Migration Toolkit for Virtualization 2.5

Installing and using the Migration Toolkit for Virtualization

Migrating from VMware vSphere or Red Hat Virtualization to Red Hat OpenShift Virtualization

Last Updated: 2024-07-29

Migration Toolkit for Virtualization 2.5 Installing and using the Migration Toolkit for Virtualization

Migrating from VMware vSphere or Red Hat Virtualization to Red Hat OpenShift Virtualization

Red Hat Modernization and Migration Documentation Team
ccs-mms-docs@redhat.com

Legal Notice

Abstract

The Migration Toolkit for Virtualization (MTV) enables you to migrate virtual machines from VMware vSphere, Red Hat Virtualization, or OpenStack to OpenShift Virtualization running on Red Hat OpenShift.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message.

CHAPTER 1. ABOUT THE MIGRATION TOOLKIT FOR VIRTUALIZATION

You can use the Migration Toolkit for Virtualization (MTV) to migrate virtual machines from the following source providers to OpenShift Virtualization destination providers:

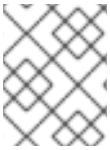
- VMware vSphere
- Red Hat Virtualization (RHV)
- OpenStack
- Open Virtual Appliances (OVAs) that were created by VMware vSphere
- Remote OpenShift Virtualization clusters

Migration using one or more Open Virtual Appliance (OVA) files as a source provider is a Technology Preview.

IMPORTANT

Migration using one or more Open Virtual Appliance (OVA) files as a source provider is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see https://access.redhat.com/support/offerings/techpreview/.



NOTE

Migration using OpenStack source providers only supports VMs that use only Cinder volumes.

Additional resources

- Performance recommendations for migrating from VMware vSphere to OpenShift Virtualization .
- Performance recommendations for migrating from Red Hat Virtualization to OpenShift Virtualization.

1.1. ABOUT COLD AND WARM MIGRATION

MTV supports cold migration from:

- VMware vSphere
- Red Hat Virtualization (RHV)
- OpenStack
- Remote OpenShift Virtualization clusters

MTV supports warm migration from VMware vSphere and from RHV.

NOTE

Migration using OpenStack source providers only supports VMs that use only Cinder volumes.

1.1.1. Cold migration

Cold migration is the default migration type. The source virtual machines are shut down while the data is copied.

1.1.2. Warm migration

Most of the data is copied during the *precopy* stage while the source virtual machines (VMs) are running.

Then the VMs are shut down and the remaining data is copied during the *cutover* stage.

Precopy stage

The VMs are not shut down during the precopy stage.

The VM disks are copied incrementally using changed block tracking (CBT) snapshots. The snapshots are created at one-hour intervals by default. You can change the snapshot interval by updating the **forklift-controller** deployment.

IMPORTANT

You must enable CBT for each source VM and each VM disk.

A VM can support up to 28 CBT snapshots. If the source VM has too many CBT snapshots and the **Migration Controller** service is not able to create a new snapshot, warm migration might fail. The **Migration Controller** service deletes each snapshot when the snapshot is no longer required.

The precopy stage runs until the cutover stage is started manually or is scheduled to start.

Cutover stage

The VMs are shut down during the cutover stage and the remaining data is migrated. Data stored in RAM is not migrated.

You can start the cutover stage manually by using the MTV console or you can schedule a cutover time in the **Migration** manifest.

CHAPTER 2. PREREQUISITES

Review the following prerequisites to ensure that your environment is prepared for migration.

2.1. SOFTWARE REQUIREMENTS

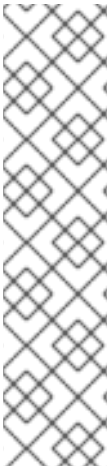
You must install compatible versions of Red Hat OpenShift and OpenShift Virtualization.

2.2. STORAGE SUPPORT AND DEFAULT MODES

MTV uses the following default volume and access modes for supported storage.

Table 2.1. Default volume and access modes

Provisioner	Volume mode	Access mode
kubernetes.io/aws-ebs	Block	ReadWriteOnce
kubernetes.io/azure-disk	Block	ReadWriteOnce
kubernetes.io/azure-file	Filesystem	ReadWriteMany
kubernetes.io/cinder	Block	ReadWriteOnce
kubernetes.io/gce-pd	Block	ReadWriteOnce
kubernetes.io/hostpath-provisioner	Filesystem	ReadWriteOnce
manila.csi.openstack.org	Filesystem	ReadWriteMany
openshift-storage.cephfs.csi.ceph.com	Filesystem	ReadWriteMany
openshift-storage.rbd.csi.ceph.com	Block	ReadWriteOnce
kubernetes.io/rbd	Block	ReadWriteOnce
kubernetes.io/vsphere-volume	Block	ReadWriteOnce



NOTE

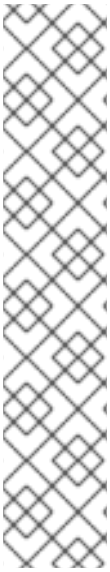
If the OpenShift Virtualization storage does not support dynamic provisioning, you must apply the following settings:

- **Filesystem** volume mode
  Filesystem volume mode is slower than **Block** volume mode.
- **ReadWriteOnce** access mode
  ReadWriteOnce access mode does not support live virtual machine migration.

See Enabling a statically-provisioned storage class for details on editing the storage profile.

NOTE

If your migration uses block storage and persistent volumes created with an EXT4 file system, increase the file system overhead in CDI to be more than 10%. The default overhead that is assumed by CDI does not completely include the reserved place for the root partition. If you do not increase the file system overhead in CDI by this amount, your migration might fail.



NOTE

When migrating from OpenStack or running a cold-migration from RHV to the OCP cluster that MTV is deployed on, the migration allocates persistent volumes without CDI. In these cases, you might need to adjust the file system overhead.

If the configured file system overhead, which has a default value of 10%, is too low, the disk transfer will fail due to lack of space. In such a case, you would want to increase to increase the file system overhead.

In some cases, however, you might want to decrease the file system overhead to reduce storage consumption.

You can change the file system overhead by changing the value of the **controller_filesystem_overhead** in the **spec** portion of the **forklift-controller** CR, as described in Configuring the MTV Operator.

2.3. NETWORK PREREQUISITES

The following prerequisites apply to all migrations:

- IP addresses, VLANs, and other network configuration settings must not be changed before or during migration. The MAC addresses of the virtual machines are preserved during migration.
- The network connections between the source environment, the OpenShift Virtualization cluster, and the replication repository must be reliable and uninterrupted.
- If you are mapping more than one source and destination network, you must create a network attachment definition for each additional destination network.

2.3.1. Ports

The firewalls must enable traffic over the following ports:

Table 2.2. Network ports required for migrating from VMware vSphere

Port	Protocol	Source	Destination	Purpose
443	TCP	OpenShift nodes	VMware vCenter	VMware provider inventory Disk transfer authentication
443	TCP	OpenShift nodes	VMware ESXi hosts	Disk transfer authentication
902	TCP	OpenShift nodes	VMware ESXi hosts	Disk transfer data copy

Table 2.3. Network ports required for migrating from Red Hat Virtualization

Port	Protocol	Source	Destination	Purpose
443	TCP	OpenShift nodes	RHV Engine	RHV provider inventory Disk transfer authentication
443	TCP	OpenShift nodes	RHV hosts	Disk transfer authentication
54322	TCP	OpenShift nodes	RHV hosts	Disk transfer data copy

2.4. SOURCE VIRTUAL MACHINE PREREQUISITES

The following prerequisites apply to all migrations:

- ISO/CDROM disks must be unmounted.
- Each NIC must contain one IPv4 and/or one IPv6 address.
- The VM operating system must be certified and supported for use as a guest operating system with OpenShift Virtualization.
- VM names must contain only lowercase letters (**a-z**), numbers (**0-9**), or hyphens (**-**), up to a maximum of 253 characters. The first and last characters must be alphanumeric. The name must not contain uppercase letters, spaces, periods (**.**), or special characters.
- VM names must not duplicate the name of a VM in the OpenShift Virtualization environment.

NOTE

Migration Toolkit for Virtualization automatically assigns a new name to a VM that does not comply with the rules.

Migration Toolkit for Virtualization makes the following changes when it automatically generates a new VM name:

- Excluded characters are removed.
- Uppercase letters are switched to lowercase letters.
- Any underscore (_) is changed to a dash (-).

This feature allows a migration to proceed smoothly even if someone entered a VM name that does not follow the rules.

2.5. RED HAT VIRTUALIZATION PREREQUISITES

The following prerequisites apply to Red Hat Virtualization migrations:

- You must use a compatible version of Red Hat Virtualization.
- You must have the Manager CA certificate, unless it was replaced by a third-party certificate, in which case, specify the Manager Apache CA certificate.
  You can obtain the Manager CA certificate by navigating to **https://<engine_host>/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA** in a browser.
- If you are migrating a virtual machine with a direct LUN disk, ensure that the nodes in the OpenShift Virtualization destination cluster that the VM is expected to run on can access the backend storage.

NOTE

- Unlike disk images that *are copied* from a source provider to a target provider, LUNs are *detached*, *but not removed*, from virtual machines in the source provider and then attached to the virtual machines (VMs) that are created in the target provider.
- LUNs are not removed from the source provider during the migration in case fallback to the source provider is required. However, before re-attaching the LUNs to VMs in the source provider, ensure that the LUNs are not used by VMs on the target environment at the same time, which might lead to data corruption.
- Migration of Fibre Channel LUNs is not supported.

2.6. OPENSTACK PREREQUISITES

The following prerequisites apply to OpenStack migrations:

- You must use a compatible version of OpenStack.

NOTE

Migration using OpenStack source providers only supports VMs that use only Cinder volumes.

2.6.1. Additional authentication methods for migrations with OpenStack source providers

MTV versions 2.5 and later support the following authentication methods for migrations with OpenStack source providers in addition to the standard username and password credential set:

- Token authentication
- Application credential authentication

You can use these methods to migrate virtual machines with OpenStack source providers using the CLI the same way you migrate other virtual machines, except for how you prepare the **Secret** manifest.

2.6.1.1. Using token authentication with an OpenStack source provider

You can use token authentication, instead of username and password authentication, when you create an OpenStack source provider.

MTV supports both of the following types of token authentication:

- Token with user ID
- Token with user name

For each type of token authentication, you need to use data from OpenStack to create a **Secret** manifest.

Prerequisites

Have an OpenStack account.

Procedure

1. In the dashboard of the OpenStack web console, click **Project > API Access**.
2. Expand **Download OpenStack RC file** and click **OpenStack RC file**.
   The file that is downloaded, referred to here as **<openstack_rc_file>**, includes the following fields used for token authentication:

   ```
   OS_AUTH_URL
   OS_PROJECT_ID
   OS_PROJECT_NAME
   OS_DOMAIN_NAME
   OS_USERNAME
   ```

3. To get the data needed for token authentication, run the following command:

   ```
   $ openstack token issue
   ```

The output, referred to here as **<openstack_token_output>**, includes the **token**, **userID**, and **projectID** that you need for authentication using a token with user ID.

4. Create a **Secret** manifest similar to the following:

   - For authentication using a token with user ID:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: openstack-secret-tokenid
  namespace: openshift-mtv
  labels:
    createdForProviderType: openstack
type: Opaque
stringData:
  authType: token
  token: <token_from_openstack_token_output>
  projectID: <projectID_from_openstack_token_output>
  userID: <userID_from_openstack_token_output>
  url: <OS_AUTH_URL_from_openstack_rc_file>
EOF
```

   - For authentication using a token with user name:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: openstack-secret-tokenname
  namespace: openshift-mtv
  labels:
    createdForProviderType: openstack
type: Opaque
stringData:
  authType: token
  token: <token_from_openstack_token_output>
  domainName: <OS_DOMAIN_NAME_from_openstack_rc_file>
  projectName: <OS_PROJECT_NAME_from_openstack_rc_file>
  username: <OS_USERNAME_from_openstack_rc_file>
  url: <OS_AUTH_URL_from_openstack_rc_file>
EOF
```

5. Continue migrating your virtual machine according to the procedure in Migrating virtual machines, starting with step 2, "Create a **Provider** manifest for the source provider."

2.6.1.2. Using application credential authentication with an OpenStack source provider

You can use application credential authentication, instead of username and password authentication, when you create an OpenStack source provider.

MTV supports both of the following types of application credential authentication:

- Application credential ID

- Application credential name

For each type of application credential authentication, you need to use data from OpenStack to create a **Secret** manifest.

Prerequisites

You have an OpenStack account.

