurada, preinstalada y de actualización automática que utiliza un servidor Prometheus . La pila de supervisión de Red Hat 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.

En versiones anteriores de IBM Cloud Pak for Integration, también podía utilizar el servicio [IBM Cloud Platform Monitoring](#) para proporcionar un servidor Prometheus en su lugar.

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
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

Métrica	Tipo	Descripción
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
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

Métrica	Tipo	Descripción
ibmmq_qmgr_log_logical_written_bytes_total	counter	Registro - bytes lógicos escritos
ibmmq_qmgr_log_max_bytes	gauge	Registro – máx bytes
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
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/MQCONN
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

Métrica	Tipo	Descripción
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
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_mqput_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

Métrica	Tipo	Descripción
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
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_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

Métrica	Tipo	Descripción
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

CP4I V 9.2.2 CD Visualización del estado de los gestores de colas de HA nativa para contenedores certificados de IBM MQ

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

Acerca de esta tarea

Importante:

Puede utilizar el mandato **dspm** 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, Replicao Unknown.

INSTANCIA

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

CONNACTV

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).

ALTTIME

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 dspm -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) 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)
```

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) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

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) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

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)

“Ejemplo: Configuración de un gestor de colas de HA nativa” en la página 96

Este ejemplo muestra cómo desplegar un gestor de colas utilizando la característica de alta disponibilidad nativa en Red Hat OpenShift Container Platform (OCP) utilizando IBM MQ Operator.

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 con `cloudctl login` (en IBM Cloud Pak for Integration) o `oc login`.

Procedimiento

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

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

a) 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
```

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

dmpmqcfig genera la configuración de MQSC del gestor de colas.

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

- Restablezca la configuración del gestor de colas.

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

a) 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
```

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

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

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

- [“Resolución de problemas: obtención de acceso a los datos del gestor de colas”](#) en la página 127

OpenShift

CP4I

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 rsh pvc-inspector-pod-name
```

Se coloca en el directorio que contiene los directorios PVC montados.

- b) Abra un shell remoto en el pod, ejecutando el mandato siguiente:

```
ls
```

