



OpenShift Container Platform 4.10

Migración de la versión 3 a la 4

Migración a OpenShift Container Platform 4

OpenShift Container Platform 4.10 Migración de la versión 3 a la 4

Migración a OpenShift Container Platform 4

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

Legal Notice

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Migrating_from_version_3_to_4.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Resumen

Este documento proporciona instrucciones para migrar su clúster de OpenShift Container Platform de la versión 3 a la versión 4.

Table of Contents

CAPÍTULO 1. DESCRIPCIÓN GENERAL DE LA MIGRACIÓN DE OPENSIFT CONTAINER PLATFORM 3 A 4	6
1.1. DIFERENCIAS ENTRE OPENSIFT CONTAINER PLATFORM 3 Y 4	6
1.2. PLANIFICACIÓN DE LAS CONSIDERACIONES DE RED	6
1.3. INSTALACIÓN DE MTC	6
1.4. ACTUALIZACIÓN DE MTC	6
1.5. REVISIÓN DE LAS LISTAS DE COMPROBACIÓN PREVIAS A LA MIGRACIÓN	7
1.6. MIGRACIÓN DE APLICACIONES	7
1.7. OPCIONES DE MIGRACIÓN AVANZADAS	7
1.8. SOLUCIÓN DE PROBLEMAS EN LAS MIGRACIONES	7
1.9. RETROCESO DE UNA MIGRACIÓN	7
1.10. DESINSTALACIÓN DE MTC Y ELIMINACIÓN DE RECURSOS	7
CAPÍTULO 2. ACERCA DE LA MIGRACIÓN DE OPENSIFT CONTAINER PLATFORM 3 A 4	9
CAPÍTULO 3. DIFERENCIAS ENTRE OPENSIFT CONTAINER PLATFORM 3 Y 4	11
3.1. ARQUITECTURA	11
Infraestructura inmutable	11
Operadores	11
3.2. INSTALACIÓN Y ACTUALIZACIÓN	12
Proceso de instalación	12
Opciones de infraestructura	12
Actualización del clúster	12
3.3. CONSIDERACIONES SOBRE LA MIGRACIÓN	12
3.3.1. Consideraciones sobre el almacenamiento	12
Almacenamiento persistente del volumen local	12
Almacenamiento persistente con FlexVolume	13
Almacenamiento persistente de la interfaz de almacenamiento de contenedores (CSI)	13
Red Hat OpenShift Container Storage	13
Opciones de almacenamiento persistente no admitidas	13
3.3.2. Consideraciones sobre la red	13
Modo de aislamiento de la red	13
3.3.3. Consideraciones sobre el registro	14
Implementación de OpenShift Logging	14
Datos de registro agregados	14
Configuraciones de registro no compatibles	14
3.3.4. Consideraciones de seguridad	14
Acceso no autenticado a los terminales de detección	14
Proveedores de identidad	14
Formato de almacenamiento de tokens OAuth	15
3.3.5. Consideraciones sobre el control	15
Alerta para supervisar la disponibilidad de la infraestructura	15
CAPÍTULO 4. CONSIDERACIONES SOBRE LA RED	16
4.1. CONSIDERACIONES DEL DNS	16
4.1.1. Aislamiento del dominio DNS del clúster de destino de los clientes	16
4.1.2. Configuración del clúster de destino para que acepte el dominio DNS de origen	16
4.2. ESTRATEGIAS DE REDIRECCIONAMIENTO DEL TRÁFICO DE RED	17
CAPÍTULO 5. ACERCA DE MIGRATION TOOLKIT FOR CONTAINERS	19
5.1. TERMINOLOGÍA	19
5.2. FLUJO DE TRABAJO DE MTC	20
5.3. ACERCA DE LOS MÉTODOS DE COPIA DE DATOS	23

5.3.1. Método de copia de sistemas de archivos	23
5.3.2. Método de copia de instantáneas	24
5.4. MIGRACIÓN DIRECTA DE VOLÚMENES Y MIGRACIÓN DIRECTA DE IMÁGENES	24
CAPÍTULO 6. INSTALACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS	26
6.1. DIRECTRICES DE COMPATIBILIDAD	26
6.2. INSTALACIÓN DEL OPERADOR DE MIGRATION TOOLKIT FOR CONTAINERS HEREDADO EN OPENSIFT CONTAINER PLATFORM 3	27
6.3. INSTALACIÓN DEL OPERADOR DE MIGRATION TOOLKIT FOR CONTAINERS EN OPENSIFT CONTAINER PLATFORM 4.10	29
6.4. CONFIGURACIÓN DE LOS PROXIES	29
6.5. CONFIGURACIÓN DE UN REPOSITORIO DE REPLICACIÓN	31
6.5.1. Requisitos previos	31
6.5.2. Recuperación de las credenciales de Multicloud Object Gateway	31
6.5.3. Configuración de Amazon Web Services	32
6.5.4. Configuración de Google Cloud Platform	34
6.5.5. Configuración de Microsoft Azure	36
6.5.6. Recursos adicionales	37
6.6. DESINSTALACIÓN DE MTC Y ELIMINACIÓN DE RECURSOS	37
CAPÍTULO 7. INSTALACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS EN UN ENTORNO DE RED RESTRINGIDO	39
7.1. DIRECTRICES DE COMPATIBILIDAD	39
7.2. INSTALACIÓN DEL OPERADOR DE MIGRATION TOOLKIT FOR CONTAINERS EN OPENSIFT CONTAINER PLATFORM 4.10	40
7.3. INSTALACIÓN DEL OPERADOR DE MIGRATION TOOLKIT FOR CONTAINERS HEREDADO EN OPENSIFT CONTAINER PLATFORM 3	41
7.4. CONFIGURACIÓN DE LOS PROXIES	43
7.5. CONFIGURACIÓN DE UN REPOSITORIO DE REPLICACIÓN	44
7.5.1. Requisitos previos	45
7.5.2. Recuperación de las credenciales de Multicloud Object Gateway	45
7.5.3. Recursos adicionales	45
7.6. DESINSTALACIÓN DE MTC Y ELIMINACIÓN DE RECURSOS	45
CAPÍTULO 8. ACTUALIZACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS	47
8.1. ACTUALIZACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS EN OPENSIFT CONTAINER PLATFORM 4.10	47
8.2. ACTUALIZACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS EN OPENSIFT CONTAINER PLATFORM 3	48
8.3. ACTUALIZACIÓN DE MTC 1.3 A 1.7	49
CAPÍTULO 9. LISTAS DE CONTROL PREVIAS A LA MIGRACIÓN	51
9.1. RECURSOS	51
9.2. CLÚSTER DE ORIGEN	51
9.3. CLÚSTER DE DESTINO	52
9.4. RENDIMIENTO	53
CAPÍTULO 10. MIGRACIÓN DE LAS APLICACIONES	54
10.1. REQUISITOS PREVIOS A LA MIGRACIÓN	54
Recursos adicionales para los requisitos previos a la migración	55
10.2. MIGRACIÓN DE LAS APLICACIONES MEDIANTE LA CONSOLA WEB DE MTC	56
10.2.1. Inicio de la consola web de MTC	56
10.2.2. Adición de un clúster a la consola web de MTC	56
10.2.3. Adición de un repositorio de replicación a la consola web de MTC	58
10.2.4. Creación de un plan de migración en la consola web de MTC	59

Recursos adicionales	62
10.2.5. Ejecución de un plan de migración en la consola web de MTC	62
CAPÍTULO 11. OPCIONES DE MIGRACIÓN AVANZADAS	64
11.1. TERMINOLOGÍA	64
11.2. MIGRACIÓN DE UNA APLICACIÓN DESDE LAS INSTALACIONES A UN CLÚSTER BASADO EN LA NUBE	65
11.3. MIGRACIÓN DE APLICACIONES MEDIANTE LA LÍNEA DE COMANDOS	68
11.3.1. Requisitos previos a la migración	68
11.3.2. Creación de una ruta de registro para la migración directa de imágenes	69
11.3.3. Configuración de los proxies	69
11.3.4. Migración de una aplicación mediante la API de MTC	71
11.3.5. Migración de estado	76
Recursos adicionales	77
11.4. ENLACES DE MIGRACIÓN	77
11.4.1. Escritura de una estrategia de Ansible para un enlace de migración	78
11.4.1.1. Módulos de Ansible	78
11.4.1.2. Variables del entorno	79
11.5. OPCIONES DEL PLAN DE MIGRACIÓN	79
11.5.1. Exclusión de los recursos	79
11.5.2. Asignación de espacios de nombres	80
11.5.3. Exclusión de reclamaciones de volúmenes persistentes	81
11.5.4. Asignación de reclamaciones de volúmenes persistentes	81
11.5.5. Edición de los atributos del volumen persistente	82
Recursos adicionales	84
11.5.6. Migración de objetos de Kubernetes	84
11.6. OPCIONES DEL CONTROLADOR DE MIGRACIÓN	85
11.6.1. Aumento de los límites para las grandes migraciones	85
11.6.2. Habilitación del redimensionamiento de volúmenes persistentes para la migración directa de volúmenes	86
11.6.3. Habilitación de clientes Kubernetes en caché	87
CAPÍTULO 12. RESOLUCIÓN DE PROBLEMAS	89
12.1. FLUJO DE TRABAJO DE MTC	89
Acerca de los recursos personalizados de MTC	91
12.2. MANIFIESTOS DE RECURSOS PERSONALIZADOS DE MTC	92
12.2.1. DirectImageMigration	92
12.2.2. DirectImageStreamMigration	93
12.2.3. DirectVolumeMigration	93
12.2.4. DirectVolumeMigrationProgress	94
12.2.5. MigAnalytic	94
12.2.6. MigCluster	95
12.2.7. MigHook	96
12.2.8. MigMigration	97
12.2.9. MigPlan	98
12.2.10. MigStorage	99
12.3. REGISTROS Y HERRAMIENTAS DE DEPURACIÓN	100
12.3.1. Vista de los recursos del plan de migración	100
12.3.2. Vista del registro de un plan de migración	101
12.3.3. Uso del lector de registros de migración	102
12.3.4. Acceso a las métricas de rendimiento	102
12.3.4.1. Métricas proporcionadas	102
12.3.4.1.1. cam_app_workload_migrations	102

12.3.4.1.2. mtc_client_request_count	103
12.3.4.1.3. mtc_client_request_elapsed	103
12.3.4.1.4. Consultas útiles	104
12.3.5. Uso de la herramienta de recopilación	104
Visualización de los datos de las métricas con la consola de Prometheus	105
12.3.6. Depuración de los recursos de Velero con la herramienta CLI de Velero	105
Sintaxis	106
Opción de ayuda	106
Comando Describe	106
Comando Logs	106
12.3.7. Depuración de un fallo de migración parcial	107
12.3.8. Uso de los recursos personalizados de MTC para la resolución de problemas	108
12.4. PREOCUPACIONES Y PROBLEMAS COMUNES	111
12.4.1. Actualización de imágenes internas obsoletas	112
12.4.2. Migración directa de volúmenes incompleta	113
12.4.3. Mensajes de error y resoluciones	114
12.4.3.1. Aparece un error de certificado de CA al acceder a la consola de MTC por primera vez	114
12.4.3.2. Error de tiempo de espera de OAuth en la consola de MTC	114
12.4.3.3. Error de certificado firmado por una autoridad desconocida	114
12.4.3.4. Errores de ubicación de la copia de seguridad en el registro del pod de Velero	115
12.4.3.5. Error de tiempo de espera del volumen del pod en el registro del pod de Velero	115
12.4.3.6. Errores de verificación de Restic en el recurso personalizado MigMigration	116
12.4.3.7. Error de permiso de Restic al migrar desde un almacenamiento del NFS con root_squash activado	117
12.4.4. Problemas conocidos	118
12.5. RETROCESO DE UNA MIGRACIÓN	119
12.5.1. Retroceso de una migración mediante la consola web de MTC	119
12.5.2. Retroceso de una migración desde la interfaz de línea de comandos	120
12.5.3. Retroceso manual de una migración	121
Recursos adicionales	122

CAPÍTULO 1. DESCRIPCIÓN GENERAL DE LA MIGRACIÓN DE OPENSIFT CONTAINER PLATFORM 3 A 4

Los clústeres de OpenShift Container Platform 4 son diferentes de los clústeres de OpenShift Container Platform 3. Los clústeres de OpenShift Container Platform 4 contienen nuevas tecnologías y funcionalidades que dan como resultado un clúster autogestionado, flexible y automatizado. Para obtener más información sobre la migración de OpenShift Container Platform 3 a 4, consulte [Acerca de la migración de OpenShift Container Platform 3 a 4](#).

1.1. DIFERENCIAS ENTRE OPENSIFT CONTAINER PLATFORM 3 Y 4

Antes de migrar de OpenShift Container Platform 3 a 4, puede comprobar las [diferencias entre OpenShift Container Platform 3 y 4](#). Revise la siguiente información:

- [Arquitectura](#)
- [Instalación y actualización](#)
- [Consideraciones de almacenamiento, redes, registros, seguridad y monitoreo](#)

1.2. PLANIFICACIÓN DE LAS CONSIDERACIONES DE RED

Antes de migrar de OpenShift Container Platform 3 a 4, revise las [diferencias entre OpenShift Container Platform 3 y 4](#) para obtener información sobre las siguientes áreas:

- [Consideraciones del DNS](#)
 - [Aísle el dominio DNS del clúster de destino de los clientes](#) .
 - [Configure el clúster de destino para que acepte el dominio DNS de origen](#) .

Puede migrar cargas de trabajo de aplicaciones con estado de OpenShift Container Platform 3 a 4 en la granularidad de un espacio de nombres. Para saber más sobre MTC, consulte [Comprensión de MTC](#).



NOTA

Si migra desde OpenShift Container Platform 3, consulte [Acerca de la migración de OpenShift Container Platform 3 a 4](#) e [Instalación del operador de Migration Toolkit for Containers heredado en OpenShift Container Platform 3](#).

1.3. INSTALACIÓN DE MTC

Revise las siguientes tareas para instalar MTC:

1. [Instale el operador de Migration Toolkit for Containers en el clúster de destino con Operator Lifecycle Manager \(OLM\)](#).
2. [Instale manualmente el operador de Migration Toolkit for Containers heredado en el clúster de origen](#).
3. [Configure el almacenamiento de objetos para utilizarlo como repositorio de replicación](#) .

1.4. ACTUALIZACIÓN DE MTC

Actualice [Migration Toolkit for Containers \(MTC\)](#) en OpenShift Container Platform 4.10 con OLM. La actualización de MTC en OpenShift Container Platform 3 se realiza reinstalando el operador de Migration Toolkit for Containers heredado.

1.5. REVISIÓN DE LAS LISTAS DE COMPROBACIÓN PREVIAS A LA MIGRACIÓN

Antes de migrar las cargas de trabajo de sus aplicaciones con Migration Toolkit for Containers (MTC), revise las [listas de comprobación previas a la migración](#).

1.6. MIGRACIÓN DE APLICACIONES

Puede migrar sus aplicaciones utilizando la [consola web](#) de MTC o la [línea de comandos](#).

1.7. OPCIONES DE MIGRACIÓN AVANZADAS

Puede automatizar las migraciones y modificar los recursos personalizados de MTC para mejorar el rendimiento de las migraciones a gran escala mediante las siguientes opciones:

- [Ejecutar una migración de estado](#)
- [Crear enlaces de migración](#)
- [Editar, excluir y asignar los recursos migrados](#)
- [Configurar el controlador de migración para las migraciones grandes](#)

1.8. SOLUCIÓN DE PROBLEMAS EN LAS MIGRACIONES

Puede realizar las siguientes tareas de resolución de problemas:

- [Visualización de los recursos del plan de migración mediante la consola web de MTC](#)
- [Visualización del archivo de registro agregado del plan de migración](#)
- [Uso del lector de registros de migración](#)
- [Acceso a las métricas de rendimiento](#)
- [Uso de la herramienta de **recopilación**](#)
- [Uso de la CLI de Velero para depurar los CR **Backup** y **Restore**](#)
- [Uso de los recursos personalizados de MTC para la resolución de problemas](#)
- [Comprobación de problemas y preocupaciones comunes](#)

1.9. RETROCESO DE UNA MIGRACIÓN

Puede [retrotraer una migración](#) con la consola web de MTC, la CLI o manualmente.

1.10. DESINSTALACIÓN DE MTC Y ELIMINACIÓN DE RECURSOS

Puede [desinstalar MTC y eliminar sus recursos](#) para limpiar el clúster.

CAPÍTULO 2. ACERCA DE LA MIGRACIÓN DE OPENSIFT CONTAINER PLATFORM 3 A 4

OpenShift Container Platform 4 contiene nuevas tecnologías y funcionalidades que dan como resultado un clúster autogestionado, flexible y automatizado. Los clústeres de OpenShift Container Platform 4 se despliegan y gestionan de forma muy diferente a los de OpenShift Container Platform 3.

La forma más eficaz de migrar de OpenShift Container Platform 3 a 4 es utilizar una canalización de CI/CD para automatizar las implementaciones en un marco de [gestión del ciclo de vida de las aplicaciones](#).

Si no dispone de una canalización de CI/CD o está migrando aplicaciones con estado, puede utilizar Migration Toolkit for Containers (MTC) para migrar las cargas de trabajo de sus aplicaciones.

Puede utilizar Red Hat Advanced Cluster Management for Kubernetes para ayudarle a importar y gestionar fácilmente sus clústeres de OpenShift Container Platform 3, aplicar políticas y volver a implementar sus aplicaciones. Aproveche la [suscripción](#) gratuita para utilizar Red Hat Advanced Cluster Management para simplificar su proceso de migración.

Para pasar con éxito a OpenShift Container Platform 4, revise la siguiente información:

Diferencias entre OpenShift Container Platform 3 y 4

- Arquitectura
- Instalación y actualización
- Consideraciones de almacenamiento, redes, registros, seguridad y monitoreo

Acerca de Migration Toolkit for Containers

- Flujo de trabajo
- Métodos de copia de instantáneas y sistemas de archivos para volúmenes persistentes (PV)
- Migración directa de volúmenes
- Migración directa de imágenes

Opciones de migración avanzadas

- Automatización de la migración con enlaces de migración
- Uso de la API de MTC
- Exclusión de recursos de un plan de migración
- Configuración del recurso personalizado **MigrationController** para migraciones a gran escala
- Habilitación del redimensionamiento automático de PV para la migración directa de volúmenes
- Habilitación de clientes de Kubernetes en caché para mejorar el rendimiento

Para conocer las nuevas funciones y mejoras, los cambios técnicos y los problemas conocidos, consulte las [notas de la versión de MTC](#).

CAPÍTULO 3. DIFERENCIAS ENTRE OPENSIFT CONTAINER PLATFORM 3 Y 4

OpenShift Container Platform 4.10 introduce cambios y mejoras en la arquitectura. Los procedimientos que utilizó para gestionar su clúster de OpenShift Container Platform 3 podrían no aplicarse a OpenShift Container Platform 4.

Para obtener información sobre la configuración de su clúster de OpenShift Container Platform 4, revise las secciones correspondientes de la documentación de OpenShift Container Platform. Para obtener información sobre las nuevas características y otros cambios técnicos notables, revise las [notas de la versión 4.10 de OpenShift Container Platform](#).

No es posible actualizar su clúster existente de OpenShift Container Platform 3 a OpenShift Container Platform 4. Debe comenzar con una nueva instalación de OpenShift Container Platform 4. Hay herramientas disponibles para ayudarlo a migrar la configuración del plano de control y las cargas de trabajo de las aplicaciones.

3.1. ARQUITECTURA

Con OpenShift Container Platform 3, los administradores implementaron individualmente los hosts de Red Hat Enterprise Linux (RHEL) y, luego, instalaron OpenShift Container Platform sobre estos hosts para formar un clúster. Los administradores fueron los responsables de configurar correctamente estos hosts y de realizar las actualizaciones.

OpenShift Container Platform 4 representa un cambio significativo respecto de la forma en que se implementan y gestionan los clústeres de OpenShift Container Platform. OpenShift Container Platform 4 incluye nuevas tecnologías y funcionalidades, como operadores, conjuntos de máquinas y Red Hat Enterprise Linux CoreOS (RHCOS), que son fundamentales para el funcionamiento del clúster. Este cambio tecnológico permite que los clústeres autogestionen algunas funciones que antes realizaban los administradores. Esto también garantiza la estabilidad y consistencia de la plataforma, y simplifica la instalación y el escalamiento.

Para más información, consulte [Arquitectura de OpenShift Container Platform](#).

Infraestructura inmutable

OpenShift Container Platform 4 utiliza Red Hat Enterprise Linux CoreOS (RHCOS), que está diseñado para ejecutar aplicaciones en contenedores, y proporciona una instalación eficiente, una gestión basada en el operador y actualizaciones simplificadas. RHCOS es un host de contenedores inmutable en lugar de un sistema operativo personalizable, como RHEL. RHCOS permite que OpenShift Container Platform 4 gestione y automatice la implementación del host de contenedores subyacente. RHCOS forma parte de OpenShift Container Platform, lo que significa que todo se ejecuta dentro de un contenedor y se implementa con OpenShift Container Platform.

En OpenShift Container Platform 4, los nodos del plano de control deben ejecutar RHCOS, lo que garantiza el mantenimiento de la automatización de toda la pila para el plano de control. Esto hace que el despliegue de actualizaciones y mejoras sea un proceso mucho más sencillo que en OpenShift Container Platform 3.

Para más información, consulte [Red Hat Enterprise Linux CoreOS \(RHCOS\)](#).

Operadores

Los operadores son un método para empaquetar, implementar y gestionar una aplicación de Kubernetes. Los operadores alivian la complejidad operativa de ejecutar otro software. Vigilan su entorno y utilizan el estado actual para tomar decisiones en tiempo real. Los operadores avanzados están diseñados para actualizar y reaccionar ante los fallos de forma automática.

Para obtener más información, consulte la sección [Comprensión de los operadores](#).

3.2. INSTALACIÓN Y ACTUALIZACIÓN

Proceso de instalación

Para instalar OpenShift Container Platform 3.11, ha preparado sus hosts de Red Hat Enterprise Linux (RHEL), ha establecido todos los valores de configuración que necesitaba su clúster y ha ejecutado una estrategia de Ansible para instalar y configurar su clúster.

En OpenShift Container Platform 4.10, se utiliza el programa de instalación de OpenShift para crear un conjunto mínimo de recursos necesarios para un clúster. Una vez que el clúster está en funcionamiento, se utilizan los operadores para seguir configurando el clúster e instalar nuevos servicios. Tras el primer arranque, el operador de configuración de la máquina (MCO) gestiona los sistemas de Red Hat Enterprise Linux CoreOS (RHCOS) que se ejecutan en el clúster de OpenShift Container Platform.

Para más información, consulte el [proceso de instalación](#).

Si desea añadir máquinas de trabajadores de Red Hat Enterprise Linux (RHEL) a su clúster de OpenShift Container Platform 4.10, utilice una estrategia de Ansible para unir las máquinas de trabajadores de RHEL una vez que el clúster esté en funcionamiento. Para más información, consulte [Añadir máquinas de computación de RHEL a un clúster de OpenShift Container Platform](#).

Opciones de infraestructura

En OpenShift Container Platform 3.11, instaló su clúster en la infraestructura que preparó y mantuvo. Además de proporcionar su propia infraestructura, OpenShift Container Platform 4 ofrece una opción para desplegar un clúster en la infraestructura que el programa de instalación de OpenShift Container Platform proporciona y el clúster mantiene.

Para obtener más información, consulte la [descripción general de la instalación de OpenShift Container Platform](#).

Actualización del clúster

En OpenShift Container Platform 3.11, se actualiza el clúster mediante la ejecución de las estrategias de Ansible. En OpenShift Container Platform 4.10, el clúster gestiona sus propias actualizaciones, incluidas las actualizaciones de Red Hat Enterprise Linux CoreOS (RHCOS) en los nodos del clúster. Puede actualizar fácilmente su clúster utilizando la consola web o el comando **oc adm upgrade** desde la CLI de OpenShift y los operadores se actualizarán automáticamente. Si su clúster de OpenShift Container Platform 4.10 tiene máquinas de trabajadores de RHEL, tendrá que ejecutar una estrategia de Ansible para actualizar esas máquinas de trabajadores.

Para más información, consulte [Actualización de clústeres](#).

3.3. CONSIDERACIONES SOBRE LA MIGRACIÓN

Revise los cambios y otras consideraciones que podrían afectar a su transición de OpenShift Container Platform 3.11 a OpenShift Container Platform 4.

3.3.1. Consideraciones sobre el almacenamiento

Revise los siguientes cambios de almacenamiento que debe tener en cuenta al pasar de OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Almacenamiento persistente del volumen local

El almacenamiento local solo es compatible con el operador de almacenamiento local en OpenShift Container Platform 4.10. No es compatible con el método de aprovisionamiento local de OpenShift Container Platform 3.11.

Para más información, consulte [Almacenamiento persistente mediante volúmenes locales](#).

Almacenamiento persistente con FlexVolume

La ubicación del complemento FlexVolume ha cambiado desde OpenShift Container Platform 3.11. La nueva ubicación en OpenShift Container Platform 4.10 es `/etc/kubernetes/kubelet-plugins/volume/exec`. Los complementos FlexVolume acoplables ya no son compatibles.

Para más información, consulte [Almacenamiento persistente con FlexVolume](#).

Almacenamiento persistente de la interfaz de almacenamiento de contenedores (CSI)

El almacenamiento persistente mediante la interfaz de almacenamiento de contenedores (CSI) fue la [vista previa de tecnología](#) en OpenShift Container Platform 3.11. OpenShift Container Platform 4.10 incluye [varios controladores de la CSI](#). También puede instalar su propio controlador.

Para más información, consulte [Almacenamiento persistente mediante la interfaz de almacenamiento de contenedores \(CSI\)](#).

Red Hat OpenShift Container Storage

Red Hat OpenShift Container Storage 3, que está disponible para su uso con OpenShift Container Platform 3.11, utiliza Red Hat Gluster Storage como almacenamiento de respaldo.

Red Hat OpenShift Container Storage 4, que está disponible para su uso con OpenShift Container Platform 4, utiliza Red Hat Ceph Storage como almacenamiento de respaldo.

Para más información, consulte [Almacenamiento persistente con Red Hat OpenShift Container Storage](#) y el artículo [Matriz de interoperabilidad](#).

Opciones de almacenamiento persistente no admitidas

La compatibilidad con las siguientes opciones de almacenamiento persistente de OpenShift Container Platform 3.11 ha cambiado en OpenShift Container Platform 4.10:

- GlusterFS ya no se admite.
- CephFS como producto independiente ya no se admite.
- Ceph RBD como producto independiente ya no se admite.

Si utilizó uno de ellos en OpenShift Container Platform 3.11, debe elegir una opción de almacenamiento persistente diferente para que sea totalmente compatible con OpenShift Container Platform 4.10.

Para más información, consulte la sección [Comprensión del almacenamiento persistente](#).

3.3.2. Consideraciones sobre la red

Revise los siguientes cambios en la red que debe tener en cuenta al pasar de OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Modo de aislamiento de la red

El modo de aislamiento de la red por defecto de OpenShift Container Platform 3.11 era **ovs-subnet**, aunque los usuarios solían cambiar a **ovn-multitenant**. El modo de aislamiento de la red por defecto para OpenShift Container Platform 4.10 está controlado por una política de red.

Si su clúster de OpenShift Container Platform 3.11 utilizaba el modo **ovs-subred** o **ovs-multitenant**, se

recomienda cambiar a una política de red para su clúster de OpenShift Container Platform 4.10. Las políticas de red se admiten en la fase previa, son más flexibles y proporcionan la funcionalidad **ovs-multitenant**. Si desea mantener el comportamiento de **ovs-multitenant** mientras utiliza una política de red en OpenShift Container Platform 4.10, siga los pasos para [configurar el aislamiento multiusuario utilizando la política de red](#).

Para más información, consulte [Acerca de la política de red](#).

3.3.3. Consideraciones sobre el registro

Revise los siguientes cambios en el registro que debe tener en cuenta al pasar de OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Implementación de OpenShift Logging

OpenShift Container Platform 4 proporciona un mecanismo de implementación sencillo para OpenShift Logging mediante el uso de un recurso personalizado de Cluster Logging.

Para obtener más información, consulte [Instalación de OpenShift Logging](#).

Datos de registro agregados

No se pueden transferir los datos de registro agregados de OpenShift Container Platform 3.11 a su nuevo clúster de OpenShift Container Platform 4.

Para obtener más información, consulte [Acerca de OpenShift Logging](#).

Configuraciones de registro no compatibles

Algunas configuraciones de registro que estaban disponibles en OpenShift Container Platform 3.11 ya no son compatibles en OpenShift Container Platform 4.10.

Para obtener más información sobre los casos de registro explícitamente no compatibles, consulte [Mantenimiento y soporte](#).