Procedure

1. In the dashboard of the OpenStack web console, click **Project** > **API Access**.
2. Expand **Download OpenStack RC file** and click **OpenStack RC file**.
   The file that is downloaded, referred to here as **<openstack_rc_file>**, includes the following fields used for application credential authentication:

   ```
   OS_AUTH_URL
   OS_PROJECT_ID
   OS_PROJECT_NAME
   OS_DOMAIN_NAME
   OS_USERNAME
   ```

3. To get the data needed for application credential authentication, run the following command:

   ```
   $ openstack application credential create --role member --role reader --secret redhat forklift
   ```

   The output, referred to here as **<openstack_credential_output>**, includes:

   - The **id** and **secret** that you need for authentication using an application credential ID
   - The **name** and **secret** that you need for authentication using an application credential name

4. Create a **Secret** manifest similar to the following:

   - For authentication using the application credential ID:

     ```
     cat << EOF | oc apply -f -
     apiVersion: v1
     kind: Secret
     metadata:
       name: openstack-secret-appid
       namespace: openshift-mtv
       labels:
         createdForProviderType: openstack
     type: Opaque
     stringData:
       authType: applicationcredential
       applicationCredentialID: <id_from_openstack_credential_output>
       applicationCredentialSecret: <secret_from_openstack_credential_output>
       url: <OS_AUTH_URL_from_openstack_rc_file>
     EOF
     ```

   - For authentication using the application credential name:

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: openstack-secret-appname
  namespace: openshift-mtv
  labels:
    createdForProviderType: openstack
type: Opaque
stringData:
  authType: applicationcredential
  applicationCredentialName: <name_from_openstack_credential_output>
  applicationCredentialSecret: <secret_from_openstack_credential_output>
  domainName: <OS_DOMAIN_NAME_from_openstack_rc_file>
  username: <OS_USERNAME_from_openstack_rc_file>
  url: <OS_AUTH_URL_from_openstack_rc_file>
EOF
```

5. Continue migrating your virtual machine according to the procedure in Migrating virtual machines, starting with step 2, "Create a **Provider** manifest for the source provider."

2.7. VMWARE PREREQUISITES

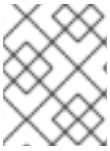
It is strongly recommended to create a VDDK image to accelerate migrations. For more information, see Creating a VDDK image .

The following prerequisites apply to VMware migrations:

- You must use a compatible version of VMware vSphere.
- You must be logged in as a user with at least the minimal set of VMware privileges.
- You must install VMware Tools on all source virtual machines (VMs).
- The VM operating system must be certified and supported for use as a guest operating system with OpenShift Virtualization *and* for conversion to KVM with **virt-v2v**.
- If you are running a warm migration, you must enable changed block tracking (CBT) on the VMs and on the VM disks.
- You must obtain the SHA-1 fingerprint of the vCenter host.
- If you are migrating more than 10 VMs from an ESXi host in the same migration plan, you must increase the NFC service memory of the host.
- It is strongly recommended to disable hibernation because Migration Toolkit for Virtualization (MTV) does not support migrating hibernated VMs.

IMPORTANT

In the event of a power outage, data might be lost for a VM with disabled hibernation. However, if hibernation is not disabled, migration will fail



NOTE

Neither MTV nor OpenShift Virtualization support conversion of Btrfs for migrating VMs from VMWare.

VMware privileges

The following minimal set of VMware privileges is required to migrate virtual machines to OpenShift Virtualization with the Migration Toolkit for Virtualization (MTV).

Table 2.4. VMware privileges

Privilege	Description
Virtual machine.Interaction privileges:	
Virtual machine.Interaction.Power Off	Allows powering off a powered-on virtual machine. This operation powers down the guest operating system.
Virtual machine.Interaction.Power On	Allows powering on a powered-off virtual machine and resuming a suspended virtual machine.
Virtual machine.Provisioning privileges:  NOTE All **Virtual machine.Provisioning** privileges are required.	
Virtual machine.Provisioning.Allow disk access	Allows opening a disk on a virtual machine for random read and write access. Used mostly for remote disk mounting.
Virtual machine.Provisioning.Allow file access	Allows operations on files associated with a virtual machine, including VMX, disks, logs, and NVRAM.
Virtual machine.Provisioning.Allow read-only disk access	Allows opening a disk on a virtual machine for random read access. Used mostly for remote disk mounting.
Virtual machine.Provisioning.Allow virtual machine download	Allows read operations on files associated with a virtual machine, including VMX, disks, logs, and NVRAM.
Virtual machine.Provisioning.Allow virtual machine files upload	Allows write operations on files associated with a virtual machine, including VMX, disks, logs, and NVRAM.
Virtual machine.Provisioning.Clone template	Allows cloning of a template.

Privilege	Description
Virtual machine.Provisioning.Clone virtual machine	Allows cloning of an existing virtual machine and allocation of resources.
Virtual machine.Provisioning.Create template from virtual machine	Allows creation of a new template from a virtual machine.
Virtual machine.Provisioning.Customize guest	Allows customization of a virtual machine's guest operating system without moving the virtual machine.
Virtual machine.Provisioning.Deploy template	Allows deployment of a virtual machine from a template.
Virtual machine.Provisioning.Mark as template	Allows marking an existing powered-off virtual machine as a template.
Virtual machine.Provisioning.Mark as virtual machine	Allows marking an existing template as a virtual machine.
Virtual machine.Provisioning.Modify customization specification	Allows creation, modification, or deletion of customization specifications.
Virtual machine.Provisioning.Promote disks	Allows promote operations on a virtual machine's disks.
Virtual machine.Provisioning.Read customization specifications	Allows reading a customization specification.
Virtual machine.Snapshot management privileges:	
Virtual machine.Snapshot management.Create snapshot	Allows creation of a snapshot from the virtual machine's current state.
Virtual machine.Snapshot management.Remove Snapshot	Allows removal of a snapshot from the snapshot history.

2.7.1. Creating a VDDK image

The Migration Toolkit for Virtualization (MTV) uses the VMware Virtual Disk Development Kit (VDDK) SDK to accelerate transferring virtual disks from VMware vSphere. Therefore, creating a VDDK image, although optional, is highly recommended.

To make use of this feature, you download the VMware Virtual Disk Development Kit (VDDK), build a VDDK image, and push the VDDK image to your image registry.

The VDDK package contains symbolic links, therefore, the procedure of creating a VDDK image must be performed on a file system that preserves symbolic links (symlinks).

NOTE

Storing the VDDK image in a public registry might violate the VMware license terms.

Prerequisites

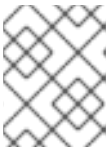
- Red Hat OpenShift image registry .
- **podman** installed.
- You are working on a file system that preserves symbolic links (symlinks).
- If you are using an external registry, OpenShift Virtualization must be able to access it.

Procedure

1. Create and navigate to a temporary directory:

```
$ mkdir /tmp/<dir_name> && cd /tmp/<dir_name>
```

2. In a browser, navigate to the VMware VDDK version 8 download page .
3. Select version 8.0.1 and click **Download**.



NOTE

In order to migrate to OpenShift Virtualization 4.12, download VDDK version 7.0.3.2 from the VMware VDDK version 7 download page .

4. Save the VDDK archive file in the temporary directory.
5. Extract the VDDK archive:

```
$ tar -xzf VMware-vix-disklib-<version>.x86_64.tar.gz
```

6. Create a **Dockerfile**:

```
$ cat > Dockerfile <<EOF
FROM registry.access.redhat.com/ubi8/ubi-minimal
USER 1001
COPY vmware-vix-disklib-distrib /vmware-vix-disklib-distrib
RUN mkdir -p /opt
ENTRYPOINT ["cp", "-r", "/vmware-vix-disklib-distrib", "/opt"]
EOF
```

7. Build the VDDK image:

```
$ podman build . -t <registry_route_or_server_path>/vddk:<tag>
```

8. Push the VDDK image to the registry:

```
$ podman push <registry_route_or_server_path>/vddk:<tag>
```

9. Ensure that the image is accessible to your OpenShift Virtualization environment.

2.7.2. Obtaining the SHA-1 fingerprint of a vCenter host

You must obtain the SHA-1 fingerprint of a vCenter host in order to create a **Secret** CR.

Procedure

- Run the following command:

```
$ openssl s_client \
    -connect <vcenter_host>:443 \ 1
    < /dev/null 2>/dev/null \
    | openssl x509 -fingerprint -noout -in /dev/stdin \
    | cut -d '=' -f 2
```

1 Specify the IP address or FQDN of the vCenter host.

Example output

```
01:23:45:67:89:AB:CD:EF:01:23:45:67:89:AB:CD:EF:01:23:45:67
```

2.7.3. Increasing the NFC service memory of an ESXi host

If you are migrating more than 10 VMs from an ESXi host in the same migration plan, you must increase the NFC service memory of the host. Otherwise, the migration will fail because the NFC service memory is limited to 10 parallel connections.

Procedure

1. Log in to the ESXi host as root.
2. Change the value of **maxMemory** to **1000000000** in **/etc/vmware/hostd/config.xml**:

```
...
    <nfcsvc>
      <path>libnfcsvc.so</path>
      <enabled>true</enabled>
      <maxMemory>1000000000</maxMemory>
      <maxStreamMemory>10485760</maxStreamMemory>
    </nfcsvc>
...
```

3. Restart **hostd**:

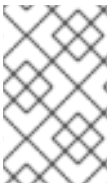
```
# /etc/init.d/hostd restart
```

You do not need to reboot the host.

2.8. OPEN VIRTUAL APPLIANCE (OVA) PREREQUISITES

The following prerequisites apply to Open Virtual Appliance (OVA) file migrations:

- All OVA files are created by VMware vSphere.



NOTE

Migration of OVA files that were not created by VMware vSphere but are compatible with vSphere might succeed. However, migration of such files is not supported by MTV. MTV supports only OVA files created by VMware vSphere.

- The OVA files are in one or more folders under an NFS shared directory in one of the following structures:
  - In one or more compressed Open Virtualization Format (OVF) packages that hold all the VM information.
    The filename of each compressed package **must** have the **.ova** extension. Several compressed packages can be stored in the same folder.

    When this structure is used, MTV scans the root folder and the first-level subfolders for compressed packages.

    For example, if the NFS share is, /**nfs**, then:
    The folder /**nfs** is scanned.
    The folder /**nfs**/**subfolder1** is scanned.
    But, /**nfs**/**subfolder1**/**subfolder2** is not scanned.

  - In extracted OVF packages.
    When this structure is used, MTV scans the root folder, first-level subfolders, and second-level subfolders for extracted OVF packages. However, there can be only one **.ovf** file in a folder. Otherwise, the migration will fail.

    For example, if the NFS share is, /**nfs**, then:
    The OVF file /**nfs**/**vm.ovf** is scanned.
    The OVF file /**nfs**/**subfolder1**/**vm.ovf** is scanned.
    The OVF file /**nfs**/**subfolder1**/**subfolder2**/**vm.ovf** is scanned.
    But, the OVF file /**nfs**/**subfolder1**/**subfolder2**/**subfolder3**/**vm.ovf** is not scanned.

2.9. SOFTWARE COMPATIBILITY GUIDELINES

You must install compatible software versions.

Table 2.5. Compatible software versions

Migration Toolkit for Virtualization	Red Hat OpenShift	OpenShift Virtualization	VMware vSphere	Red Hat Virtualization	OpenStack
2.5.7	4.12 or later	4.12 or later	6.5 or later	4.4 SP1 or later	16.1 or later

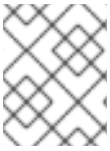
MIGRATION FROM RED HAT VIRTUALIZATION 4.3

MTV 2.5 was tested only with Red Hat Virtualization (RHV) 4.4 SP1. Migration from Red Hat Virtualization (RHV) 4.3 has not been tested with MTV 2.5.

As RHV 4.3 lacks the improvements that were introduced in RHV 4.4 for MTV, and new features were not tested with RHV 4.3, migrations from RHV 4.3 may not function at the same level as migrations from RHV 4.4, with some functionality may be missing.

Therefore, it is recommended to upgrade RHV to the supported version above before the migration to OpenShift Virtualization.

However, migrations from RHV 4.3.11 were tested with MTV 2.3, and may work in practice in many environments using MTV 2.5. In this case, we advise upgrading Red Hat Virtualization Manager (RHVM) to the previously mentioned supported version before the migration to OpenShift Virtualization.



NOTE

Deployment of MTV 2.5.3 and later is enabled on OpenShift Kubernetes Engine (OKE). For more information, see About OpenShift Kubernetes Engine .

2.9.1. OpenShift Operator Life Cycles

For more information about the software maintenance Life Cycle classifications for Operators shipped by Red Hat for use with OpenShift Container Platform, see OpenShift Operator Life Cycles.

CHAPTER 3. INSTALLING AND CONFIGURING THE MTV OPERATOR

You can install the MTV Operator by using the Red Hat OpenShift web console or the command line interface (CLI).

In Migration Toolkit for Virtualization (MTV) version 2.4 and later, the MTV Operator includes the MTV plugin for the Red Hat OpenShift web console.

After you install the MTV Operator by using either the Red Hat OpenShift web console or the CLI, you can configure the Operator.

3.1. INSTALLING THE MTV OPERATOR BY USING THE RED HAT OPENSHIFT WEB CONSOLE

You can install the MTV Operator by using the Red Hat OpenShift web console.

Prerequisites

- Red Hat OpenShift 4.12 or later installed.
- OpenShift Virtualization Operator installed on an OpenShift migration target cluster.
- You must be logged in as a user with **cluster-admin** permissions.

Procedure

1. In the Red Hat OpenShift web console, click **Operators → OperatorHub**.
2. Use the **Filter by keyword** field to search for **mtv-operator**.
3. Click **Migration Toolkit for Virtualization Operator** and then click **Install**.
4. Click **Create ForkliftController** when the button becomes active.
5. Click **Create**.
   Your ForkliftController appears in the list that is displayed.
6. Click **Workloads → Pods** to verify that the MTV pods are running.
7. Click **Operators → Installed Operators** to verify that **Migration Toolkit for Virtualization Operator** appears in the **openshift-mtv** project with the status **Succeeded**.
   When the plugin is ready you will be prompted to reload the page. The **Migration** menu item is automatically added to the navigation bar, displayed on the left of the Red Hat OpenShift web console.

3.2. INSTALLING THE MTV OPERATOR FROM THE COMMAND LINE INTERFACE

You can install the MTV Operator from the command line interface (CLI).

Prerequisites

- Red Hat OpenShift 4.12 or later installed.
- OpenShift Virtualization Operator installed on an OpenShift migration target cluster.
- You must be logged in as a user with **cluster-admin** permissions.

Procedure

1. Create the openshift-mtv project:

```
$ cat << EOF | oc apply -f -
apiVersion: project.openshift.io/v1
kind: Project
metadata:
  name: openshift-mtv
EOF
```

2. Create an **OperatorGroup** CR called **migration**:

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

3. Create a **Subscription** CR for the Operator:

```
$ cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: mtv-operator
  namespace: openshift-mtv
spec:
  channel: release-v2.5
  installPlanApproval: Automatic
  name: mtv-operator
  source: redhat-operators
  sourceNamespace: openshift-marketplace
  startingCSV: "mtv-operator.v2.5.7"
EOF
```

4. Create a **ForkliftController** CR:

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: ForkliftController
metadata:
  name: forklift-controller
  namespace: openshift-mtv
```

```
spec:
  olm_managed: true
EOF
```

5. Verify that the MTV pods are running:

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

Example output

```
NAME                                   READY   STATUS    RESTARTS   AGE
forklift-api-bb45b8db4-cpzlg           1/1     Running   0          6m34s
forklift-controller-7649db6845-zd25p   2/2     Running   0          6m38s
forklift-must-gather-api-78fb4bcdf6-h2r4m   1/1     Running   0          6m28s
forklift-operator-59c87cfbdc-pmkfc     1/1     Running   0          28m
forklift-ui-plugin-5c5564f6d6-zpd85    1/1     Running   0          6m24s
forklift-validation-7d84c74c6f-fj9xg   1/1     Running   0          6m30s
forklift-volume-populator-controller-85d5cb64b6-mrlmc   1/1     Running   0          6m36s
```

3.3. CONFIGURING THE MTV OPERATOR

You can configure the following settings of the MTV Operator using either the CLI or the user interface.