- c) Puede ver los directorios con el mismo nombre que las PVC que se han montado. Acceda a los archivos de las PVC del gestor de colas examinando estos directorios. Para ver una lista de las PVC, ejecute 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
```

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

EUS

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-RJON-C7QG3S	Production o NonProduction	IBM Cloud Pak for Integration 2021.4.1	9.2.4 o 9.2.5
L-RJON-C7QFZX	Production o NonProduction	IBM Cloud Pak for Integration Edición limitada 2021.4.1	9.2.4 o 9.2.5
L-RJON-C5CSNH	Production o NonProduction	IBM Cloud Pak for Integration 2021.3.1	9.2.3 o 9.2.4
L-RJON-C5CSM2	Production o NonProduction	IBM Cloud Pak for Integration Edición limitada 2021.3.1	9.2.3 o 9.2.4
L-RJON-BZFQU2	Production o NonProduction	IBM Cloud Pak for Integration 2021.2.1	9.2.3
L-RJON-BZFQSB	Production o NonProduction	IBM Cloud Pak for Integration Edición limitada 2021.2.1	9.2.3
L-RJON-BUVMQX	Production o NonProduction	IBM Cloud Pak for Integration 2020.4.1	9.2.0 EUS o 9.2.1
L-RJON-BUVMYB	Production o NonProduction	IBM Cloud Pak for Integration Edición limitada 2020.4.1	9.2.0 EUS o 9.2.1
L-APIG-BZDDDY	Production	IBM MQ Advanced y IBM MQ Advanced para entorno no productivo 9.2 -07/2021	9.2.3, 9.2.4 o 9.2.5
L-APIG-BYHCL7	Development	IBM MQ Advanced for Developers (sin garantía) V9.2 -07/2021	9.2.3, 9.2.4 o 9.2.5
L-APIG-BVJJB3	Production	IBM MQ Advanced y IBM MQ Advanced para entorno no productivo 9.2 -03/2021	9.2.2
L-APIG-BMJJBM	Production	IBM MQ Advanced V9.2	9.2.0 CD o 9.2.1
L-APIG-BMKG5H	Development	IBM MQ Advanced for Developers (sin garantía) V9.2	9.2.0 CD, 9.2.1 o 9.2.2

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

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-RJON-BXUPZ2	Production o NonProduction	IBM Cloud Pak for Integration 2021.1.1	9.2.2
L-RJON-BXUQ34	Production o NonProduction	IBM Cloud Pak for Integration Edición limitada 2021.1.1	9.2.2

Valor de spec.license.license	Valor de spec.license.use	Información de licencia	Versiones de IBM MQ aplicables
L-RJON-BYRMYW	NonProduction	IBM Cloud Pak for Integration Eval-Demo 2021.1.1 . Release anterior para su uso con HA nativa solo con IBM MQ Operator 1.5 .	9.2.2
L-RJON-BQPGWD	Production o NonProduction	IBM Cloud Pak for Integration 2020.3.1	CD de 9.2.0
L-RJON-BN7PN3	Production o NonProduction	IBM Cloud Pak for Integration 2020.2.1	9.1.5 o 9.2.0 CD
L-RJON-BPHL2Y	Production o NonProduction	IBM Cloud Pak for Integration Edición limitada 2020.2.1	9.1.5
L-APIG-BJAKBF	Production	IBM MQ Advanced V9.1 -04/2020	9.1.5
L-APIG-BM7GDH	Development	IBM MQ Advanced for Developers (sin garantía) V9.1 -04/2020	9.1.5

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

  **Referencia de API para QueueManager (mq.ibm.com/v1beta1)**

QueueManager

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

Campo	Descripción
Serie apiVersion	APIVersion define el esquema con versión de esta representación de un objeto. Los servidores deben convertir los esquemas reconocidos al valor interno más reciente y pueden rechazar valores no reconocidos. Más información: 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-clases .
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 130](#)

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 <u>Anotaciones</u>	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 <u>imagePullSecrets</u> <u>LocalObjectReference</u>	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 <u>Etiquetas</u>	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.
license <u>Licencia</u>	Valores que controlan la aceptación de la licencia y qué métricas de licencia utilizar.
pki <u>PKI</u>	Valores de infraestructura de claves públicas, para definir claves y certificados para utilizarlos con TLS (Transport Layer Security) o AMS (MQ Advanced Message Security).
queueManager <u>Configuración de QueueManager</u>	Valores para el contenedor del gestor de colas y el gestor de colas subyacente.
securityContext <u>SecurityContext</u>	Valores de seguridad para añadir al securityContext del pod del gestor de colas.
template <u>Plantilla</u>	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.
terminationGracePeriod <u>Seconds entero</u>	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.
tracing <u>TracingConfig</u>	Valores para rastrear la integración con el panel de control de operaciones de Cloud Pak for Integration.
Serie <u>version</u>	Valor que controla la versión de MQ que se utilizará (necesario). Por ejemplo: 9.1.5.0-r2 especificaría MQ versión 9.1.5.0, utilizando la segunda revisión de la imagen de contenedor. Los arreglos específicos de contenedor se aplican a menudo en revisiones, como por ejemplo arreglos en la imagen base.
web <u>Configuración de WebServer</u>	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 130](#)

.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 130](#)

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 130](#)

.spec.license

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

Aparece en:

- [“especificación” en la página 130](#)

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 http://ibm.biz/BdqvCF 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 http://ibm.biz/BdqvCF 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 130](#)

Campo	Descripción
Matriz <code>keys PKISource</code>	Claves privadas para añadir al repositorio de claves del gestor de colas.
Matriz <code>trust PKISource</code>	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 133](#)

Campo	Descripción
Serie <code>name</code>	El nombre se utiliza como etiqueta para la clave o el certificado. Debe ser una serie alfanumérica en minúsculas.
<code>secret</code> <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 133](#)

Campo	Descripción
<code>items</code> matriz	Claves dentro del secreto de Kubernetes que se debe añadir al contenedor del gestor de colas.
Serie <code>secretName</code>	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 133](#)

Campo	Descripción
Serie <code>name</code>	El nombre se utiliza como etiqueta para la clave o el certificado. Debe ser una serie alfanumérica en minúsculas.
<code>secret</code> <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 133](#)

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 130](#)

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.
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.

Campo	Descripción
storage <u>Almacenamiento de QueueManager</u>	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 134](#)

Campo	Descripción
tls <u>Tls</u>	Valores de TLS opcionales para configurar la comunicación segura entre réplicas de NativeHA . Requiere MQ Operator 1.5.0 o superior.
Serie type	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.
Serie updateStrategy	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 135](#)

Campo	Descripción
Serie cipherSpec	El nombre de la CipherSpec para NativeHA TLS.
Serie secretName	El nombre del secreto de Kubernetes .

.spec.queueManager.ini

Origen de los archivos de configuración de INI.

Aparece en:

- [“.spec.queueManager” en la página 134](#)

Campo	Descripción
configMap <u>ConfigMapINISource</u>	ConfigMap representa un Kubernetes ConfigMap que contiene información de INI.

Campo	Descripción
secret SecretINISource	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 135

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 135

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 134

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 134](#)

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 134](#)

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 137](#)

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 137](#)

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 134](#)

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 .
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.resources

Valores que controlan los requisitos de recursos.

Aparece en:

- [“.spec.queueManager”](#) en la página 134

Campo	Descripción
limits <u>Límites</u>	Valores de CPU y memoria.
requests <u>Solicitudes</u>	Valores de CPU y memoria.

.spec.queueManager.resources.limits

Valores de CPU y memoria.

Aparece en:

- [“.spec.queueManager.resources”](#) en la página 138

Campo	Descripción
cpu	
memory	

.spec.queueManager.resources.requests

Valores de CPU y memoria.

Aparece en:

- [“.spec.queueManager.resources”](#) en la página 138

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 134

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 134

Campo	Descripción