3.3.4. Consideraciones de seguridad

Revise los siguientes cambios de seguridad que debe tener en cuenta al pasar de OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Acceso no autenticado a los terminales de detección

En OpenShift Container Platform 3.11, un usuario no autenticado podía acceder a los terminales de detección (por ejemplo, `/api/*` y `/apis/*`). Por motivos de seguridad, en OpenShift Container Platform 4.10 ya no se permite el acceso no autenticado a los terminales de detección. Si necesita permitir el acceso no autenticado, puede configurar los ajustes de RBAC como sea necesario; sin embargo, asegúrese de considerar las implicaciones de seguridad ya que esto puede exponer los componentes internos del clúster a la red externa.

Proveedores de identidad

La configuración de los proveedores de identidad ha cambiado para OpenShift Container Platform 4, incluidos los siguientes cambios notables:

- El proveedor de identidad del encabezado de solicitud en OpenShift Container Platform 4.10 requiere TLS mutuo, mientras que en OpenShift Container Platform 3.11 no lo requería.
- La configuración del proveedor de identidades OpenID Connect se ha simplificado en OpenShift Container Platform 4.10. Ahora obtiene los datos, que antes había que especificar en OpenShift Container Platform 3.11, del terminal `/.well-known/openid-configuration` del proveedor.

Para obtener más información, consulte [Comprensión de la configuración del proveedor de identidades](#) .

Formato de almacenamiento de tokens OAuth

Los tokens de portador HTTP OAuth recién creados ya no coinciden con los nombres de los objetos del token de acceso OAuth. Los nombres de los objetos son ahora un hash del token del portador y ya no son sensibles. Esto reduce el riesgo de filtrar información sensible.

3.3.5. Consideraciones sobre el control

Revise los siguientes cambios de monitoreo que debe tener en cuenta al pasar de OpenShift Container Platform 3.11 a OpenShift Container Platform 4.10.

Alerta para supervisar la disponibilidad de la infraestructura

La alerta por defecto que se dispara para asegurar la disponibilidad de la estructura de monitorización se llamaba **DeadMansSwitch** en OpenShift Container Platform 3.11. En OpenShift Container Platform 4 se cambió el nombre a **Watchdog**. Si tenía configurada la integración de PagerDuty con esta alerta en OpenShift Container Platform 3.11, debe configurar la integración de PagerDuty para la alerta **Watchdog** en OpenShift Container Platform 4.

Para más información, consulte [Aplicación de la configuración personalizada de Alertmanager](#) .

CAPÍTULO 4. CONSIDERACIONES SOBRE LA RED

Revise las estrategias para redirigir el tráfico de la red de aplicaciones después de la migración.

4.1. CONSIDERACIONES DEL DNS

El dominio DNS del clúster de destino es diferente del dominio del clúster de origen. Por defecto, las aplicaciones obtienen los FQDN del clúster de destino tras la migración.

Para conservar el dominio DNS de origen de las aplicaciones migradas, seleccione una de las dos opciones descritas a continuación.

4.1.1. Aislamiento del dominio DNS del clúster de destino de los clientes

Puede permitir que las solicitudes de los clientes enviadas al dominio DNS del clúster de origen lleguen al dominio DNS del clúster de destino sin exponer el clúster de destino a los clientes.

Procedimiento

1. Coloque un componente de red exterior, como un equilibrador de carga de aplicaciones o un proxy inverso, entre los clientes y el clúster de destino.
2. Actualice el FQDN de la aplicación en el clúster de origen del servidor DNS para que devuelva la dirección IP del componente de red exterior.
3. Configure el componente de red para enviar las solicitudes recibidas para la aplicación en el dominio de origen al equilibrador de carga en el dominio del clúster de destino.
4. Cree un registro DNS comodín para el dominio ***.apps.source.example.com** que apunte a la dirección IP del equilibrador de carga del clúster de origen.
5. Cree un registro DNS para cada aplicación que apunte a la dirección IP del componente de red exterior frente al clúster de destino. Un registro DNS específico tiene mayor prioridad que un registro comodín, por lo que no surge ningún conflicto cuando se resuelve el FQDN de la aplicación.



NOTA

- El componente de red exterior debe terminar todas las conexiones TLS seguras. Si las conexiones pasan por el equilibrador de carga del clúster de destino, el FQDN de la aplicación de destino queda expuesto al cliente y se producen errores de certificación.
- Las aplicaciones no deben devolver a los clientes enlaces que hagan referencia al dominio del clúster de destino. De lo contrario, es posible que algunas partes de la aplicación no se carguen o funcionen correctamente.

4.1.2. Configuración del clúster de destino para que acepte el dominio DNS de origen

Puede configurar el clúster de destino para que acepte solicitudes de una aplicación migrada en el dominio DNS del clúster de origen.

Procedimiento

Tanto para el acceso HTTP no seguro como para el acceso HTTPS seguro, realice los siguientes pasos:

1. Cree una ruta en el proyecto del clúster de destino que esté configurada para aceptar solicitudes dirigidas al FQDN de la aplicación en el clúster de origen:

```
$ oc expose svc <app1-svc> --hostname <app1.apps.source.example.com> \
-n <app1-namespace>
```

Con esta nueva ruta, el servidor acepta cualquier solicitud para ese FQDN y la envía a los pods de la aplicación correspondientes. Además, cuando se migra la aplicación, se crea otra ruta en el dominio del clúster de destino. Las solicitudes llegan a la aplicación migrada utilizando cualquiera de estos nombres de host.

2. Cree un registro DNS con su proveedor de DNS que apunte el FQDN de la aplicación en el clúster de origen a la dirección IP del equilibrador de carga predeterminado del clúster de destino. Esto redirigirá el tráfico de su clúster de origen a su clúster de destino. El FQDN de la aplicación se resuelve en el equilibrador de carga del clúster de destino. El router del Controlador de Entrada por defecto acepta solicitudes para ese FQDN porque una ruta para ese nombre de host está expuesta.

Para el acceso HTTPS seguro, realice el siguiente paso adicional:

1. Sustituya el certificado x509 del controlador de entrada predeterminado creado durante el proceso de instalación por un certificado personalizado.
2. Configure este certificado para que incluya los dominios DNS comodín de los clústeres de origen y destino en el campo **subjectAltName**. El nuevo certificado es válido para asegurar las conexiones realizadas con cualquiera de los dos dominios DNS.

Recursos adicionales

- Para más información, consulte [Sustitución del certificado de entrada predeterminado](#).

4.2. ESTRATEGIAS DE REDIRECCIONAMIENTO DEL TRÁFICO DE RED

Después de una migración exitosa, debe redirigir el tráfico de red de sus aplicaciones sin estado desde el clúster de origen al clúster de destino.

Las estrategias para redirigir el tráfico de la red se basan en los siguientes supuestos:

- Los pods de aplicación se ejecutan tanto en el clúster de origen como en el de destino.
- Cada aplicación tiene una ruta que contiene el nombre de host del clúster de origen.
- La ruta con el nombre de host del clúster de origen contiene un certificado de CA.
- En el caso de HTTPS, el certificado de CA del enrutador de destino contiene un nombre alternativo del sujeto para el registro DNS comodín del clúster de origen.

Considere las siguientes estrategias y seleccione la que se ajuste a sus objetivos.

- Redirigir todo el tráfico de red para todas las aplicaciones al mismo tiempo. Cambie el registro DNS comodín del clúster de origen para que apunte a la dirección IP virtual (VIP) del enrutador del clúster de destino.

Esta estrategia es adecuada para aplicaciones sencillas o pequeñas migraciones.

- Redirigir el tráfico de red para las aplicaciones individuales
Cree un registro DNS para cada aplicación con el nombre de host del clúster de origen apuntando a la VIP del enrutador del clúster de destino. Este registro DNS tiene prioridad sobre el registro DNS comodín del clúster de origen.
- Redirigir el tráfico de red gradualmente para las aplicaciones individuales
 1. Cree un proxy que pueda dirigir el tráfico tanto a la VIP del enrutador de origen como a la VIP del enrutador de destino para cada aplicación.
 2. Cree un registro DNS para cada aplicación con el nombre de host del clúster de origen apuntando al proxy.
 3. Configure la entrada del proxy para que la aplicación enrute un porcentaje del tráfico a la VIP del enrutador de destino y el resto del tráfico a la VIP del enrutador de origen.
 4. Aumente gradualmente el porcentaje de tráfico que dirige a la VIP del enrutador de destino hasta redirigir todo el tráfico de red.
- Redirigir el tráfico basado en el usuario para las aplicaciones individuales
Con esta estrategia puede filtrar los encabezados TCP/IP de las solicitudes de los usuarios para redirigir el tráfico de red para los grupos predefinidos de usuarios. Esto le permite probar el proceso de redireccionamiento en poblaciones específicas de usuarios antes de redirigir todo el tráfico de la red.
 1. Cree un proxy que pueda dirigir el tráfico tanto a la VIP del enrutador de origen como a la VIP del enrutador de destino para cada aplicación.
 2. Cree un registro DNS para cada aplicación con el nombre de host del clúster de origen apuntando al proxy.
 3. Configure la entrada del proxy para que la aplicación enrute el tráfico que coincide con un patrón de encabezado determinado, como los **clientes de prueba**, a la VIP del enrutador de destino y el resto del tráfico a la VIP del enrutador de origen.
 4. Redirija el tráfico a la VIP del enrutador de destino por etapas hasta que todo el tráfico esté en la VIP del enrutador de destino.

CAPÍTULO 5. ACERCA DE MIGRATION TOOLKIT FOR CONTAINERS

Migration Toolkit for Containers (MTC) le permite migrar cargas de trabajo de aplicaciones con estado de OpenShift Container Platform 3 a 4.10 en la granularidad de un espacio de nombres.



IMPORTANTE

Antes de comenzar la migración, asegúrese de revisar las [diferencias entre OpenShift Container Platform 3 y 4](#).

MTC proporciona una consola web y una API basada en los recursos personalizados de Kubernetes para ayudarlo a controlar la migración y minimizar el tiempo de inactividad de las aplicaciones.

La consola de MTC se instala por defecto en el clúster de destino. Puede configurar el operador de Migration Toolkit for Containers para instalar la consola en un [clúster de origen de OpenShift Container Platform 3](#) o en un [clúster remoto](#).

MTC admite los métodos de copia de datos de instantáneas y sistemas de archivos para migrar los datos del clúster de origen al clúster de destino. Puede seleccionar un método que se adapte a su entorno y que sea compatible con su proveedor de almacenamiento.

El catálogo de servicios es obsoleto en OpenShift Container Platform 4. Puede migrar los recursos de carga de trabajo aprovisionados con el catálogo de servicios de OpenShift Container Platform 3 a 4, pero no puede realizar acciones del catálogo de servicios, como el **aprovisionamiento**, el **desaprovisionamiento** o la **actualización**, en estas cargas de trabajo después de la migración. La consola de MTC muestra un mensaje si los recursos del catálogo de servicios no pueden migrarse.

5.1. TERMINOLOGÍA

Tabla 5.1. Terminología de MTC

Término	Definición
Clúster de origen	Clúster desde el que se migran las aplicaciones.
Cluster de destino ^[1]	Clúster al que se migran las aplicaciones.
Repositorio de replicación	Almacenamiento de objetos utilizado para copiar imágenes, volúmenes y objetos de Kubernetes durante la migración indirecta o para objetos de Kubernetes durante la migración directa de volúmenes o la migración directa de imágenes. El repositorio de replicación debe ser accesible para todos los clústeres.
Clúster del host	Clúster en el que se ejecuta el pod migration-controller y la consola web. El clúster del host suele ser el clúster de destino, pero no es necesario. El clúster del host no requiere una ruta de registro expuesta para la migración directa de imágenes.

Término	Definición
Clúster remoto	<p>El clúster remoto suele ser el clúster de origen, pero no es necesario.</p> <p>El clúster remoto requiere un recurso personalizado secreto que contenga el token de la cuenta de servicio del controlador de migración.</p> <p>El clúster remoto requiere una ruta de registro segura expuesta para la migración directa de imágenes.</p>
Migración indirecta	Las imágenes, los volúmenes y los objetos de Kubernetes se copian del clúster de origen al repositorio de replicación y, luego, del repositorio de replicación al clúster de destino.
Migración directa de volúmenes	Los volúmenes persistentes se copian directamente del clúster de origen al de destino.
Migración directa de imágenes	Las imágenes se copian directamente del clúster de origen al de destino.
Migración por etapas	<p>Los datos se copian en el clúster de destino sin detener la aplicación.</p> <p>Ejecutar una migración por etapas varias veces reduce la duración de la migración por transición.</p>
Migración por transición	La aplicación se detiene en el clúster de origen y sus recursos se migran al clúster de destino.
Migración de estado	El estado de la aplicación se migra copiando reclamaciones de volúmenes persistentes específicas y objetos de Kubernetes al clúster de destino.
Migración de retroceso	La migración de retroceso retrotrae toda la migración.

¹ Llame al clúster de *destino* en la consola web de MTC.

5.2. FLUJO DE TRABAJO DE MTC

Puede migrar los recursos de Kubernetes, los datos de volúmenes persistentes y las imágenes de contenedores internos a OpenShift Container Platform 4.10 utilizando la consola web de Migration Toolkit for Containers (MTC) o la API de Kubernetes.

MTC migra los siguientes recursos:

- Un espacio de nombres especificado en un plan de migración.
- Recursos de espacios de nombres: cuando MTC migra un espacio de nombres, migra todos los objetos y recursos asociados a ese espacio de nombres, como los servicios o los pods. Además, si un recurso que existe en el espacio de nombres, pero no en el nivel del clúster, depende de un recurso que existe en el nivel del clúster, MTC migra ambos recursos.
Por ejemplo, una restricción de contexto de seguridad (SCC) es un recurso que existe a nivel del

clúster y una cuenta de servicio (SA) es un recurso que existe a nivel del espacio de nombres. Si existe una SA en un espacio de nombres que MTC migra, MTC localiza automáticamente cualquier SCC vinculada a la SA y también la migra. Del mismo modo, MTC migra las reclamaciones de volúmenes persistentes que están vinculadas a los volúmenes persistentes del espacio de nombres.



NOTA

Es posible que deba migrar los recursos con alcance del clúster de forma manual según el recurso.

- Recursos personalizados (CR) y definiciones de recursos personalizados (CRD): MTC migra automáticamente los CR y las CRD a nivel del espacio de nombres.

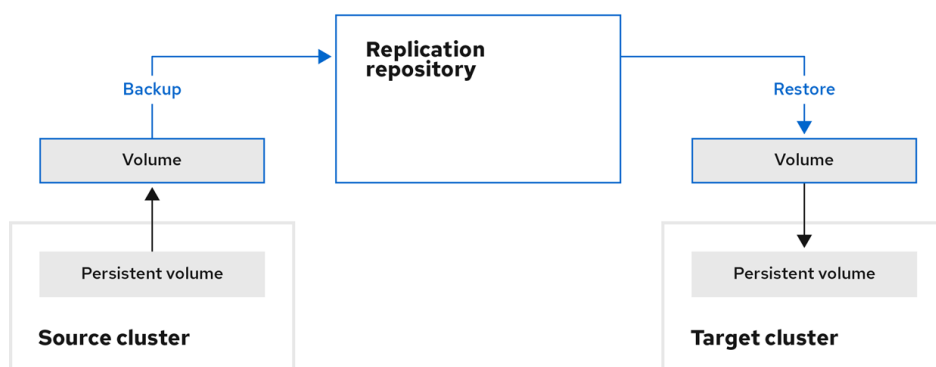
La migración de una aplicación con la consola web de MTC implica los siguientes pasos:

1. Instale el operador de Migration Toolkit for Containers en todos los clústeres.
Puede instalar el operador de Migration Toolkit for Containers en un entorno restringido con acceso a Internet limitado o nulo. Los clústeres de origen y destino deben tener acceso a la red entre sí y a un registro de réplica.
2. Configure el repositorio de replicación, un almacenamiento de objetos intermedio que MTC utiliza para migrar los datos.
Los clústeres de origen y destino deben tener acceso a la red del repositorio de replicación durante la migración. Si utiliza un servidor proxy, debe configurarlo para permitir el tráfico de red entre el repositorio de replicación y los clústeres.
3. Añada el clúster de origen a la consola web de MTC.
4. Añada el repositorio de replicación a la consola web de MTC.
5. Cree un plan de migración con una de las siguientes opciones de migración de datos:
 - **Copiar:** MTC copia los datos del clúster de origen en el repositorio de replicación y del repositorio de replicación en el clúster de destino.



NOTA

Si utiliza la migración directa de imágenes o la migración directa de volúmenes, las imágenes o los volúmenes se copian directamente del clúster de origen al de destino.



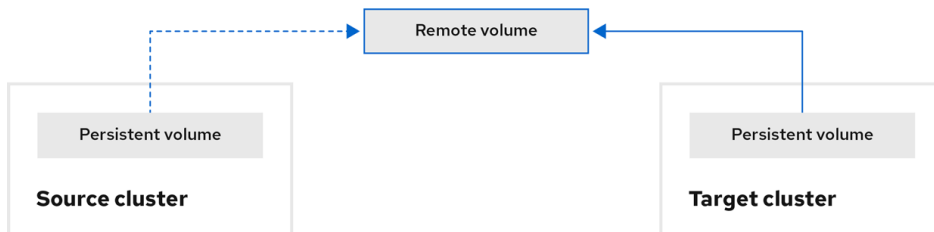
OpenShift_45_1019

- **Mover:** MTC desmonta un volumen remoto, por ejemplo, NFS, del clúster de origen, crea un recurso de PV en el clúster de destino que apunta al volumen remoto y, luego, monta el volumen remoto en el clúster de destino. Las aplicaciones que se ejecutan en el clúster de destino utilizan el mismo volumen remoto que utilizaba el clúster de origen. El volumen remoto debe ser accesible para los clústeres de origen y destino.



NOTA

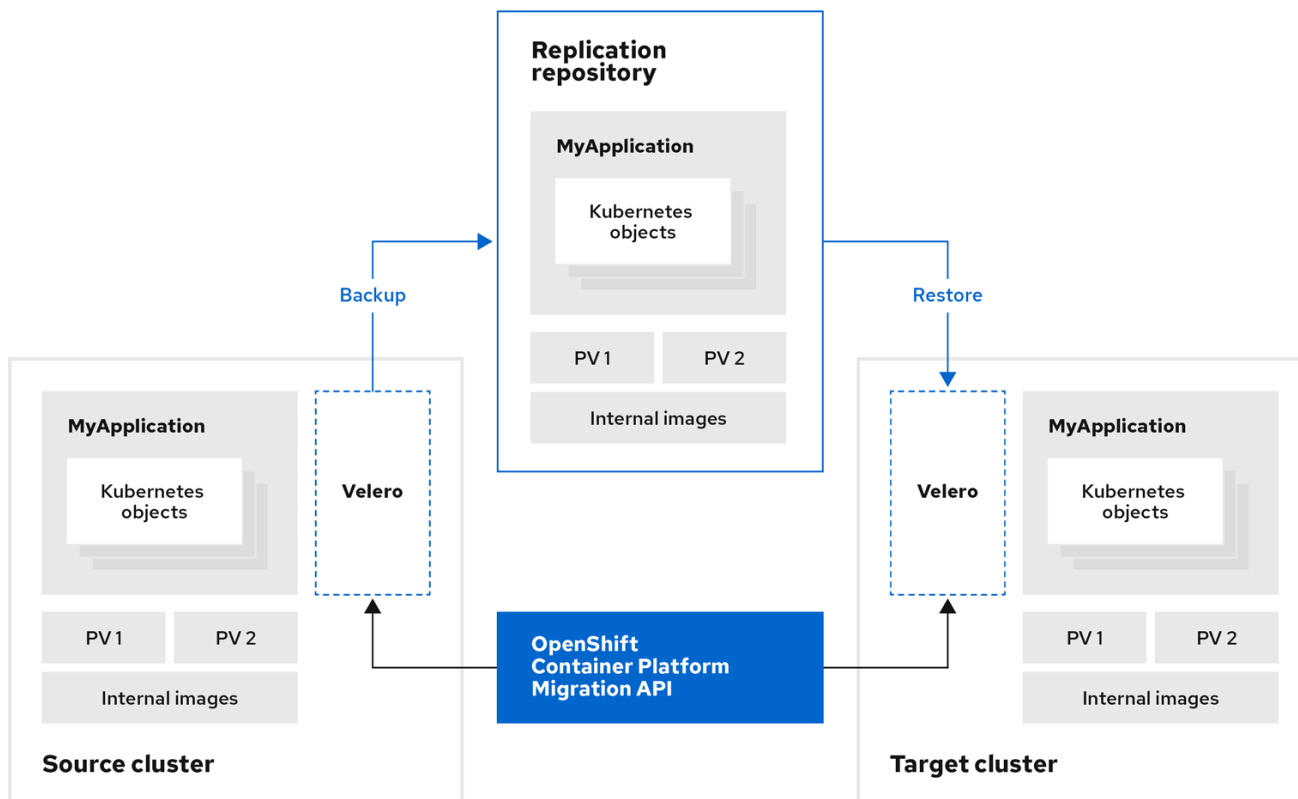
Aunque el repositorio de replicación no aparece en este diagrama, es necesario para la migración.



OpenShift_45_1019

6. Ejecute el plan de migración con una de las siguientes opciones:

- La **migración por etapas** copia los datos en el clúster de destino sin detener la aplicación. La migración por etapas puede ejecutarse varias veces para que la mayor parte de los datos se copien en el destino antes de la migración. Ejecutar una o más migraciones por etapas reduce la duración de la migración de transición.
- La **transición** detiene la aplicación en el clúster de origen y mueve los recursos al clúster de destino.
Opcional: puede desactivar la casilla de verificación **Detener transacciones en el clúster de origen durante la migración**.



OpenShift_45_1019

5.3. ACERCA DE LOS MÉTODOS DE COPIA DE DATOS

Migration Toolkit for Containers (MTC) admite los métodos de copia de datos de instantáneas y sistemas de archivos para migrar los datos del clúster de origen al clúster de destino. Puede seleccionar un método que se adapte a su entorno y que sea compatible con su proveedor de almacenamiento.

5.3.1. Método de copia de sistemas de archivos

MTC copia los archivos de datos del clúster de origen en el repositorio de replicación y, luego, en el clúster de destino.

El método de copia de sistemas de archivos utiliza Restic para la migración indirecta o Rsync para la migración directa de volúmenes.

Tabla 5.2. Resumen del método de copia de sistemas de archivos

Beneficios	Limitaciones
<ul style="list-style-type: none"> ● Los clústeres pueden tener diferentes clases de almacenamiento. ● Es compatible con todos los proveedores de almacenamiento S3. ● La verificación de datos opcional cuenta con la suma de comprobación. ● Admite la migración directa de volúmenes, lo que aumenta considerablemente el rendimiento. 	<ul style="list-style-type: none"> ● Es más lento que el método de copia de instantáneas. ● La verificación de datos opcional reduce significativamente el rendimiento.

5.3.2. Método de copia de instantáneas

MTC copia una instantánea de los datos del clúster de origen en el repositorio de replicación de un proveedor de la nube. Los datos se restauran en el clúster de destino.

El método de copia de instantáneas puede utilizarse con Amazon Web Services, Google Cloud Provider y Microsoft Azure.

Tabla 5.3. Resumen del método de copia de instantáneas

Beneficios	Limitaciones
<ul style="list-style-type: none"> ● Es más rápido que el método de copia del sistema de archivos. 	<ul style="list-style-type: none"> ● El proveedor de la nube debe admitir las instantáneas. ● Los clústeres deben estar en el mismo proveedor de la nube. ● Los clústeres deben estar en la misma localidad o región. ● Los clústeres deben tener la misma clase de almacenamiento. ● La clase de almacenamiento debe ser compatible con las instantáneas. ● No admite la migración directa de volúmenes.

5.4. MIGRACIÓN DIRECTA DE VOLÚMENES Y MIGRACIÓN DIRECTA DE IMÁGENES

Puede utilizar la migración directa de imágenes (DIM) y la migración directa de volúmenes (DVM) para migrar imágenes y datos directamente del clúster de origen al de destino.

Si ejecuta la DVM con nodos que están en diferentes zonas de disponibilidad, la migración podría fallar porque los pods migrados no pueden acceder a la reclamación de volúmenes persistentes.

La DIM y la DVM tienen importantes ventajas de rendimiento porque se omiten los pasos intermedios de copia de seguridad de los archivos del clúster de origen al repositorio de replicación y de restauración de los archivos del repositorio de replicación al clúster de destino. Los datos se transfieren con [Rsync](#).

La DIM y la DVM tienen requisitos adicionales.

CAPÍTULO 6. INSTALACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS

Puede instalar Migration Toolkit for Containers (MTC) en OpenShift Container Platform 3 y 4.

Después de instalar el operador de Migration Toolkit for Containers en OpenShift Container Platform 4.10 mediante Operator Lifecycle Manager, instale manualmente el operador de Migration Toolkit for Containers heredado en OpenShift Container Platform 3.

Por defecto, la consola web de MTC y el pod **Migration Controller** se ejecutan en el clúster de destino. Puede configurar el manifiesto de recursos personalizados de **Migration Controller** para que ejecute la consola web de MTC y el pod **Migration Controller** en el [clúster de origen o en un clúster remoto](#).

Después de haber instalado MTC, debe configurar un almacenamiento de objetos para utilizarlo como repositorio de replicación.

Para desinstalar MTC, consulte [Desinstalación de MTC y eliminación de recursos](#).

6.1. DIRECTRICES DE COMPATIBILIDAD

Debe instalar el operador de Migration Toolkit for Containers (MTC) compatible con su versión de OpenShift Container Platform.

Definiciones

plataforma heredada

OpenShift Container Platform 4.5 y anteriores.

plataforma moderna

OpenShift Container Platform 4.6 y posteriores.

operador heredado

El Operador MTC diseñado para plataformas heredadas.

operador moderno

El Operador MTC diseñado para plataformas modernas.

grupo de control


El clúster que ejecuta el controlador MTC y la GUI.

grupo remoto

Un clúster de origen o destino para una migración que ejecuta Velero. El clúster de control se comunica con los clústeres remotos a través de la API de Velero para impulsar las migraciones.

Tabla 6.1. Compatibilidad con el MTC: Migración desde una plataforma heredada

	OpenShift Container Platform 4.5 o anterior	OpenShift Container Platform 4.6 posterior
--	---	--

	OpenShift Container Platform 4.5 o anterior	OpenShift Container Platform 4.6 posterior
Última versión del MTC	<p>MTC 1.7.z</p> <p>Operador legado 1.7: Instalar manualmente con el archivo operator.yml.</p> <div style="display: flex; align-items: center;">  <div> <p>IMPORTANTE</p> <p>Este clúster no puede ser el de control.</p> </div> </div>	<p>MTC 1.7.z</p> <p>Instalación con OLM, canal de liberación release-v1.7</p>
Versión estable de MTC	<p>MTC 1.5</p> <p>Operador heredado 1.5: Instalar manualmente con el archivo operator.yml.</p>	<p>MTC 1.6.z</p> <p>Instalación con OLM, canal de liberación release-v1.6</p>



NOTA

Existen casos en los que las restricciones de red impiden que los clusters modernos se conecten a otros clusters que participan en la migración. Por ejemplo, al migrar de un clúster de OpenShift Container Platform 3.11 en las instalaciones a un clúster moderno de OpenShift Container Platform en la nube, donde el clúster moderno no puede conectarse al clúster de OpenShift Container Platform 3.11.

Con MTC 1.7, si uno de los clústeres remotos no puede comunicarse con el clúster de control debido a restricciones de red, utilice el comando **crane tunnel-api**.

Con la versión estable de MTC, aunque siempre se debe designar el clúster más moderno como clúster de control, en este caso concreto es posible designar el clúster heredado como clúster de control y empujar las cargas de trabajo al clúster remoto.

6.2. INSTALACIÓN DEL OPERADOR DE MIGRATION TOOLKIT FOR CONTAINERS HEREDADO EN OPENSIFT CONTAINER PLATFORM 3

Puede instalar manualmente el operador de Migration Toolkit for Containers heredado en OpenShift Container Platform 3.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.
- Debe tener acceso a **registry.redhat.io**.
- Debe tener instalado **Podman**.
- Debe crear un [secreto de flujo de imágenes](#) y copiarlo en cada nodo del clúster.

Procedimiento

1. Inicie sesión en **registry.redhat.io** con sus credenciales del Portal del cliente de Red Hat:

```
$ sudo podman login registry.redhat.io
```

2. Descargue el archivo **operator.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/operator.yml ./
```

3. Descargue el archivo **controller.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/controller.yml ./
```

4. Inicie sesión en su clúster de OpenShift Container Platform 3.

5. Verifique que el clúster pueda autenticarse con **registry.redhat.io**:

```
$ oc run test --image registry.redhat.io/ubi8 --command sleep infinity
```

6. Cree el objeto del operador de Migration Toolkit for Containers:

```
$ oc create -f operator.yml
```

Ejemplo de salida

```
namespace/openshift-migration created
rolebinding.rbac.authorization.k8s.io/system:deployers created
serviceaccount/migration-operator created
customresourcedefinition.apiextensions.k8s.io/migrationcontrollers.migration.openshift.io
created
role.rbac.authorization.k8s.io/migration-operator created
rolebinding.rbac.authorization.k8s.io/migration-operator created
clusterrolebinding.rbac.authorization.k8s.io/migration-operator created
deployment.apps/migration-operator created
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-builders" already exists 1
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-pullers" already exists
```

- 1** Puede ignorar los mensajes de **error del servidor (AlreadyExists)**. Son causados por el operador de Migration Toolkit for Containers que crea recursos para versiones anteriores de OpenShift Container Platform 3 que se proporcionan en versiones posteriores.