- Maximum number of virtual machines (VMs) per plan that can be migrated simultaneously
- How long **must gather** reports are retained before being automatically deleted
- CPU limit allocated to the main controller container
- Memory limit allocated to the main controller container
- Interval at which a new snapshot is requested before initiating a warm migration
- Frequency with which the system checks the status of snapshot creation or removal during a warm migration
- Percentage of space in persistent volumes allocated as file system overhead when the **storageclass** is **filesystem** (CLI only)

These settings are configured by changing the default of the appropriate parameter in the **spec** part of the **forklift-controller** CR.

The procedure for configuring these settings using the user interface is presented in Configuring MTV settings. The procedure for configuring these settings using the CLI is presented following.

Procedure

- Change a parameter's value in the **spec** portion of the **forklift-controller** CR by adding the label and value as follows:

```
spec:
  label: value (1)
```

1 Labels you can configure using the CLI are shown in the table that follows, along with a description of each label and its default value.

Table 3.1. MTV Operator labels

Label	Description	Default value
controller_max_vm_inflight	The maximum number of VMs per plan that can be migrated simultaneously.	**20**
must_gather_api_cleanup_max_age	The duration in hours for retaining **must gather** reports before they are automatically deleted.	**-1** (disabled)
controller_container_limits_cpu	The CPU limit allocated to the main controller container.	**500m**
controller_container_limits_memory	The memory limit allocated to the main controller container.	**800Mi**
controller_precopy_interval	The interval in minutes at which a new snapshot is requested before initiating a warm migration.	**60**
controller_snapshot_status_check_rate_seconds	The frequency in seconds with which the system checks the status of snapshot creation or removal during a warm migration.	**10**
controller_filesystem_overhead	Percentage of space in persistent volumes allocated as file system overhead when the **storageclass** is **filesystem**. Note, this setting can only be changed using the CLI.	**10**
controller_block_overhead	Fixed amount of additional space allocated in persistent block volumes. This setting is applicable for any **storageclass** that is block-based. It can be used when data, such as encryption headers, is written to the persistent volumes in addition to the content of the virtual disk. Note, this setting can only be changed using the CLI.	**0**

CHAPTER 4. MIGRATING VIRTUAL MACHINES BY USING THE RED HAT OPENSHIFT WEB CONSOLE

You can migrate virtual machines (VMs) to OpenShift Virtualization by using the Red Hat OpenShift web console.

IMPORTANT

You must ensure that all prerequisites are met.

VMware only: You must have the minimal set of VMware privileges.

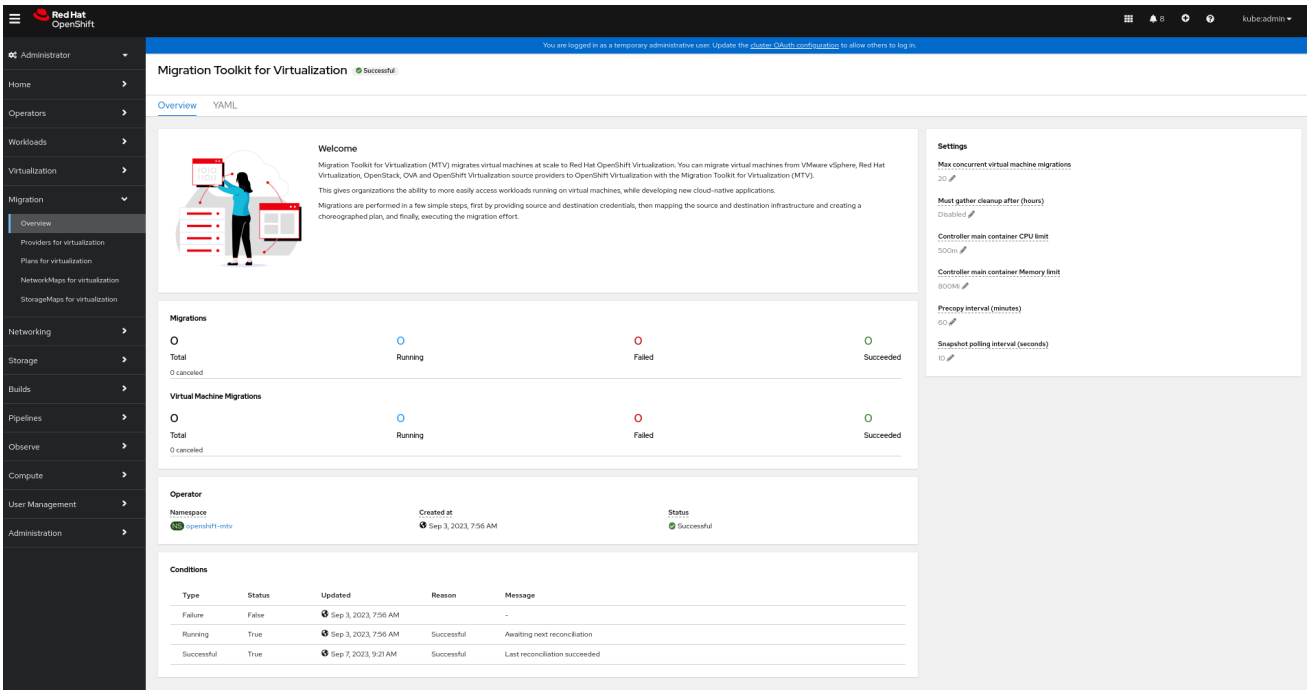
VMware only: Creating a VMware Virtual Disk Development Kit (VDDK) image will increase migration speed.

4.1. THE MTV USER INTERFACE

The Migration Toolkit for Virtualization (MTV) user interface is integrated into the OpenShift web console.

In the left-hand panel, you can choose a page related to a component of the migration progress, for example, **Providers for Migration**, or, if you are an administrator, you can choose **Overview**, which contains information about migrations and lets you configure MTV settings.

Figure 4.1. MTV extension interface



In pages related to components, you can click on the **Projects** list, which is in the upper-left portion of the page, and see which projects (namespaces) you are allowed to work with.

- If you are an administrator, you can see all projects.
- If you are a non-administrator, you can see only the projects that you have permissions to work with.

4.2. THE MTV OVERVIEW PAGE

The Migration Toolkit for Virtualization (MTV) **Overview** page displays system-wide information about migrations and a list of **Settings** you can change.

If you have Administrator privileges, you can access the **Overview** page by clicking **Migration → Overview** in the Red Hat OpenShift web console.

The **Overview** page displays the following information:

- Migrations: The number of migrations performed using MTV:
  - Total
  - Running
  - Failed
  - Succeeded
  - Canceled
- Virtual Machine Migrations: The number of VMs migrated using MTV:
  - Total
  - Running
  - Failed
  - Succeeded
  - Canceled
- Operator: The namespace on which the MTV Operator is deployed and the status of the Operator.
- Conditions: Status of the MTV Operator:
  - Failure: Last failure. **False** indicates no failure since deployment.
  - Running: Whether the Operator is currently running and waiting for the next reconciliation.
  - Successful: Last successful reconciliation.

4.3. CONFIGURING MTV SETTINGS

If you have Administrator privileges, you can access the **Overview** page and change the following settings in it:

Table 4.1. MTV settings

Setting	Description	Default value

Setting	Description	Default value
Max concurrent virtual machine migrations	The maximum number of VMs per plan that can be migrated simultaneously	20
Must gather cleanup after (hours)	The duration for retaining **must gather** reports before they are automatically deleted	Disabled
Controller main container CPU limit	The CPU limit allocated to the main controller container	500 m
Controller main container Memory limit	The memory limit allocated to the main controller container	800 Mi
Precopy internal (minutes)	The interval at which a new snapshot is requested before initiating a warm migration	60
Snapshot polling interval (seconds)	The frequency with which the system checks the status of snapshot creation or removal during a warm migration	10

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Overview**. The **Settings** list is on the right-hand side of the page.
2. In the **Settings** list, click the Edit icon of the setting you want to change.
3. Choose a setting from the list.
4. Click **Save**.

4.4. ADDING PROVIDERS

You can add source providers and destination providers for a virtual machine migration by using the Red Hat OpenShift web console.

4.4.1. Adding source providers

You can use MTV to migrate VMs from the following source providers:

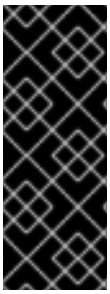
- VMware vSphere
- Red Hat Virtualization
- OpenStack
- Open Virtual Appliances (OVAs) that were created by VMware vSphere

- OpenShift Virtualization

You can add a source provider by using the Red Hat OpenShift web console.

4.4.1.1. Adding a VMware vSphere source provider

You can add a VMware vSphere source provider by using the Red Hat OpenShift web console.



IMPORTANT

EMS enforcement is disabled for migrations with VMware vSphere source providers in order to enable migrations from versions of vSphere that are supported by Migration Toolkit for Virtualization but do not comply with the 2023 FIPS requirements. Therefore, users should consider whether migrations from vSphere source providers risk their compliance with FIPS. Supported versions of vSphere are specified in Software compatibility guidelines.

Prerequisites

- VMware Virtual Disk Development Kit (VDDK) image in a secure registry that is accessible to all clusters.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Providers for virtualization**.
2. Click **Create Provider**.
3. Click **vSphere**.
4. Specify the following fields:
   - **Provider resource name**: Name of the source provider.
   - **URL**: URL of the SDK endpoint of the vCenter on which the source VM is mounted. Ensure that the URL includes the **sdk** path, usually **/sdk**. For example, **https://vCenter-host-example.com/sdk**. If a certificate for FQDN is specified, the value of this field needs to match the FQDN in the certificate.
   - **VDDK init image**: **VDDKInitImage** path. It is strongly recommended to create a VDDK init image to accelerate migrations. For more information, see Creating a VDDK image .
   - **Username**: vCenter user. For example, **user@vsphere.local**.
   - **Password**: vCenter user password.
   - **SHA-1 fingerprint**: The provider currently requires the SHA-1 fingerprint of the vCenter Server's TLS certificate in all circumstances. vSphere calls this the server's thumbprint.
5. Choose one of the following options for validating CA certificates:
   - **Skip certificate validation** : Migrate without validating a CA certificate.
   - **Use the system CA certificates**: Migrate after validating the system CA certificates.
     a. To skip certificate validation, select the **Skip certificate validation** check box.

b. To validate the system CA certificates, leave the **Skip certificate validation** check box cleared.

6. Click **Create** to add and save the provider.
   The provider appears in the list of providers.

4.4.1.2. Adding a Red Hat Virtualization source provider

You can add a Red Hat Virtualization source provider by using the Red Hat OpenShift web console.

Prerequisites

- Manager CA certificate, unless it was replaced by a third-party certificate, in which case, specify the Manager Apache CA certificate

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Providers for virtualization**.
2. Click **Create Provider**.
3. Click **Red Hat Virtualization**
4. Specify the following fields:
   - **Provider resource name**: Name of the source provider.
   - **URL**: URL of the API endpoint of the Red Hat Virtualization Manager (RHVM) on which the source VM is mounted. Ensure that the URL includes the path leading to the RHVM API server, usually **/ovirt-engine/api**. For example, **https://rhv-host-example.com/ovirt-engine/api**.
   - **Username**: Username.
   - **Password**: Password.
5. Choose one of the following options for validating CA certificates:
   - **Skip certificate validation** : Migrate without validating a CA certificate.
   - **Use a custom CA certificate**: Migrate after validating a custom CA certificate.
     a. To skip certificate validation, select the **Skip certificate validation** check box.
     b. To validate a custom CA certificate, leave the **Skip certificate validation** check box cleared and *either* drag the CA certificate to the text box *or* browse for it and click **Select**.
6. Click **Create** to add and save the provider.
   The provider appears in the list of providers.

4.4.1.3. Adding an OpenStack source provider

You can add an OpenStack source provider by using the Red Hat OpenShift web console.

NOTE

Migration using OpenStack source providers only supports VMs that use only Cinder volumes.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Providers for virtualization**.
2. Click **Create Provider**.
3. Click **OpenStack**.
4. Specify the following fields:
   - **Provider resource name**: Name of the source provider.
   - **URL**: URL of the OpenStack Identity (Keystone) endpoint. For example, **http://controller:5000/v3**.
   - **Authentication type**: Choose one of the following methods of authentication and supply the information related to your choice. For example, if you choose **Application credential ID** as the authentication type, the **Application credential ID** and the **Application credential secret** fields become active, and you need to supply the ID and the secret.
     - **Application credential ID**
       - **Application credential ID**: OpenStack application credential ID
       - **Application credential secret**: OpenStack https://github.com/kubev2v/forklift-documentation/pull/402pplication credential **Secret**
     - **Application credential name**
       - **Application credential name**: OpenStack application credential name
       - **Application credential secret**: : OpenStack application credential **Secret**
       - **Username**: OpenStack username
       - **Domain**: OpenStack domain name
     - **Token with user ID**
       - **Token**: OpenStack token
       - **User ID**: OpenStack user ID
       - **Project ID**: OpenStack project ID
     - **Token with user Name**
       - **Token**: OpenStack token
       - **Username**: OpenStack username
       - **Project**: OpenStack project
       - **Domain name**: OpenStack domain name

- **Password**
  - **Username**: OpenStack username
  - **Password**: OpenStack password
  - **Project**: OpenStack project
  - **Domain**: OpenStack domain name

5. Choose one of the following options for validating CA certificates:
   - **Skip certificate validation** : Migrate without validating a CA certificate.
   - **Use a custom CA certificate**: Migrate after validating a custom CA certificate.
     a. To skip certificate validation, select the **Skip certificate validation** check box.
     b. To validate a custom CA certificate, leave the **Skip certificate validation** check box cleared and *either* drag the CA certificate to the text box *or* browse for it and click **Select**.
6. Click **Create** to add and save the provider.
   The provider appears in the list of providers.

4.4.1.4. Adding an Open Virtual Appliance (OVA) source provider

You can add Open Virtual Appliance (OVA) files that were created by VMware vSphere as a source provider by using the Red Hat OpenShift web console.

Migration using one or more Open Virtual Appliance (OVA) files as a source provider is a Technology Preview.

IMPORTANT

Migration using one or more Open Virtual Appliance (OVA) files as a source provider is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see https://access.redhat.com/support/offerings/techpreview/.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Providers for virtualization**.
2. Click **Create Provider**.
3. Click **Open Virtual Appliance (OVA)**
4. Specify the following fields:
   - **Provider resource name**: Name of the source provider

- **URL**: URL of the NFS file share that serves the OVA

5. Click **Create** to add and save the provider.
   The provider appears in the list of providers.

NOTE

An error message might appear that states that an error has occurred. You can ignore this message.

4.4.1.5. Adding a Red Hat OpenShift Virtualization source provider

You can use a Red Hat OpenShift Virtualization provider as both a source provider and destination provider.

Specifically, the host cluster that is automatically added as a OpenShift Virtualization provider can be used as both a source provider and a destination provider.

You can migrate VMs from the cluster that MTV is deployed on to another cluster, or from a remote cluster to the cluster that MTV is deployed on.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Providers for virtualization**.
2. Click **Create Provider**.
3. Click **OpenShift Virtualization**.
4. Specify the following fields:
   - **Provider resource name**: Name of the source provider
   - **URL**: URL of the endpoint of the API server
   - **Service account bearer token**: Token for a service account with **cluster-admin** privileges If both **URL** and **Service account bearer token** are left blank, the local OpenShift cluster is used.
5. Click **Create** to add and save the provider.
   The provider appears in the list of providers.

4.4.2. Adding destination providers

You can add a OpenShift Virtualization destination provider by using the Red Hat OpenShift web console.

4.4.2.1. Adding an OpenShift Virtualization destination provider

You can use a Red Hat OpenShift Virtualization provider as both a source provider and destination provider.

Specifically, the host cluster that is automatically added as a OpenShift Virtualization provider can be used as both a source provider and a destination provider.

You can also add another OpenShift Virtualization destination provider to the Red Hat OpenShift web console in addition to the default OpenShift Virtualization destination provider, which is the cluster where you installed MTV.

You can migrate VMs from the cluster that MTV is deployed on to another cluster, or from a remote cluster to the cluster that MTV is deployed on.

Prerequisites

- You must have an OpenShift Virtualization service account token with **cluster-admin** privileges.

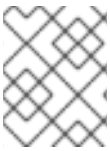
Procedure

1. In the Red Hat OpenShift web console, click **Migration → Providers for virtualization**.
2. Click **Create Provider**.
3. Click **OpenShift Virtualization**.
4. Specify the following fields:
   - **Provider resource name**: Name of the source provider
   - **URL**: URL of the endpoint of the API server
   - **Service account bearer token**: Token for a service account with **cluster-admin** privileges If both **URL** and **Service account bearer token** are left blank, the local OpenShift cluster is used.
5. Click **Create** to add and save the provider.
   The provider appears in the list of providers.

4.4.2.2. Selecting a migration network for an OpenShift Virtualization provider

You can select a default migration network for an OpenShift Virtualization provider in the Red Hat OpenShift web console to improve performance. The default migration network is used to transfer disks to the namespaces in which it is configured.

If you do not select a migration network, the default migration network is the **pod** network, which might not be optimal for disk transfer.



NOTE

You can override the default migration network of the provider by selecting a different network when you create a migration plan.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Providers for virtualization**.
2. On the right side of the provider, select **Select migration network** from the Options menu ⋮.
3. Select a network from the list of available networks and click **Select**.

4.5. CREATING A NETWORK MAPPING

You can create one or more network mappings by using the Red Hat OpenShift web console to map source networks to OpenShift Virtualization networks.

Prerequisites

- Source and target providers added to the Red Hat OpenShift web console.
- If you map more than one source and target network, each additional OpenShift Virtualization network requires its own network attachment definition.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → NetworkMaps for virtualization**.
2. Click **Create NetworkMap**.
3. Specify the following fields:
   - **Name**: Enter a name to display in the network mappings list.
   - **Source provider**: Select a source provider.
   - **Target provider**: Select a target provider.
4. Select a **Source network** and a **Target namespace/network**.
5. Optional: Click **Add** to create additional network mappings or to map multiple source networks to a single target network.
6. If you create an additional network mapping, select the network attachment definition as the target network.
7. Click **Create**.
   The network mapping is displayed on the **NetworkMaps** screen.

4.6. CREATING A STORAGE MAPPING

You can create a storage mapping by using the Red Hat OpenShift web console to map source disk storages to OpenShift Virtualization storage classes.

Prerequisites

- Source and target providers added to the Red Hat OpenShift web console.
- Local and shared persistent storage that support VM migration.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → StorageMaps for virtualization**.
2. Click **Create StorageMap**.
3. Specify the following fields:

- **Name**: Enter a name to display in the storage mappings list.
- **Source provider**: Select a source provider.
- **Target provider**: Select a target provider.

4. To create a storage mapping, click **Add** and map storage sources to target storage classes as follows:
   a. If your source provider is VMware vSphere, select a **Source datastore** and a **Target storage class**.
   b. If your source provider is Red Hat Virtualization, select a **Source storage domain** and a **Target storage class**.
   c. If your source provider is OpenStack, select a **Source volume type** and a **Target storage class**.
   d. If your source provider is a set of one or more OVA files, select a **Source** and a **Target storage class** for the dummy storage that applies to all virtual disks within the OVA files.
   e. If your storage provider is OpenShift Virtualization. select a **Source storage class** and a **Target storage class**.
   f. Optional: Click **Add** to create additional storage mappings, including mapping multiple storage sources to a single target storage class.
5. Click **Create**.
   The mapping is displayed on the **StorageMaps** page.

4.7. CREATING A MIGRATION PLAN

You can create a migration plan by using the Red Hat OpenShift web console.

A migration plan allows you to group virtual machines to be migrated together or with the same migration parameters, for example, a percentage of the members of a cluster or a complete application.

You can configure a hook to run an Ansible playbook or custom container image during a specified stage of the migration plan.

Prerequisites

- If MTV is not installed on the target cluster, you must add a target provider on the **Providers** page of the web console.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Plans for virtualization**.
2. Click **Create plan**.
3. Specify the following fields:
   - **Plan name**: Enter a migration plan name to display in the migration plan list.
   - **Plan description**: Optional: Brief description of the migration plan.

- **Source provider**: Select a source provider.
- **Target provider**: Select a target provider.
- **Target namespace**: Do one of the following:
  - Select a target namespace from the list
  - Create a target namespace by typing its name in the text box, and then clicking **create "<the_name_you_entered>"**
- You can change the migration transfer network for this plan by clicking **Select a different network**, selecting a network from the list, and then clicking **Select**.
  If you defined a migration transfer network for the OpenShift Virtualization provider and if the network is in the target namespace, the network that you defined is the default network for all migration plans. Otherwise, the **pod** network is used.

4. Click **Next**.
5. Select options to filter the list of source VMs and click **Next**.
6. Select the VMs to migrate and then click **Next**.
7. Select an existing network mapping or create a new network mapping.
8. . Optional: Click **Add** to add an additional network mapping.
   To create a new network mapping:
   - Select a target network for each source network.
   - Optional: Select **Save current mapping as a template**and enter a name for the network mapping.
9. Click **Next**.
10. Select an existing storage mapping, which you can modify, or create a new storage mapping.
    To create a new storage mapping:
    a. If your source provider is VMware, select a **Source datastore** and a **Target storage class**.
    b. If your source provider is Red Hat Virtualization, select a **Source storage domain** and a **Target storage class**.
    c. If your source provider is OpenStack, select a **Source volume type** and a **Target storage class**.
11. Optional: Select **Save current mapping as a template**and enter a name for the storage mapping.
12. Click **Next**.
13. Select a migration type and click **Next**.
    - Cold migration: The source VMs are stopped while the data is copied.
    - Warm migration: The source VMs run while the data is copied incrementally. Later, you will run the cutover, which stops the VMs and copies the remaining VM data and metadata.

NOTE

Warm migration is supported only from vSphere and Red Hat Virtualization.

14. Click **Next**.
15. Optional: You can create a migration hook to run an Ansible playbook before or after migration:
    a. Click **Add hook**.
    b. Select the **Step when the hook will be run**: pre-migration or post-migration.
    c. Select a **Hook definition**:
    - **Ansible playbook**: Browse to the Ansible playbook or paste it into the field.
    - **Custom container image**: If you do not want to use the default **hook-runner** image, enter the image path: **<registry_path>/<image_name>:<tag>**.

NOTE

The registry must be accessible to your Red Hat OpenShift cluster.

16. Click **Next**.
17. Review your migration plan and click **Finish**.
    The migration plan is saved on the **Plans** page.

    You can click the Options menu ⋮ of the migration plan and select **View details** to verify the migration plan details.

4.8. RUNNING A MIGRATION PLAN

You can run a migration plan and view its progress in the Red Hat OpenShift web console.

Prerequisites

- Valid migration plan.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Plans for virtualization**.
   The **Plans** list displays the source and target providers, the number of virtual machines (VMs) being migrated, the status, and the description of each plan.
2. Click **Start** beside a migration plan to start the migration.
3. Click **Start** in the confirmation window that opens.
   The **Migration details by VM** screen opens, displaying the migration's progress

   Warm migration only:

   - The precopy stage starts.

- Click **Cutover** to complete the migration.

4. If the migration fails:
   a. Click **Get logs** to retrieve the migration logs.
   b. Click **Get logs** in the confirmation window that opens.
   c. Wait until **Get logs** changes to **Download logs** and then click the button to download the logs.
5. Click a migration's **Status**, whether it failed or succeeded or is still ongoing, to view the details of the migration.
   The **Migration details by VM** screen opens, displaying the start and end times of the migration, the amount of data copied, and a progress pipeline for each VM being migrated.
6. Expand an individual VM to view its steps and the elapsed time and state of each step.

4.9. MIGRATION PLAN OPTIONS

On the **Plans for virtualization** page of the Red Hat OpenShift web console, you can click the Options menu ⋮ beside a migration plan to access the following options:

- **Get logs**: Retrieves the logs of a migration. When you click **Get logs**, a confirmation window opens. After you click **Get logs** in the window, wait until **Get logs** changes to **Download logs** and then click the button to download the logs.
- **Edit**: Edit the details of a migration plan. You cannot edit a migration plan while it is running or after it has completed successfully.
- **Duplicate**: Create a new migration plan with the same virtual machines (VMs), parameters, mappings, and hooks as an existing plan. You can use this feature for the following tasks:
  - Migrate VMs to a different namespace.
  - Edit an archived migration plan.
  - Edit a migration plan with a different status, for example, failed, canceled, running, critical, or ready.
- **Archive**: Delete the logs, history, and metadata of a migration plan. The plan cannot be edited or restarted. It can only be viewed.

NOTE

The **Archive** option is irreversible. However, you can duplicate an archived plan.

- **Delete**: Permanently remove a migration plan. You cannot delete a running migration plan.

NOTE

The **Delete** option is irreversible.

Deleting a migration plan does not remove temporary resources such as **importer** pods, **conversion** pods, config maps, secrets, failed VMs, and data volumes. (BZ#2018974) You must archive a migration plan before deleting it in order to clean up the temporary resources.

- **View details**: Display the details of a migration plan.
- **Restart**: Restart a failed or canceled migration plan.
- **Cancel scheduled cutover**: Cancel a scheduled cutover migration for a warm migration plan.

4.10. CANCELING A MIGRATION

You can cancel the migration of some or all virtual machines (VMs) while a migration plan is in progress by using the Red Hat OpenShift web console.

Procedure

1. In the Red Hat OpenShift web console, click **Plans for virtualization**.
2. Click the name of a running migration plan to view the migration details.
3. Select one or more VMs and click **Cancel**.
4. Click **Yes, cancel** to confirm the cancellation.
   In the **Migration details by VM** list, the status of the canceled VMs is **Canceled**. The unmigrated and the migrated virtual machines are not affected.

You can restart a canceled migration by clicking **Restart** beside the migration plan on the **Migration plans** page.

CHAPTER 5. MIGRATING VIRTUAL MACHINES FROM THE COMMAND LINE

You can migrate virtual machines to OpenShift Virtualization from the command line.

IMPORTANT

- VMware only: You must have the minimal set of VMware privileges.
- VMware only: You must have the vCenter SHA-1 fingerprint.
- VMware only: Creating a VMware Virtual Disk Development Kit (VDDK) image will increase migration speed.
- You must ensure that all prerequisites are met.

5.1. PERMISSIONS NEEDED BY NON-ADMINISTRATORS TO WORK WITH MIGRATION PLAN COMPONENTS

If you are an administrator, you can work with all components of migration plans (for example, providers, network mappings, and migration plans).

By default, non-administrators have limited ability to work with migration plans and their components. As an administrator, you can modify their roles to allow them full access to all components, or you can give them limited permissions.

For example, administrators can assign non-administrators one or more of the following cluster roles for migration plans:

Table 5.1. Example migration plan roles and their privileges

Role	Description
plans.forklift.konveyor.io-v1beta1-view	Can view migration plans but not to create, delete or modify them
plans.forklift.konveyor.io-v1beta1-edit	Can create, delete or modify (all parts of **edit** permissions) individual migration plans
plans.forklift.konveyor.io-v1beta1-admin	All **edit** privileges and the ability to delete the entire collection of migration plans

Note that pre-defined cluster roles include a resource (for example, **plans**), an API group (for example, **forklift.konveyor.io-v1beta1**) and an action (for example, **view**, **edit**).

As a more comprehensive example, you can grant non-administrators the following set of permissions per namespace:

- Create and modify storage maps, network maps, and migration plans for the namespaces they have access to
- Attach providers created by administrators to storage maps, network maps, and migration plans

- Not be able to create providers or to change system settings

Table 5.2. Example permissions required for non-adminstrators to work with migration plan components but not create providers

Actions	API group	Resource
get, **list**, **watch**, **create**, **update**, **patch**, **delete**	**forklift.konveyer.io**	**plans**
get, **list**, **watch**, **create**, **update**, **patch**, **delete**	**forklift.konveyer.io**	**migrations**
get, **list**, **watch**, **create**, **update**, **patch**, **delete**	**forklift.konveyer.io**	**hooks**
get, **list**, **watch**	**forklift.konveyer.io**	**providers**
get, **list**, **watch**, **create**, **update**, **patch**, **delete**	**forklift.konveyer.io**	**networkmaps**
get, **list**, **watch**, **create**, **update**, **patch**, **delete**	**forklift.konveyer.io**	**storagemaps**
get, **list**, **watch**	**forklift.konveyer.io**	**forkliftcontrollers**

NOTE

Non-administrators need to have the **create** permissions that are part of **edit** roles for network maps and for storage maps to create migration plans, even when using a template for a network map or a storage map.

5.2. RETRIEVING A VMWARE VSPHERE MOREF

When you migrate VMs with a VMware vSphere source provider using Migration Toolkit for Virtualization (MTV) from the CLI, you need to know the managed object reference (moRef) of certain entities in vSphere, such as datastores, networks, and VMs.

You can retrieve the moRef of one or more vSphere entities from the Inventory service. You can then use each moRef as a reference for retrieving the moRef of another entity.

Procedure

1. Retrieve the routes for the project:

```
oc get route -n openshift-mtv
```

2. Retrieve the **Inventory** service route:

```
$ oc get route <inventory_service> -n openshift-mtv
```

3. Retrieve the access token:

```
$ TOKEN=$(oc whoami -t)
```

4. Retrieve the moRef of a VMware vSphere provider:

```
$ curl -H "Authorization: Bearer $TOKEN"
https://<inventory_service_route>/providers/vsphere -k
```

5. Retrieve the datastores of a VMware vSphere source provider:

```
$ curl -H "Authorization: Bearer $TOKEN"
https://<inventory_service_route>/providers/vsphere/<provider id>/datastores/ -k
```

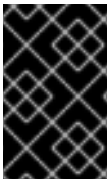
Example output

```
[
  {
    "id": "datastore-11",
    "parent": {
      "kind": "Folder",
      "id": "group-s5"
    },
    "path": "/Datacenter/datastore/v2v_general_porpuse_ISCSI_DC",
    "revision": 46,
    "name": "v2v_general_porpuse_ISCSI_DC",
    "selfLink": "providers/vsphere/01278af6-e1e4-4799-b01b-
d5ccc8dd0201/datastores/datastore-11"
  },
  {
    "id": "datastore-730",
    "parent": {
      "kind": "Folder",
      "id": "group-s5"
    },
    "path": "/Datacenter/datastore/f01-h27-640-SSD_2",
    "revision": 46,
    "name": "f01-h27-640-SSD_2",
    "selfLink": "providers/vsphere/01278af6-e1e4-4799-b01b-
d5ccc8dd0201/datastores/datastore-730"
  },
 ...
```

In this example, the moRef of the datastore **v2v_general_porpuse_ISCSI_DC** is **datastore-11** and the moRef of the datastore **f01-h27-640-SSD_2** is **datastore-730**.

5.3. MIGRATING VIRTUAL MACHINES

You migrate virtual machines (VMs) from the command line (CLI) by creating MTV custom resources (CRs).



IMPORTANT

You must specify a name for cluster-scoped CRs.

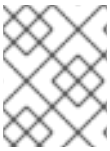
You must specify both a name and a namespace for namespace-scoped CRs.

Migration using one or more Open Virtual Appliance (OVA) files as a source provider is a Technology Preview.

IMPORTANT

Migration using one or more Open Virtual Appliance (OVA) files as a source provider is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see https://access.redhat.com/support/offerings/techpreview/.

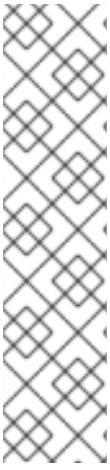


NOTE

Migration using OpenStack source providers only supports VMs that use only Cinder volumes.

Prerequisites

- VMware only: You must have a VMware Virtual Disk Development Kit (VDDK) image in a secure registry that is accessible to all clusters.
- Red Hat Virtualization (RHV) only: If you are migrating a virtual machine with a direct LUN disk, ensure that the nodes in the OpenShift Virtualization destination cluster that the VM is expected to run on can access the backend storage.



NOTE

- Unlike disk images that *are copied* from a source provider to a target provider, LUNs are *detached*, *but not removed*, from virtual machines in the source provider and then attached to the virtual machines (VMs) that are created in the target provider.
- LUNs are not removed from the source provider during the migration in case fallback to the source provider is required. However, before re-attaching the LUNs to VMs in the source provider, ensure that the LUNs are not used by VMs on the target environment at the same time, which might lead to data corruption.
- Migration of Fibre Channel LUNs is not supported.

Procedure

1. Create a **Secret** manifest for the source provider credentials:

```
$ cat << EOF | oc apply -f -
apiVersion: v1
```

```
kind: Secret
metadata:
  name: <secret>
  namespace: <namespace>
  ownerReferences: 1
    - apiVersion: forklift.konveyor.io/v1beta1
      kind: Provider
      name: <provider_name>
      uid: <provider_uid>
  labels:
    createdForProviderType: <provider_type> 2
    createdForResourceType: providers
type: Opaque
stringData: 3
  user: <user> 4
  password: <password> 5
  insecureSkipVerify: <true/false> 6
  domainName: <domain_name> 7
  projectName: <project_name> 8
  regionName: <region name> 9
  cacert: | 10
    <ca_certificate>
  url: <api_end_point> 11
  thumbprint: <vcenter_fingerprint> 12
  token: <service_account_bearer_token> 13
EOF
```

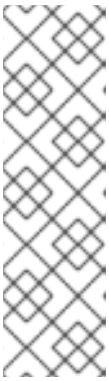
1. The **ownerReferences** section is optional.
2. Specify the type of source provider. Allowed values are **ovirt**, **vsphere**, **openstack**, **ova**, and **openshift**. This label is needed to verify the credentials are correct when the remote system is accessible and, for RHV, to retrieve the Manager CA certificate when a third-party certificate is specified.
3. The **stringData** section for OVA is different and is described in a note that follows the description of the **Secret** manifest.
4. Specify the vCenter user, the RHV Manager user, or the OpenStack user.
5. Specify the user password.
6. Specify **<true>** to skip certificate verification, which proceeds with an insecure migration and then the certificate is not required. Insecure migration means that the transferred data is sent over an insecure connection and potentially sensitive data could be exposed. Specifying **<false>** verifies the certificate.
7. OpenStack only: Specify the domain name.
8. OpenStack only: Specify the project name.
9. OpenStack only: Specify the name of the OpenStack region.
10. RHV and OpenStack only: For RHV, enter the Manager CA certificate unless it was replaced by a third-party certificate, in which case, enter the Manager Apache CA certificate. You can retrieve the Manager CA certificate at https://<engine_host>/ovirt-

engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA. For OpenStack, enter the CA certificate for connecting to the source environment. The certificate is not used when **insecureSkipVerify** is set to **<true>**.

11 Specify the API end point URL, for example, **https://<vCenter_host>/sdk** for vSphere, **https://<engine_host>/ovirt-engine/api** for RHV, or **https://<identity_service>/v3** for OpenStack.

12 VMware only: Specify the vCenter SHA-1 fingerprint.

13 OpenShift only: Token for a service account with **cluster-admin** privileges.



NOTE

The **stringData** section for an OVA **Secret** manifest is as follows:

```
stringData:
  url: <nfs_server:/nfs_path>
```

where:
nfs_server: An IP or hostname of the server where the share was created.
nfs_path : The path on the server where the OVA files are stored.

2. Create a **Provider** manifest for the source provider:

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Provider
metadata:
  name: <source_provider>
  namespace: <namespace>
spec:
  type: <provider_type> 1
  url: <api_end_point> 2
  settings:
    vddkInitImage: <registry_route_or_server_path>/vddk:<tag> 3
  secret:
    name: <secret> 4
    namespace: <namespace>
EOF
```

1 Specify the type of source provider. Allowed values are **ovirt**, **vsphere**, **openstack**, **ova**, and **openshift**.

2 Specify the API end point URL, for example, **https://<vCenter_host>/sdk** for vSphere, **https://<engine_host>/ovirt-engine/api** for RHV, or **https://<identity_service>/v3** for OpenStack.

3 VMware only: Specify the VDDK image that you created.

4 Specify the name of provider **Secret** CR.

3. VMware only: Create a **Host** manifest:

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Host
metadata:
  name: <vmware_host>
  namespace: <namespace>
spec:
  provider:
    namespace: <namespace>
    name: <source_provider> 1
  id: <source_host_mor> 2
  ipAddress: <source_network_ip> 3
EOF
```

1. Specify the name of the VMware **Provider** CR.
2. Specify the managed object reference (moRef) of the VMware host. To retrieve the moRef, see Retrieving a VMware vSphere moRef.
3. Specify the IP address of the VMware migration network.

4. Create a **NetworkMap** manifest to map the source and destination networks:

```
$  cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: NetworkMap
metadata:
  name: <network_map>
  namespace: <namespace>
spec:
  map:
    - destination:
        name: <network_name>
        type: pod 1
      source: 2
        id: <source_network_id> 3
        name: <source_network_name>
    - destination:
        name: <network_attachment_definition> 4
        namespace: <network_attachment_definition_namespace> 5
        type: multus
      source:
        name: <network_attachment_definition> 6
        namespace: <network_attachment_definition_namespace> 7
        type: multus 8
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

1. Allowed values are **pod** and **multus**.
2. You can use either the **id** *or* the **name** parameter to specify the source network.
3. Specify the VMware network moRef, the RHV network UUID, or the OpenStack network UUID. To retrieve the moRef, see Retrieving a VMware vSphere moRef.
4. Specify a network attachment definition for each additional OpenShift Virtualization network.
5. Required only when **type** is **multus**. Specify the namespace of the OpenShift Virtualization network attachment definition.
6. Specify a network attachment definition for each additional OpenShift Virtualization network.
7. Required only when **type** is **multus**. Here, **namespace** can either be specified using the namespace property or with a name built as follows: **<network_namespace>/<network_name>**.
8. OpenShift only.

5. Create a **StorageMap** manifest to map source and destination storage:

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: StorageMap
metadata:
  name: <storage_map>
  namespace: <namespace>
spec:
  map:
    - destination:
        storageClass: <storage_class>
        accessMode: <access_mode> 1
      source:
        id: <source_datastore> 2
    - destination:
        storageClass: <storage_class>
        accessMode: <access_mode>
      source:
        id: <source_datastore>
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

1. Allowed values are **ReadWriteOnce** and **ReadWriteMany**.
2. Specify the VMware datastore moRef, the RHV storage domain UUID, or the OpenStack **volume_type** UUID. For example, **f2737930-b567-451a-9ceb-2887f6207009**. To retrieve the moRef, see Retrieving a VMware vSphere moRef.

NOTE

For OVA, the **StorageMap** can map only a single storage, which all the disks from the OVA are associated with, to a storage class at the destination. For this reason, the storage is referred to in the UI as "Dummy storage for source provider <provider_name>".

6. Optional: Create a **Hook** manifest to run custom code on a VM during the phase specified in the **Plan** CR:

```
$  cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Hook
metadata:
  name: <hook>
  namespace: <namespace>
spec:
  image: quay.io/konveyor/hook-runner (1)
  playbook: | (2)

LS0tCi0gbmFtZTogTWFpbgogIGhvc3RzOiBsb2NhbGhvc3QKICB0YXNrczoKICAtIG5hbWU6I
Exv

YWQgUGxhbgogICAgaW5jbHVkZV92YXJzOgogICAgICBmaWxlOiAiL3RtC9ob29rL3BsYW4u
eW1s

IgogICAgICBuYW1lOiBwbGFuCiAgLSBuYW1lOiBMb2FkIFdvcmtsb2FkCiAgICBpbmNsdWRlX
3Zh

cnM6CiAgICAgIGZpbGU6ICIvdG1wL2hvb2svd29ya2xvYWQueW1sIgogICAgICBuYW1lOiB3b
3Jr
    bG9hZAoK
EOF
```

(1) You can use the default **hook-runner** image or specify a custom image. If you specify a custom image, you do not have to specify a playbook.

(2) Optional: Base64-encoded Ansible Playbook. If you specify a playbook, the **image** must be **hook-runner**.

7. Create a **Plan** manifest for the migration:

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Plan
metadata:
  name: <plan> (1)
  namespace: <namespace>
spec:
  warm: true (2)
  provider:
```

```
  source:
    name: <source_provider>
    namespace: <namespace>
  destination:
    name: <destination_provider>
    namespace: <namespace>
 map: 3
  network: 4
    name: <network_map> 5
    namespace: <namespace>
  storage: 6
    name: <storage_map> 7
    namespace: <namespace>
 targetNamespace: <target_namespace>
 vms: 8
  - id: <source_vm> 9
  - name: <source_vm>
    namespace: <namespace> 10
    hooks: 11
      - hook:
          namespace: <namespace>
          name: <hook> 12
        step: <step> 13
EOF
```

1. Specify the name of the **Plan** CR.
2. Specify whether the migration is warm or cold. If you specify a warm migration without specifying a value for the **cutover** parameter in the **Migration** manifest, only the precopy stage will run.
3. Specify only one network map and one storage map per plan.
4. Specify a network mapping even if the VMs to be migrated are not assigned to a network. The mapping can be empty in this case.
5. Specify the name of the **NetworkMap** CR.
6. Specify a storage mapping even if the VMs to be migrated are not assigned with disk images. The mapping can be empty in this case.
7. Specify the name of the **StorageMap** CR.
8. For all source providers except for OpenShift Virtualization, you can use either the **id** *or* the **name** parameter to specify the source VMs.
   OpenShift Virtualization source provider only: You can use only the **name** parameter, not the **id.** parameter to specify the source VMs.
9. Specify the VMware VM moRef, RHV VM UUID or the OpenStack VM UUID. To retrieve the moRef, see Retrieving a VMware vSphere moRef .
10. OpenShift Virtualization source provider only.
11. Optional: You can specify up to two hooks for a VM. Each hook must run during a separate migration step.

12 Specify the name of the **Hook** CR.

13 Allowed values are **PreHook**, before the migration plan starts, or **PostHook**, after the migration is complete.

8. Create a **Migration** manifest to run the **Plan** CR:

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Migration
metadata:
  name: <migration> 1
  namespace: <namespace>
spec:
  plan:
    name: <plan> 2
    namespace: <namespace>
  cutover: <cutover_time> 3
EOF
```

1 Specify the name of the **Migration** CR.

2 Specify the name of the **Plan** CR that you are running. The **Migration** CR creates a **VirtualMachine** CR for each VM that is migrated.

3 Optional: Specify a cutover time according to the ISO 8601 format with the UTC time offset, for example, **2021-04-04T01:23:45.678+09:00**.

You can associate multiple **Migration** CRs with a single **Plan** CR. If a migration does not complete, you can create a new **Migration** CR, without changing the **Plan** CR, to migrate the remaining VMs.

9. Retrieve the **Migration** CR to monitor the progress of the migration:

```
$ oc get migration/<migration> -n <namespace> -o yaml
```

5.4. OBTAINING THE SHA-1 FINGERPRINT OF A VCENTER HOST

You must obtain the SHA-1 fingerprint of a vCenter host in order to create a **Secret** CR.

Procedure

- Run the following command:

```
$ openssl s_client \
    -connect <vcenter_host>:443 \ 1
    < /dev/null 2>/dev/null \
    | openssl x509 -fingerprint -noout -in /dev/stdin \
    | cut -d '=' -f 2
```

1 Specify the IP address or FQDN of the vCenter host.

Example output

```
01:23:45:67:89:AB:CD:EF:01:23:45:67:89:AB:CD:EF:01:23:45:67
```

5.5. CANCELING A MIGRATION

You can cancel an entire migration or individual virtual machines (VMs) while a migration is in progress from the command line interface (CLI).

Canceling an entire migration

- Delete the **Migration** CR:

```
$ oc delete migration <migration> -n <namespace>
```
(1)

(1) Specify the name of the **Migration** CR.

Canceling the migration of individual VMs

1. Add the individual VMs to the **spec.cancel** block of the **Migration** manifest:

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Migration
metadata:
  name: <migration>
  namespace: <namespace>
...
spec:
  cancel:
  - id: vm-102
  - id: vm-203
  - name: rhel8-vm
EOF
```
(1)

(1) You can specify a VM by using the **id** key or the **name** key.

The value of the **id** key is the *managed object reference*, for a VMware VM, or the *VM UUID*, for a RHV VM.

2. Retrieve the **Migration** CR to monitor the progress of the remaining VMs:

```
$ oc get migration/<migration> -n <namespace> -o yaml
```

CHAPTER 6. ADVANCED MIGRATION OPTIONS

6.1. CHANGING PRECOPY INTERVALS FOR WARM MIGRATION

You can change the snapshot interval by patching the **ForkliftController** custom resource (CR).

Procedure

- Patch the **ForkliftController** CR:

```
$ oc patch forkliftcontroller/<forklift-controller> -n openshift-mtv -p '{"spec":
{"controller_precopy_interval": <60>}}' --type=merge 1
```

1 Specify the precopy interval in minutes. The default value is **60**.

You do not need to restart the **forklift-controller** pod.

6.2. CREATING CUSTOM RULES FOR THE VALIDATION SERVICE

The **Validation** service uses Open Policy Agent (OPA) policy rules to check the suitability of each virtual machine (VM) for migration. The **Validation** service generates a list of *concerns* for each VM, which are stored in the **Provider Inventory** service as VM attributes. The web console displays the concerns for each VM in the provider inventory.

You can create custom rules to extend the default ruleset of the **Validation** service. For example, you can create a rule that checks whether a VM has multiple disks.

6.2.1. About Rego files

Validation rules are written in Rego, the Open Policy Agent (OPA) native query language. The rules are stored as **.rego** files in the **/usr/share/opa/policies/io/konveyor/forklift/<provider>** directory of the **Validation** pod.

Each validation rule is defined in a separate **.rego** file and tests for a specific condition. If the condition evaluates as **true**, the rule adds a **{"category", "label", "assessment"}** hash to the **concerns**. The **concerns** content is added to the **concerns** key in the inventory record of the VM. The web console displays the content of the **concerns** key for each VM in the provider inventory.

The following **.rego** file example checks for distributed resource scheduling enabled in the cluster of a VMware VM:

drs_enabled.rego example

```
package io.konveyor.forklift.vmware 1

has_drs_enabled {
   input.host.cluster.drsEnabled 2
}

concerns[flag] {
   has_drs_enabled
   flag := {
```

```
        "category": "Information",
        "label": "VM running in a DRS-enabled cluster",
        "assessment": "Distributed resource scheduling is not currently supported by OpenShift
Virtualization. The VM can be migrated but it will not have this feature in the target environment."
    }
}
```

1. Each validation rule is defined within a package. The package namespaces are **io.konveyor.forklift.vmware** for VMware and **io.konveyor.forklift.ovirt** for Red Hat Virtualization.
2. Query parameters are based on the **input** key of the **Validation** service JSON.

6.2.2. Checking the default validation rules

Before you create a custom rule, you must check the default rules of the **Validation** service to ensure that you do not create a rule that redefines an existing default value.

Example: If a default rule contains the line **default valid_input = false** and you create a custom rule that contains the line **default valid_input = true**, the **Validation** service will not start.

Procedure

1. Connect to the terminal of the **Validation** pod:

```
$ oc rsh <validation_pod>
```

2. Go to the OPA policies directory for your provider:

```
$ cd /usr/share/opa/policies/io/konveyor/forklift/<provider> 1
```

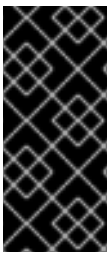
1. Specify **vmware** or **ovirt**.

3. Search for the default policies:

```
$ grep -R "default" *
```

6.2.3. Creating a validation rule

You create a validation rule by applying a config map custom resource (CR) containing the rule to the **Validation** service.



IMPORTANT

- If you create a rule with the same *name* as an existing rule, the **Validation** service performs an **OR** operation with the rules.
- If you create a rule that contradicts a default rule, the **Validation** service will not start.

Validation rule example

Validation rules are based on virtual machine (VM) attributes collected by the **Provider Inventory** service.

For example, the VMware API uses this path to check whether a VMware VM has NUMA node affinity configured: **MOR:VirtualMachine.config.extraConfig["numa.nodeAffinity"]**.

The **Provider Inventory** service simplifies this configuration and returns a testable attribute with a list value:

```
"numaNodeAffinity": [
    "0",
    "1"
],
```

You create a Rego query, based on this attribute, and add it to the **forklift-validation-config** config map:

```
`count(input.numaNodeAffinity) != 0`
```

Procedure

1. Create a config map CR according to the following example:

```
$ cat << EOF | oc apply -f -
apiVersion: v1
kind: ConfigMap
metadata:
  name: <forklift-validation-config>
  namespace: openshift-mtv
data:
  vmware_multiple_disks.rego: |-
    package <provider_package> 1

    has_multiple_disks { 2
      count(input.disks) > 1
    }

    concerns[flag] {
      has_multiple_disks 3
        flag := {
          "category": "<Information>", 4
          "label": "Multiple disks detected",
          "assessment": "Multiple disks detected on this VM."
        }
    }
EOF
```

1. Specify the provider package name. Allowed values are **io.konveyor.forklift.vmware** for VMware and **io.konveyor.forklift.ovirt** for Red Hat Virtualization.
2. Specify the **concerns** name and Rego query.
3. Specify the **concerns** name and **flag** parameter values.

4 Allowed values are **Critical**, **Warning**, and **Information**.

2. Stop the **Validation** pod by scaling the **forklift-controller** deployment to **0**:

```
$ oc scale -n openshift-mtv --replicas=0 deployment/forklift-controller
```

3. Start the **Validation** pod by scaling the **forklift-controller** deployment to **1**:

```
$ oc scale -n openshift-mtv --replicas=1 deployment/forklift-controller
```

4. Check the **Validation** pod log to verify that the pod started:

```
$ oc logs -f <validation_pod>
```

If the custom rule conflicts with a default rule, the **Validation** pod will not start.

5. Remove the source provider:

```
$ oc delete provider <provider> -n openshift-mtv
```

6. Add the source provider to apply the new rule:

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Provider
metadata:
  name: <provider>
  namespace: openshift-mtv
spec:
  type: <provider_type> 1
  url: <api_end_point> 2
  secret:
    name: <secret> 3
    namespace: openshift-mtv
EOF
```

1 Allowed values are **ovirt**, **vsphere**, and **openstack**.

2 Specify the API end point URL, for example, **https://<vCenter_host>/sdk** for vSphere, **https://<engine_host>/ovirt-engine/api** for RHV, or **https://<identity_service>/v3** for OpenStack.

3 Specify the name of the provider **Secret** CR.

You must update the rules version after creating a custom rule so that the **Inventory** service detects the changes and validates the VMs.

6.2.4. Updating the inventory rules version

You must update the inventory rules version each time you update the rules so that the **Provider Inventory** service detects the changes and triggers the **Validation** service.

The rules version is recorded in a **rules_version.rego** file for each provider.

Procedure

1. Retrieve the current rules version:

```
$ GET https://forklift-validation/v1/data/io/konveyor/forklift/<provider>/rules_version
```
(1)

Example output

```
{
  "result": {
     "rules_version": 5
  }
}
```

2. Connect to the terminal of the **Validation** pod:

```
$ oc rsh <validation_pod>
```

3. Update the rules version in the **/usr/share/opa/policies/io/konveyor/forklift/<provider>/rules_version.rego** file.

4. Log out of the **Validation** pod terminal.

5. Verify the updated rules version:

```
$ GET https://forklift-validation/v1/data/io/konveyor/forklift/<provider>/rules_version
```
(1)

Example output

```
{
  "result": {
     "rules_version": 6
  }
}
```

6.3. RETRIEVING THE INVENTORY SERVICE JSON

You retrieve the **Inventory** service JSON by sending an **Inventory** service query to a virtual machine (VM). The output contains an **"input"** key, which contains the inventory attributes that are queried by the **Validation** service rules.

You can create a validation rule based on any attribute in the **"input"** key, for example, **input.snapshot.kind**.

Procedure

1. Retrieve the routes for the project:

```
oc get route -n openshift-mtv
```

2. Retrieve the **Inventory** service route:

```
$ oc get route <inventory_service> -n openshift-mtv
```

3. Retrieve the access token:

```
$ TOKEN=$(oc whoami -t)
```

4. Trigger an HTTP GET request (for example, using Curl):