failureThreshold entero	Número mínimo de anomalías consecutivas para que el analizador se considere anómalo. El valor predeterminado es 60.
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 134

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.

Campo	Descripción
defaultDeleteClaim boolean	Indica si se deben suprimir 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.

.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 139

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.
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 139](#)

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.
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 139](#)

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.
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.securityContext

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

Aparece en:

- [“especificación” en la página 130](#)

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 .
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.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 130](#)

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 130](#)

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 142

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.agent.livenessProbe

Valores que controlan el análisis de actividad.

Aparece en:

- [“.spec.tracing.agent”](#) en la página 143

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.agent.readinessProbe

Valores que controlan el sondeo de preparación.

Aparece en:

- [“.spec.tracing.agent”](#) en la página 143

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.tracing.collector

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

Aparece en:

- [“.spec.tracing”](#) en la página 142

Campo	Descripción
image	La imagen de contenedor que se utilizará.
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 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 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.readinessProbe

Valores que controlan el sondeo de preparación.

Aparece en:

- [“.spec.tracing.collector”](#) 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 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 130

Campo	Descripción
enabled boolean	Indica si se debe habilitar o no el servidor web. El valor predeterminado es false.

.estado

El estado observado de QueueManager.

Aparece en:

- [“QueueManager” en la página 130](#)

Campo	Descripción
Serie adminUiUrl	URL para la interfaz de usuario de administración.
availability Disponibilidad	Estado de disponibilidad del gestor de colas.
Matriz conditions QueueManagerStatusCondition	Las condiciones representan las últimas observaciones disponibles del estado del gestor de colas.
Matriz endpoints QueueManagerStatusEndpoint	Información sobre los puntos finales que este gestor de colas está exponiendo, como por ejemplo puntos finales de interfaz de usuario o API.
Serie name	Nombre del gestor de colas.
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 146](#)

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 146](#)

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.

Campo	Descripción
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 146](#)

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.versions

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

Aparece en:

- [“estado” en la página 146](#)

Campo	Descripción
available QueueManagerStatusVersion Disponible	Otras versiones de MQ disponibles en IBM Entitled Registry.
Serie reconciled	La versión específica de IBM MQ que se está utilizando. Si se especifica una imagen personalizada, es posible que no coincida con la versión de MQ que se está utilizando realmente.

.status.versions.available

Otras versiones de MQ disponibles en IBM Entitled Registry.

Aparece en:

- [“status.versions” en la página 147](#)

Campo	Descripción
channels matriz	Canales que están disponibles para actualizar automáticamente la versión de MQ .
Matriz versions 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 147](#)

Campo	Descripción
Serie name	Versión name para esta versión de QueueManager. Son valores válidos para el campo <code>spec.version</code> .



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 ha introducido soporte para condiciones en IBM MQ Operator 1.2.

Se definen las condiciones siguientes para un recurso QueueManager:

Tabla 1. Condiciones de estado del gestor de colas

Componente	Tipo de condición	Código de razón	Aviso de mensaje
QueueManager ⁷	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
	Error	Fallido	El gestor de colas de MQ no se ha podido desplegar
	Aviso	UnsupportedVersion	⁸ 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 EUS	⁹ Se ha instalado un operando de EUS < mq_ version > pero lo está gestionando 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 EUS	¹⁰ Se ha instalado un operando EUS < mq_ version > pero el OCP versión 4 < ocp_ version > 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 EUS	¹¹ Se ha instalado un operando de EUS < mq_ version > pero la versión de OCP < ocp_ version > no cumple los requisitos para la duración de soporte ampliada.

⁷ Las condiciones Creating y Failed supervisan el progreso global del despliegue del gestor de colas. Si está utilizando una licencia de IBM Cloud Pak for Integration y la consola web de MQ está configurada, la condición OidcPending registra el estado del gestor de colas mientras espera el registro de cliente OIDC con IAM.

Tabla 1. Condiciones de estado del gestor de colas (continuación)

Componente	Tipo de condición	Código de razón	Aviso de mensaje
Pod ¹²	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 ¹³	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
	Error	StorageFailed	El almacenamiento para el gestor de colas de MQ no se ha podido suministrar

Multi Creación de su propio contenedor de IBM MQ y código de despliegue

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 alguno de los contenedores preeempaquetados que proporciona IBM. Consulte [IBM MQ en contenedores](#)

Acerca de esta tarea

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

⁸ Operador 1.4.0 y posterior

⁹ Operador 1.4.0 y posterior

¹⁰ Operador 1.4.0 y posterior

¹¹ Solo operador 1.3.0

¹² 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.

¹³ 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. Si se produce algún error durante el suministro de almacenamiento, la condición StorageFailed se añadirá a la lista de condiciones y la condición general del gestor de colas también se establecerá en Failed.

Procedimiento

- [“Planificación de una imagen propia de gestor de colas de IBM MQ utilizando un contenedor” en la página 151](#)
- [“Creación de una imagen de contenedor de gestor de colas de IBM MQ de ejemplo” en la página 151](#)
- [“Ejecución de aplicaciones de enlaces locales en contenedores separados” en la página 154](#)

Conceptos relacionados

[IBM MQ en contenedores](#)

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

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

Supervisión del proceso

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

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

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

Llenar /var/mqm

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

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

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

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.

Multi 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 152
 - [“Creación de una imagen de gestor de colas de IBM MQ configurada de muestra”](#) en la página 152

Multi 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. Si tiene previsto configurar el acceso seguro utilizando la restricción de contexto de seguridad (SCC) de Red Hat OpenShift Container Platform "restringido" , debe utilizar el paquete IBM MQ 'Sin instalación'.

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](#).

Multi 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

V 9.2.0 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 IBM MQ 9.2, 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 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}**.

Multi Ejecución de aplicaciones de enlaces locales en contenedores separados

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

Acerca de esta tarea

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

Hay que atenerse a las restricciones siguientes:

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

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

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

Procedimiento

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