7. Cree el objeto **MigrationController**:

```
$ oc create -f controller.yml
```

8. Compruebe que los pods de MTC estén en funcionamiento:


```
$ oc get pods -n openshift-migration
```

6.3. INSTALACIÓN DEL OPERADOR DE MIGRATION TOOLKIT FOR CONTAINERS EN OPENSIFT CONTAINER PLATFORM 4.10

Instale el operador de Migration Toolkit for Containers en OpenShift Container Platform 4.10 con Operator Lifecycle Manager.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.

Procedimiento

1. En la consola web de OpenShift Container Platform, haga clic en **Operators** (Operadores) → **OperatorHub**.
2. Utilice el campo **Filter by keyword** (Filtrar por palabra clave) para encontrar el **operador de Migration Toolkit for Containers**.
3. Seleccione el **operador de Migration Toolkit for Containers** y haga clic en **Install** (Instalar).
4. Haga clic en **Install** (Instalar).
En la página **Installed Operators** (Operadores instalados), el **operador de Migration Toolkit for Containers** aparece en el proyecto **openshift-migration** con el estado **Succeeded** (Correcto).
5. Haga clic en el **operador de Migration Toolkit for Containers**.
6. En **Provided APIs** (API proporcionadas), localice el mosaico **Migration Controller** (Controlador de migración) y haga clic en **Create Instance** (Crear instancia).
7. Haga clic en **Create** (Crear).
8. Haga clic en **Workloads** (Cargas de trabajo) → **Pods** para comprobar que los pods de MTC se están ejecutando.

6.4. CONFIGURACIÓN DE LOS PROXIES

Para OpenShift Container Platform 4.1 y versiones anteriores, debe configurar los proxies en el manifiesto de recursos personalizados (CR) de **MigrationController** después de instalar el operador de Migration Toolkit for Containers dado que estas versiones no admiten un objeto **proxy** para todo el clúster.

Para OpenShift Container Platform 4.2 a 4.10, Migration Toolkit for Containers (MTC) hereda la configuración del proxy de todo el clúster. Puede cambiar los parámetros del proxy si desea anular la configuración del proxy en todo el clúster.

Debe configurar los proxies para que permitan el protocolo SPDY y reenvíen el encabezado **HTTP de actualización** al servidor de la API. De lo contrario, se muestra un error de **solicitud de actualización requerida**. El CR **MigrationController** utiliza SPDY para ejecutar comandos dentro de los pods remotos. El encabezado **HTTP Upgrade** es necesario para abrir una conexión websocket con el servidor de la API.

Migración directa de volúmenes

Si realiza una migración directa de volúmenes (DVM) desde un clúster de origen detrás de un proxy, debe configurar el proxy Stunnel. Stunnel crea un túnel transparente entre los clústeres de origen y destino para la conexión TCP sin cambiar los certificados.

La DVM solo admite un proxy. El clúster de origen no puede acceder a la ruta del clúster de destino si éste también está detrás de un proxy.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.

Procedimiento

1. Obtenga el manifiesto del CR **MigrationController**:

```
$ oc get migrationcontroller <migration_controller> -n openshift-migration
```

2. Actualice los parámetros del proxy:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigrationController
metadata:
  name: <migration_controller>
  namespace: openshift-migration
...
spec:
  stunnel_tcp_proxy: http://<username>:<password>@<ip>:<port> 1
  httpProxy: http://<username>:<password>@<ip>:<port> 2
  httpsProxy: http://<username>:<password>@<ip>:<port> 3
  noProxy: example.com 4
```

- 1 URL del proxy Stunnel para la migración directa de volúmenes.
- 2 URL del proxy para crear conexiones HTTP fuera del clúster. El esquema de la URL debe ser **http**.
- 3 URL del proxy para crear conexiones HTTPS fuera del clúster. Si no se especifica, se utiliza **httpProxy** para las conexiones HTTP y HTTPS.
- 4 Lista separada por comas de nombres de dominio de destino, dominios, direcciones IP u otros CIDR de red para excluir el proxy.

Introduzca un dominio con `.` para que coincida solo con los subdominios. Por ejemplo, **.y.com** coincide con **x.y.com**, pero no con **y.com**. Utilice `*` para evitar el proxy en todos los destinos. Si escala a los trabajadores que no están incluidos en la red definida por el campo **networking.machineNetwork[].cidr** de la configuración de instalación, debe añadirlos a esta lista para evitar problemas de conexión.

Este campo se ignora si no se establecen los campos **httpProxy** y **httpsProxy**.

3. Guarde el manifiesto como **migration-controller.yaml**.

4. Aplique el manifiesto actualizado:

```
$ oc replace -f migration-controller.yaml -n openshift-migration
```

Para obtener más información, consulte [Configuración del proxy para todo el clúster](#) .

6.5. CONFIGURACIÓN DE UN REPOSITORIO DE REPLICACIÓN

Debe configurar un almacenamiento de objetos para utilizarlo como repositorio de replicación. Migration Toolkit for Containers (MTC) copia los datos del clúster de origen en el repositorio de replicación y, luego, del repositorio de replicación en el clúster de destino.

MTC admite los [métodos de copia de datos de instantáneas y sistemas de archivos](#) para migrar los datos del clúster de origen al clúster de destino. Puede seleccionar un método que se adapte a su entorno y que sea compatible con su proveedor de almacenamiento.

Se admiten los siguientes proveedores de almacenamiento:

- [Multicloud Object Gateway](#)
- [Amazon Web Services S3](#)
- [Google Cloud Platform](#)
- [Microsoft Azure Blob](#)
- Almacenamiento de objetos genérico S3, por ejemplo, Minio o Ceph S3

6.5.1. Requisitos previos

- Todos los clústeres deben tener un acceso de red ininterrumpido al repositorio de replicación.
- Si utiliza un servidor proxy con un repositorio de replicación alojado internamente, debe asegurarse de que el proxy permita el acceso al repositorio de replicación.

6.5.2. Recuperación de las credenciales de Multicloud Object Gateway

Debe recuperar las credenciales de Multicloud Object Gateway (MCG) y el terminal de S3 para configurar MCG como repositorio de replicación para Migration Toolkit for Containers (MTC). Debe recuperar las credenciales de Multicloud Object Gateway (MCG) para crear un recurso personalizado (CR) **secreto** para OpenShift API for Data Protection (OADP).

MCG es un componente de OpenShift Container Storage.

Requisitos previos

- Debe implementar OpenShift Container Storage utilizando la [guía de implementación de OpenShift Container Storage](#) adecuada.

Procedimiento

1. Obtenga el terminal S3, **AWS_ACCESS_KEY_ID** y **AWS_SECRET_ACCESS_KEY** ejecutando el comando **describe** en el recurso personalizado **NooBaa**. Estas credenciales se utilizan para añadir MCG como repositorio de replicación.

6.5.3. Configuración de Amazon Web Services

Configure el almacenamiento de objetos S3 de Amazon Web Services (AWS) como un repositorio de replicación para Migration Toolkit for Containers (MTC).

Requisitos previos

- Debe tener instalada la [CLI de AWS](#).
- El bucket de almacenamiento S3 de AWS debe ser accesible para los clústeres de origen y destino.
- Si utiliza el método de copia de instantáneas:
 - Debe tener acceso a EC2 Elastic Block Storage (EBS).
 - Los clústeres de origen y destino deben estar en la misma región.
 - Los clústeres de origen y destino deben tener la misma clase de almacenamiento.
 - La clase de almacenamiento debe ser compatible con las instantáneas.

Procedimiento

1. Establezca la variable **BUCKET**:

```
$ BUCKET=<your_bucket>
```

2. Establezca la variable **REGION**:

```
$ REGION=<your_region>
```

3. Cree un bucket S3 de AWS:

```
$ aws s3api create-bucket \
  --bucket $BUCKET \
  --region $REGION \
  --create-bucket-configuration LocationConstraint=$REGION 1
```

- 1** **us-east-1** no admite **LocationConstraint**. Si su región es **us-east-1**, omita **--create-bucket-configuration LocationConstraint=\$REGION**.

4. Cree el usuario IAM:

```
$ aws iam create-user --user-name velero 1
```

- 1** Si quiere usar Velero para hacer copias de seguridad de varios clústeres con múltiples buckets de S3, cree un nombre de usuario único para cada clúster.

5. Cree un archivo **velero-policy.json**:

```
$ cat > velero-policy.json <<EOF
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVolumes",
      "ec2:DescribeSnapshots",
      "ec2:CreateTags",
      "ec2:CreateVolume",
      "ec2:CreateSnapshot",
      "ec2>DeleteSnapshot"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3>DeleteObject",
      "s3:PutObject",
      "s3:AbortMultipartUpload",
      "s3:ListMultipartUploadParts"
    ],
    "Resource": [
      "arn:aws:s3:::${BUCKET}/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::${BUCKET}"
    ]
  }
]
}
EOF

```

6. Adjunte las políticas para conceder los permisos necesarios al usuario **velero**:

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero \
  --policy-document file://velero-policy.json

```

7. Cree la clave de acceso para el usuario **velero**:

```

$ aws iam create-access-key --user-name velero

```

Ejemplo de salida

```

{
  "AccessKey": {

```

```

    "UserName": "velero",
    "Status": "Active",
    "CreateDate": "2017-07-31T22:24:41.576Z",
    "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>,
    "AccessKeyId": <AWS_ACCESS_KEY_ID>
  }
}

```

Registre **AWS_SECRET_ACCESS_KEY** y **AWS_ACCESS_KEY_ID**. Utilice las credenciales para añadir AWS como repositorio de replicación.

6.5.4. Configuración de Google Cloud Platform

Configure un bucket de almacenamiento de Google Cloud Platform (GCP) como repositorio de replicación para Migration Toolkit for Containers (MTC).

Requisitos previos

- Debe tener instaladas las herramientas de la CLI **gcloud** y **gsutil**. Consulte la [documentación de Google Cloud](#) para obtener más detalles.
- El bucket de almacenamiento de GCP debe ser accesible para los clústeres de origen y destino.
- Si utiliza el método de copia de instantáneas:
 - Los clústeres de origen y destino deben estar en la misma región.
 - Los clústeres de origen y destino deben tener la misma clase de almacenamiento.
 - La clase de almacenamiento debe ser compatible con las instantáneas.

Procedimiento

1. Inicie sesión en GCP:

```
$ gcloud auth login
```

2. Establezca la variable **BUCKET**:

```
$ BUCKET=<bucket> 1
```

- 1** Especifica el nombre del bucket.

3. Cree el bucket de almacenamiento:

```
$ gsutil mb gs://$BUCKET/
```

4. Establezca la variable **PROJECT_ID** para su proyecto activo:

```
$ PROJECT_ID=$(gcloud config get-value project)
```

5. Cree una cuenta de servicio:

```
$ gcloud iam service-accounts create velero \
  --display-name "Velero service account"
```

6. Enumere sus cuentas de servicio:

```
$ gcloud iam service-accounts list
```

7. Establezca la variable **SERVICE_ACCOUNT_EMAIL** para que coincida con el valor de su **correo electrónico**:

```
$ SERVICE_ACCOUNT_EMAIL=$(gcloud iam service-accounts list \
  --filter="displayName:Velero service account" \
  --format 'value(email)')
```

8. Adjunte las políticas para conceder los permisos necesarios al usuario **velero**:

```
$ ROLE_PERMISSIONS=(
  compute.disks.get
  compute.disks.create
  compute.disks.createSnapshot
  compute.snapshots.get
  compute.snapshots.create
  compute.snapshots.useReadOnly
  compute.snapshots.delete
  compute.zones.get
)
```

9. Cree el rol personalizado **velero.server**:

```
$ gcloud iam roles create velero.server \
  --project $PROJECT_ID \
  --title "Velero Server" \
  --permissions "${IFS=","; echo "${ROLE_PERMISSIONS[*]}")"
```

10. Añada la vinculación de la política IAM al proyecto:

```
$ gcloud projects add-iam-policy-binding $PROJECT_ID \
  --member serviceAccount:$SERVICE_ACCOUNT_EMAIL \
  --role projects/$PROJECT_ID/roles/velero.server
```

11. Actualice la cuenta de servicio IAM:

```
$ gsutil iam ch serviceAccount:$SERVICE_ACCOUNT_EMAIL:objectAdmin gs://${BUCKET}
```

12. Guarde las claves de la cuenta de servicio IAM en el archivo **credentials-velero** en el directorio actual:

```
$ gcloud iam service-accounts keys create credentials-velero \
  --iam-account $SERVICE_ACCOUNT_EMAIL
```

Utilice el archivo **credentials-velero** para añadir GCP como repositorio de replicación.

6.5.5. Configuración de Microsoft Azure

Configure un contenedor de almacenamiento de Microsoft Azure Blob como repositorio de replicación para Migration Toolkit for Containers (MTC).

Requisitos previos

- Debe tener instalada la [CLI de Azure](#).
- El contenedor de almacenamiento de Azure Blob debe ser accesible para los clústeres de origen y destino.
- Si utiliza el método de copia de instantáneas:
 - Los clústeres de origen y destino deben estar en la misma región.
 - Los clústeres de origen y destino deben tener la misma clase de almacenamiento.
 - La clase de almacenamiento debe ser compatible con las instantáneas.

Procedimiento

1. Inicie sesión en Azure:

```
$ az login
```

2. Establezca la variable **AZURE_RESOURCE_GROUP**:

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

3. Cree un grupo de recursos de Azure:

```
$ az group create -n $AZURE_RESOURCE_GROUP --location CentralUS 1
```

- 1** Especifique su ubicación.

4. Establezca la variable **AZURE_STORAGE_ACCOUNT_ID**:

```
$ AZURE_STORAGE_ACCOUNT_ID="velero$(uuidgen | cut -d '-' -f5 | tr '[A-Z]' '[a-z]')"
```

5. Cree una cuenta de almacenamiento de Azure:

```
$ az storage account create \  
  --name $AZURE_STORAGE_ACCOUNT_ID \  
  --resource-group $AZURE_BACKUP_RESOURCE_GROUP \  
  --sku Standard_GRS \  
  --encryption-services blob \  
  --https-only true \  
  --kind BlobStorage \  
  --access-tier Hot
```

6. Establezca la variable **BLOB_CONTAINER**:


```
$ BLOB_CONTAINER=velero
```

7. Cree un contenedor de almacenamiento de Azure Blob:

```
$ az storage container create \
  -n $BLOB_CONTAINER \
  --public-access off \
  --account-name $AZURE_STORAGE_ACCOUNT_ID
```

8. Cree un servicio principal y credenciales para **velero**:

```
$ AZURE_SUBSCRIPTION_ID=`az account list --query '[?isDefault].id' -o tsv` \
  AZURE_TENANT_ID=`az account list --query '[?isDefault].tenantId' -o tsv` \
  AZURE_CLIENT_SECRET=`az ad sp create-for-rbac --name "velero" \
  --role "Contributor" --query 'password' -o tsv` \
  AZURE_CLIENT_ID=`az ad sp list --display-name "velero" \
  --query '[0].appId' -o tsv`
```

9. Guarde las credenciales de la entidad principal de servicio en el archivo **credentials-velero**:

```
$ cat << EOF > ./credentials-velero
  AZURE_SUBSCRIPTION_ID=${AZURE_SUBSCRIPTION_ID}
  AZURE_TENANT_ID=${AZURE_TENANT_ID}
  AZURE_CLIENT_ID=${AZURE_CLIENT_ID}
  AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
  AZURE_RESOURCE_GROUP=${AZURE_RESOURCE_GROUP}
  AZURE_CLOUD_NAME=AzurePublicCloud
  EOF
```

Utilice el archivo **credentials-velero** para añadir Azure como repositorio de replicación.

6.5.6. Recursos adicionales

- [Flujo de trabajo de MTC](#)
- [Acerca de los métodos de copia de datos](#)
- [Adición de un repositorio de replicación a la consola web de MTC](#)

6.6. DESINSTALACIÓN DE MTC Y ELIMINACIÓN DE RECURSOS

Puede desinstalar Migration Toolkit for Containers (MTC) y eliminar sus recursos para limpiar el clúster.



NOTA

Al borrar las CRD de **velero** se elimina Velero del clúster.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster**.

Procedimiento

1. Elimine el recurso personalizado (CR) **MigrationController** de todos los clústeres:

```
$ oc delete migrationcontroller <migration_controller>
```

2. Desinstale el operador de Migration Toolkit for Containers de OpenShift Container Platform 4 utilizando Operator Lifecycle Manager.

3. Elimine los recursos del clúster de todos los clústeres ejecutando los siguientes comandos:

- Definiciones de recursos personalizados (CRD) de **migration**:

```
$ oc delete $(oc get crds -o name | grep 'migration.openshift.io')
```

- CRD de **velero**:

```
$ oc delete $(oc get crds -o name | grep 'velero')
```

- Roles del clúster de **migration**:

```
$ oc delete $(oc get clusterroles -o name | grep 'migration.openshift.io')
```

- Roles del clúster de **migration-operator**:

```
$ oc delete clusterrole migration-operator
```

- Roles del clúster de **velero**:

```
$ oc delete $(oc get clusterroles -o name | grep 'velero')
```

- Vinculaciones de roles del clúster de **migration**:

```
$ oc delete $(oc get clusterrolebindings -o name | grep 'migration.openshift.io')
```

- Vinculaciones de roles del clúster de **migration-operator**:

```
$ oc delete clusterrolebindings migration-operator
```

- Vinculaciones de roles del clúster de **velero**:

```
$ oc delete $(oc get clusterrolebindings -o name | grep 'velero')
```

CAPÍTULO 7. INSTALACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS EN UN ENTORNO DE RED RESTRINGIDO

Puede instalar Migration Toolkit for Containers (MTC) en OpenShift Container Platform 3 y 4 en un entorno de red restringido realizando los siguientes procedimientos:

1. Cree un [catálogo de operadores en espejo](#).
Este proceso crea un archivo **mapping.txt** que contiene la asignación entre la imagen **registry.redhat.io** y su imagen de registro en espejo. El archivo **mapping.txt** es necesario para instalar el operador en el clúster de origen.
2. Instale el operador de Migration Toolkit for Containers en el clúster de destino de OpenShift Container Platform 4.10 mediante Operator Lifecycle Manager.
Por defecto, la consola web de MTC y el pod **Migration Controller** se ejecutan en el clúster de destino. Puede configurar el manifiesto de recursos personalizados de **Migration Controller** para que ejecute la consola web de MTC y el pod **Migration Controller** en el [clúster de origen o en un clúster remoto](#).
3. Instale el operador de Migration Toolkit for Containers *heredado* en el clúster de origen de OpenShift Container Platform 3 desde la interfaz de la línea de comandos.
4. Configure el almacenamiento de objetos para utilizarlo como repositorio de replicación.

Para desinstalar MTC, consulte [Desinstalación de MTC y eliminación de recursos](#).

7.1. DIRECTRICES DE COMPATIBILIDAD

Debe instalar el operador de Migration Toolkit for Containers (MTC) compatible con su versión de OpenShift Container Platform.

Definiciones

plataforma heredada

OpenShift Container Platform 4.5 y anteriores.

plataforma moderna

OpenShift Container Platform 4.6 y posteriores.

operador heredado

El Operador MTC diseñado para plataformas heredadas.

operador moderno

El Operador MTC diseñado para plataformas modernas.


grupo de control

El clúster que ejecuta el controlador MTC y la GUI.

grupo remoto

Un clúster de origen o destino para una migración que ejecuta Velero. El clúster de control se comunica con los clústeres remotos a través de la API de Velero para impulsar las migraciones.

Tabla 7.1. Compatibilidad con el MTC: Migración desde una plataforma heredada

	OpenShift Container Platform 4.5 o anterior	OpenShift Container Platform 4.6 posterior
Última versión del MTC	<p>MTC 1.7.z</p> <p>Operador legado 1.7: Instalar manualmente con el archivo operator.yml.</p> <div style="display: flex; align-items: center;">  <div style="margin-left: 10px;"> <p>IMPORTANTE</p> <p>Este clúster no puede ser el de control.</p> </div> </div>	<p>MTC 1.7.z</p> <p>Instalación con OLM, canal de liberación release-v1.7</p>
Versión estable de MTC	<p>MTC 1.5</p> <p>Operador heredado 1.5: Instalar manualmente con el archivo operator.yml.</p>	<p>MTC 1.6.z</p> <p>Instalación con OLM, canal de liberación release-v1.6</p>



NOTA

Existen casos en los que las restricciones de red impiden que los clusters modernos se conecten a otros clusters que participan en la migración. Por ejemplo, al migrar de un clúster de OpenShift Container Platform 3.11 en las instalaciones a un clúster moderno de OpenShift Container Platform en la nube, donde el clúster moderno no puede conectarse al clúster de OpenShift Container Platform 3.11.

Con MTC 1.7, si uno de los clústeres remotos no puede comunicarse con el clúster de control debido a restricciones de red, utilice el comando **crane tunnel-api**.

Con la versión estable de MTC, aunque siempre se debe designar el clúster más moderno como clúster de control, en este caso concreto es posible designar el clúster heredado como clúster de control y empujar las cargas de trabajo al clúster remoto.

7.2. INSTALACIÓN DEL OPERADOR DE MIGRATION TOOLKIT FOR CONTAINERS EN OPENSIFT CONTAINER PLATFORM 4.10

Instale el operador de Migration Toolkit for Containers en OpenShift Container Platform 4.10 con Operator Lifecycle Manager.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.
- Debe crear un catálogo de operadores a partir de una imagen en espejo en un registro local.

Procedimiento

1. En la consola web de OpenShift Container Platform, haga clic en **Operators** (Operadores) → **OperatorHub**.

2. Utilice el campo **Filter by keyword** (Filtrar por palabra clave) para encontrar el **operador de Migration Toolkit for Containers**.
3. Seleccione el **operador de Migration Toolkit for Containers** y haga clic en **Install** (Instalar).
4. Haga clic en **Install** (Instalar).
En la página **Installed Operators** (Operadores instalados), el **operador de Migration Toolkit for Containers** aparece en el proyecto **openshift-migration** con el estado **Succeeded** (Correcto).
5. Haga clic en el **operador de Migration Toolkit for Containers**
6. En **Provided APIs** (API proporcionadas), localice el mosaico **Migration Controller** (Controlador de migración) y haga clic en **Create Instance** (Crear instancia).
7. Haga clic en **Create** (Crear).
8. Haga clic en **Workloads** (Cargas de trabajo) → **Pods** para comprobar que los pods de MTC se están ejecutando.

7.3. INSTALACIÓN DEL OPERADOR DE MIGRATION TOOLKIT FOR CONTAINERS HEREDADO EN OPENSIFT CONTAINER PLATFORM 3

Puede instalar manualmente el operador de Migration Toolkit for Containers heredado en OpenShift Container Platform 3.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.
- Debe tener acceso a **registry.redhat.io**.
- Debe tener instalado **Podman**.
- Debe crear un [secreto de flujo de imágenes](#) y copiarlo en cada nodo del clúster.
- Debe tener una estación de trabajo Linux con acceso a la red para poder descargar archivos de **registry.redhat.io**.
- Debe crear una imagen en espejo del catálogo de operadores.
- Debe instalar el operador de Migration Toolkit for Containers desde el catálogo de operadores en espejo en OpenShift Container Platform 4.10.

Procedimiento

1. Inicie sesión en **registry.redhat.io** con sus credenciales del Portal del cliente de Red Hat:

```
$ sudo podman login registry.redhat.io
```

2. Descargue el archivo **operator.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/operator.yml ./
```

3. Descargue el archivo **controller.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/controller.yml ./
```

4. Obtenga la asignación de la imagen del operador ejecutando el siguiente comando:

```
$ grep openshift-migration-legacy-rhel8-operator ./mapping.txt | grep rhmtc
```

El archivo **mapping.txt** se creó cuando se replicó el catálogo de operadores. El resultado muestra la asignación entre la imagen **registry.redhat.io** y su imagen de registro en espejo.

Ejemplo de salida

```
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-
operator@sha256:468a6126f73b1ee12085ca53a312d1f96ef5a2ca03442bcb63724af5e2614e8
a=<registry.apps.example.com>/rhmtc/openshift-migration-legacy-rhel8-operator
```

5. Actualice los valores de la **imagen** para los contenedores **ansible** y **operator** y el valor **REGISTRY** en el archivo **operator.yml**:

```
containers:
  - name: ansible
    image: <registry.apps.example.com>/rhmtc/openshift-migration-legacy-rhel8-
operator@sha256:
<468a6126f73b1ee12085ca53a312d1f96ef5a2ca03442bcb63724af5e2614e8a> 1
  ...
  - name: operator
    image: <registry.apps.example.com>/rhmtc/openshift-migration-legacy-rhel8-
operator@sha256:
<468a6126f73b1ee12085ca53a312d1f96ef5a2ca03442bcb63724af5e2614e8a> 2
  ...
  env:
    - name: REGISTRY
      value: <registry.apps.example.com> 3
```

1 2 Especifique su registro de réplica y el valor **sha256** de la imagen del operador.

3 Especifique su registro de réplica.

6. Inicie sesión en su clúster de OpenShift Container Platform 3.
7. Cree el objeto del operador de Migration Toolkit for Containers:

```
$ oc create -f operator.yml
```

Ejemplo de salida

```
namespace/openshift-migration created
rolebinding.rbac.authorization.k8s.io/system:deployers created
serviceaccount/migration-operator created
customresourcedefinition.apiextensions.k8s.io/migrationcontrollers.migration.openshift.io
created
```

```

role.rbac.authorization.k8s.io/migration-operator created
rolebinding.rbac.authorization.k8s.io/migration-operator created
clusterrolebinding.rbac.authorization.k8s.io/migration-operator created
deployment.apps/migration-operator created
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-builders" already exists 1
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-pullers" already exists
    
```

- 1** Puede ignorar los mensajes de **error del servidor (AlreadyExists)**. Son causados por el operador de Migration Toolkit for Containers que crea recursos para versiones anteriores de OpenShift Container Platform 3 que se proporcionan en versiones posteriores.

8. Cree el objeto **MigrationController**:

```
$ oc create -f controller.yml
```

9. Compruebe que los pods de MTC estén en funcionamiento:

```
$ oc get pods -n openshift-migration
```

7.4. CONFIGURACIÓN DE LOS PROXIES

Para OpenShift Container Platform 4.1 y versiones anteriores, debe configurar los proxies en el manifiesto de recursos personalizados (CR) de **MigrationController** después de instalar el operador de Migration Toolkit for Containers dado que estas versiones no admiten un objeto **proxy** para todo el clúster.

Para OpenShift Container Platform 4.2 a 4.10, Migration Toolkit for Containers (MTC) hereda la configuración del proxy de todo el clúster. Puede cambiar los parámetros del proxy si desea anular la configuración del proxy en todo el clúster.

Debe configurar los proxies para que permitan el protocolo SPDY y reenvíen el encabezado **HTTP de actualización** al servidor de la API. De lo contrario, se muestra un error de **solicitud de actualización requerida**. El CR **MigrationController** utiliza SPDY para ejecutar comandos dentro de los pods remotos. El encabezado **HTTP Upgrade** es necesario para abrir una conexión websocket con el servidor de la API.

Migración directa de volúmenes

Si realiza una migración directa de volúmenes (DVM) desde un clúster de origen detrás de un proxy, debe configurar el proxy Stunnel. Stunnel crea un túnel transparente entre los clústeres de origen y destino para la conexión TCP sin cambiar los certificados.

La DVM solo admite un proxy. El clúster de origen no puede acceder a la ruta del clúster de destino si éste también está detrás de un proxy.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.

Procedimiento

1. Obtenga el manifiesto del CR **MigrationController**:

```
$ oc get migrationcontroller <migration_controller> -n openshift-migration
```

2. Actualice los parámetros del proxy:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigrationController
metadata:
  name: <migration_controller>
  namespace: openshift-migration
...
spec:
  stunnel_tcp_proxy: http://<username>:<password>@<ip>:<port> 1
  httpProxy: http://<username>:<password>@<ip>:<port> 2
  httpsProxy: http://<username>:<password>@<ip>:<port> 3
  noProxy: example.com 4
```

- 1 URL del proxy Stunnel para la migración directa de volúmenes.
- 2 URL del proxy para crear conexiones HTTP fuera del clúster. El esquema de la URL debe ser **http**.
- 3 URL del proxy para crear conexiones HTTPS fuera del clúster. Si no se especifica, se utiliza **httpProxy** para las conexiones HTTP y HTTPS.
- 4 Lista separada por comas de nombres de dominio de destino, dominios, direcciones IP u otros CIDR de red para excluir el proxy.

Introduzca un dominio con `.` para que coincida solo con los subdominios. Por ejemplo, `.y.com` coincide con `x.y.com`, pero no con `y.com`. Utilice `*` para evitar el proxy en todos los destinos. Si escala a los trabajadores que no están incluidos en la red definida por el campo `networking.machineNetwork[].cidr` de la configuración de instalación, debe añadirlos a esta lista para evitar problemas de conexión.

Este campo se ignora si no se establecen los campos **httpProxy** y **httpsProxy**.

3. Guarde el manifiesto como **migration-controller.yaml**.
4. Aplique el manifiesto actualizado:

```
$ oc replace -f migration-controller.yaml -n openshift-migration
```

Para obtener más información, consulte [Configuración del proxy para todo el clúster](#) .

7.5. CONFIGURACIÓN DE UN REPOSITORIO DE REPLICACIÓN

Multicloud Object Gateway es la única opción admitida para un entorno de red restringido.

MTC admite los [métodos de copia de datos de instantáneas y sistemas de archivos](#) para migrar los datos del clúster de origen al clúster de destino. Puede seleccionar un método que se adapte a su entorno y que sea compatible con su proveedor de almacenamiento.

7.5.1. Requisitos previos

- Todos los clústeres deben tener un acceso de red ininterrumpido al repositorio de replicación.
- Si utiliza un servidor proxy con un repositorio de replicación alojado internamente, debe asegurarse de que el proxy permita el acceso al repositorio de replicación.

7.5.2. Recuperación de las credenciales de Multicloud Object Gateway

Debe recuperar las credenciales de Multicloud Object Gateway (MCG) para crear un recurso personalizado (CR) **secreto** para OpenShift API for Data Protection (OADP).

MCG es un componente de OpenShift Container Storage.

Requisitos previos

- Debe implementar OpenShift Container Storage utilizando la [guía de implementación de OpenShift Container Storage](#) adecuada.

Procedimiento

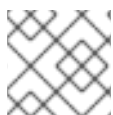
1. Obtenga el terminal S3, **AWS_ACCESS_KEY_ID** y **AWS_SECRET_ACCESS_KEY** ejecutando el comando **describe** en el recurso personalizado **NooBaa**.

7.5.3. Recursos adicionales

- [Documentación de entorno desconectado en Red Hat OpenShift Container Storage](#)
- [Flujo de trabajo de MTC](#)
- [Acerca de los métodos de copia de datos](#)
- [Adición de un repositorio de replicación a la consola web de MTC](#)

7.6. DESINSTALACIÓN DE MTC Y ELIMINACIÓN DE RECURSOS

Puede desinstalar Migration Toolkit for Containers (MTC) y eliminar sus recursos para limpiar el clúster.



NOTA

Al borrar las CRD de **velero** se elimina Velero del clúster.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster**.

Procedimiento

1. Elimine el recurso personalizado (CR) **MigrationController** de todos los clústeres:

```
$ oc delete migrationcontroller <migration_controller>
```

2. Desinstale el operador de Migration Toolkit for Containers de OpenShift Container Platform 4 utilizando Operator Lifecycle Manager.

3. Elimine los recursos del clúster de todos los clústeres ejecutando los siguientes comandos:

- Definiciones de recursos personalizados (CRD) de **migration**:

```
$ oc delete $(oc get crds -o name | grep 'migration.openshift.io')
```

- CRD de **velero**:

```
$ oc delete $(oc get crds -o name | grep 'velero')
```

- Roles del clúster de **migration**:

```
$ oc delete $(oc get clusterroles -o name | grep 'migration.openshift.io')
```

- Roles del clúster de **migration-operator**:

```
$ oc delete clusterrole migration-operator
```

- Roles del clúster de **velero**:

```
$ oc delete $(oc get clusterroles -o name | grep 'velero')
```

- Vinculaciones de roles del clúster de **migration**:

```
$ oc delete $(oc get clusterrolebindings -o name | grep 'migration.openshift.io')
```

- Vinculaciones de roles del clúster de **migration-operator**:

```
$ oc delete clusterrolebindings migration-operator
```

- Vinculaciones de roles del clúster de **velero**:

```
$ oc delete $(oc get clusterrolebindings -o name | grep 'velero')
```

CAPÍTULO 8. ACTUALIZACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS

Puede actualizar Migration Toolkit for Containers (MTC) en OpenShift Container Platform 4.10 mediante Operator Lifecycle Manager.

Puede actualizar MTC en OpenShift Container Platform 3 reinstalando el antiguo operador de Migration Toolkit for Containers.



IMPORTANTE

Si actualiza desde la versión 1.3 de MTC, debe realizar un procedimiento adicional para actualizar el recurso personalizado (CR) de **MigPlan**.

8.1. ACTUALIZACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS EN OPENSIFT CONTAINER PLATFORM 4.10

Puede actualizar Migration Toolkit for Containers (MTC) en OpenShift Container Platform 4.10 mediante Operator Lifecycle Manager.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster**.

Procedimiento

1. En la consola de OpenShift Container Platform, navegue hasta **Operators** (Operadores) → **Installed Operators** (Operadores instalados).
Los operadores que tienen una actualización pendiente muestran el estado **Upgrade available** (Actualización disponible).
2. Haga clic en el **operador de Migration Toolkit for Containers**
3. Haga clic en la pestaña **Subscription** (Suscripción). Cualquier actualización que requiera aprobación se muestra junto a **Upgrade Status** (Estado de la actualización). Por ejemplo, puede mostrar **1 requires approval** (1 requiere aprobación).
4. Haga clic en **1 requires approval** (1 requiere aprobación) y, luego, en **Preview Install Plan** (Vista previa del plan de instalación).
5. Revise los recursos que aparecen como disponibles para la actualización y haga clic en **Approve** (Aprobar).
6. Vuelva a navegar a la página **Operators → Installed Operators** (Operadores → Operadores instalados) para supervisar el progreso de la actualización. Una vez completado, el estado cambia a **Succeeded** (Correcto) y **Up to date** (Actualizado).
7. Haga clic en el **operador de Migration Toolkit for Containers**
8. En **Provided APIs** (API proporcionadas), localice el mosaico **Migration Controller** (Controlador de migración) y haga clic en **Create Instance** (Crear instancia).
9. Haga clic en **Workloads** (Cargas de trabajo) → **Pods** para comprobar que los pods de MTC se están ejecutando.

8.2. ACTUALIZACIÓN DE MIGRATION TOOLKIT FOR CONTAINERS EN OPENSIFT CONTAINER PLATFORM 3

Puede actualizar Migration Toolkit for Containers (MTC) en OpenShift Container Platform 3 instalando manualmente el operador de Migration Toolkit for Containers heredado.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster**.
- Debe tener acceso a **registry.redhat.io**.
- Debe tener instalado **Podman**.

Procedimiento

1. Inicie sesión en **registry.redhat.io** con sus credenciales del Portal del cliente de Red Hat:

```
$ sudo podman login registry.redhat.io
```

2. Descargue el archivo **operator.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/operator.yml ./
```

3. Sustituya el operador de Migration Toolkit for Containers:

```
$ oc replace --force -f operator.yml
```

4. Escale la implementación de **migration-operator** a **0** para detenerla:

```
$ oc scale -n openshift-migration --replicas=0 deployment/migration-operator
```

5. Escale la implementación de **migration-operator** a **1** para iniciarla y aplicar los cambios:

```
$ oc scale -n openshift-migration --replicas=1 deployment/migration-operator
```

6. Verifique que **migration-operator** se haya actualizado:

```
$ oc -o yaml -n openshift-migration get deployment/migration-operator | grep image: | awk -F
":" '{ print $NF }'
```

7. Descargue el archivo **controller.yml**:

```
$ sudo podman cp $(sudo podman create \
registry.redhat.io/rhmtc/openshift-migration-legacy-rhel8-operator:v1.5.3):/controller.yml ./
```

8. Cree el objeto **migration-controller**:

```
$ oc create -f controller.yml
```

9. Si ha añadido previamente el clúster de OpenShift Container Platform 3 a la consola web de MTC, debe actualizar el token de la cuenta de servicio en la consola web porque el proceso de actualización elimina y restaura el espacio de nombres **openshift-migration**:
 - a. Obtenga el token de la cuenta de servicio:


```
$ oc sa get-token migration-controller -n openshift-migration
```
 - b. En la consola web de MTC, haga clic en **Clusters** (Clústeres).
 - c. Haga clic en el menú Options (Opciones)  junto al clúster y seleccione **Edit** (Editar).
 - d. Introduzca el nuevo token de cuenta de servicio en el campo **Service account token** (Token de cuenta de servicio).
 - e. Haga clic en **Update cluster** (Actualizar clúster) y, luego, en **Close** (Cerrar).
10. Compruebe que los pods de MTC estén en funcionamiento:

```
$ oc get pods -n openshift-migration
```

8.3. ACTUALIZACIÓN DE MTC 1.3 A 1.7

Si está actualizando la versión 1.3.x de Migration Toolkit for Containers (MTC) a la 1.7, debe actualizar el manifiesto de recursos personalizados (CR) **de MigPlan** en el clúster en el que se ejecuta el pod de **MigrationController**.

Dado que los parámetros **indirectImageMigration** e **indirectVolumeMigration** no existen en MTC 1.3, su valor por defecto en la versión 1.4 es **false** (falso), lo que significa que la migración directa de imágenes y la migración directa de volúmenes están activadas. Debido a que los requisitos de migración directa no se cumplen, el plan de migración no puede alcanzar el estado **Ready** (Listo) a menos que los valores de estos parámetros se cambien a **true** (verdadero).

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster**.

Procedimiento

1. Inicie sesión en el clúster en el que se ejecuta el pod de **MigrationController**.
2. Obtenga el manifiesto de CR de **MigPlan**:

```
$ oc get migplan <migplan> -o yaml -n openshift-migration
```

3. Actualice los valores de los siguientes parámetros y guarde el archivo como **migplan.yaml**:

```
...
spec:
  indirectImageMigration: true
  indirectVolumeMigration: true
```

4. Reemplace el manifiesto del CR de **MigPlan** para aplicar los cambios:

```
$ oc replace -f migplan.yaml -n openshift-migration
```

5. Obtenga el manifiesto actualizado del CR de **MigPlan** para verificar los cambios:

```
$ oc get migplan <migplan> -o yaml -n openshift-migration
```

CAPÍTULO 9. LISTAS DE CONTROL PREVIAS A LA MIGRACIÓN

Antes de migrar las cargas de trabajo de sus aplicaciones con Migration Toolkit for Containers (MTC), revise las siguientes listas de comprobación.

9.1. RECURSOS

- Si su aplicación utiliza una red de servicios interna o una ruta externa para comunicarse con los servicios, la ruta correspondiente existe.
- Si su aplicación utiliza recursos a nivel del clúster, los ha vuelto a crear en el clúster de destino.
- Ha [excluido](#) los volúmenes persistentes (PV), los flujos de imágenes y otros recursos que no desea migrar.
- Ha hecho una copia de seguridad de los datos de FV por si una aplicación muestra un comportamiento inesperado tras la migración y corrompe los datos.

9.2. CLÚSTER DE ORIGEN

- El clúster cumple los [requisitos mínimos de hardware](#) .
- Ha instalado la versión correcta del operador de Migration Toolkit for Containers:
 - **operator-3.7.yml** en la versión 3.7 de OpenShift Container Platform.
 - **operator.yml** en las versiones 3.9 a 4.5 de OpenShift Container Platform.
- La versión de MTC es la misma en todos los clústeres.
- Todos los nodos tienen una suscripción activa en OpenShift Container Platform.
- Ha realizado todas las [tareas de ejecución](#) .
- Ha realizado todas las [comprobaciones de estado del entorno](#) .
- Ha comprobado la existencia de PV con configuraciones anormales atascadas en el estado **Terminating** (Terminación) ejecutando el siguiente comando:

```
$ oc get pv
```

- Ha comprobado la existencia de pods cuyo estado es distinto a **Running** (Ejecutándose) o **Completed** (Completado) ejecutando el siguiente comando:

```
$ oc get pods --all-namespaces | egrep -v 'Running | Completed'
```

- Ha comprobado la existencia de pods con un elevado número de reinicios ejecutando el siguiente comando:

```
$ oc get pods --all-namespaces --field-selector=status.phase=Running \
-o json | jq '.items[]|select(any( .status.containerStatuses[]; \
.restartCount > 3))|.metadata.name'
```

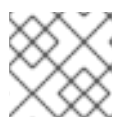
Incluso si los pods están en el estado **Running** (Ejecutándose), un alto número de reinicios podría indicar problemas subyacentes.

- Ha eliminado las construcciones, implementaciones e imágenes antiguas de cada espacio de nombres que se va a migrar mediante [pruning](#) (recorte).
- El registro interno utiliza un [tipo de almacenamiento compatible](#).
- Solo migración directa de imágenes: el registro interno está [expuesto](#) al tráfico externo.
- Puede leer y escribir imágenes en el registro.
- [etcd cluster](#) es correcto.
- El [tiempo medio de respuesta del servidor API](#) en el clúster de origen es inferior a 50 ms.
- Los certificados del clúster son [válidos](#) mientras dure el proceso de migración.
- Ha comprobado si hay solicitudes de firma de certificados pendientes ejecutando el siguiente comando:


```
$ oc get csr -A | grep pending -i
```
- El [proveedor de identidad](#) está en funcionamiento.

9.3. CLÚSTER DE DESTINO

- Ha instalado la versión 1.5.1 del operador de Migration Toolkit for Containers.
- Se cumplen todos los [requisitos previos de MTC](#).
- El clúster cumple los requisitos mínimos de hardware para la plataforma y el método de instalación específicos, por ejemplo, en [bare metal](#).
- El clúster tiene [clases de almacenamiento](#) definidas para los tipos de almacenamiento utilizados por el clúster de origen, por ejemplo, volumen de bloques, sistema de archivos o almacenamiento de objetos.



NOTA

NFS no requiere una clase de almacenamiento definida.

- El clúster tiene la configuración de red y los permisos correctos para acceder a los servicios externos, por ejemplo, bases de datos, repositorios de código fuente, registros de imágenes de contenedores y herramientas de CI/CD.
- Las aplicaciones y los servicios externos que utilizan los servicios proporcionados por el clúster tienen la configuración de red y los permisos correctos para acceder al clúster.
- Se cumplen las dependencias internas de la imagen del contenedor.
Si una aplicación utiliza una imagen interna en el espacio de nombres **openshift** que no es compatible con OpenShift Container Platform 4.10, puede actualizar manualmente la [etiqueta de flujo de imágenes de OpenShift Container Platform 3](#) con **podman**.

- El clúster de destino y el repositorio de replicación tienen suficiente espacio de almacenamiento.
- El [proveedor de identidad](#) está en funcionamiento.
- Existen registros del DNS para su aplicación en el clúster de destino.
- Establezca el valor del parámetro **annotation.openshift.io/host.generated** en **true** (verdadero) para que cada ruta de OpenShift Container Platform actualice su nombre de host para el clúster de destino. En caso contrario, las rutas migradas conservan el nombre del clúster de origen.
- Los certificados que utiliza su aplicación existen en el clúster de destino.
- Ha configurado las reglas de firewall adecuadas en el clúster de destino.
- Ha configurado correctamente el equilibrio de carga en el clúster de destino.
- Si migra objetos a un espacio de nombres existente en el clúster de destino que tiene el mismo nombre que el espacio de nombres que se migra desde el origen, el espacio de nombres de destino no contiene objetos del mismo nombre y tipo que los objetos que se migran.

**NOTA**

No cree espacios de nombres para su aplicación en el clúster de destino antes de la migración, ya que esto podría hacer que los cupos cambien.

9.4. RENDIMIENTO

- La red de migración tiene un rendimiento mínimo de 10 Gbps.
- Los clústeres tienen recursos suficientes para la migración.

**NOTA**

Los clústeres requieren memoria, CPU y almacenamiento adicionales para ejecutar una migración además de las cargas de trabajo normales. Los requisitos reales de los recursos dependen del número de recursos de Kubernetes que se migren en un único plan de migración. Debe probar las migraciones en un entorno que no sea de producción para estimar las necesidades de los recursos.

- La [memoria y el uso de la CPU](#) de los nodos son correctos.
- Se ha comprobado el [rendimiento del disco etcd](#) de los clústeres con **fiio**.

CAPÍTULO 10. MIGRACIÓN DE LAS APLICACIONES

Puede migrar sus aplicaciones utilizando la consola web de Migration Toolkit for Containers (MTC) o desde la [línea de comandos](#).

Puede utilizar la migración por etapas y la migración de transición para migrar una aplicación entre clústeres:

- La migración por etapas copia los datos del clúster de origen en el de destino sin detener la aplicación. Puede ejecutar una migración por etapas varias veces para reducir la duración de la migración de transición.
- La migración de transición detiene las transacciones en el clúster de origen y mueve los recursos al clúster de destino.

Puede utilizar la migración de estado para migrar el estado de una aplicación:

- La migración por etapas copia las reclamaciones de volúmenes persistentes (PVC) seleccionadas y los recursos de Kubernetes.
- Puede utilizar la migración por etapas para migrar un espacio de nombres dentro del mismo clúster.

La mayoría de los recursos de los clústeres aún no son gestionados por MTC. Si sus aplicaciones requieren recursos en el clúster, es posible que tenga que crearlos manualmente en el clúster de destino.

Durante la migración, MTC conserva las siguientes anotaciones del espacio de nombres:

- **openshift.io/sa.scc.mcs**
- **openshift.io/sa.scc.supplemental-groups**
- **openshift.io/sa.scc.uid-range**

Estas anotaciones conservan el rango de UID, asegurando que los contenedores conserven sus permisos del sistema de archivos en el clúster de destino. Existe el riesgo de que los UID migrados puedan duplicar los UID dentro de un espacio de nombres existente o futuro en el clúster de destino.

10.1. REQUISITOS PREVIOS A LA MIGRACIÓN

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.

Migración directa de imágenes

- Debe asegurarse de que el registro interno seguro del clúster de origen esté expuesto.
- Debe crear una ruta hacia el registro expuesto.

Migración directa de volúmenes

- Si sus clústeres utilizan proxies, debe configurar el proxy TCP Stunnel.

Imágenes internas

- Si su aplicación utiliza imágenes internas del espacio de nombres de **openshift**, debe asegurarse de que las versiones necesarias de las imágenes estén presentes en el clúster de destino. Puede actualizar manualmente una etiqueta de flujo de imágenes para utilizar una imagen obsoleta de OpenShift Container Platform 3 en un clúster de OpenShift Container Platform 4.10.

Clústeres

- El clúster de origen debe actualizarse a la última versión de MTC z-stream.
- La versión de MTC debe ser la misma en todos los clústeres.

Red

- Los clústeres tienen acceso a la red sin restricciones entre sí y al repositorio de replicación.
- Si se copian los volúmenes persistentes con **move**, los clústeres deben tener un acceso de red sin restricciones a los volúmenes remotos.
- Debe habilitar los siguientes puertos en un clúster de OpenShift Container Platform 3:
 - **8443** (servidor API)
 - **443** (rutas)
 - **53** (DNS)
- Debe habilitar los siguientes puertos en un clúster de OpenShift Container Platform 4:
 - **6443** (servidor API)
 - **443** (rutas)
 - **53** (DNS)
- Debe habilitar el puerto **443** en el repositorio de replicación si está utilizando TLS.

Volúmenes persistentes (PV)

- Los PV deben ser válidos.
- Los PV deben estar vinculados a reclamaciones de volúmenes persistentes.
- Si utiliza instantáneas para copiar los PV, se aplican los siguientes requisitos adicionales:
 - El proveedor de la nube debe admitir las instantáneas.
 - Los PV deben tener el mismo proveedor de nube.
 - Los FV deben estar situados en la misma región geográfica.
 - Los PV deben tener la misma clase de almacenamiento.

Recursos adicionales para los requisitos previos a la migración

- [Exposición manual de un registro seguro para OpenShift Container Platform 3](#)

- [Actualización de imágenes internas obsoletas](#)

10.2. MIGRACIÓN DE LAS APLICACIONES MEDIANTE LA CONSOLA WEB DE MTC

Puede configurar clústeres y un repositorio de replicación utilizando la consola web de MTC. A continuación, puede crear y ejecutar un plan de migración.

10.2.1. Inicio de la consola web de MTC

Puede iniciar la consola web de Migration Toolkit for Containers (MTC) en un navegador.

Requisitos previos

- La consola web de MTC debe tener acceso de red a la consola web de OpenShift Container Platform.
- La consola web de MTC debe tener acceso a la red del servidor de autorización OAuth.

Procedimiento

1. Inicie sesión en el clúster de OpenShift Container Platform en el que ha instalado MTC.
2. Obtenga la URL de la consola web de MTC introduciendo el siguiente comando:

```
$ oc get -n openshift-migration route/migration -o go-template='https://{ .spec.host }'
```

El resultado se parece al siguiente: **https://migration-openshift-migration.apps.cluster.openshift.com**.

3. Inicie un navegador y navegue hasta la consola web de MTC.



NOTA

Si intenta acceder a la consola web de MTC inmediatamente después de instalar el operador de Migration Toolkit for Containers, es posible que la consola no se cargue porque el operador todavía está configurando el clúster. Espere unos minutos y vuelva a intentarlo.

4. Si utiliza certificados de CA autofirmados, se le pedirá que acepte el certificado de CA del servidor API del clúster de origen. La página web lo guiará a través del proceso de aceptación de los certificados restantes.
5. Inicie sesión con su **nombre de usuario** y **contraseña** de OpenShift Container Platform.

10.2.2. Adición de un clúster a la consola web de MTC

Puede añadir un clúster a la consola web de Migration Toolkit for Containers (MTC).

Requisitos previos

- Si utiliza las instantáneas de Azure para copiar datos:

- Para OpenShift Container Platform 4:

```
$ oc create route passthrough --service=image-registry --port=5000 -n openshift-image-registry
```

- **Clúster de Azure:** debe seleccionar esta opción si utiliza las instantáneas de Azure para copiar sus datos.
 - **Grupo de recursos de Azure:** este campo se muestra si se selecciona el **clúster de Azure**. Especifique el grupo de recursos de Azure.
 - **Requiere verificación de SSL:** (opcional) seleccione esta opción para verificar las conexiones SSL al clúster.
 - **Archivo de paquete de CA:** este campo se muestra si se selecciona **Require SSL verification** (Requiere verificación de SSL). Si ha creado un archivo de paquete de certificados de CA personalizado para los certificados autofirmados, haga clic en **Browse** (Examinar), seleccione el archivo de paquete de CA y cárguelo.
6. Haga clic en **Add cluster** (Añadir clúster).
El clúster aparece en la lista **Clusters** (Clústeres).

10.2.3. Adición de un repositorio de replicación a la consola web de MTC

Puede añadir un almacenamiento de objetos como repositorio de replicación a la consola web de Migration Toolkit for Containers (MTC).

MTC es compatible con los siguientes proveedores de almacenamiento:

- Amazon Web Services (AWS) S3
- Multi-Cloud Object Gateway (MCG)
- Almacenamiento de objetos genérico S3, por ejemplo, Minio o Ceph S3
- Google Cloud Provider (GCP)
- Microsoft Azure Blob

Requisitos previos

- Debe configurar el almacenamiento de objetos como repositorio de replicación.

Procedimiento

1. En la consola web de MTC, haga clic en **Replication repositories** (Repositorios de replicación).
2. Haga clic en **Add repository** (Añadir repositorio).
3. Seleccione un **tipo de proveedor de almacenamiento** y complete los siguientes campos:
 - **AWS** para proveedores de S3, incluidos AWS y MCG:
 - **Nombre del repositorio de replicación:** especifique el nombre del repositorio de replicación en la consola web de MTC.

- **Nombre del bucket de S3:** especifique el nombre del bucket de S3.
 - **Región del bucket de S3:** especifique la región del bucket de S3. Es **obligatorio** para AWS S3. Es **opcional** para algunos proveedores de S3. Consulte la documentación del producto de su proveedor de S3 para conocer los valores esperados.
 - **Terminal de S3:** especifique la URL del servicio S3, no del bucket, por ejemplo, **https://<s3-storage.apps.cluster.com>**. Es **obligatorio** para un proveedor genérico de S3. Debe utilizar el prefijo **https://**.
 - **Clave de acceso del proveedor de S3** especifique **<AWS_SECRET_ACCESS_KEY>** para AWS o la clave de acceso del proveedor de S3 para MCG y otros proveedores de S3.
 - **Clave de acceso secreta del proveedor de S3** especifique **<AWS_ACCESS_KEY_ID>** para AWS o la clave de acceso secreta del proveedor de S3 para MCG y otros proveedores de S3.
 - **Requiere verificación de SSL:** desactive esta casilla si utiliza un proveedor genérico de S3.
 - Si ha creado un paquete de certificados de CA personalizado para los certificados autofirmados, haga clic en **Browse** (Examinar) y busque el archivo codificado en Base64.
- **GCP:**
 - **Nombre del repositorio de replicación:** especifique el nombre del repositorio de replicación en la consola web de MTC.
 - **Nombre del bucket de GCP:** especifique el nombre del bucket de GCP.
 - **Credencial de GCP de JSON Blob** especifique la cadena en el archivo **credentials-velero**.
- **Azure:**
 - **Nombre del repositorio de replicación:** especifique el nombre del repositorio de replicación en la consola web de MTC.
 - **Grupo de recursos de Azure** especifique el grupo de recursos de almacenamiento de Azure Blob.
 - **Nombre de la cuenta de almacenamiento de Azure** especifique el nombre de la cuenta de almacenamiento de Azure Blob.
 - **Credenciales de Azure (contenido del archivo INI):** especifique la cadena en el archivo **credentials-velero**.
4. Haga clic en **Add repository** (Añadir repositorio) y espere la validación de la conexión.
 5. Haga clic en **Close** (Cerrar).
El nuevo repositorio aparece en la lista de **repositorios de replicación**.

10.2.4. Creación de un plan de migración en la consola web de MTC

Puede crear un plan de migración en la consola web de Migration Toolkit for Containers (MTC).

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.
- Debe asegurarse de que se instala la misma versión de MTC en todos los clústeres.
- Debe añadir los clústeres y el repositorio de replicación a la consola web de MTC.
- Si desea utilizar el método de copia de datos *en movimiento* para migrar un volumen persistente (PV), los clústeres de origen y destino deben tener un acceso de red ininterrumpido al volumen remoto.
- Si desea utilizar la migración directa de imágenes, debe especificar la ruta expuesta al registro de imágenes del clúster de origen. Esto puede hacerse utilizando la consola web de MTC o actualizando el manifiesto de recursos personalizado de **MigCluster**.

Procedimiento

1. En la consola web de MTC, haga clic en **Migration plans** (Planes de migración).
2. Haga clic en **Add migration plan** (Añadir plan de migración).
3. Introduzca el **nombre del plan**.
El nombre del plan de migración no debe superar los 253 caracteres alfanuméricos en minúsculas (**a-z, 0-9**) y no debe contener espacios ni guiones bajos (**_**).
4. Seleccione un **clúster de origen**, un **clúster de destino** y un **repositorio**.
5. Haga clic en **Next** (Siguiendo).
6. Seleccione los proyectos para la migración.
7. Opcional: haga clic en el icono de edición junto a un proyecto para cambiar el espacio de nombres de destino.
8. Haga clic en **Next** (Siguiendo).
9. Seleccione un **tipo de migración** para cada PV:
 - La opción **Copy** (Copiar) copia los datos del PV de un clúster de origen en el repositorio de replicación y, luego, restaura los datos en un PV recién creado con características similares en el clúster de destino.
 - La opción **Move** (Mover) desmonta un volumen remoto, por ejemplo, NFS, del clúster de origen, crea un recurso de PV en el clúster de destino que apunta al volumen remoto y, a continuación, monta el volumen remoto en el clúster de destino. Las aplicaciones que se ejecutan en el clúster de destino utilizan el mismo volumen remoto que utilizaba el clúster de origen.
10. Haga clic en **Next** (Siguiendo).
11. Seleccione un **método de copia** para cada PV:
 - **Lacopia de instantáneas** hace una copia de seguridad y restaura los datos utilizando la funcionalidad de instantáneas del proveedor de la nube. Es significativamente más rápido que la **copia de sistemas de archivos**

- La **copia de sistemas de archivos** hace una copia de seguridad de los archivos en el clúster de origen y los restaura en el clúster de destino.
El método de copia de sistemas de archivos es necesario para la migración directa de volúmenes.
12. Puede seleccionar **Verify copy** (Verificar copia) para verificar los datos migrados con la **copia de sistemas de archivos**. Los datos se verifican generando una suma de comprobación para cada archivo de origen y verificando la suma de comprobación después de la restauración. La verificación de los datos reduce considerablemente el rendimiento.
 13. Seleccione una **clase de almacenamiento de destino**.
Si ha seleccionado **Copia de sistemas de archivos** puede cambiar la clase de almacenamiento de destino.
 14. Haga clic en **Next** (Siguiente).
 15. En la página de **opciones de migración**, se selecciona la opción de **migración de imagen directa** si se ha especificado una ruta de registro de imagen expuesta para el clúster de origen. La opción de **migración directa de PV** se selecciona si está migrando datos con la **copia de sistemas de archivos**.
Las opciones de migración directa copian imágenes y archivos directamente del clúster de origen al de destino. Esta opción es mucho más rápida que copiar imágenes y archivos del cluster de origen al repositorio de replicación y, luego, del repositorio de replicación al clúster de destino.
 16. Haga clic en **Next** (Siguiente).
 17. Opcional: haga clic en **Add Hook** (Añadir enlace) para añadir un enlace al plan de migración. Un enlace ejecuta un código personalizado. Puede añadir hasta cuatro enlaces a un mismo plan de migración. Cada enlace se ejecuta durante un paso de migración diferente.
 - a. Introduzca el nombre del enlace que se mostrará en la consola web.
 - b. Si el enlace es una estrategia de Ansible, seleccione **Ansible playbook** (Estrategia de Ansible) y haga clic en **Browse** (Examinar) para cargar la estrategia o pegue el contenido de la estrategia en el campo.
 - c. Opcional: especifique una imagen de tiempo de ejecución de Ansible si no está utilizando la imagen de enlace predeterminada.
 - d. Si el enlace no es una estrategia de Ansible, seleccione **Custom container image** (Imagen de contenedor personalizada) y especifique el nombre de la imagen y la ruta.
Una imagen de contenedor personalizada puede incluir estrategias de Ansible.
 - e. Seleccione el **clúster de origen** o el **clúster de destino**.
 - f. Introduzca el **nombre de la cuenta de servicio** y el **espacio de nombres de la cuenta de servicio**.
 - g. Seleccione el paso de migración para el enlace:
 - **Antes de la copia de seguridad** antes de realizar la copia de seguridad de la carga de trabajo de la aplicación en el clúster de origen
 - **Después de la copia de seguridad** después de que la carga de trabajo de la aplicación se respalde en el clúster de origen

- **Antes de la restauración:** antes de restaurar la carga de trabajo de la aplicación en el clúster de destino
- **Después de la restauración:** después de restaurar la carga de trabajo de la aplicación en el clúster de destino

h. Haga clic en **Add** (Añadir).

18. Haga clic en **Finish** (Finalizar).

El plan de migración se muestra en la lista de **planes de migración**.

Recursos adicionales

- [Método de copia de sistemas de archivos de MTC](#)
- [Método de copia de instantáneas de MTC](#)

10.2.5. Ejecución de un plan de migración en la consola web de MTC

Puede migrar aplicaciones y datos con el plan de migración que creó en la consola web de Migration Toolkit for Containers (MTC).



NOTA

Durante la migración, MTC establece la política de recuperación de los volúmenes persistentes (PV) migrados como **Retain** en el clúster de destino.

El recurso personalizado **Backup** contiene una anotación **PVOriginalReclaimPolicy** que indica la política de recuperación original. Puede restaurar manualmente la política de recuperación de los PV migrados.


Requisitos previos

La consola web de MTC debe contener lo siguiente:

- Clúster de origen con el estado **Ready** (Listo)
- Clúster de destino con el estado **Ready** (Listo)
- Repositorio de replicación
- Plan de migración válido

Procedimiento

1. Inicie sesión en la consola web de MTC y haga clic en **Migration plans** (Planes de migración).

2. Haga clic en el menú Options (Opciones)  junto al plan de migración y seleccione una de las siguientes opciones en **Migration** (Migración):

- La **migración por etapas** copia los datos del clúster de origen en el clúster de destino sin detener la aplicación.
- La **migración por transición** detiene las transacciones en el clúster de origen y mueve los recursos al clúster de destino.

Opcional: en el cuadro de diálogo **Cutover migration** (Migración por transición) puede desactivar la casilla de verificación **Halt transactions on the source cluster during migration** (Detener transacciones en el clúster de origen durante la migración).

- La **migración por etapas** copia las reclamaciones de volúmenes persistentes (PVC) seleccionadas y los recursos de Kubernetes. Puede utilizar la migración por etapas para migrar un espacio de nombres dentro del mismo clúster.



IMPORTANTE

No utilice la migración por etapas para migrar un espacio de nombres entre clústeres. Utilice en su lugar la migración por etapas o transición.

- Seleccione una o más PVC en el cuadro de diálogo **State migration** (Migración de estado) y haga clic en **Migrate** (Migrar).
3. Una vez finalizada la migración, compruebe que la aplicación ha migrado correctamente en la consola web de OpenShift Container Platform:
 - a. Haga clic en **Home** (Inicio) → **Projects** (Proyectos).
 - b. Haga clic en el proyecto migrado para ver su estado.
 - c. En la sección **Routes** (Rutas), haga clic en **Location** (Ubicación) para verificar que la aplicación esté funcionando, si es el caso.
 - d. Haga clic en **Workloads** (Cargas de trabajo) → **Pods** para comprobar que los pods se ejecutan en el espacio de nombres migrado.
 - e. Haga clic en **Storage** (Almacenamiento) → **Persistent volumes** (Volúmenes persistentes) para verificar que los volúmenes persistentes migrados estén correctamente provisionados.

CAPÍTULO 11. OPCIONES DE MIGRACIÓN AVANZADAS

Puede automatizar sus migraciones y modificar los recursos personalizados **MigPlan** y **MigrationController** para realizar migraciones a gran escala y mejorar el rendimiento.

11.1. TERMINOLOGÍA

Tabla 11.1. Terminología de MTC

Término	Definición
Clúster de origen	Clúster desde el que se migran las aplicaciones.
Cluster de destino ^[1]	Clúster al que se migran las aplicaciones.
Repositorio de replicación	Almacenamiento de objetos utilizado para copiar imágenes, volúmenes y objetos de Kubernetes durante la migración indirecta o para objetos de Kubernetes durante la migración directa de volúmenes o la migración directa de imágenes. El repositorio de replicación debe ser accesible para todos los clústeres.
Clúster del host	Clúster en el que se ejecuta el pod migration-controller y la consola web. El clúster del host suele ser el clúster de destino, pero no es necesario. El clúster del host no requiere una ruta de registro expuesta para la migración directa de imágenes.
Clúster remoto	El clúster remoto suele ser el clúster de origen, pero no es necesario. El clúster remoto requiere un recurso personalizado secreto que contenga el token de la cuenta de servicio del controlador de migración . El clúster remoto requiere una ruta de registro segura expuesta para la migración directa de imágenes.
Migración indirecta	Las imágenes, los volúmenes y los objetos de Kubernetes se copian del clúster de origen al repositorio de replicación y, luego, del repositorio de replicación al clúster de destino.
Migración directa de volúmenes	Los volúmenes persistentes se copian directamente del clúster de origen al de destino.
Migración directa de imágenes	Las imágenes se copian directamente del clúster de origen al de destino.
Migración por etapas	Los datos se copian en el clúster de destino sin detener la aplicación. Ejecutar una migración por etapas varias veces reduce la duración de la migración por transición.

Término	Definición
Migración por transición	La aplicación se detiene en el clúster de origen y sus recursos se migran al clúster de destino.
Migración de estado	El estado de la aplicación se migra copiando reclamaciones de volúmenes persistentes específicas y objetos de Kubernetes al clúster de destino.
Migración de retroceso	La migración de retroceso retrotrae toda la migración.

¹ Llame al clúster de *destino* en la consola web de MTC.

11.2. MIGRACIÓN DE UNA APLICACIÓN DESDE LAS INSTALACIONES A UN CLÚSTER BASADO EN LA NUBE

Puede migrar desde un clúster de origen que esté detrás de un cortafuegos a un clúster de destino basado en la nube estableciendo un túnel de red entre los dos clústeres. El comando **crane tunnel-api** establece un túnel de este tipo creando un túnel VPN en el clúster de origen y luego conectándose a un servidor VPN que se ejecuta en el clúster de destino. El servidor VPN se expone al cliente utilizando una dirección de balanceo de carga en el cluster de destino.

Un servicio creado en el clúster de destino expone la API del clúster de origen a MTC, que se ejecuta en el clúster de destino.

Requisitos previos

- El sistema que crea el túnel VPN debe tener acceso y estar conectado a ambos clusters.
- Debe ser posible crear un equilibrador de carga en el clúster de destino. Consulte a su proveedor de la nube para asegurarse de que esto es posible.
- Tenga nombres preparados para asignar a los espacios de nombres, tanto en el clúster de origen como en el de destino, en los que ejecutar el túnel VPN. Estos espacios de nombres no deben ser creados de antemano. Para obtener información sobre las reglas del espacio de nombres, consulte <https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#dns-subdomain-names>.
- Cuando se conectan varios clústeres de origen protegidos por cortafuegos al clúster de la nube, cada clúster de origen requiere su propio espacio de nombres.
- El servidor OpenVPN está instalado en el clúster de destino.
- El cliente OpenVPN está instalado en el clúster de origen.
- Al configurar el clúster de origen en MTC, la URL de la API toma la forma de **https://proxied-cluster.<namespace>.svc.cluster.local:8443**.
 - Si utiliza la API, consulte *Crear un manifiesto MigCluster CR para cada clúster remoto* .
 - Si utiliza la consola web MTC, consulte *Migración de sus aplicaciones mediante la consola web MTC*.

- La consola web MTC y el controlador de migración deben estar instalados en el clúster de destino.

Procedimiento

1. Instala la utilidad de **la grúa**:

```
$ podman cp $(podman create registry.redhat.io/rhmtc/openshift-migration-controller-rhel8:v1.7.0):/crane ./
```

2. Inicie sesión de forma remota en un nodo del clúster de origen y en un nodo del clúster de destino.
3. Obtenga el contexto del clúster para ambos clústeres después de iniciar la sesión:

```
$ oc config view
```

4. Establezca un túnel introduciendo el siguiente comando en el sistema de comandos:

```
$ crane tunnel-api [--namespace <namespace>] \  
  --destination-context <destination-cluster> \  
  --source-context <source-cluster>
```

Si no se especifica un espacio de nombres, el comando utiliza el valor por defecto **openvpn**.

Por ejemplo:

```
$ crane tunnel-api --namespace my_tunnel \  
  --destination-context openshift-migration/c131-e-us-east-containers-cloud-ibm-com/admin \  
  --source-context default/192-168-122-171-nip-io:8443/admin
```

SUGERENCIA

Consulte todos los parámetros disponibles para el comando **crane tunnel-api** introduciendo **crane tunnel-api --help**.

El comando genera certificados TSL/SSL. Este proceso puede durar varios minutos. Cuando el proceso finaliza, aparece un mensaje.

El servidor OpenVPN se inicia en el clúster de destino y el cliente OpenVPN se inicia en el clúster de origen.

Después de unos minutos, el equilibrador de carga se resuelve en el nodo de origen.

SUGERENCIA

Puede ver el registro de los pods de OpenVPN para comprobar el estado de este proceso introduciendo los siguientes comandos con privilegios de root:

```
# oc get po -n <namespace>
```

Ejemplo de salida

```
NAME          READY   STATUS    RESTARTS   AGE
<pod_name>    2/2    Running   0           44s
```

```
# oc logs -f -n <namespace> <pod_name> -c openvpn
```

Cuando se resuelve la dirección del equilibrador de carga, aparece el mensaje **Secuencia de inicialización completada** al final del registro.

5. En el servidor OpenVPN, que se encuentra en un nodo de control de destino, verifique que el servicio **openvpn** y el servicio **proxied-cluster** se están ejecutando:

```
$ oc get service -n <namespace>
```

6. En el nodo de origen, obtenga el token de la cuenta de servicio (SA) para el controlador de migración:

```
# oc sa get-token -n openshift-migration migration-controller
```

7. Abra la consola web de MTC y añada el clúster de origen, utilizando los siguientes valores:

- **Nombre del clúster:** El nombre del clúster de origen.
- **URL:** **proxied-cluster.<namespace>.svc.cluster.local:8443**. Si no ha definido un valor para **<namespace>**, utilice **openvpn**.
- **Token de la cuenta de servicio:** El token de la cuenta de servicio del controlador de migración.
- **Host de ruta expuesto al registro de imágenes proxied-cluster.** **<namespace>.svc.cluster.local:5000**. Si no ha definido un valor para **<namespace>**, utilice **openvpn**.

Después de que MTC haya validado la conexión con éxito, puede proceder a crear y ejecutar un plan de migración. El espacio de nombres del clúster de origen debería aparecer en la lista de espacios de nombres.

Recursos adicionales

- Para obtener información sobre la creación de un manifiesto MigCluster CR para cada clúster remoto, consulte [Migración de una aplicación mediante la API MTC](#).
- Para obtener información sobre cómo añadir un clúster mediante la consola web, consulte [Migración de sus aplicaciones mediante la consola web de MTC](#)

11.3. MIGRACIÓN DE APLICACIONES MEDIANTE LA LÍNEA DE COMANDOS

Puede migrar aplicaciones con la API de MTC utilizando la interfaz de línea de comandos (CLI) para automatizar la migración.

11.3.1. Requisitos previos a la migración

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.

Migración directa de imágenes

- Debe asegurarse de que el registro interno seguro del clúster de origen esté expuesto.
- Debe crear una ruta hacia el registro expuesto.

Migración directa de volúmenes

- Si sus clústeres utilizan proxies, debe configurar el proxy TCP Stunnel.

Imágenes internas

- Si su aplicación utiliza imágenes internas del espacio de nombres de **openshift**, debe asegurarse de que las versiones necesarias de las imágenes estén presentes en el clúster de destino. Puede actualizar manualmente una etiqueta de flujo de imágenes para utilizar una imagen obsoleta de OpenShift Container Platform 3 en un clúster de OpenShift Container Platform 4.10.

Clústeres

- El clúster de origen debe actualizarse a la última versión de MTC z-stream.
- La versión de MTC debe ser la misma en todos los clústeres.

Red

- Los clústeres tienen acceso a la red sin restricciones entre sí y al repositorio de replicación.
- Si se copian los volúmenes persistentes con **move**, los clústeres deben tener un acceso de red sin restricciones a los volúmenes remotos.
- Debe habilitar los siguientes puertos en un clúster de OpenShift Container Platform 3:
 - **8443** (servidor API)
 - **443** (rutas)
 - **53** (DNS)
- Debe habilitar los siguientes puertos en un clúster de OpenShift Container Platform 4:
 - **6443** (servidor API)
 - **443** (rutas)

- **53** (DNS)
- Debe habilitar el puerto **443** en el repositorio de replicación si está utilizando TLS.

Volúmenes persistentes (PV)

- Los PV deben ser válidos.
- Los PV deben estar vinculados a reclamaciones de volúmenes persistentes.
- Si utiliza instantáneas para copiar los PV, se aplican los siguientes requisitos adicionales:
 - El proveedor de la nube debe admitir las instantáneas.
 - Los PV deben tener el mismo proveedor de nube.
 - Los FV deben estar situados en la misma región geográfica.
 - Los PV deben tener la misma clase de almacenamiento.

11.3.2. Creación de una ruta de registro para la migración directa de imágenes

Para la migración directa de imágenes, debe crear una ruta al registro interno expuesto en todos los clústeres remotos.

Requisitos previos

- El registro interno debe estar expuesto al tráfico externo en todos los clústeres remotos. El registro de OpenShift Container Platform 4 está expuesto por defecto.

El registro de OpenShift Container Platform 3 debe [exponerse manualmente](#).

Procedimiento

- Para crear una ruta a un registro de OpenShift Container Platform 3, ejecute el siguiente comando:

```
$ oc create route passthrough --service=docker-registry -n default
```

- Para crear una ruta a un registro de OpenShift Container Platform 4, ejecute el siguiente comando:

```
$ oc create route passthrough --service=image-registry -n openshift-image-registry
```

11.3.3. Configuración de los proxies

Para OpenShift Container Platform 4.1 y versiones anteriores, debe configurar los proxies en el manifiesto de recursos personalizados (CR) de **MigrationController** después de instalar el operador de Migration Toolkit for Containers dado que estas versiones no admiten un objeto **proxy** para todo el clúster.

Para OpenShift Container Platform 4.2 a 4.10, Migration Toolkit for Containers (MTC) hereda la configuración del proxy de todo el clúster. Puede cambiar los parámetros del proxy si desea anular la configuración del proxy en todo el clúster.

Debe configurar los proxies para que permitan el protocolo SPDY y reenvíen el encabezado **HTTP de actualización** al servidor de la API. De lo contrario, se muestra un error de **solicitud de actualización requerida**. El CR **MigrationController** utiliza SPDY para ejecutar comandos dentro de los pods remotos. El encabezado **HTTP Upgrade** es necesario para abrir una conexión websocket con el servidor de la API.

Migración directa de volúmenes

Si realiza una migración directa de volúmenes (DVM) desde un clúster de origen detrás de un proxy, debe configurar el proxy Stunnel. Stunnel crea un túnel transparente entre los clústeres de origen y destino para la conexión TCP sin cambiar los certificados.

La DVM solo admite un proxy. El clúster de origen no puede acceder a la ruta del clúster de destino si éste también está detrás de un proxy.

Requisitos previos

- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster** en todos los clústeres.

Procedimiento

1. Obtenga el manifiesto del CR **MigrationController**:

```
$ oc get migrationcontroller <migration_controller> -n openshift-migration
```

2. Actualice los parámetros del proxy:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigrationController
metadata:
  name: <migration_controller>
  namespace: openshift-migration
...
spec:
  stunnel_tcp_proxy: http://<username>:<password>@<ip>:<port> 1
  httpProxy: http://<username>:<password>@<ip>:<port> 2
  httpsProxy: http://<username>:<password>@<ip>:<port> 3
  noProxy: example.com 4
```

- 1 URL del proxy Stunnel para la migración directa de volúmenes.
- 2 URL del proxy para crear conexiones HTTP fuera del clúster. El esquema de la URL debe ser **http**.
- 3 URL del proxy para crear conexiones HTTPS fuera del clúster. Si no se especifica, se utiliza **httpProxy** para las conexiones HTTP y HTTPS.
- 4 Lista separada por comas de nombres de dominio de destino, dominios, direcciones IP u otros CIDR de red para excluir el proxy.

Introduzca un dominio con **.** para que coincida solo con los subdominios. Por ejemplo, **.y.com** coincide con **x.y.com**, pero no con **y.com**. Utilice ***** para evitar el proxy en todos los destinos. Si escala a los trabajadores que no están incluidos en la red definida por el campo

`networking.machineNetwork[].cidr` de la configuración de instalación, debe añadirlos a esta lista para evitar problemas de conexión.

Este campo se ignora si no se establecen los campos `httpProxy` y `httpsProxy`.

3. Guarde el manifiesto como `migration-controller.yaml`.
4. Aplique el manifiesto actualizado:

```
$ oc replace -f migration-controller.yaml -n openshift-migration
```

11.3.4. Migración de una aplicación mediante la API de MTC

Puede migrar una aplicación desde la línea de comandos utilizando la API de Migration Toolkit for Containers (MTC).

Procedimiento

1. Cree un manifiesto de CR de **MigCluster** para el clúster del host:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigCluster
metadata:
  name: <host_cluster>
  namespace: openshift-migration
spec:
  isHostCluster: true
EOF
```

2. Cree un manifiesto de objetos **Secret** para cada clúster remoto:

```
$ cat << EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: <cluster_secret>
  namespace: openshift-config
type: Opaque
data:
  saToken: <sa_token> 1
EOF
```

- 1 Especifique el token de la cuenta de servicio (SA) **migration-controller** codificado en base64 del clúster remoto. Puede obtener el token ejecutando el siguiente comando:

```
$ oc sa get-token migration-controller -n openshift-migration | base64 -w 0
```

3. Cree un manifiesto de CR de **MigCluster** para cada clúster remoto:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigCluster
```

```

metadata:
  name: <remote_cluster> ❶
  namespace: openshift-migration
spec:
  exposedRegistryPath: <exposed_registry_route> ❷
  insecure: false ❸
  isHostCluster: false
  serviceAccountSecretRef:
    name: <remote_cluster_secret> ❹
    namespace: openshift-config
  url: <remote_cluster_url> ❺
EOF

```

- ❶ Especifique el CR de **clúster** del clúster remoto.
- ❷ Opcional: para la migración directa de imágenes, especifique la ruta de registro expuesta.
- ❸ La verificación de SSL se activa si es **falsa**. Los certificados de CA no son necesarios ni se comprueban si son **verdaderos**.
- ❹ Especifique el objeto **secreto** del clúster remoto.
- ❺ Especifique la URL del clúster remoto.

4. Compruebe que todos los clústeres tengan el estado **Ready** (Listo):

```
$ oc describe cluster <cluster>
```

5. Cree un manifiesto de objetos **Secret** para el repositorio de replicación:

```

$ cat << EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  namespace: openshift-config
  name: <migstorage_creds>
type: Opaque
data:
  aws-access-key-id: <key_id_base64> ❶
  aws-secret-access-key: <secret_key_base64> ❷
EOF

```

- ❶ Especifique el ID de la clave en formato base64.
- ❷ Especifique la clave secreta en formato base64.

Las credenciales de AWS están codificadas en base64 por defecto. Para otros proveedores de almacenamiento, debe codificar sus credenciales ejecutando el siguiente comando con cada clave:

```
$ echo -n "<key>" | base64 -w 0 ❶
```

- 1 Especifique el ID de la clave o la clave secreta. Ambas claves deben estar codificadas en base64.

6. Cree un manifiesto de CR de **MigStorage** para el repositorio de replicación:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigStorage
metadata:
  name: <migstorage>
  namespace: openshift-migration
spec:
  backupStorageConfig:
    awsBucketName: <bucket> 1
    credsSecretRef:
      name: <storage_secret> 2
      namespace: openshift-config
  backupStorageProvider: <storage_provider> 3
  volumeSnapshotConfig:
    credsSecretRef:
      name: <storage_secret> 4
      namespace: openshift-config
  volumeSnapshotProvider: <storage_provider> 5
EOF
```

- 1 Especifique el nombre del bucket.
- 2 Especifique el CR **Secrets** del almacenamiento de objetos. Debe asegurarse de que las credenciales almacenadas en el CR **Secrets** del almacenamiento de objetos sean correctas.
- 3 Especifique el proveedor de almacenamiento.
- 4 Opcional: si copia los datos mediante el uso de instantáneas, especifique el CR **Secrets** del almacenamiento de objetos. Debe asegurarse de que las credenciales almacenadas en el CR **Secrets** del almacenamiento de objetos sean correctas.
- 5 Opcional: si copia los datos mediante el uso de instantáneas, especifique el proveedor de almacenamiento.

7. Compruebe que el CR **MigStorage** tenga el estado **Ready** (Listo):

```
$ oc describe migstorage <migstorage>
```

8. Cree un manifiesto del CR **MigPlan**:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
  namespace: openshift-migration
spec:
```

```

destMigClusterRef:
  name: <host_cluster>
  namespace: openshift-migration
indirectImageMigration: true 1
indirectVolumeMigration: true 2
migStorageRef:
  name: <migstorage> 3
  namespace: openshift-migration
namespaces:
  - <source_namespace_1> 4
  - <source_namespace_2>
  - <source_namespace_3>:<destination_namespace> 5
srcMigClusterRef:
  name: <remote_cluster> 6
  namespace: openshift-migration
EOF

```

- 1 La migración directa de imágenes se activa si es **falsa**.
- 2 La migración directa de volúmenes se activa si es **falsa**.
- 3 Especifique el nombre de la instancia de CR **MigStorage**.
- 4 Especifique uno o más espacios de nombres de origen. Por defecto, el espacio de nombres de destino tiene el mismo nombre.
- 5 Especifique un espacio de nombres de destino si es diferente del espacio de nombres de origen.
- 6 Especifique el nombre de la instancia **MigCluster** del clúster de origen.

9. Compruebe que la instancia **MigPlan** se encuentre en el estado **Ready** (Listo):

```
$ oc describe migplan <migplan> -n openshift-migration
```

10. Cree un manifiesto de CR de **MigMigration** para iniciar la migración definida en la instancia **MigPlan**:

```

$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigMigration
metadata:
  name: <migmigration>
  namespace: openshift-migration
spec:
  migPlanRef:
    name: <migplan> 1
    namespace: openshift-migration
  quiescePods: true 2
  stage: false 3
  rollback: false 4
EOF

```

- 1 Especifique el nombre del CR **MigPlan**.
- 2 Los pods del clúster de origen se detienen antes de la migración si son **verdaderos**.
- 3 Se realiza una migración por etapas que copia la mayor parte de los datos sin detener la aplicación si es **verdadera**.
- 4 Una migración completada se revierte si es **verdadera**.

11. Verifique la migración observando el progreso del CR **MigMigration**:

```
$ oc watch migmigration <migmigration> -n openshift-migration
```

El resultado se parece a lo siguiente:

Ejemplo de salida

```
Name:      c8b034c0-6567-11eb-9a4f-0bc004db0fbc
Namespace: openshift-migration
Labels:    migration.openshift.io/migplan-name=django
Annotations: openshift.io/touch: e99f9083-6567-11eb-8420-0a580a81020c
API Version: migration.openshift.io/v1alpha1
Kind:      MigMigration
...
Spec:
  Mig Plan Ref:
    Name:      migplan
    Namespace: openshift-migration
    Stage:     false
Status:
  Conditions:
    Category:      Advisory
    Last Transition Time: 2021-02-02T15:04:09Z
    Message:       Step: 19/47
    Reason:        InitialBackupCreated
    Status:        True
    Type:          Running
    Category:      Required
    Last Transition Time: 2021-02-02T15:03:19Z
    Message:       The migration is ready.
    Status:        True
    Type:          Ready
    Category:      Required
    Durable:       true
    Last Transition Time: 2021-02-02T15:04:05Z
    Message:       The migration registries are healthy.
    Status:        True
    Type:          RegistriesHealthy
  Itinerary:      Final
Observed Digest:
7fae9d21f15979c71ddc7dd075cb97061895caac5b936d92fae967019ab616d5
Phase:           InitialBackupCreated
Pipeline:
  Completed: 2021-02-02T15:04:07Z
  Message:    Completed
```

```

Name:    Prepare
Started: 2021-02-02T15:03:18Z
Message: Waiting for initial Velero backup to complete.
Name:    Backup
Phase:   InitialBackupCreated
Progress:
  Backup openshift-migration/c8b034c0-6567-11eb-9a4f-0bc004db0fbc-wpc44: 0 out of
estimated total of 0 objects backed up (5s)
  Started: 2021-02-02T15:04:07Z
  Message: Not started
  Name:    StageBackup
  Message: Not started
  Name:    StageRestore
  Message: Not started
  Name:    DirectImage
  Message: Not started
  Name:    DirectVolume
  Message: Not started
  Name:    Restore
  Message: Not started
  Name:    Cleanup
Start Timestamp: 2021-02-02T15:03:18Z
Events:
Type Reason Age      From      Message
---- -
Normal Running 57s      migmigration_controller Step: 2/47
Normal Running 57s      migmigration_controller Step: 3/47
Normal Running 57s (x3 over 57s) migmigration_controller Step: 4/47
Normal Running 54s      migmigration_controller Step: 5/47
Normal Running 54s      migmigration_controller Step: 6/47
Normal Running 52s (x2 over 53s) migmigration_controller Step: 7/47
Normal Running 51s (x2 over 51s) migmigration_controller Step: 8/47
Normal Ready 50s (x12 over 57s) migmigration_controller The migration is ready.
Normal Running 50s      migmigration_controller Step: 9/47
Normal Running 50s      migmigration_controller Step: 10/47

```

11.3.5. Migración de estado

Puede realizar migraciones repetibles, solo de estado, utilizando Migration Toolkit for Containers (MTC) para migrar las reclamaciones de volúmenes persistentes (PVC) que constituyen el estado de una aplicación. Se migran las PVC especificadas y se excluyen otras PVC del plan de migración. Los datos del volumen persistente (PV) se copian en el clúster de destino. Las referencias del PV no se mueven. Los pods de aplicación siguen ejecutándose en el clúster de origen. Puede asignar las PVC para asegurarse de que las PVC de origen y destino estén sincronizadas.

Puede realizar una migración única de los objetos de Kubernetes que constituyen el estado de una aplicación.

Si tiene una canalización de CI/CD, puede migrar componentes sin estado implementándolos en el clúster de destino. A continuación, puede migrar los componentes con estado utilizando MTC.

Puede realizar una migración de estado entre clústeres o dentro del mismo clúster.



IMPORTANTE

La migración de estado migra solo los componentes que constituyen el estado de una aplicación. Si desea migrar un espacio de nombres completo, utilice la migración por etapas o transición.

Recursos adicionales

- Consulte [Exclusión de las PVC de la migración](#) para seleccionar las PVC para la migración de estado.
- Consulte [Asignación de PVC](#) para migrar los datos de los PV de origen a las PVC provisionadas en el clúster de destino.
- Consulte [Migración de objetos de Kubernetes](#) para migrar los objetos de Kubernetes que constituyen el estado de una aplicación.

11.4. ENLACES DE MIGRACIÓN

Puede añadir hasta cuatro enlaces de migración a un único plan de migración y cada enlace se ejecutará en una fase diferente de la migración. Los enlaces de migración realizan tareas tales como la personalización de la inactividad de la aplicación, la migración manual de tipos de datos no compatibles y la actualización de las aplicaciones después de la migración.

Un enlace de migración se ejecuta en un clúster de origen o de destino en uno de los siguientes pasos de migración:

- **Antes de la copia de seguridad:** antes de realizar la copia de seguridad de los recursos en el clúster de origen
- **Después de la copia de seguridad:** después de que los recursos se respalden en el clúster de origen
- **Antes de la restauración:** antes de restaurar los recursos en el clúster de destino
- **Después de la restauración:** después de restaurar los recursos en el clúster de destino

Puede crear un enlace creando una estrategia de Ansible que se ejecute con la imagen predeterminada de Ansible o con un contenedor de enlaces personalizado.

Estrategias de Ansible

Las estrategias de Ansible se montan en un contenedor de enlaces como mapa de configuración. El contenedor de enlaces se ejecuta como un trabajo, utilizando el clúster, la cuenta de servicio y el espacio de nombres especificados en el recurso personalizado **MigPlan**. El trabajo continúa ejecutándose hasta que alcanza el límite por defecto de 6 reintentos o hasta que se completa con éxito. Esto continúa incluso si se desaloja o elimina el pod inicial.

La imagen de ejecución de Ansible por defecto es **registry.redhat.io/rhmtc/openshift-migration-hook-runner-rhel7:1.7**. Esta imagen se basa en la imagen de Ansible Runner e incluye **python-openshift** para los recursos de Ansible Kubernetes y el binario **oc** actualizado.

Contenedor de enlaces personalizado

Puede utilizar un contenedor de enlaces personalizado en lugar de la imagen predeterminada de Ansible.

11.4.1. Escritura de una estrategia de Ansible para un enlace de migración

Puede escribir una estrategia de Ansible para utilizarla como enlace de migración. El enlace se añade a un plan de migración utilizando la consola web de MTC o especificando valores para los parámetros **spec.hooks** en el manifiesto de recursos personalizados (CR) de **MigPlan**.

La estrategia de Ansible se monta en un contenedor de enlaces como mapa de configuración. El contenedor de enlaces se ejecuta como un trabajo, utilizando el clúster, la cuenta de servicio y el espacio de nombres especificados en el CR **MigPlan**. El contenedor de enlaces utiliza un token de cuenta de servicio especificado para que las tareas no requieran autenticación antes de ejecutarse en el clúster.

11.4.1.1. Módulos de Ansible

Puede utilizar el módulo **shell** de Ansible para ejecutar comandos **oc**.

Ejemplo de módulo **shell**

```
- hosts: localhost
gather_facts: false
tasks:
- name: get pod name
  shell: oc get po --all-namespaces
```

Puede utilizar los módulos **kubernetes.core**, como **k8s_info**, para interactuar con los recursos de Kubernetes.

Ejemplo de módulo **k8s_info**

```
- hosts: localhost
gather_facts: false
tasks:
- name: Get pod
  k8s_info:
    kind: pods
    api: v1
    namespace: openshift-migration
    name: "{{ lookup('env', 'HOSTNAME') }}"
    register: pods

- name: Print pod name
  debug:
    msg: "{{ pods.resources[0].metadata.name }}"
```

Puede utilizar el módulo **fail** para producir un estado de salida distinto de cero en casos en los que normalmente no se produciría un estado de salida distinto de cero, garantizando que se detecte el éxito o el fracaso de un enlace. Los enlaces se ejecutan como trabajos y el estado de éxito o fracaso de un enlace se basa en el estado de salida del contenedor de trabajos.

Ejemplo de módulo **fail**

```
- hosts: localhost
gather_facts: false
tasks:
```

```
- name: Set a boolean
  set_fact:
    do_fail: true

- name: "fail"
  fail:
    msg: "Cause a failure"
  when: do_fail
```

11.4.1.2. Variables del entorno

El nombre del CR **MigPlan** y los espacios de nombres de migración se pasan como variables de entorno al contenedor de enlaces. Se accede a estas variables mediante el complemento de **búsqueda**.

Variables de entorno adicionales

```
- hosts: localhost
  gather_facts: false
  tasks:
  - set_fact:
      namespaces: "{{ (lookup('env', 'migration_namespaces')).split(',') }}"

  - debug:
      msg: "{{ item }}"
      with_items: "{{ namespaces }}"

  - debug:
      msg: "{{ lookup('env', 'migplan_name') }}"
```

11.5. OPCIONES DEL PLAN DE MIGRACIÓN

Puede excluir, editar y asignar componentes en el recurso personalizado (CR) de **MigPlan**.

11.5.1. Exclusión de los recursos

Puede excluir recursos, por ejemplo, flujos de imágenes, volúmenes persistentes (PV) o suscripciones, de un plan de migración de Migration Toolkit for Containers (MTC) para reducir la carga de recursos para la migración o para migrar imágenes o PV con una herramienta diferente.

Por defecto, MTC excluye de la migración los recursos del catálogo de servicios y los recursos de Operator Lifecycle Manager (OLM). Estos recursos forman parte del grupo de API del catálogo de servicios y el grupo de API de OLM, ninguno de los cuales es compatible con la migración en este momento.

Procedimiento

1. Edite el manifiesto de recursos personalizados de **MigrationController**:

```
$ oc edit migrationcontroller <migration_controller> -n openshift-migration
```

2. Actualice la sección de **especificaciones** añadiendo parámetros para excluir recursos específicos. Para aquellos recursos que no tienen sus propios parámetros de exclusión, añada el parámetro **additional_excluded_resources**:

-

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigrationController
metadata:
  name: migration-controller
  namespace: openshift-migration
spec:
  disable_image_migration: true 1
  disable_pv_migration: true 2
  additional_excluded_resources: 3
  - resource1
  - resource2
  ...

```

- 1 Añada **disable_image_migration: true** para excluir los flujos de imágenes de la migración. **imagestreams** se añade a la lista **excluded_resources** en **main.yml** cuando se reinicia el pod de **MigrationController**.
- 2 Añada **disable_pv_migration: true** para excluir los PV del plan de migración. **persistentvolumes** y **persistentvolumeclaims** se añaden a la lista **excluded_resources** en **main.yml** cuando se reinicia el pod de **MigrationController**. Al desactivar la migración de FV también se desactiva la detección de FV cuando se crea el plan de migración.
- 3 Puede añadir los recursos de OpenShift Container Platform que desee excluir a la lista **additional_excluded_resources**.

3. Espere dos minutos a que se reinicie el pod de **MigrationController** para que se apliquen los cambios.
4. Compruebe que el recurso esté excluido:

```

$ oc get deployment -n openshift-migration migration-controller -o yaml | grep
EXCLUDED_RESOURCES -A1

```

El resultado contiene los recursos excluidos:

Ejemplo de salida

```

name: EXCLUDED_RESOURCES
value:
resource1,resource2,imagetags,templateinstances,clusterserviceversions,packagemanifests,sul
scriptions,servicebrokers,servicebindings,serviceclasses,serviceinstances,serviceplans,imagest
ams,persistentvolumes,persistentvolumeclaims

```

11.5.2. Asignación de espacios de nombres

Si asigna espacios de nombres en el recurso personalizado (CR) de **MigPlan**, debe asegurarse de que los espacios de nombres no estén duplicados en los clústeres de origen o de destino, ya que los rangos UID y GID de los espacios de nombres se copian durante la migración.

Dos espacios de nombres de origen asignados al mismo espacio de nombres de destino

```

spec:
  namespaces:

```

- namespace_2
- namespace_1:namespace_2

Si desea que el espacio de nombres de origen se asigne a un espacio de nombres del mismo nombre, no es necesario crear una asignación. Por defecto, un espacio de nombres de origen y un espacio de nombres de destino tienen el mismo nombre.

Asignación incorrecta del espacio de nombres

```
spec:
  namespaces:
    - namespace_1:namespace_1
```

Referencia correcta al espacio de nombres

```
spec:
  namespaces:
    - namespace_1
```

11.5.3. Exclusión de reclamaciones de volúmenes persistentes

Seleccione las reclamaciones de volúmenes persistentes (PVC) para la migración de estado excluyendo las PVC que no se desean migrar. Para excluir las PVC, se debe establecer el parámetro **spec.persistentVolumes.pvc.selection.action** del recurso personalizado (CR) de **MigPlan** después de descubrir los volúmenes persistentes (PV).

Requisitos previos

- El CR **MigPlan** debe tener el estado **Ready** (Listo).

Procedimiento

- Añada el parámetro **spec.persistentVolumes.pvc.selection.action** al CR **MigPlan** y configúrelo como **skip**:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
  namespace: openshift-migration
spec:
  ...
  persistentVolumes:
    - capacity: 10Gi
      name: <pv_name>
      pvc:
        ...
        selection:
          action: skip
```

11.5.4. Asignación de reclamaciones de volúmenes persistentes

Puede migrar datos de un volumen persistente (PV) del clúster de origen a las reclamaciones de

volúmenes persistentes (PVC) que ya están provisionadas en el clúster de destino en el CR **MigPlan** mediante la asignación de las PVC. Esta asignación garantiza que las PVC de destino de las aplicaciones migradas estén sincronizadas con las PVC de origen.

Las PVC se asignan actualizando el parámetro **spec.persistentVolumes.pvc.name** en el recurso personalizado (CR) **MigPlan** una vez descubiertas los PV.

Requisitos previos

- El CR **MigPlan** debe tener el estado **Ready** (Listo).

Procedimiento

- Actualice el parámetro **spec.persistentVolumes.pvc.name** en el CR **MigPlan**:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
  namespace: openshift-migration
spec:
  ...
  persistentVolumes:
  - capacity: 10Gi
    name: <pv_name>
    pvc:
      name: <source_pvc>:<destination_pvc> ❶
```

- ❶ Especifique la PVC en el clúster de origen y la PVC en el clúster de destino. Si la PVC de destino no existe, se creará. Puede utilizar esta asignación para cambiar el nombre de la PVC durante la migración.

11.5.5. Edición de los atributos del volumen persistente

Después de crear un recurso personalizado (CR) de **MigPlan**, el CR **MigrationController** detecta los volúmenes persistentes (PV). El bloque **spec.persistentVolumes** y el bloque **status.destStorageClasses** se añaden al CR **MigPlan**.

Puede editar los valores en el bloque **spec.persistentVolumes.selection**. Si se cambian los valores fuera del bloque **spec.persistentVolumes.selection**, los valores se sobrescriben cuando el CR **MigPlan** es reconciliado por el CR **MigrationController**.



NOTA

El valor por defecto del parámetro **spec.persistentVolumes.selection.storageClass** está determinado por la siguiente lógica:

1. Si el PV del clúster de origen es Gluster o NFS, el valor por defecto es **cephfs** para **accessMode: ReadWriteMany** o **cephrbd** para **accessMode: ReadWriteOnce**.
2. Si el PV no es ni Gluster ni NFS o si **cephfs** o **cephrbd** no están disponibles, el valor por defecto es una clase de almacenamiento para el mismo proveedor.
3. Si una clase de almacenamiento para el mismo proveedor no está disponible, el valor predeterminado es la clase de almacenamiento por defecto del clúster de destino.

Puede cambiar el valor de **storageClass** por el valor de cualquier parámetro de **nombre** en el bloque **status.destStorageClasses** del CR **MigPlan**.

Si el valor de **storageClass** está vacío, el PV no tendrá clase de almacenamiento después de la migración. Esta opción es apropiada si, por ejemplo, desea mover el PV a un volumen del NFS en el clúster de destino.

Requisitos previos

- El CR **MigPlan** debe tener el estado **Ready** (Listo).

Procedimiento

- Edite los valores **spec.persistentVolumes.selection** en el CR **MigPlan**:

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
  namespace: openshift-migration
spec:
  persistentVolumes:
  - capacity: 10Gi
    name: pvc-095a6559-b27f-11eb-b27f-021bddcaf6e4
    proposedCapacity: 10Gi
    pvc:
      accessModes:
      - ReadWriteMany
      hasReference: true
      name: mysql
      namespace: mysql-persistent
    selection:
      action: <copy> 1
      copyMethod: <filesystem> 2
      verify: true 3
      storageClass: <gp2> 4
      accessMode: <ReadWriteMany> 5
      storageClass: cephfs

```

- 1 Los valores permitidos son **move**, **copy** y **skip**. Si solo se admite una acción, el valor por defecto es la acción admitida. Si se admiten varias acciones, el valor por defecto es **copy**.
- 2 Los valores permitidos son **snapshot** y **filesystem**. El valor por defecto es **filesystem**.
- 3 El parámetro **verify** se muestra si se selecciona la opción de verificación para la copia de sistemas de archivos en la consola web de MTC. Puede ponerlo en **false**.
- 4 Puede cambiar el valor predeterminado por el valor de cualquier parámetro de **nombre** en el bloque **status.destStorageClasses** del CR **MigPlan**. Si no se especifica ningún valor, el PV no tendrá clase de almacenamiento después de la migración.
- 5 Los valores permitidos son **ReadWriteOnce** y **ReadWriteMany**. Si no se especifica este valor, el valor predeterminado es el modo de acceso de la PVC del clúster de origen. Solo se puede editar el modo de acceso en el CR **MigPlan**. No se puede editar mediante la consola web de MTC.

Recursos adicionales

- Para más detalles sobre las acciones de **mover** y **copiar**, consulte [Flujo de trabajo de MTC](#) .
- Para obtener más información sobre la acción de **omisión**, consulte [Exclusión de las PVC de la migración](#).
- Para obtener más detalles sobre los métodos de copia instantáneas y de sistemas de archivos, consulte [Acerca de los métodos de copia de datos](#) .

11.5.6. Migración de objetos de Kubernetes

Puede realizar una migración única de los objetos de Kubernetes que constituyen el estado de una aplicación.



NOTA

Después de la migración, el parámetro **closed** del CR **MigPlan** se establece como **true** (verdadero). No se puede crear otro CR **MigMigration** para el CR **MigPlan**.

Los objetos de Kubernetes se añaden al CR **MigPlan** mediante una de las siguientes opciones:

- Añada los objetos Kubernetes a la sección **includedResources**.
- Use el parámetro **labelSelector** para hacer referencia a los objetos de Kubernetes etiquetados.
- Añada objetos de Kubernetes a la sección **includedResources** y, luego, fíltrelos con el parámetro **labelSelector**, por ejemplo, los recursos **Secret** y **ConfigMap** con la etiqueta **app: frontend**.

Procedimiento

- Actualice el CR **MigPlan**:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  name: <migplan>
```



```

namespace: openshift-migration
spec:
  includedResources:
    - kind: <kind> 1
      group: ""
    - kind: <kind>
      group: ""
  ...
  labelSelector:
    matchLabels:
      <label> 2

```

1 Especifique el objeto de Kubernetes, por ejemplo, **Secret** o **ConfigMap**.

2 Especifique la etiqueta de los recursos a migrar, por ejemplo, **app: frontend**.

11.6. OPCIONES DEL CONTROLADOR DE MIGRACIÓN

Puede editar los límites del plan de migración, habilitar el cambio de tamaño del volumen persistente o habilitar los clientes de Kubernetes en la caché en el recurso personalizado (CR) **MigrationController** para realizar grandes migraciones y mejorar el rendimiento.

11.6.1. Aumento de los límites para las grandes migraciones

Puede aumentar los límites de los objetos de migración y los recursos del contenedor para las migraciones de gran tamaño con Migration Toolkit for Containers (MTC).



IMPORTANTE

Debe probar estos cambios antes de realizar una migración en un entorno de producción.

Procedimiento

1. Edite el manifiesto de recursos personalizados (CR) de **MigrationController**:

```
$ oc edit migrationcontroller -n openshift-migration
```

2. Actualice los siguientes parámetros:

```

...
mig_controller_limits_cpu: "1" 1
mig_controller_limits_memory: "10Gi" 2
...
mig_controller_requests_cpu: "100m" 3
mig_controller_requests_memory: "350Mi" 4
...
mig_pv_limit: 100 5
mig_pod_limit: 100 6
mig_namespace_limit: 10 7
...

```

1. Especifique el número de CPU disponibles para el CR **MigrationController**.
 2. Especifique la cantidad de memoria disponible para el CR **MigrationController**.
 3. Especifique el número de CPU disponibles para las solicitudes del CR **MigrationController**. **100m** representa 0,1 unidades de CPU ($100 * 1e-3$).
 4. Especifique la cantidad de memoria disponible para las solicitudes de CR de **MigrationController**.
 5. Especifique el número de volúmenes persistentes que se pueden migrar.
 6. Especifique el número de pods que se pueden migrar.
 7. Especifique el número de espacios de nombres que se pueden migrar.
3. Cree un plan de migración que utilice los parámetros actualizados para verificar los cambios. Si su plan de migración supera los límites de CR de **MigrationController**, la consola de MTC muestra un mensaje de advertencia al guardar el plan de migración.

11.6.2. Habilitación del redimensionamiento de volúmenes persistentes para la migración directa de volúmenes

Puede habilitar el redimensionamiento de los volúmenes persistentes (PV) para la migración directa de volúmenes para evitar que se agote el espacio en disco en el clúster de destino.

Cuando el uso del disco de un PV alcanza un nivel configurado, el recurso personalizado (CR) **MigrationController** compara la capacidad de almacenamiento solicitada de una reclamación de volumen persistente (PVC) con su capacidad real aprovisionada. A continuación, calcule el espacio necesario en el clúster de destino.

El parámetro **pv_resizing_threshold** determina cuándo se utiliza el redimensionamiento de PV. El umbral por defecto es del **3 %**. Esto significa que el redimensionamiento de PV se produce cuando el uso del disco de un PV es superior al **97 %**. Puede aumentar este umbral para que el redimensionamiento de PV se produzca a nivel de uso del disco más bajo.

La capacidad de PVC se calcula según los siguientes criterios:

- Si la capacidad de almacenamiento solicitada (**spec.resources.requests.storage**) de la PVC no es igual a la capacidad real provisionada (**status.capacity.storage**), se utiliza el valor mayor.
- Si un PV se aprovisiona a través de una PVC y posteriormente se modifica de forma que sus capacidades de PV y PVC ya no coinciden, se utiliza el valor mayor.

Requisitos previos

- Las PVC deben estar conectadas a uno o más pods en funcionamiento para que el CR **MigrationController** pueda ejecutar los comandos.

Procedimiento

1. Inicie la sesión en el clúster del host.
2. Habilite el redimensionamiento de los PV parcheando el CR **MigrationController**:

```
$ oc patch migrationcontroller migration-controller -p '{"spec":
{"enable_dvm_pv_resizing":true}}' \ 1
--type='merge' -n openshift-migration
```

- 1** Establezca el valor en **false** para desactivar el redimensionamiento de PV.

3. Opcional: actualice el parámetro **pv_resizing_threshold** para aumentar el umbral:

```
$ oc patch migrationcontroller migration-controller -p '{"spec":{"pv_resizing_threshold":41}}' \
1
--type='merge' -n openshift-migration
```

- 1** El valor por defecto es **3**.

Cuando se supera el umbral, se muestra el siguiente mensaje de estado en el estado del CR **MigPlan**:

```
status:
conditions:
...
- category: Warn
  durable: true
  lastTransitionTime: "2021-06-17T08:57:01Z"
  message: 'Capacity of the following volumes will be automatically adjusted to avoid disk
capacity issues in the target cluster: [pvc-b800eb7b-cf3b-11eb-a3f7-0eae3e0555f3]'
  reason: Done
  status: "False"
  type: PvCapacityAdjustmentRequired
```



NOTA

En el caso del almacenamiento de AWS gp2, este mensaje no aparece a menos que **pv_resizing_threshold** sea del 42 % o superior debido a la forma en que gp2 calcula el uso y el tamaño del volumen. ([BZ#1973148](#))

11.6.3. Habilitación de clientes Kubernetes en caché

Puede habilitar clientes Kubernetes en caché en el recurso personalizado (CR) **MigrationController** para mejorar el rendimiento durante la migración. El mayor beneficio de rendimiento se muestra cuando se migra entre clústeres en diferentes regiones o con una latencia de red significativa.



NOTA

Las tareas delegadas, por ejemplo, la copia de seguridad de Rsync para la migración directa de volúmenes o la copia de seguridad y restauración de Velero, no muestran un rendimiento mejorado con los clientes en caché.

Los clientes en caché requieren memoria adicional porque el CR **MigrationController** almacena en caché todos los recursos de la API necesarios para interactuar con los CR de **MigCluster**. Las solicitudes que normalmente se envían al servidor de la API se dirigen a la caché en su lugar. La caché vigila el servidor de la API para las actualizaciones.

Puede aumentar los límites de memoria y las solicitudes de CR **MigrationController** si se producen errores **OOMKilled** después de habilitar los clientes en caché.

Procedimiento

1. Habilite los clientes en caché ejecutando el siguiente comando:

```
$ oc -n openshift-migration patch migrationcontroller migration-controller --type=json --patch \
'[{ "op": "replace", "path": "/spec/mig_controller_enable_cache", "value": true}]'
```

2. Opcional: aumente los límites de memoria del CR **MigrationController** ejecutando el siguiente comando:

```
$ oc -n openshift-migration patch migrationcontroller migration-controller --type=json --patch \
'[{ "op": "replace", "path": "/spec/mig_controller_limits_memory", "value": <10Gi>}]'
```

3. Opcional: aumente las solicitudes de memoria del CR **MigrationController** ejecutando el siguiente comando:

```
$ oc -n openshift-migration patch migrationcontroller migration-controller --type=json --patch \
'[{ "op": "replace", "path": "/spec/mig_controller_requests_memory", "value": <350Mi>}]'
```

CAPÍTULO 12. RESOLUCIÓN DE PROBLEMAS

Esta sección describe los recursos para la resolución de problemas de Migration Toolkit for Containers (MTC).

Para ver los problemas conocidos, consulte las [notas de la versión de MTC](#).

12.1. FLUJO DE TRABAJO DE MTC

Puede migrar los recursos de Kubernetes, los datos de volúmenes persistentes y las imágenes de contenedores internos a OpenShift Container Platform 4.10 utilizando la consola web de Migration Toolkit for Containers (MTC) o la API de Kubernetes.

MTC migra los siguientes recursos:

- Un espacio de nombres especificado en un plan de migración.
- Recursos de espacios de nombres: cuando MTC migra un espacio de nombres, migra todos los objetos y recursos asociados a ese espacio de nombres, como los servicios o los pods. Además, si un recurso que existe en el espacio de nombres, pero no en el nivel del clúster, depende de un recurso que existe en el nivel del clúster, MTC migra ambos recursos.
Por ejemplo, una restricción de contexto de seguridad (SCC) es un recurso que existe a nivel del clúster y una cuenta de servicio (SA) es un recurso que existe a nivel del espacio de nombres. Si existe una SA en un espacio de nombres que MTC migra, MTC localiza automáticamente cualquier SCC vinculada a la SA y también la migra. Del mismo modo, MTC migra las reclamaciones de volúmenes persistentes que están vinculadas a los volúmenes persistentes del espacio de nombres.



NOTA

Es posible que deba migrar los recursos con alcance del clúster de forma manual según el recurso.

- Recursos personalizados (CR) y definiciones de recursos personalizados (CRD): MTC migra automáticamente los CR y las CRD a nivel del espacio de nombres.

La migración de una aplicación con la consola web de MTC implica los siguientes pasos:

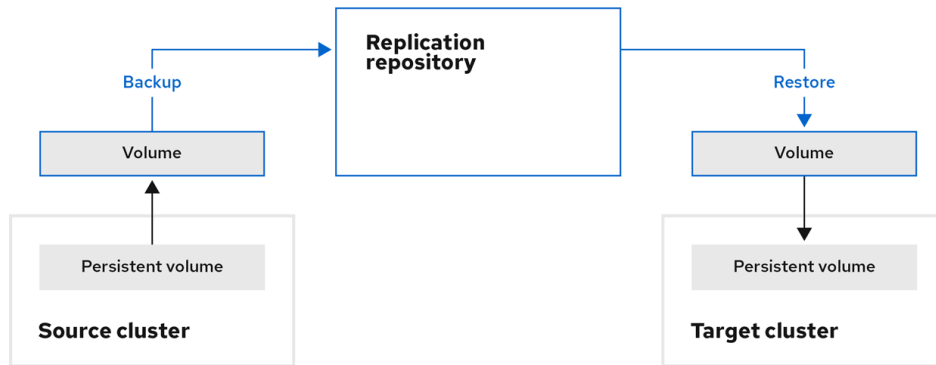
1. Instale el operador de Migration Toolkit for Containers en todos los clústeres.
Puede instalar el operador de Migration Toolkit for Containers en un entorno restringido con acceso a Internet limitado o nulo. Los clústeres de origen y destino deben tener acceso a la red entre sí y a un registro de réplica.
2. Configure el repositorio de replicación, un almacenamiento de objetos intermedio que MTC utiliza para migrar los datos.
Los clústeres de origen y destino deben tener acceso a la red del repositorio de replicación durante la migración. Si utiliza un servidor proxy, debe configurarlo para permitir el tráfico de red entre el repositorio de replicación y los clústeres.
3. Añada el clúster de origen a la consola web de MTC.
4. Añada el repositorio de replicación a la consola web de MTC.
5. Cree un plan de migración con una de las siguientes opciones de migración de datos:

- **Copiar:** MTC copia los datos del clúster de origen en el repositorio de replicación y del repositorio de replicación en el clúster de destino.



NOTA

Si utiliza la migración directa de imágenes o la migración directa de volúmenes, las imágenes o los volúmenes se copian directamente del clúster de origen al de destino.



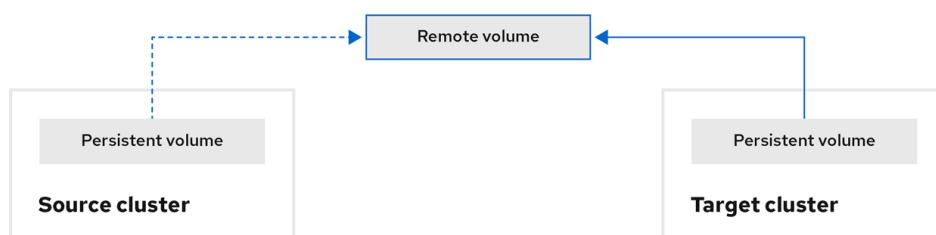
OpenShift_45_1019

- **Mover:** MTC desmonta un volumen remoto, por ejemplo, NFS, del clúster de origen, crea un recurso de PV en el clúster de destino que apunta al volumen remoto y, luego, monta el volumen remoto en el clúster de destino. Las aplicaciones que se ejecutan en el clúster de destino utilizan el mismo volumen remoto que utilizaba el clúster de origen. El volumen remoto debe ser accesible para los clústeres de origen y destino.



NOTA

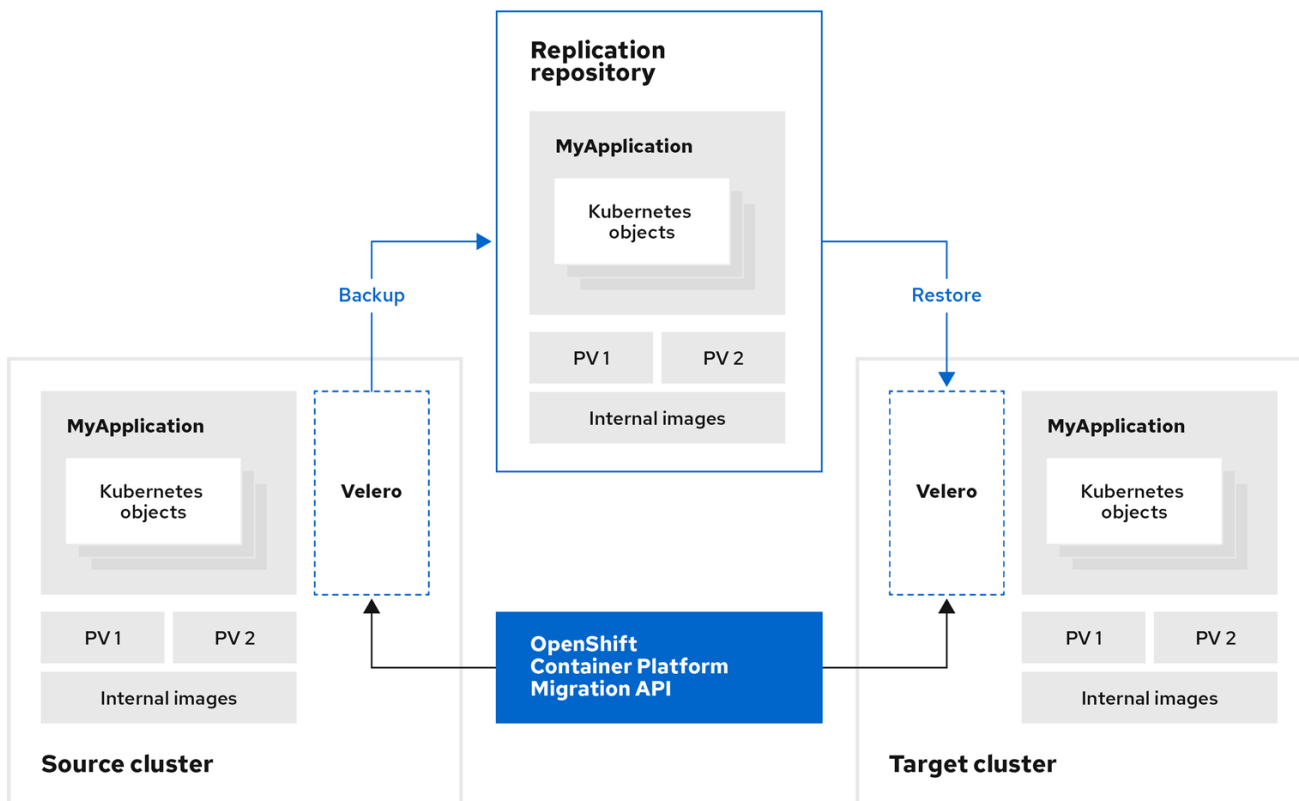
Aunque el repositorio de replicación no aparece en este diagrama, es necesario para la migración.



OpenShift_45_1019

6. Ejecute el plan de migración con una de las siguientes opciones:

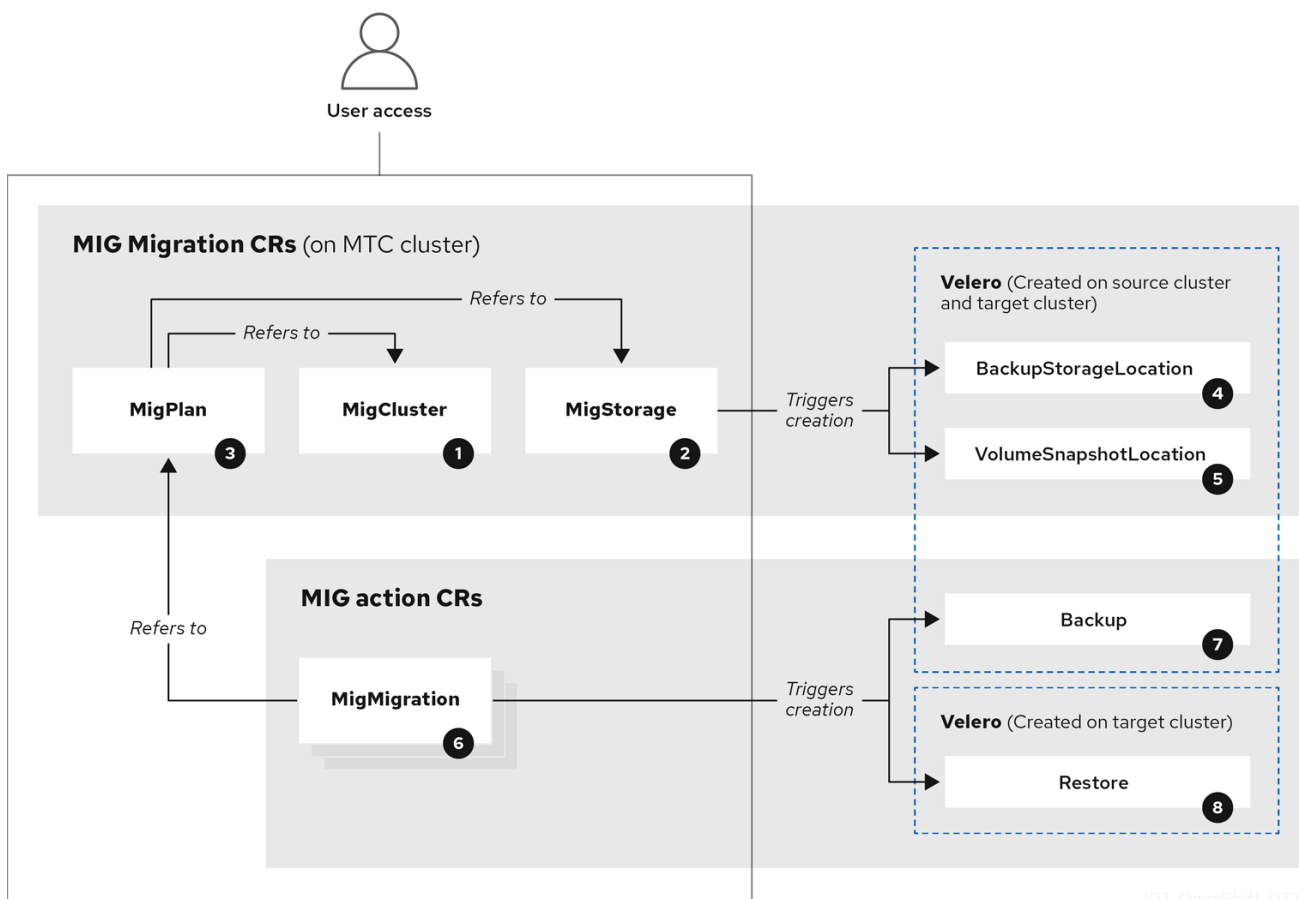
- La **migración por etapas** copia los datos en el clúster de destino sin detener la aplicación. La migración por etapas puede ejecutarse varias veces para que la mayor parte de los datos se copien en el destino antes de la migración. Ejecutar una o más migraciones por etapas reduce la duración de la migración de transición.
- La **transición** detiene la aplicación en el clúster de origen y mueve los recursos al clúster de destino.
Opcional: puede desactivar la casilla de verificación **Detener transacciones en el clúster de origen durante la migración**.



OpenShift_45_1019

Acerca de los recursos personalizados de MTC

Migration Toolkit for Containers (MTC) crea los siguientes recursos personalizados (CR):



102_OpenShift_0720

- 1 **MigCluster** (configuración, clúster de MTC): definición de clúster
- 2 **MigStorage** (configuración, clúster de MTC): definición de almacenamiento
- 3 **MigPlan** (configuración, clúster de MTC): plan de migración

El CR **MigPlan** describe los clústeres de origen y destino, el repositorio de replicación y los espacios de nombres que se migran. Se asocia a 0, 1 o muchos CR de **MigMigration**.



NOTA

Al eliminar un CR de **MigPlan** se eliminan los CR de **MigMigration** asociados.

- 4 **BackupStorageLocation** (configuración, clúster de MTC): ubicación de los objetos de copia de seguridad de **Velero**.
- 5 **VolumeSnapshotLocation** (configuración, clúster de MTC): ubicación de las instantáneas de volumen de **Velero**.
- 6 **MigMigration** (acción, clúster de MTC): migración creada cada vez que se escenifica o migran datos. Cada CR de **MigMigration** está asociado a un CR de **MigPlan**.
- 7 **Copia de seguridad** (acción, clúster de origen): cuando se ejecuta un plan de migración, el CR **MigMigration** crea dos CR de copia de seguridad de **Velero** en cada clúster de origen:
 - Copia de seguridad de CR #1 para objetos de Kubernetes
 - Copia de seguridad de CR #2 para los datos de PV
- 8 **Restauración** (acción, clúster de destino): cuando se ejecuta un plan de migración, el CR **MigMigration** crea dos CR de restauración de **Velero** en el clúster de destino:
 - Restauración de CR #1 (utilizando la copia de seguridad de CR #2) para los datos de PV
 - Restauración de CR #2 (utilizando la copia de seguridad de CR #1) para los objetos de Kubernetes

12.2. MANIFIESTOS DE RECURSOS PERSONALIZADOS DE MTC

Migration Toolkit for Containers (MTC) utiliza los siguientes manifiestos de recursos personalizados (CR) para la migración de aplicaciones.

12.2.1. DirectImageMigration

El CR **DirectImageMigration** copia las imágenes directamente del clúster de origen al de destino.

```
apiVersion: migration.openshift.io/v1alpha1
kind: DirectImageMigration
metadata:
  labels:
```



```

  controller-tools.k8s.io: "1.0"
  name: <direct_image_migration>
spec:
  srcMigClusterRef:
    name: <source_cluster>
    namespace: openshift-migration
  destMigClusterRef:
    name: <destination_cluster>
    namespace: openshift-migration
  namespaces: ❶
    - <source_namespace_1>
    - <source_namespace_2>:<destination_namespace_3> ❷

```

- ❶ Uno o varios espacios de nombres que contengan imágenes para migrar. Por defecto, el espacio de nombres de destino tiene el mismo nombre que el espacio de nombres de origen.
- ❷ Espacio de nombres de origen asignado a un espacio de nombres de destino con un nombre diferente.

12.2.2. DirectImageStreamMigration

El CR **DirectImageStreamMigration** copia las referencias de los flujos de imágenes directamente del clúster de origen al de destino.

```

apiVersion: migration.openshift.io/v1alpha1
kind: DirectImageStreamMigration
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <direct_image_stream_migration>
spec:
  srcMigClusterRef:
    name: <source_cluster>
    namespace: openshift-migration
  destMigClusterRef:
    name: <destination_cluster>
    namespace: openshift-migration
  imageStreamRef:
    name: <image_stream>
    namespace: <source_image_stream_namespace>
  destNamespace: <destination_image_stream_namespace>

```

12.2.3. DirectVolumeMigration

El CR **DirectVolumeMigration** copia los volúmenes persistentes (PV) directamente del clúster de origen al de destino.

```

apiVersion: migration.openshift.io/v1alpha1
kind: DirectVolumeMigration
metadata:
  name: <direct_volume_migration>
  namespace: openshift-migration
spec:

```

```

createDestinationNamespaces: false 1
deleteProgressReportingCRs: false 2
destMigClusterRef:
  name: <host_cluster> 3
  namespace: openshift-migration
persistentVolumeClaims:
- name: <pvc> 4
  namespace: <pvc_namespace>
srcMigClusterRef:
  name: <source_cluster>
  namespace: openshift-migration

```

- 1** Establezca en **true** (verdadero) para crear espacios de nombres para los PV en el clúster de destino.
- 2** Establezca en **true** (verdadero) para eliminar los CR de **DirectVolumeMigrationProgress** después de la migración. El valor predeterminado es **false** (falso) para que los CR de **DirectVolumeMigrationProgress** se conserven para la resolución de problemas.
- 3** Actualice el nombre del clúster si el clúster de destino no es el clúster del host.
- 4** Especifique una o más PVC para migrar.

12.2.4. DirectVolumeMigrationProgress

El CR **DirectVolumeMigrationProgress** muestra el progreso del CR **DirectVolumeMigration**.

```

apiVersion: migration.openshift.io/v1alpha1
kind: DirectVolumeMigrationProgress
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <direct_volume_migration_progress>
spec:
  clusterRef:
    name: <source_cluster>
    namespace: openshift-migration
  podRef:
    name: <rsync_pod>
    namespace: openshift-migration

```

12.2.5. MigAnalytic

El CR **MigAnalytic** recoge el número de imágenes, los recursos de Kubernetes y la capacidad del volumen persistente (PV) de un CR **MigPlan** asociado.

Puede configurar los datos que recoge.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigAnalytic
metadata:
  annotations:
    migplan: <migplan>

```

```

name: <miganalytic>
namespace: openshift-migration
labels:
  migplan: <migplan>
spec:
  analyzeImageCount: true 1
  analyzeK8SResources: true 2
  analyzePVCapacity: true 3
  listImages: false 4
  listImagesLimit: 50 5
  migPlanRef:
    name: <migplan>
    namespace: openshift-migration

```

- 1 Opcional: devuelve el número de imágenes.
- 2 Opcional: devuelve el número, el tipo y la versión de la API de los recursos de Kubernetes.
- 3 Opcional: devuelve la capacidad del PV.
- 4 Devuelve una lista de nombres de imágenes. El valor por defecto es **false** (falso) para que la salida no sea excesivamente larga.
- 5 Opcional: especifique el número máximo de nombres de imágenes para devolver si **listImages** está en **true** (verdadero).

12.2.6. MigCluster

El CR **MigCluster** define un clúster de host, local o remoto.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigCluster
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <host_cluster> 1
  namespace: openshift-migration
spec:
  isHostCluster: true 2
  # The 'azureResourceGroup' parameter is relevant only for Microsoft Azure.
  azureResourceGroup: <azure_resource_group> 3
  caBundle: <ca_bundle_base64> 4
  insecure: false 5
  refresh: false 6
  # The 'restartRestic' parameter is relevant for a source cluster.
  restartRestic: true 7
  # The following parameters are relevant for a remote cluster.
  exposedRegistryPath: <registry_route> 8
  url: <destination_cluster_url> 9
  serviceAccountSecretRef:
    name: <source_secret> 10
    namespace: openshift-config

```

- 1 Actualice el nombre del clúster si el pod **migration-controller** no se está ejecutando en este clúster.
- 2 El pod **migration-controller** se ejecuta en este clúster si está en **true** (verdadero).
- 3 Solo para Microsoft Azure: especifique el grupo de recursos.
- 4 Opcional: si ha creado un paquete de certificados para los certificados de CA autofirmados y el valor del parámetro **insecure** es **false** (falso), especifique el paquete de certificados codificado en base64.
- 5 Establézcalo en **true** (verdadero) para deshabilitar la verificación de SSL.
- 6 Establézcalo como **true** (verdadero) para validar el clúster.
- 7 Establézcalo en **true** (verdadero) para reiniciar los pods de **Restic** en el clúster de origen después de crear los pods de **Stage**.
- 8 Solo clúster remoto y migración directa de imágenes: especifique la ruta de registro segura expuesta.
- 9 Solo clúster remoto: especifique la URL.
- 10 Solo clúster remoto: especifique el nombre del objeto **Secret**.

12.2.7. MigHook

El CR **MigHook** define un enlace de migración que ejecuta un código personalizado en una etapa específica de la migración. Puede crear hasta cuatro enlaces de migración. Cada enlace se ejecuta durante una fase de migración diferente.

Puede configurar el nombre del enlace, la duración del tiempo de ejecución, una imagen personalizada y el clúster donde se ejecutará el enlace.

Las fases de migración y los espacios de nombres de los enlaces se configuran en el CR **MigPlan**.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigHook
metadata:
  generateName: <hook_name_prefix> 1
  name: <mighook> 2
  namespace: openshift-migration
spec:
  activeDeadlineSeconds: 1800 3
  custom: false 4
  image: <hook_image> 5
  playbook: <ansible_playbook_base64> 6
  targetCluster: source 7

```

- 1 Opcional: se añade un hash único al valor de este parámetro para que cada enlace de migración tenga un nombre único. No es necesario especificar el valor del parámetro **name**.
- 2 Especifique el nombre del enlace de migración, a menos que especifique el valor del parámetro **generateName**.

- 3 Opcional: especifique el número máximo de segundos en que puede ejecutarse un enlace. El valor por defecto es **1800**.
- 4 El enlace es una imagen personalizada si está en **true** (verdadero). La imagen personalizada puede incluir Ansible o puede estar escrita en un lenguaje de programación diferente.
- 5 Especifique la imagen personalizada, por ejemplo, **quay.io/konveyor/hook-runner:latest**. Es obligatorio si **custom** está en **true** (verdadero).
- 6 Estrategias de Ansible codificadas en base64. Obligatorias si **custom** está en **false** (falso).
- 7 Especifique el clúster en el que se ejecutará el enlace. Los valores válidos son **source** o **destination**.

12.2.8. MigMigration

El CR **MigMigration** ejecuta un CR de **MigPlan**.

Puede configurar un CR de **Migmigration** para ejecutar una migración por etapas o incremental a fin de cancelar una migración en curso o para retrotraer una migración completada.

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigMigration
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <migmigration>
  namespace: openshift-migration
spec:
  canceled: false 1
  rollback: false 2
  stage: false 3
  quiescePods: true 4
  keepAnnotations: true 5
  verify: false 6
  migPlanRef:
    name: <migplan>
    namespace: openshift-migration
```

- 1 Establezca como **true** (verdadero) para cancelar una migración en curso.
- 2 Establezca como **true** (verdadero) para retrotraer una migración completada.
- 3 Establezca como **true** (verdadero) para ejecutar una migración por etapas. Los datos se copian de forma incremental y los pods del clúster de origen no se detienen.
- 4 Establezca en **true** (verdadero) para detener la aplicación durante la migración. Los pods del clúster de origen se escalan a **0** después de la etapa **Backup** (Copia de seguridad).
- 5 Establezca como **true** (verdadero) para conservar las etiquetas y anotaciones aplicadas durante la migración.
- 6 Establezca como **true** (verdadero) para comprobar que el estado de los pods migrados en el clúster de destino se verifique y para devolver los nombres de los pods que no están en el estado **Running** (Ejecutándose).

12.2.9. MigPlan

El CR **MigPlan** define los parámetros de un plan de migración.

Puede configurar los espacios de nombres de destino, las fases de enlace y la migración directa o indirecta.



NOTA

Por defecto, el espacio de nombres de destino tiene el mismo nombre que el espacio de nombres de origen. Si configura un espacio de nombres de destino diferente, debe asegurarse de que los espacios de nombres no estén duplicados en los clústeres de origen o de destino, ya que los rangos UID y GID se copian durante la migración.

```

apiVersion: migration.openshift.io/v1alpha1
kind: MigPlan
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <migplan>
  namespace: openshift-migration
spec:
  closed: false 1
  srcMigClusterRef:
    name: <source_cluster>
    namespace: openshift-migration
  destMigClusterRef:
    name: <destination_cluster>
    namespace: openshift-migration
  hooks: 2
  - executionNamespace: <namespace> 3
    phase: <migration_phase> 4
    reference:
      name: <hook> 5
      namespace: <hook_namespace> 6
      serviceAccount: <service_account> 7
  indirectImageMigration: true 8
  indirectVolumeMigration: false 9
  migStorageRef:
    name: <migstorage>
    namespace: openshift-migration
  namespaces:
    - <source_namespace_1> 10
    - <source_namespace_2>
    - <source_namespace_3>:<destination_namespace_4> 11
  refresh: false 12

```

1 La migración se ha completado si está en **true** (verdadero). No se puede crear otro CR **MigMigration** para el CR **MigPlan**.

2

Opcional: puede especificar hasta cuatro enlaces de migración. Cada enlace debe ejecutarse durante una fase de migración diferente.

- 3 Opcional: especifique el espacio de nombres en el que se ejecutará el enlace.
- 4 Opcional: especifique la fase de migración durante la cual se ejecuta un enlace. Se puede asignar un enlace a una fase. Los valores válidos son **PreBackup**, **PostBackup**, **PreRestore** y **PostRestore**.
- 5 Opcional: especifique el nombre del CR de **MigHook**.
- 6 Opcional: especifique el espacio de nombres del CR de **MigHook**.
- 7 Opcional: especifique una cuenta de servicio con privilegios de **administrador de clúster**.
- 8 La migración directa de imágenes se deshabilita si está en **false** (falso). Las imágenes se copian del clúster de origen al repositorio de replicación, y del repositorio de replicación al clúster de destino.
- 9 La migración directa de volúmenes se deshabilita si está en **false** (falso). Los PV se copian del clúster de origen al repositorio de replicación y del repositorio de replicación al clúster de destino.
- 10 Especifique uno o más espacios de nombres de origen. Si se especifica solo el espacio de nombres de origen, el espacio de nombres de destino es el mismo.
- 11 Especifique el espacio de nombres de destino si es diferente del espacio de nombres de origen.
- 12 El CR **MigPlan** se valida si está en **true** (verdadero).

12.2.10. MigStorage

El CR **MigStorage** describe el almacenamiento de objetos para el repositorio de replicación.

Son compatibles Amazon Web Services (AWS), Microsoft Azure, Google Cloud Storage, Multi-Cloud Object Gateway y el almacenamiento en la nube genérico S3.

AWS y el método de copia de instantáneas tienen parámetros adicionales.

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigStorage
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <migstorage>
  namespace: openshift-migration
spec:
  backupStorageProvider: <backup_storage_provider> 1
  volumeSnapshotProvider: <snapshot_storage_provider> 2
  backupStorageConfig:
    awsBucketName: <bucket> 3
    awsRegion: <region> 4
    credsSecretRef:
      namespace: openshift-config
      name: <storage_secret> 5
    awsKmsKeyId: <key_id> 6
```

```

awsPublicUrl: <public_url> 7
awsSignatureVersion: <signature_version> 8
volumeSnapshotConfig:
  awsRegion: <region> 9
  credsSecretRef:
    namespace: openshift-config
    name: <storage_secret> 10
refresh: false 11

```

- 1 Especifique el proveedor de almacenamiento.
- 2 Solo para el método de copia de instantáneas: especifique el proveedor de almacenamiento.
- 3 Solo para AWS: especifique el nombre del bucket.
- 4 Solo para AWS: especifique la región del bucket, por ejemplo, **us-east-1**.
- 5 Especifique el nombre del objeto **Secret** que ha creado para el almacenamiento.
- 6 Solo para AWS: si utiliza el servicio de gestión de claves de AWS, especifique el identificador único de la clave.
- 7 Solo para AWS: si ha concedido acceso público al bucket de AWS, especifique la URL del bucket.
- 8 Solo para AWS: especifique la versión de la firma de AWS para autenticar las solicitudes al bucket, por ejemplo, **4**.
- 9 Solo para el método de copia de instantáneas: especifique la región geográfica de los clústeres.
- 10 Solo para el método de copia de instantáneas: especifique el nombre del objeto **Secret** que creó para el almacenamiento.
- 11 Establézcalo como **true** (verdadero) para validar el clúster.

12.3. REGISTROS Y HERRAMIENTAS DE DEPURACIÓN

En esta sección se describen los registros y las herramientas de depuración que se pueden utilizar para la resolución de problemas.

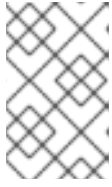
12.3.1. Vista de los recursos del plan de migración

Puede ver los recursos del plan de migración para monitorear una migración en curso o para solucionar problemas de una migración fallida mediante la consola web de MTC y la interfaz de línea de comandos (CLI).

Procedimiento


1. En la consola web de MTC, haga clic en **Migration Plans** (Planes de migración).
2. Haga clic en el número de **migraciones** junto a un plan de migración para ver la página **Migrations** (Migraciones).
3. Haga clic en una migración para ver los **detalles de la migración**.

4. Expanda los **recursos de migración** para ver los recursos de migración y su estado en la vista de árbol.



NOTA

Para solucionar una migración fallida, comience con un recurso de alto nivel que haya fallado y, luego, baje por el árbol de recursos hacia los recursos de nivel inferior.

5. Haga clic en el menú Options (Opciones)  junto a un recurso y seleccione una de las siguientes opciones:

- El comando **Copy oc describe** copia el comando en el portapapeles.
 - Inicie sesión en el clúster correspondiente y ejecute el comando. Las condiciones y los eventos del recurso se muestran en formato YAML.
- El comando **Copy oc logs** copia el comando en el portapapeles.
 - Inicie sesión en el clúster correspondiente y ejecute el comando. Si el recurso admite el filtrado de registros, se muestra un registro filtrado.
- En **View JSON** (Ver JSON) se muestran los datos del recurso en formato JSON en un navegador web. Los datos son los mismos que la salida del comando **oc get <resource>**.

12.3.2. Vista del registro de un plan de migración

Puede ver un registro agregado para un plan de migración. Utilice la consola web de MTC para copiar un comando en el portapapeles y, a continuación, ejecute el comando desde la interfaz de línea de comandos (CLI).

El comando muestra los registros filtrados de los siguientes pods:

- **Migration Controller**
- **Velero**
- **Restic**
- **Rsync**
- **Stunnel**
- **Registry**

Procedimiento

1. En la consola web de MTC, haga clic en **Migration Plans** (Planes de migración).
2. Haga clic en el número de **migraciones** junto a un plan de migración.
3. Haga clic en **View logs** (Ver registros).

4. Haga clic en el icono de copia para copiar el comando **oc logs** en el portapapeles.
5. Inicie sesión en el clúster correspondiente e introduzca el comando en la CLI.
Se muestra el registro agregado del plan de migración.

12.3.3. Uso del lector de registros de migración

Puede utilizar el lector de registros de migración para mostrar una única vista filtrada de todos los registros de migración.

Procedimiento

1. Obtenga el pod **mig-log-reader**:

```
$ oc -n openshift-migration get pods | grep log
```

2. Introduzca el siguiente comando para mostrar un único registro de migración:

```
$ oc -n openshift-migration logs -f <mig-log-reader-pod> -c color 1
```

- 1** En la opción **-c plain** se muestra el registro sin colores.

12.3.4. Acceso a las métricas de rendimiento

El recurso personalizado (CR) **MigrationController** registra las métricas y las extrae en el almacenamiento de monitoreo del clúster. Puede consultar las métricas utilizando Prometheus Query Language (PromQL) para diagnosticar problemas de rendimiento de la migración. Todas las métricas se restablecen cuando se reinicia el pod del controlador de migraciones.

Puede acceder a las métricas de rendimiento y ejecutar consultas mediante la consola web de OpenShift Container Platform.

Procedimiento

1. En la consola web de OpenShift Container Platform, haga clic en **Observe** (Observar) → **Metrics** (Métricas).
2. Introduzca una consulta PromQL, seleccione una ventana de tiempo para mostrar y haga clic en **Run Queries** (Ejecutar consultas).
Si su navegador web no muestra todos los resultados, utilice la consola de Prometheus.

12.3.4.1. Métricas proporcionadas

El recurso personalizado (CR) **MigrationController** proporciona métricas para el recuento de CR de **MigMigration** y para sus solicitudes de API.

12.3.4.1.1. cam_app_workload_migrations

Esta métrica es un recuento de los CR **MigMigration** a lo largo del tiempo. Es útil para ver junto a las métricas **mtc_client_request_count** y **mtc_client_request_elapsed** a fin de cotejar la información de las solicitudes de la API con los cambios de estado de la migración. Esta métrica está incluida en la telemetría.

Tabla 12.1. cam_app_workload_migrations metric

Nombre de la etiqueta consultable	Ejemplo de valores de la etiqueta	Descripción de la etiqueta
status	running, idle, failed, completed	Estado del RC MigMigration
type	stage, final	Tipo de CR de MigMigration

12.3.4.1.2. mtc_client_request_count

Esta métrica es un recuento acumulado de las solicitudes de la API de Kubernetes que emitió **MigrationController**. No está incluida en la telemetría.

Tabla 12.2. mtc_client_request_count metric

Nombre de la etiqueta consultable	Ejemplo de valores de la etiqueta	Descripción de la etiqueta
cluster	https://migcluster-url:443	Clúster para el que se emitió la solicitud
component	MigPlan, MigCluster	API del subcontrolador que emitió la solicitud
function	(*ReconcileMigPlan).Reconcile	Función desde la que se emitió la solicitud
kind	SecretList, Deployment	Tipo de Kubernetes para el que se emitió la solicitud

12.3.4.1.3. mtc_client_request_elapsed

Esta métrica es una latencia acumulada, en milisegundos, de las solicitudes de la API de Kubernetes que emitió **MigrationController**. No está incluida en la telemetría.

Tabla 12.3. mtc_client_request_elapsed metric

Nombre de la etiqueta consultable	Ejemplo de valores de la etiqueta	Descripción de la etiqueta
cluster	https://cluster-url.com:443	Clúster para el que se emitió la solicitud
component	migplan, migcluster	API del subcontrolador que emitió la solicitud
function	(*ReconcileMigPlan).Reconcile	Función desde la que se emitió la solicitud

Nombre de la etiqueta consultable	Ejemplo de valores de la etiqueta	Descripción de la etiqueta
kind	SecretList, Deployment	Recurso de Kubernetes para el que se emitió la solicitud

12.3.4.1.4. Consultas útiles

En la tabla se enumeran algunas consultas útiles que pueden utilizarse para monitorear el rendimiento.

Tabla 12.4. Consultas útiles

Consulta	Descripción
mtc_client_request_count	Número de solicitudes de API emitidas, clasificadas por tipo de solicitud
sum(mtc_client_request_count)	Número total de solicitudes de API emitidas
mtc_client_request_elapsed	Latencia de las solicitudes de la API, ordenada por tipo de solicitud
sum(mtc_client_request_elapsed)	Latencia total de las solicitudes de la API
sum(mtc_client_request_elapsed) / sum(mtc_client_request_count)	Latencia media de las solicitudes de la API
mtc_client_request_elapsed / mtc_client_request_count	Latencia media de las solicitudes de la API, clasificadas por tipo de solicitud
cam_app_workload_migrations{status="running"} * 100	Recuento de las migraciones en curso, multiplicado por 100 para facilitar su visualización junto al recuento de solicitudes

12.3.5. Uso de la herramienta de recopilación

Puede recopilar registros, métricas e información sobre los recursos personalizados de MTC utilizando la herramienta de **recopilación**.

Los datos **obligatorios** deben adjuntarse a todos los casos de clientes.

Puede recopilar datos para un período de una hora o de 24 horas y ver los datos con la consola de Prometheus.

Requisitos previos

- Debe iniciar sesión en el clúster de OpenShift Container Platform como usuario con el rol de **administrador de clúster**.

- Debe tener instalada la CLI de OpenShift (**oc**).

Procedimiento

1. Navegue hasta el directorio en el que desea almacenar los datos **que deben recopilarse**.
2. Ejecute el comando **oc adm must-gather** para una de las siguientes opciones de recopilación de datos:

- Para recopilar los datos de la última hora:

```
$ oc adm must-gather --image=registry.redhat.io/rhmtc/openshift-migration-must-gather-rhel8:v1.7
```

Los datos se guardan como **must-gather/must-gather.tar.gz**. Puede subir este archivo a un caso de soporte en el [Portal del cliente de Red Hat](#) .

- Para recopilar los datos de las últimas 24 horas:

```
$ oc adm must-gather --image=registry.redhat.io/rhmtc/openshift-migration-must-gather-rhel8:v1.7 \
  -- /usr/bin/gather_metrics_dump
```

Esta operación puede llevar mucho tiempo. Los datos se guardan como **must-gather/metrics/prom_data.tar.gz**.

Visualización de los datos de las métricas con la consola de Prometheus

Puede ver los datos de las métricas con la consola de Prometheus.

Procedimiento

1. Descomprima el archivo **prom_data.tar.gz**:

```
$ tar -xvzf must-gather/metrics/prom_data.tar.gz
```

2. Cree una instancia local de Prometheus:

```
$ make prometheus-run
```

El comando muestra la URL de Prometheus.

Resultado

```
Started Prometheus on http://localhost:9090
```

3. Inicie un navegador web y navegue hasta la URL para ver los datos mediante la consola web de Prometheus.
4. Después de ver los datos, elimine la instancia y los datos de Prometheus:

```
$ make prometheus-cleanup
```

12.3.6. Depuración de los recursos de Velero con la herramienta CLI de Velero

Puede depurar los recursos personalizados (CR) **Backup** y **Restore** y recuperar los registros con la herramienta CLI de Velero.

La herramienta CLI de Velero proporciona información más detallada que la herramienta CLI de OpenShift.

Sintaxis

Utilice el comando **oc exec** para ejecutar un comando CLI de Velero:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \  
-- ./velero <backup_restore_cr> <command> <cr_name>
```

Ejemplo

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \  
-- ./velero backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

Puede especificar **velero-<pod> -n openshift-migration** en lugar de **\$(oc get pods -n openshift-migration -o name | grep velero)**.

Ejemplo

```
$ oc exec velero-<pod> -n openshift-migration -- ./velero backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

Opción de ayuda

Utilice la opción **velero --help** para enumerar todos los comandos de la CLI de Velero:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) -- ./velero --help
```

Comando Describe

Utilice el comando **velero describe** para recuperar un resumen de las advertencias y los errores asociados a los CR **Backup** o **Restore**:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \  
-- ./velero <backup_restore_cr> describe <cr_name>
```

Ejemplo

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \  
-- ./velero backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

Comando Logs

Utilice el comando **velero logs** para recuperar los registros del CR **Backup** o **Restore**:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \  
-- ./velero <backup_restore_cr> logs <cr_name>
```

Ejemplo

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) \  
-- ./velero restore logs ccc7c2d0-6017-11eb-afab-85d0007f5a19-x4lbf
```

12.3.7. Depuración de un fallo de migración parcial

Puede depurar un mensaje de advertencia de fallo de migración parcial utilizando la CLI de Velero para examinar los registros del recurso personalizado (CR) **Restore**.

Un fallo parcial se produce cuando Velero encuentra un problema que no hace que la migración falle. Por ejemplo, si falta una definición de recurso personalizada (CRD) o si hay una discrepancia entre las versiones de CRD en los clústeres de origen y de destino, la migración se completa pero el CR no se crea en el clúster de destino.

Velero registra el problema como un fallo parcial y, luego, procesa el resto de los objetos en el CR **Backup**.

Procedimiento

1. Compruebe el estado del CR **MigMigration**:

```
$ oc get migmigration <migmigration> -o yaml
```

Ejemplo de salida

```
status:
  conditions:
  - category: Warn
    durable: true
    lastTransitionTime: "2021-01-26T20:48:40Z"
    message: 'Final Restore openshift-migration/ccc7c2d0-6017-11eb-afab-85d0007f5a19-
x4lbf: partially failed on destination cluster'
    status: "True"
    type: VeleroFinalRestorePartiallyFailed
  - category: Advisory
    durable: true
    lastTransitionTime: "2021-01-26T20:48:42Z"
    message: The migration has completed with warnings, please look at `Warn` conditions.
    reason: Completed
    status: "True"
    type: SucceededWithWarnings
```

2. Compruebe el estado del CR **Restore** mediante el comando **describe** de Velero:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) -n openshift-migration -
- ./velero restore describe <restore>
```

Ejemplo de salida

```
Phase: PartiallyFailed (run 'velero restore logs ccc7c2d0-6017-11eb-afab-85d0007f5a19-
x4lbf' for more information)

Errors:
  Velero: <none>
  Cluster: <none>
  Namespaces:
    migration-example: error restoring example.com/migration-example/migration-example:
the server could not find the requested resource
```

3. Compruebe los registros del CR **Restore** utilizando el comando **logs** Velero:

```
$ oc exec $(oc get pods -n openshift-migration -o name | grep velero) -n openshift-migration -
- ./velero restore logs <restore>
```

Ejemplo de salida

```
time="2021-01-26T20:48:37Z" level=info msg="Attempting to restore migration-example:
migration-example" logSource="pkg/restore/restore.go:1107" restore=openshift-
migration/ccc7c2d0-6017-11eb-afab-85d0007f5a19-x4lbf
time="2021-01-26T20:48:37Z" level=info msg="error restoring migration-example: the server
could not find the requested resource" logSource="pkg/restore/restore.go:1170"
restore=openshift-migration/ccc7c2d0-6017-11eb-afab-85d0007f5a19-x4lbf
```

El mensaje de error de registro del CR **Restore**, **el servidor no pudo encontrar el recurso solicitado**, indica la causa de la migración parcialmente fallida.

12.3.8. Uso de los recursos personalizados de MTC para la resolución de problemas

Puede comprobar los siguientes recursos personalizados (CR) de Migration Toolkit for Containers (MTC) para solucionar una migración fallida:

- **MigCluster**
- **MigStorage**
- **MigPlan**

- **BackupStorageLocation**

El CR **BackupStorageLocation** contiene la etiqueta **migrationcontroller** para identificar la instancia de MTC que creó el CR:

```
labels:
  migrationcontroller: ebe13bee-c803-47d0-a9e9-83f380328b93
```

- **VolumeSnapshotLocation**

El CR **VolumeSnapshotLocation** contiene la etiqueta **migrationcontroller** para identificar la instancia de MTC que creó el CR:

```
labels:
  migrationcontroller: ebe13bee-c803-47d0-a9e9-83f380328b93
```

- **MigMigration**

- **Backup**

MTC cambia la política de recuperación de los volúmenes persistentes (PV) migrados a **Retain** en el clúster de destino. El CR **Backup** contiene la anotación **openshift.io/orig-reclaim-policy** que indica la política de recuperación original. Puede restaurar manualmente la política de recuperación de los PV migrados.

- **Restore**

Procedimiento

1. Enumera los CR de **MigMigration** en el espacio de nombres **openshift-migration**:

```
$ oc get migmigration -n openshift-migration
```

Ejemplo de salida

```
NAME                                AGE
88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s
```

2. Inspeccione el CR **MigMigration**:

```
$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
```

El resultado es similar a los siguientes ejemplos.

Ejemplo de resultado de MigMigration

```
name:      88435fe0-c9f8-11e9-85e6-5d593ce65e10
namespace: openshift-migration
labels:    <none>
annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
apiVersion: migration.openshift.io/v1alpha1
kind:      MigMigration
metadata:
  creationTimestamp: 2019-08-29T01:01:29Z
  generation:       20
  resourceVersion:  88179
  selfLink:         /apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
  uid:              8886de4c-c9f8-11e9-95ad-0205fe66cbb6
spec:
  migPlanRef:
    name:      socks-shop-mig-plan
    namespace: openshift-migration
  quiescePods: true
  stage:      false
status:
  conditions:
    category:      Advisory
    durable:       True
    lastTransitionTime: 2019-08-29T01:03:40Z
    message:       The migration has completed successfully.
    reason:        Completed
    status:        True
    type:          Succeeded
  phase:         Completed
  startTimestamp: 2019-08-29T01:01:29Z
  events:        <none>
```

Ejemplo de resultado de la copia de seguridad del CR #2 de Velero que describe los datos del PV

```
apiVersion: velero.io/v1
kind: Backup
```

```

metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.105.179:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-0205fe66cbb6
    openshift.io/orig-reclaim-policy: delete
  creationTimestamp: "2019-08-29T01:03:15Z"
  generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
  generation: 1
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    velero.io/storage-location: myrepo-vpzq9
  name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  namespace: openshift-migration
  resourceVersion: "87313"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:
  excludedNamespaces: []
  excludedResources: []
  hooks:
    resources: []
  includeClusterResources: null
  includedNamespaces:
  - sock-shop
  includedResources:
  - persistentvolumes
  - persistentvolumeclaims
  - namespaces
  - imagestreams
  - imagestreamtags
  - secrets
  - configmaps
  - pods
  labelSelector:
    matchLabels:
      migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
  storageLocation: myrepo-vpzq9
  ttl: 720h0m0s
  volumeSnapshotLocations:
  - myrepo-wv6fx
status:
  completionTimestamp: "2019-08-29T01:02:36Z"
  errors: 0
  expiration: "2019-09-28T01:02:35Z"
  phase: Completed
  startTimestamp: "2019-08-29T01:02:35Z"
  validationErrors: null
  version: 1

```

```

volumeSnapshotsAttempted: 0
volumeSnapshotsCompleted: 0
warnings: 0

```

Ejemplo de resultado del CR #2 de Velero que describe los recursos de Kubernetes

```

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
  generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
    migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
  name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  namespace: openshift-migration
  resourceVersion: "82329"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
    - nodes
    - events
    - events.events.k8s.io
    - backups.velero.io
    - restores.velero.io
    - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
  namespaceMapping: null
  restorePVs: true
status:
  errors: 0
  failureReason: ""
  phase: Completed
  validationErrors: null
  warnings: 15

```

12.4. PREOCUPACIONES Y PROBLEMAS COMUNES

En esta sección se describen los problemas más comunes que pueden causar inconvenientes durante la migración.

12.4.1. Actualización de imágenes internas obsoletas

Si su aplicación usa imágenes del espacio de nombres **openshift**, las versiones necesarias de las imágenes deben estar presentes en el clúster de destino.

Si una imagen de OpenShift Container Platform 3 es obsoleta en OpenShift Container Platform 4.10, puede actualizar manualmente la etiqueta de flujo de imágenes con **podman**.

Requisitos previos

- Debe tener instalado **Podman**.
- Debe iniciar la sesión como usuario con privilegios de **administrador de clúster**.
- Si utiliza registros inseguros, añada los valores de su host de registro a la sección **[registries.insecure]** de **/etc/container/registries.conf** para asegurarse de que **podman** no encuentre un error de verificación de TLS.
- Los registros internos deben estar expuestos en los clústeres de origen y destino.

Procedimiento

1. Asegúrese de que los registros internos estén expuestos en los clústeres de OpenShift Container Platform 3 y 4.

El registro interno está expuesto por defecto en OpenShift Container Platform 4.

2. Si utiliza registros inseguros, añada los valores de su host de registro a la sección **[registries.insecure]** de **/etc/container/registries.conf** para asegurarse de que **podman** no encuentre un error de verificación de TLS.

3. Inicie sesión en el registro de OpenShift Container Platform 3:

```
$ podman login -u $(oc whoami) -p $(oc whoami -t) --tls-verify=false <registry_url>:<port>
```

4. Inicie sesión en el registro de OpenShift Container Platform 4:

```
$ podman login -u $(oc whoami) -p $(oc whoami -t) --tls-verify=false <registry_url>:<port>
```

5. Extraiga la imagen de OpenShift Container Platform 3:

```
$ podman pull <registry_url>:<port>/openshift/<image>
```

6. Etiquete la imagen de OpenShift Container Platform 3 para el registro de OpenShift Container Platform 4:

```
$ podman tag <registry_url>:<port>/openshift/<image> \ 1  
  <registry_url>:<port>/openshift/<image> 2
```

- 1** Especifique la URL del registro y el puerto para el clúster de OpenShift Container Platform 3.

- 2** Especifique la URL del registro y el puerto para el clúster de OpenShift Container Platform 4.

7. Inserte la imagen en el registro de OpenShift Container Platform 4:

```
$ podman push <registry_url>:<port>/openshift/<image> ❶
```

❶ Especifique el clúster de OpenShift Container Platform 4.

8. Compruebe que la imagen tenga un flujo de imagen válido:

```
$ oc get imagestream -n openshift | grep <image>
```

Ejemplo de salida

```
NAME      IMAGE REPOSITORY          TAGS  UPDATED
my_image  image-registry.openshift-image-registry.svc:5000/openshift/my_image  latest  32
seconds ago
```

12.4.2. Migración directa de volúmenes incompleta

Si la migración directa de volúmenes no se completa, es posible que el clúster de destino no tenga las mismas anotaciones **node-selector** que el clúster de origen.

Migration Toolkit for Containers (MTC) migra los espacios de nombres con todas las anotaciones para preservar las restricciones de contexto de seguridad y los requisitos de programación. Durante la migración directa de volúmenes, MTC crea pods de transferencia de Rsync en el clúster de destino en los espacios de nombres que se migraron desde el clúster de origen. Si un espacio de nombres del clúster de destino no tiene las mismas anotaciones que el espacio de nombres del clúster de origen, no se pueden programar los pods de transferencia de Rsync. Los pods de Rsync permanecen en el estado **Pending** (Pendiente).

Puede identificar y solucionar este problema realizando el siguiente procedimiento.

Procedimiento

1. Compruebe el estado del CR **MigMigration**:

```
$ oc describe migmigration <pod> -n openshift-migration
```

El resultado incluye el siguiente mensaje de estado:

Ejemplo de salida

```
Some or all transfer pods are not running for more than 10 mins on destination cluster
```

2. En el clúster de origen, obtenga los detalles del espacio de nombres migrado:

```
$ oc get namespace <namespace> -o yaml ❶
```

❶ Especifique el espacio de nombres migrado.

3. En el clúster de destino, edite el espacio de nombres migrado:

```
$ oc edit namespace <namespace>
```

- Añada las anotaciones **openshift.io/node-selector** que faltan al espacio de nombres migrado, como en el siguiente ejemplo:

```
apiVersion: v1
kind: Namespace
metadata:
  annotations:
    openshift.io/node-selector: "region=east"
...
```

- Vuelva a ejecutar el plan de migración.

12.4.3. Mensajes de error y resoluciones

Esta sección describe los mensajes de error más comunes que puede encontrar en Migration Toolkit for Containers (MTC) y cómo resolver sus causas subyacentes.

12.4.3.1. Aparece un error de certificado de CA al acceder a la consola de MTC por primera vez

Si se muestra un mensaje de **error de certificado de CA** la primera vez que se intenta acceder a la consola de MTC, la causa probable es el uso de certificados de CA autofirmados en uno de los clústeres.

Para resolver este problema, navegue hasta la URL **oauth-authorization-server** que aparece en el mensaje de error y acepte el certificado. Para resolver este problema de forma permanente, añada el certificado al almacén de confianza de su navegador web.

Si aparece el mensaje de **No autorizado** después de haber aceptado el certificado, navegue hasta la consola de MTC y actualice la página web.

12.4.3.2. Error de tiempo de espera de OAuth en la consola de MTC

Si en la consola de MTC aparece un mensaje que indica que **la conexión se ha interrumpido** después de haber aceptado un certificado autofirmado, es probable que las causas sean las siguientes:

- Acceso de red interrumpido al servidor OAuth.
- Acceso de red interrumpido a la consola de OpenShift Container Platform.
- Configuración del proxy que bloquea el acceso a la URL **oauth-authorization-server**. Consulte [Consola de MTC inaccesible debido a un error de tiempo de espera de OAuth](#) para obtener más detalles.

Para determinar la causa del tiempo de espera:

- Inspeccione la página web de la consola de MTC con un inspector web del navegador.
- Compruebe si hay errores en el registro del pod de **la interfaz de usuario de la migración**.

12.4.3.3. Error de certificado firmado por una autoridad desconocida

Si utiliza un certificado autofirmado para proteger un clúster o un repositorio de replicación para Migration Toolkit for Containers (MTC), la verificación del certificado puede fallar con el siguiente mensaje de error: **Certificado firmado por una autoridad desconocida**.

Puede crear un archivo de paquetes de certificados de CA personalizado y cargarlo en la consola web de MTC cuando añada un clúster o un repositorio de replicación.

Procedimiento

Descargue un certificado de CA desde un terminal remoto y guárdelo como archivo de paquete de CA:

```
$ echo -n | openssl s_client -connect <host_FQDN>:<port> \ 1
| sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > <ca_bundle.cert> 2
```

- 1 Especifique el FQDN del host y el puerto del terminal, por ejemplo, **api.mi-cluster.ejemplo.com:6443**.
- 2 Especifique el nombre del archivo del paquete de CA.

12.4.3.4. Errores de ubicación de la copia de seguridad en el registro del pod de Velero

Si el recurso personalizado **Backup** de **Velero** contiene una referencia a una ubicación de almacenamiento de copias de seguridad (BSL) que no existe, el registro del pod de **Velero** podría mostrar los siguientes mensajes de error:

```
$ oc logs <MigrationUI_Pod> -n openshift-migration
```

Puede ignorar estos mensajes de error. La falta de una BSL no puede hacer fracasar una migración.

12.4.3.5. Error de tiempo de espera del volumen del pod en el registro del pod de Velero

Si una migración falla porque Restic se agota, se muestra el siguiente error en el registro del pod de **Velero**.

```
level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all PodVolumeBackups to complete"
error.file="/go/src/github.com/heptio/velero/pkg/restic/backupper.go:165"
error.function="github.com/heptio/velero/pkg/restic.(*backupper).BackupPodVolumes" group=v1
```

El valor por defecto de **restic_timeout** es de una hora. Puede aumentar este parámetro para migraciones grandes, teniendo en cuenta que un valor más alto puede retrasar el retorno de los mensajes de error.

Procedimiento

1. En la consola web de OpenShift Container Platform, navegue hasta **Operators** (Operadores) → **Installed Operators** (Operadores instalados).
2. Haga clic en el **operador de Migration Toolkit for Containers**
3. En la pestaña **MigrationController**, haga clic en **migration-controller**.
4. En la pestaña **YAML**, actualice el siguiente valor del parámetro:

```
spec:
  restic_timeout: 1h 1
```

- 1** Las unidades válidas son **h** (horas), **m** (minutos) y **s** (segundos), por ejemplo, **3h30m15s**.

5. Haga clic en **Save** (Guardar).

12.4.3.6. Errores de verificación de Restic en el recurso personalizado MigMigration

Si la verificación de datos falla al migrar un volumen persistente con el método de copia de datos de sistemas de archivos, se muestra el siguiente error en el CR de **MigMigration**.

Ejemplo de salida

```
status:
  conditions:
  - category: Warn
    durable: true
    lastTransitionTime: 2020-04-16T20:35:16Z
    message: There were verify errors found in 1 Restic volume restores. See restore `registry-
example-migration-rvwcm>`
    for details 1
    status: "True"
    type: ResticVerifyErrors 2
```

- 1** El mensaje de error identifica el nombre de CR de **Restore**.
- 2** **ResticVerifyErrors** es un tipo de aviso de error general que incluye errores de verificación.



NOTA

Un error de verificación de datos no hace que el proceso de migración falle.

Puede comprobar el CR **Restore** para identificar el origen del error de verificación de datos.

Procedimiento

1. Inicie sesión en el clúster de destino.
2. Vea el CR **Restore**:

```
$ oc describe <registry-example-migration-rvwcm> -n openshift-migration
```

El resultado identifica el volumen persistente con errores **PodVolumeRestore**.

Ejemplo de salida

```
status:
  phase: Completed
  podVolumeRestoreErrors:
  - kind: PodVolumeRestore
```



```

name: <registry-example-migration-rvwcm-98t49>
namespace: openshift-migration
podVolumeRestoreResticErrors:
- kind: PodVolumeRestore
  name: <registry-example-migration-rvwcm-98t49>
  namespace: openshift-migration

```

3. Vea el CR **PodVolumeRestore**:

```
$ oc describe <migration-example-rvwcm-98t49>
```

El resultado identifica el pod de **Restic** que registró los errores.

Ejemplo de salida

```

completionTimestamp: 2020-05-01T20:49:12Z
errors: 1
resticErrors: 1
...
resticPod: <restic-nr2v5>

```

4. Vea el registro del pod de **Restic** para localizar los errores:

```
$ oc logs -f <restic-nr2v5>
```

12.4.3.7. Error de permiso de Restic al migrar desde un almacenamiento del NFS con **root_squash** activado

Si está migrando datos desde un almacenamiento del NFS y **root_squash** está activado, **Restic** se asigna a **nfsnobody** y no tiene permiso para realizar la migración. El siguiente error aparece en el registro del pod de **Restic**.

Ejemplo de salida

```

backup=openshift-migration/<backup_id> controller=pod-volume-backup error="fork/exec
/usr/bin/restic: permission denied" error.file="/go/src/github.com/vmware-
tanzu/velero/pkg/controller/pod_volume_backup_controller.go:280"
error.function="github.com/vmware-tanzu/velero/pkg/controller.
(*podVolumeBackupController).processBackup"
logSource="pkg/controller/pod_volume_backup_controller.go:280" name=<backup_id>
namespace=openshift-migration

```

Puede resolver este problema creando un grupo suplementario para Restic y añadiendo el ID del grupo al manifiesto del CR de **MigrationController**.

Procedimiento

1. Cree un grupo suplementario para Restic en el almacenamiento del NFS.
2. Establezca el bit **setgid** en los directorios del NFS para que se herede la propiedad del grupo.
3. Añade el parámetro **restic_supplemental_groups** al manifiesto del CR de **MigrationController** en los clústeres de origen y destino:

■

```
spec:
  restic_supplemental_groups: <group_id> 1
```

- 1 Especifique el ID del grupo suplementario.

4. Espere a que se reinicien los pods de **Restic** para que se apliquen los cambios.

12.4.4. Problemas conocidos

Esta versión tiene los siguientes problemas conocidos:

- Durante la migración, Migration Toolkit for Containers (MTC) conserva las siguientes anotaciones del espacio de nombres:
 - **openshift.io/sa.scc.mcs**
 - **openshift.io/sa.scc.supplemental-groups**
 - **openshift.io/sa.scc.uid-range**

Estas anotaciones conservan el rango de UID, asegurando que los contenedores conserven sus permisos del sistema de archivos en el clúster de destino. Existe el riesgo de que los UID migrados puedan duplicar los UID dentro de un espacio de nombres existente o futuro en el clúster de destino. ([BZ#1748440](#))
- La mayoría de los recursos de los clústeres aún no son gestionados por MTC. Si sus aplicaciones requieren recursos en el clúster, es posible que tenga que crearlos manualmente en el clúster de destino.
- Si una migración falla, el plan de migración no conserva las configuraciones personalizadas del PV para los pods en reposo. Debe retrotraer manualmente la migración, eliminar el plan de migración y crear un nuevo plan de migración con su configuración de FV. ([BZ#1784899](#))
- Si una migración grande falla porque Restic se agota, puede aumentar el valor del parámetro **restic_timeout** (por defecto: **1h**) en el manifiesto del recurso personalizado (CR) de **MigrationController**.
- Si selecciona la opción de verificación de datos para los PV que se migran con el método de copia de sistemas de archivos, el rendimiento es significativamente más lento.
- Si está migrando datos desde un almacenamiento del NFS y **root_squash** está activado, **Restic** se asigna a **nfsnobody**. La migración falla y se muestra un error de permiso en el registro del pod de **Restic**. ([BZ#1873641](#))

Puede resolver este problema añadiendo grupos suplementarios para **Restic** al manifiesto del CR de **MigrationController**:

```
spec:
  ...
  restic_supplemental_groups:
  - 5555
  - 6666
```

- Si ejecuta la migración directa de volúmenes con nodos que están en diferentes zonas de disponibilidad o conjuntos de disponibilidad, la migración podría fallar porque los pods migrados no pueden acceder a la PVC. ([BZ#1947487](#))

12.5. RETROCESO DE UNA MIGRACIÓN

Puede retrotraer una migración con la consola web de MTC o la CLI.

También puede [retrotraer una migración manualmente](#).

12.5.1. Retroceso de una migración mediante la consola web de MTC

Puede retrotraer una migración en la consola web de Migration Toolkit for Containers (MTC).



NOTA

Los siguientes recursos permanecen en los espacios de nombres migrados para la depuración después de una migración directa de volúmenes (DVM) fallida:

- Mapas de configuración (clústeres de origen y destino)
- Objetos **Secret** (clústeres de origen y destino)
- CR **Rsync** (clúster de origen)


Estos recursos no afectan al retroceso. Puede eliminarlos manualmente.

Si posteriormente se ejecuta el mismo plan de migración con éxito, los recursos de la migración fallida se eliminan automáticamente.

Si su aplicación se detuvo durante una migración fallida, debe retrotraer la migración para evitar la corrupción de datos en el volumen persistente.

El retroceso no es necesario si la aplicación no se detuvo durante la migración porque la aplicación original sigue ejecutándose en el clúster de origen.

Procedimiento

1. En la consola web de MTC, haga clic en **Migration plans** (Planes de migración).
2. Haga clic en el menú Options (Opciones)  al lado del plan de migración y seleccione **Rollback** (Retrotraer) en **Migration** (Migración).
3. Haga clic en **Rollback** (Retrotraer) y espere a que se complete el retroceso.
En los detalles del plan de migración, se muestra **Rollback succeeded** (Retroceso exitoso).
4. Compruebe que el retroceso se haya realizado correctamente en la consola web de OpenShift Container Platform del clúster de origen:
 - a. Haga clic en **Home** (Inicio) → **Projects** (Proyectos).
 - b. Haga clic en el proyecto migrado para ver su estado.
 - c. En la sección **Routes** (Rutas), haga clic en **Location** (Ubicación) para verificar que la aplicación esté funcionando, si es el caso.
 - d. Haga clic en **Workloads** (Cargas de trabajo) → **Pods** para comprobar que los pods se ejecutan en el espacio de nombres migrado.

- e. Haga clic en **Storage** (Almacenamiento) → **Persistent volumes** (Volúmenes persistentes) para verificar que los volúmenes persistentes migrados estén correctamente aprovisionados.

12.5.2. Retroceso de una migración desde la interfaz de línea de comandos

Puede retrotraer una migración creando el recurso personalizado (CR) **MigMigration** desde la interfaz de línea de comandos.



NOTA

Los siguientes recursos permanecen en los espacios de nombres migrados para la depuración después de una migración directa de volúmenes (DVM) fallida:

- Mapas de configuración (clústeres de origen y destino)
- Objetos **Secret** (clústeres de origen y destino)
- CR **Rsync** (clúster de origen)

Estos recursos no afectan al retroceso. Puede eliminarlos manualmente.

Si posteriormente se ejecuta el mismo plan de migración con éxito, los recursos de la migración fallida se eliminan automáticamente.

Si su aplicación se detuvo durante una migración fallida, debe retrotraer la migración para evitar la corrupción de datos en el volumen persistente.

El retroceso no es necesario si la aplicación no se detuvo durante la migración porque la aplicación original sigue ejecutándose en el clúster de origen.

Procedimiento

1. Cree un CR **MigMigration** basado en el siguiente ejemplo:

```
$ cat << EOF | oc apply -f -
apiVersion: migration.openshift.io/v1alpha1
kind: MigMigration
metadata:
  labels:
    controller-tools.k8s.io: "1.0"
  name: <migmigration>
  namespace: openshift-migration
spec:
  ...
  rollback: true
  ...
  migPlanRef:
    name: <migplan> 1
    namespace: openshift-migration
EOF
```

- 1 Especifique el nombre del CR **MigPlan** asociado.

2. En la consola web de MTC, compruebe que los recursos del proyecto migrado se hayan eliminado del clúster de destino.
3. Compruebe que los recursos del proyecto migrado estén presentes en el clúster de origen y que la aplicación se esté ejecutando.

12.5.3. Retroceso manual de una migración

Puede retrotraer una migración fallida manualmente borrando los pods de la **etapa** y desinstalando la aplicación.

Si ejecuta el mismo plan de migración con éxito, los recursos de la migración fallida se eliminan automáticamente.



NOTA

Los siguientes recursos permanecen en los espacios de nombres migrados después de una migración directa de volúmenes (DVM) fallida:

- Mapas de configuración (clústeres de origen y destino)
- Objetos **Secret** (clústeres de origen y destino)
- CR **Rsync** (clúster de origen)

Estos recursos no afectan al retroceso. Puede eliminarlos manualmente.

Procedimiento

1. Elimine los pods de la **etapa** en todos los clústeres:

```
$ oc delete $(oc get pods -l migration.openshift.io/is-stage-pod -n <namespace>) 1
```

- 1** Espacios de nombres especificados en el CR **MigPlan**.

2. Desactive la aplicación en el clúster de origen escalando las réplicas a su número antes de la migración:

```
$ oc scale deployment <deployment> --replicas=<premigration_replicas>
```

La anotación **migration.openshift.io/preQuiesceReplicas** en el CR **Deployment** muestra el número de réplicas antes de la migración:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  annotations:
    deployment.kubernetes.io/revision: "1"
    migration.openshift.io/preQuiesceReplicas: "1"
```

3. Verifique que los pods de la aplicación se ejecuten en el clúster de origen:

```
$ oc get pod -n <namespace>
```

Recursos adicionales

- [Eliminación de operadores de un clúster mediante la consola web](#)