```
$ curl -H "Authorization: Bearer $TOKEN" https://<inventory_service_route>/providers -k
```

5. Retrieve the **UUID** of a provider:

```
$ curl -H "Authorization: Bearer $TOKEN"
https://<inventory_service_route>/providers/<provider> -k 1
```

1 1 1 Allowed values for the provider are **vsphere**, **ovirt**, and **openstack**.

6. Retrieve the VMs of a provider:

```
$ curl -H "Authorization: Bearer $TOKEN"
https://<inventory_service_route>/providers/<provider>/<UUID>/vms -k
```

7. Retrieve the details of a VM:

```
$ curl -H "Authorization: Bearer $TOKEN"
https://<inventory_service_route>/providers/<provider>/<UUID>/workloads/<vm> -k
```

Example output


```
{
   "input": {
      "selfLink": "providers/vsphere/c872d364-d62b-46f0-bd42-16799f40324e/workloads/vm-
431",
      "id": "vm-431",
      "parent": {
         "kind": "Folder",
         "id": "group-v22"
      },
      "revision": 1,
      "name": "iscsi-target",
      "revisionValidated": 1,
      "isTemplate": false,
      "networks": [
         {
            "kind": "Network",
            "id": "network-31"
         },
         {
            "kind": "Network",
            "id": "network-33"
```

```
        }
    ],
    "disks": [
        {
            "key": 2000,
            "file": "[iSCSI_Datastore] iscsi-target/iscsi-target-000001.vmdk",
            "datastore": {
                "kind": "Datastore",
                "id": "datastore-63"
            },
            "capacity": 17179869184,
            "shared": false,
            "rdm": false
        },
        {
            "key": 2001,
            "file": "[iSCSI_Datastore] iscsi-target/iscsi-target_1-000001.vmdk",
            "datastore": {
                "kind": "Datastore",
                "id": "datastore-63"
            },
            "capacity": 10737418240,
            "shared": false,
            "rdm": false
        }
    ],
    "concerns": [],
    "policyVersion": 5,
    "uuid": "42256329-8c3a-2a82-54fd-01d845a8bf49",
    "firmware": "bios",
    "powerState": "poweredOn",
    "connectionState": "connected",
    "snapshot": {
        "kind": "VirtualMachineSnapshot",
        "id": "snapshot-3034"
    },
    "changeTrackingEnabled": false,
    "cpuAffinity": [
        0,
        2
    ],
    "cpuHotAddEnabled": true,
    "cpuHotRemoveEnabled": false,
    "memoryHotAddEnabled": false,
    "faultToleranceEnabled": false,
    "cpuCount": 2,
    "coresPerSocket": 1,
    "memoryMB": 2048,
    "guestName": "Red Hat Enterprise Linux 7 (64-bit)",
    "balloonedMemory": 0,
    "ipAddress": "10.19.2.96",
    "storageUsed": 30436770129,
    "numaNodeAffinity": [
        "0",
        "1"
    ],
```

```
        "devices": [
            {
                "kind": "RealUSBController"
            }
        ],
        "host": {
            "id": "host-29",
            "parent": {
                "kind": "Cluster",
                "id": "domain-c26"
            },
            "revision": 1,
            "name": "IP address or host name of the vCenter host or RHV Engine host",
            "selfLink": "providers/vsphere/c872d364-d62b-46f0-bd42-16799f40324e/hosts/host-
29",
            "status": "green",
            "inMaintenance": false,
            "managementServerIp": "10.19.2.96",
            "thumbprint": <thumbprint>,
            "timezone": "UTC",
            "cpuSockets": 2,
            "cpuCores": 16,
            "productName": "VMware ESXi",
            "productVersion": "6.5.0",
            "networking": {
                "pNICs": [
                    {
                        "key": "key-vim.host.PhysicalNic-vmnic0",
                        "linkSpeed": 10000
                    },
                    {
                        "key": "key-vim.host.PhysicalNic-vmnic1",
                        "linkSpeed": 10000
                    },
                    {
                        "key": "key-vim.host.PhysicalNic-vmnic2",
                        "linkSpeed": 10000
                    },
                    {
                        "key": "key-vim.host.PhysicalNic-vmnic3",
                        "linkSpeed": 10000
                    }
                ],
                "vNICs": [
                    {
                        "key": "key-vim.host.VirtualNic-vmk2",
                        "portGroup": "VM_Migration",
                        "dPortGroup": "",
                        "ipAddress": "192.168.79.13",
                        "subnetMask": "255.255.255.0",
                        "mtu": 9000
                    },
                    {
                        "key": "key-vim.host.VirtualNic-vmk0",
                        "portGroup": "Management Network",
                        "dPortGroup": "",
```

```
            "ipAddress": "10.19.2.13",
            "subnetMask": "255.255.255.128",
            "mtu": 1500
        },
        {
            "key": "key-vim.host.VirtualNic-vmk1",
            "portGroup": "Storage Network",
            "dPortGroup": "",
            "ipAddress": "172.31.2.13",
            "subnetMask": "255.255.0.0",
            "mtu": 1500
        },
        {
            "key": "key-vim.host.VirtualNic-vmk3",
            "portGroup": "",
            "dPortGroup": "dvportgroup-48",
            "ipAddress": "192.168.61.13",
            "subnetMask": "255.255.255.0",
            "mtu": 1500
        },
        {
            "key": "key-vim.host.VirtualNic-vmk4",
            "portGroup": "VM_DHCP_Network",
            "dPortGroup": "",
            "ipAddress": "10.19.2.231",
            "subnetMask": "255.255.255.128",
            "mtu": 1500
        }
    ],
    "portGroups": [
        {
            "key": "key-vim.host.PortGroup-VM Network",
            "name": "VM Network",
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch0"
        },
        {
            "key": "key-vim.host.PortGroup-Management Network",
            "name": "Management Network",
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch0"
        },
        {
            "key": "key-vim.host.PortGroup-VM_10G_Network",
            "name": "VM_10G_Network",
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch1"
        },
        {
            "key": "key-vim.host.PortGroup-VM_Storage",
            "name": "VM_Storage",
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch1"
        },
        {
            "key": "key-vim.host.PortGroup-VM_DHCP_Network",
            "name": "VM_DHCP_Network",
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch1"
        },
        {
```

```
            "key": "key-vim.host.PortGroup-Storage Network",
            "name": "Storage Network",
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch1"
         },
         {
            "key": "key-vim.host.PortGroup-VM_Isolated_67",
            "name": "VM_Isolated_67",
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch2"
         },
         {
            "key": "key-vim.host.PortGroup-VM_Migration",
            "name": "VM_Migration",
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch2"
         }
      ],
      "switches": [
         {
            "key": "key-vim.host.VirtualSwitch-vSwitch0",
            "name": "vSwitch0",
            "portGroups": [
               "key-vim.host.PortGroup-VM Network",
               "key-vim.host.PortGroup-Management Network"
            ],
            "pNICs": [
               "key-vim.host.PhysicalNic-vmnic4"
            ]
         },
         {
            "key": "key-vim.host.VirtualSwitch-vSwitch1",
            "name": "vSwitch1",
            "portGroups": [
               "key-vim.host.PortGroup-VM_10G_Network",
               "key-vim.host.PortGroup-VM_Storage",
               "key-vim.host.PortGroup-VM_DHCP_Network",
               "key-vim.host.PortGroup-Storage Network"
            ],
            "pNICs": [
               "key-vim.host.PhysicalNic-vmnic2",
               "key-vim.host.PhysicalNic-vmnic0"
            ]
         },
         {
            "key": "key-vim.host.VirtualSwitch-vSwitch2",
            "name": "vSwitch2",
            "portGroups": [
               "key-vim.host.PortGroup-VM_Isolated_67",
               "key-vim.host.PortGroup-VM_Migration"
            ],
            "pNICs": [
               "key-vim.host.PhysicalNic-vmnic3",
               "key-vim.host.PhysicalNic-vmnic1"
            ]
         }
      ]
   },
   "networks": [
```

```
            {
                "kind": "Network",
                "id": "network-31"
            },
            {
                "kind": "Network",
                "id": "network-34"
            },
            {
                "kind": "Network",
                "id": "network-57"
            },
            {
                "kind": "Network",
                "id": "network-33"
            },
            {
                "kind": "Network",
                "id": "dvportgroup-47"
            }
        ],
        "datastores": [
            {
                "kind": "Datastore",
                "id": "datastore-35"
            },
            {
                "kind": "Datastore",
                "id": "datastore-63"
            }
        ],
        "vms": null,
        "networkAdapters": [],
        "cluster": {
            "id": "domain-c26",
            "parent": {
                "kind": "Folder",
                "id": "group-h23"
            },
            "revision": 1,
            "name": "mycluster",
            "selfLink": "providers/vsphere/c872d364-d62b-46f0-bd42-
16799f40324e/clusters/domain-c26",
            "folder": "group-h23",
            "networks": [
                {
                    "kind": "Network",
                    "id": "network-31"
                },
                {
                    "kind": "Network",
                    "id": "network-34"
                },
                {
                    "kind": "Network",
                    "id": "network-57"
```

```
                },
                {
                    "kind": "Network",
                    "id": "network-33"
                },
                {
                    "kind": "Network",
                    "id": "dvportgroup-47"
                }
            ],
            "datastores": [
                {
                    "kind": "Datastore",
                    "id": "datastore-35"
                },
                {
                    "kind": "Datastore",
                    "id": "datastore-63"
                }
            ],
            "hosts": [
                {
                    "kind": "Host",
                    "id": "host-44"
                },
                {
                    "kind": "Host",
                    "id": "host-29"
                }
            ],
            "dasEnabled": false,
            "dasVms": [],
            "drsEnabled": true,
            "drsBehavior": "fullyAutomated",
            "drsVms": [],
            "datacenter": null
        }
    }
  }
}
```

CHAPTER 7. UPGRADING THE MIGRATION TOOLKIT FOR VIRTUALIZATION

You can upgrade the MTV Operator by using the Red Hat OpenShift web console to install the new version.

Procedure

1. In the Red Hat OpenShift web console, click **Operators → Installed Operators → Migration Toolkit for Virtualization Operator → Subscription**.
2. Change the update channel to the correct release.
   See Changing update channel in the Red Hat OpenShift documentation.
3. Confirm that **Upgrade status** changes from **Up to date** to **Upgrade available**. If it does not, restart the **CatalogSource** pod:
   a. Note the catalog source, for example, **redhat-operators**.
   b. From the command line, retrieve the catalog source pod:

   ```
   $ oc get pod -n openshift-marketplace | grep <catalog_source>
   ```

   c. Delete the pod:

   ```
   $ oc delete pod -n openshift-marketplace <catalog_source_pod>
   ```

   Upgrade status changes from **Up to date** to **Upgrade available**.

   If you set **Update approval** on the **Subscriptions** tab to **Automatic**, the upgrade starts automatically.
4. If you set **Update approval** on the **Subscriptions** tab to **Manual**, approve the upgrade.
   See Manually approving a pending upgrade in the Red Hat OpenShift documentation.
5. If you are upgrading from MTV 2.2 and have defined VMware source providers, edit the VMware provider by adding a VDDK **init** image. Otherwise, the update will change the state of any VMware providers to **Critical**. For more information, see Addding a VMSphere source provider.
6. If you mapped to NFS on the Red Hat OpenShift destination provider in MTV 2.2, edit the **AccessModes** and **VolumeMode** parameters in the NFS storage profile. Otherwise, the upgrade will invalidate the NFS mapping. For more information, see Customizing the storage profile.

CHAPTER 8. UNINSTALLING THE MIGRATION TOOLKIT FOR VIRTUALIZATION

You can uninstall the Migration Toolkit for Virtualization (MTV) by using the Red Hat OpenShift web console or the command line interface (CLI).

8.1. UNINSTALLING MTV BY USING THE RED HAT OPENSHIFT WEB CONSOLE

You can uninstall Migration Toolkit for Virtualization (MTV) by using the Red Hat OpenShift web console to delete the **openshift-mtv** project and custom resource definitions (CRDs).

Prerequisites

- You must be logged in as a user with **cluster-admin** privileges.

Procedure

1. Click **Home** → **Projects**.
2. Locate the **openshift-mtv** project.
3. On the right side of the project, select **Delete Project** from the Options menu ⋮.
4. In the **Delete Project** pane, enter the project name and click **Delete**.
5. Click **Administration** → **CustomResourceDefinitions**.
6. Enter **forklift** in the **Search** field to locate the CRDs in the **forklift.konveyor.io** group.
7. On the right side of each CRD, select **Delete CustomResourceDefinition** from the Options menu ⋮.

8.2. UNINSTALLING MTV FROM THE COMMAND LINE INTERFACE

You can uninstall Migration Toolkit for Virtualization (MTV) from the command line interface (CLI) by deleting the **openshift-mtv** project and the **forklift.konveyor.io** custom resource definitions (CRDs).

Prerequisites

- You must be logged in as a user with **cluster-admin** privileges.

Procedure

1. Delete the project:

```
$ oc delete project openshift-mtv
```

2. Delete the CRDs:

```
$ oc get crd -o name | grep 'forklift' | xargs oc delete
```

3. Delete the OAuthClient:

```
$ oc delete oauthclient/forklift-ui
```

CHAPTER 9. TROUBLESHOOTING

This section provides information for troubleshooting common migration issues.

9.1. ERROR MESSAGES

This section describes error messages and how to resolve them.

warm import retry limit reached

The **warm import retry limit reached** error message is displayed during a warm migration if a VMware virtual machine (VM) has reached the maximum number (28) of changed block tracking (CBT) snapshots during the precopy stage.

To resolve this problem, delete some of the CBT snapshots from the VM and restart the migration plan.

Unable to resize disk image to required size

The **Unable to resize disk image to required size** error message is displayed when migration fails because a virtual machine on the target provider uses persistent volumes with an EXT4 file system on block storage. The problem occurs because the default overhead that is assumed by CDI does not completely include the reserved place for the root partition.

To resolve this problem, increase the file system overhead in CDI to be more than 10%.

9.2. USING THE MUST-GATHER TOOL

You can collect logs and information about MTV custom resources (CRs) by using the **must-gather** tool. You must attach a **must-gather** data file to all customer cases.

You can gather data for a specific namespace, migration plan, or virtual machine (VM) by using the filtering options.

NOTE

If you specify a non-existent resource in the filtered **must-gather** command, no archive file is created.

Prerequisites

- You must be logged in to the OpenShift Virtualization cluster as a user with the **cluster-admin** role.
- You must have the Red Hat OpenShift CLI (**oc**) installed.

Collecting logs and CR information

1. Navigate to the directory where you want to store the **must-gather** data.
2. Run the **oc adm must-gather** command:

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.5.7
```

The data is saved as **/must-gather/must-gather.tar.gz**. You can upload this file to a support case on the Red Hat Customer Portal .

3. Optional: Run the **oc adm must-gather** command with the following options to gather filtered data:

   - Namespace:

     ```
     $ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.5.7 \
       -- NS=<namespace> /usr/bin/targeted
     ```

   - Migration plan:

     ```
     $ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.5.7 \
       -- PLAN=<migration_plan> /usr/bin/targeted
     ```

   - Virtual machine:

     ```
     $ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.5.7 \
       -- VM=<vm_id> NS=<namespace> /usr/bin/targeted 1
     ```

     1 Specify the VM *ID* as it appears in the **Plan** CR.

9.3. ARCHITECTURE

This section describes MTV custom resources, services, and workflows.

9.3.1. MTV custom resources and services

The Migration Toolkit for Virtualization (MTV) is provided as an Red Hat OpenShift Operator. It creates and manages the following custom resources (CRs) and services.

MTV custom resources

- **Provider** CR stores attributes that enable MTV to connect to and interact with the source and target providers.
- **NetworkMapping** CR maps the networks of the source and target providers.
- **StorageMapping** CR maps the storage of the source and target providers.
- **Plan** CR contains a list of VMs with the same migration parameters and associated network and storage mappings.
- **Migration** CR runs a migration plan.
  Only one **Migration** CR per migration plan can run at a given time. You can create multiple **Migration** CRs for a single **Plan** CR.

MTV services

- The **Inventory** service performs the following actions:
  - Connects to the source and target providers.
  - Maintains a local inventory for mappings and plans.
  - Stores VM configurations.
  - Runs the **Validation** service if a VM configuration change is detected.
- The **Validation** service checks the suitability of a VM for migration by applying rules.
- The **Migration Controller** service orchestrates migrations.
  When you create a migration plan, the **Migration Controller** service validates the plan and adds a status label. If the plan fails validation, the plan status is **Not ready** and the plan cannot be used to perform a migration. If the plan passes validation, the plan status is **Ready** and it can be used to perform a migration. After a successful migration, the **Migration Controller** service changes the plan status to **Completed**.
- The **Populator Controller** service orchestrates disk transfers using Volume Populators.
- The **Kubevirt Controller** and **Containerized Data Import (CDI) Controller** services handle most technical operations.

9.3.2. High-level migration workflow

The high-level workflow shows the migration process from the point of view of the user:

1. You create a source provider, a target provider, a network mapping, and a storage mapping.
2. You create a **Plan** custom resource (CR) that includes the following resources:
   - Source provider
   - Target provider, if MTV is not installed on the target cluster
   - Network mapping
   - Storage mapping
   - One or more virtual machines (VMs)
3. You run a migration plan by creating a **Migration** CR that references the **Plan** CR.
   If you cannot migrate all the VMs for any reason, you can create multiple **Migration** CRs for the same **Plan** CR until all VMs are migrated.
4. For each VM in the **Plan** CR, the **Migration Controller** service records the VM migration progress in the **Migration** CR.
5. Once the data transfer for each VM in the **Plan** CR completes, the **Migration Controller** service creates a **VirtualMachine** CR.
   When all VMs have been migrated, the **Migration Controller** service updates the status of the **Plan** CR to **Completed**. The power state of each source VM is maintained after migration.

9.3.3. Detailed migration workflow

You can use the detailed migration workflow to troubleshoot a failed migration.

The workflow describes the following steps:

Warm Migration or migration to a remote OpenShift cluster:

1. When you create the **Migration** custom resource (CR) to run a migration plan, the **Migration Controller** service creates a **DataVolume** CR for each source VM disk.
   For each VM disk:

2. The **Containerized Data Importer (CDI) Controller** service creates a persistent volume claim (PVC) based on the parameters specified in the **DataVolume** CR.

3. If the **StorageClass** has a dynamic provisioner, the persistent volume (PV) is dynamically provisioned by the **StorageClass** provisioner.

4. The **CDI Controller** service creates an **importer** pod.

5. The **importer** pod streams the VM disk to the PV.
   After the VM disks are transferred:

6. The **Migration Controller** service creates a **conversion** pod with the PVCs attached to it when importing from VMWare.
   The **conversion** pod runs **virt-v2v**, which installs and configures device drivers on the PVCs of the target VM.

7. The **Migration Controller** service creates a **VirtualMachine** CR for each source virtual machine (VM), connected to the PVCs.

8. If the VM ran on the source environment, the **Migration Controller** powers on the VM, the **KubeVirt Controller** service creates a **virt-launcher** pod and a **VirtualMachineInstance** CR.
   The **virt-launcher** pod runs **QEMU-KVM** with the PVCs attached as VM disks.

Cold migration from RHV or OpenStack to the local OpenShift cluster:

1. When you create a **Migration** custom resource (CR) to run a migration plan, the **Migration Controller** service creates for each source VM disk a **PersistentVolumeClaim** CR, and an **OvirtVolumePopulator** when the source is RHV, or an **OpenstackVolumePopulator** CR when the source is OpenStack.
   For each VM disk:

2. The **Populator Controller** service creates a temporarily persistent volume claim (PVC).

3. If the **StorageClass** has a dynamic provisioner, the persistent volume (PV) is dynamically provisioned by the **StorageClass** provisioner.

   - The **Migration Controller** service creates a dummy pod to bind **all PVCs**. The name of the pod contains **pvcinit**.

4. The **Populator Controller** service creates a **populator** pod.

5. The **populator** pod transfers the disk data to the PV.
   After the VM disks are transferred:

6. The temporary PVC is deleted, and the initial PVC points to the PV with the data.

7. The **Migration Controller** service creates a **VirtualMachine** CR for each source virtual machine (VM), connected to the PVCs.

8. If the VM ran on the source environment, the **Migration Controller** powers on the VM, the **KubeVirt Controller** service creates a **virt-launcher** pod and a **VirtualMachineInstance** CR. The **virt-launcher** pod runs **QEMU-KVM** with the PVCs attached as VM disks.

Cold migration from VMWare to the local OpenShift cluster:

1. When you create a **Migration** custom resource (CR) to run a migration plan, the **Migration Controller** service creates a **DataVolume** CR for each source VM disk.
For each VM disk:

2. The **Containerized Data Importer (CDI) Controller** service creates a blank persistent volume claim (PVC) based on the parameters specified in the **DataVolume** CR.

3. If the **StorageClass** has a dynamic provisioner, the persistent volume (PV) is dynamically provisioned by the **StorageClass** provisioner.

For all VM disks:

1. The **Migration Controller** service creates a dummy pod to bind **all PVCs**. The name of the pod contains **pvcinit**.

2. The **Migration Controller** service creates a **conversion** pod for all PVCs.

3. The **conversion** pod runs **virt-v2v**, which converts the VM to the KVM hypervisor and transfers the disks' data to their corresponding PVs.
After the VM disks are transferred:

4. The **Migration Controller** service creates a **VirtualMachine** CR for each source virtual machine (VM), connected to the PVCs.

5. If the VM ran on the source environment, the **Migration Controller** powers on the VM, the **KubeVirt Controller** service creates a **virt-launcher** pod and a **VirtualMachineInstance** CR. The **virt-launcher** pod runs **QEMU-KVM** with the PVCs attached as VM disks.

9.4. LOGS AND CUSTOM RESOURCES

You can download logs and custom resource (CR) information for troubleshooting. For more information, see the detailed migration workflow.

9.4.1. Collected logs and custom resource information

You can download logs and custom resource (CR) **yaml** files for the following targets by using the Red Hat OpenShift web console or the command line interface (CLI):

- Migration plan: Web console or CLI.
- Virtual machine: Web console or CLI.
- Namespace: CLI only.

The **must-gather** tool collects the following logs and CR files in an archive file:

- CRs:
  - **DataVolume** CR: Represents a disk mounted on a migrated VM.

- **VirtualMachine** CR: Represents a migrated VM.
- **Plan** CR: Defines the VMs and storage and network mapping.
- **Job** CR: Optional: Represents a pre-migration hook, a post-migration hook, or both.

- Logs:
  - **importer** pod: Disk-to-data-volume conversion log. The **importer** pod naming convention is **importer-<migration_plan>-<vm_id><5_char_id>**, for example, **importer-mig-plan-ed90dfc6-9a17-4a8btnfh**, where **ed90dfc6-9a17-4a8** is a truncated RHV VM ID and **btnfh** is the generated 5-character ID.
  - **conversion** pod: VM conversion log. The **conversion** pod runs **virt-v2v**, which installs and configures device drivers on the PVCs of the VM. The **conversion** pod naming convention is **<migration_plan>-<vm_id><5_char_id>**.
  - **virt-launcher** pod: VM launcher log. When a migrated VM is powered on, the **virt-launcher** pod runs **QEMU-KVM** with the PVCs attached as VM disks.
  - **forklift-controller** pod: The log is filtered for the migration plan, virtual machine, or namespace specified by the **must-gather** command.
  - **forklift-must-gather-api** pod: The log is filtered for the migration plan, virtual machine, or namespace specified by the **must-gather** command.
  - **hook-job** pod: The log is filtered for hook jobs. The **hook-job** naming convention is **<migration_plan>-<vm_id><5_char_id>**, for example, **plan2j-vm-3696-posthook-4mx85** or **plan2j-vm-3696-prehook-mwqnl**.

NOTE

Empty or excluded log files are not included in the **must-gather** archive file.

Example must-gather archive structure for a VMware migration plan

```
must-gather
└── namespaces
    ├── target-vm-ns
    │   ├── crs
    │   │   ├── datavolume
    │   │   │   ├── mig-plan-vm-7595-tkhdz.yaml
    │   │   │   ├── mig-plan-vm-7595-5qvqp.yaml
    │   │   │   └── mig-plan-vm-8325-xccfw.yaml
    │   │   └── virtualmachine
    │   │       ├── test-test-rhel8-2disks2nics.yaml
    │   │       └── test-x2019.yaml
    │   └── logs
    │       ├── importer-mig-plan-vm-7595-tkhdz
    │       │   └── current.log
    │       ├── importer-mig-plan-vm-7595-5qvqp
    │       │   └── current.log
    │       ├── importer-mig-plan-vm-8325-xccfw
    │       │   └── current.log
    │       ├── mig-plan-vm-7595-4glzd
    │       │   └── current.log
```

```
│       └── mig-plan-vm-8325-4zw49
│           └── current.log
└── openshift-mtv
    ├── crs
    │   └── plan
    │       └── mig-plan-cold.yaml
    └── logs
        ├── forklift-controller-67656d574-w74md
        │   └── current.log
        └── forklift-must-gather-api-89fc7f4b6-hlwb6
            └── current.log
```

9.4.2. Downloading logs and custom resource information from the web console

You can download logs and information about custom resources (CRs) for a completed, failed, or canceled migration plan or for migrated virtual machines (VMs) by using the Red Hat OpenShift web console.

Procedure

1. In the Red Hat OpenShift web console, click **Migration → Plans for virtualization**.
2. Click **Get logs** beside a migration plan name.
3. In the **Get logs** window, click **Get logs**.
   The logs are collected. A **Log collection complete** message is displayed.
4. Click **Download logs** to download the archive file.
5. To download logs for a migrated VM, click a migration plan name and then click **Get logs** beside the VM.

9.4.3. Accessing logs and custom resource information from the command line interface

You can access logs and information about custom resources (CRs) from the command line interface by using the **must-gather** tool. You must attach a **must-gather** data file to all customer cases.

You can gather data for a specific namespace, a completed, failed, or canceled migration plan, or a migrated virtual machine (VM) by using the filtering options.

NOTE

If you specify a non-existent resource in the filtered **must-gather** command, no archive file is created.

Prerequisites

- You must be logged in to the OpenShift Virtualization cluster as a user with the **cluster-admin** role.
- You must have the Red Hat OpenShift CLI (**oc**) installed.

Procedure

1. Navigate to the directory where you want to store the **must-gather** data.
2. Run the **oc adm must-gather** command:

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.5.7
```

The data is saved as **/must-gather/must-gather.tar.gz**. You can upload this file to a support case on the Red Hat Customer Portal .

3. Optional: Run the **oc adm must-gather** command with the following options to gather filtered data:

- Namespace:

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.5.7 \
  -- NS=<namespace> /usr/bin/targeted
```

- Migration plan:

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.5.7 \
  -- PLAN=<migration_plan> /usr/bin/targeted
```

- Virtual machine:

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.5.7 \
  -- VM=<vm_name> NS=<namespace> /usr/bin/targeted 1
```

1 You must specify the VM *name*, not the VM ID, as it appears in the **Plan** CR.