```
mkdir /tmp/dockerVolume
```

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

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

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

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

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

```
docker exec secondaryContainer dspmq
```

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

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

Resultados

Ahora tiene aplicaciones locales que se ejecutan en contenedores separados y ahora puede ejecutar correctamente mandatos como **dspmq**, **amqspu**, **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 155 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 154 para obtener la lista de restricciones a la que hay que atenderse.

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]
lpi0btainQMDetails --> 545261715

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

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

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

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

docker exec container strmqweb
<file path>/mqrc.c[1152]
lpi0btainQMDetails --> 545261715
```

CP4I 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.

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:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

Kubernetes 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.

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 re arranque 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 5

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.

CP4I Visualización del estado de los gestores de colas de HA nativa para contenedores creados de forma personalizada

Para contenedores creados de forma personalizada, puede ver el estado de las instancias de HA nativa utilizando el mandato **dspmqr**.

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 **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.

ALTDATA

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) 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)
```

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) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
```

```
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

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) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

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

dspmq

CP4I 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.

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

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

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

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

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

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

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

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

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

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

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

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

Rochester, MN 55901
U.S.A.

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

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

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

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

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

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

LICENCIA DE COPYRIGHT:

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

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

Información acerca de las interfaces de programación

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

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

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

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

Marcas registradas

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

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

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

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

Este producto incluye software desarrollado por Eclipse Project (<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: