Migration Toolkit for Virtualization 2.6

安装并使用 Migration Toolkit for Virtualization

从 VMware vSphere 或 Red Hat Virtualization 迁移到 Red Hat OpenShift Virtualization

Last Updated: 2024-05-31

Migration Toolkit for Virtualization 2.6 安装并使用 Migration Toolkit for Virtualization

从 VMware vSphere 或 Red Hat Virtualization 迁移到 Red Hat OpenShift Virtualization

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

法律通告

摘要

Migration Toolkit for Virtualization (MTV)可让您将虚拟机从 VMware vSphere、Red Hat Virtualization 或 OpenStack 迁移到在 Red Hat OpenShift 上运行的 OpenShift Virtualization。

目录

使开源包含更多 3

第 1 章 关于 MIGRATION TOOLKIT FOR VIRTUALIZATION 4
1.1. 关于冷迁移和温迁移 4

第 2 章 先决条件 6
2.1. 软件要求 6
2.2. 存储支持和默认模式 6
2.3. 网络先决条件 7
2.4. 源虚拟机先决条件 8
2.5. RED HAT VIRTUALIZATION 的先决条件 8
2.6. OPENSTACK 的先决条件 9
2.7. VMWARE 的先决条件 12
2.8. 开放虚拟设备(OVA)先决条件 16
2.9. 软件兼容性指南 17

第 3 章 安装和配置 MTV OPERATOR 19
3.1. 使用 RED HAT OPENSHIFT WEB 控制台安装 MTV OPERATOR 19
3.2. 使用命令行界面安装 MTV OPERATOR 19
3.3. 配置 MTV OPERATOR 21

第 4 章 使用 RED HAT OPENSHIFT WEB 控制台迁移虚拟机 24
4.1. MTV 用户界面 24
4.2. MTV OVERVIEW 页 24
4.3. 配置 MTV 设置 26
4.4. 添加供应商 26
4.5. 创建迁移计划 34
4.6. 运行迁移计划 36
4.7. 迁移计划选项 36
4.8. 取消迁移 37

第 5 章 从命令行迁移虚拟机 38
5.1. 非管理员用于迁移计划组件所需的权限 38
5.2. 迁移虚拟机 39
5.3. 取消迁移 66

第 6 章 高级迁移选项 68
6.1. 为 WARM 迁移更改预复制间隔 68
6.2. 为 VALIDATION 服务创建自定义规则 68
6.3. 在迁移计划中添加 HOOK 82

第 7 章 升级 MIGRATION TOOLKIT FOR VIRTUALIZATION 89

第 8 章 卸载 MIGRATION TOOLKIT FOR VIRTUALIZATION 91
8.1. 使用 RED HAT OPENSHIFT WEB 控制台卸载 MTV 91
8.2. 使用命令行界面卸载 MTV 92

第 9 章 故障排除 93
9.1. É"™È¯¯Ä¿iÆ�¯ 93
9.2. 使用 MUST-GATHER 工具 93
9.3. 架构 95
9.4. 日志和自定义资源 100

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅我们的首席技术官 Chris Wright 提供的消息。

第 1 章 关于 MIGRATION TOOLKIT FOR VIRTUALIZATION

您可以使用 Migration Toolkit for Virtualization (MTV)将虚拟机从以下源供应商迁移到 OpenShift Virtualization 目标供应商：

- VMware vSphere
- Red Hat Virtualization（RHV）
- OpenStack
- 由 VMware vSphere 创建的开放虚拟设备(OVA)
- 远程 OpenShift Virtualization 集群

其他资源

- 从 VMware vSphere 迁移到 OpenShift Virtualization 的性能建议 。
- 从 Red Hat Virtualization 迁移到 OpenShift Virtualization 的性能建议 。

1.1. 关于冷迁移和温迁移

MTV 支持从以下迁移进行冷迁移：

- VMware vSphere
- Red Hat Virtualization（RHV）
- OpenStack
- 远程 OpenShift Virtualization 集群

MTV 支持从 VMware vSphere 和 RHV 进行温迁移。

1.1.1. 冷迁移

冷迁移是默认的迁移类型。源虚拟机会在复制数据期间被关闭。

1.1.2. 温迁移

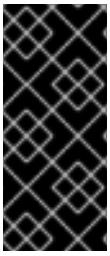
在源虚拟机 (VM) 正在运行时，大多数数据都会在 *precopy* 阶段复制。

然后，虚拟机将关闭，并在 *cutover* 阶段复制剩余的数据。

复制前（Precopy）阶段

在 precopy 阶段不会关闭虚拟机。

VM 磁盘使用更改的块跟踪(CBT)快照逐步进行复制。默认情况下，快照以一小时的间隔创建。您可以通过更新 **forklift-controller** 部署来更改快照间隔。



重要

您必须为每个源虚拟机和每个虚拟机磁盘启用 CBT。

虚拟机可以最多支持 28 CBT 快照。如果源虚拟机有太多 CBT 快照，且 **Migration Controller** 服务无法创建新快照，则温迁移可能会失败。当不再需要快照时，**Migration Controller** 服务会删除每个快照。

precopy 阶段会运行，直到手动启动了或通过调度启动了 cutover 阶段为止。

cutover 阶段

虚拟机在 cutover 阶段关闭，剩余的数据被迁移。存储在 RAM 中的数据不会迁移。

您可以使用 MTV 控制台手动启动 cutover 阶段，或者在 **Migration** 清单中调度剪切时间。

第 2 章 先决条件

查看以下先决条件，以确保您的环境为迁移准备。

2.1. 软件要求

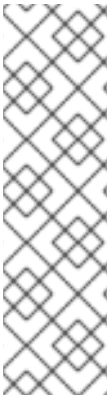
您必须安装 兼容版本的 Red Hat OpenShift 和 OpenShift Virtualization。

2.2. 存储支持和默认模式

MTV 对支持的存储使用以下默认卷和访问模式。

表 2.1. 默认卷和访问模式

Provisioner	卷模式	访问模式
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



注意

如果 OpenShift Virtualization 存储不支持 动态置备，您必须应用以下设置：

- **文件系统卷模式**
  文件系统卷模式比 Block 卷模式慢。
- **ReadWriteOnce** 访问模式
  ReadWriteOnce 访问模式不支持实时迁移。

有关编辑存储配置集的详情，请参阅启用静态置备的存储类。

注意

如果您的迁移使用块存储和持久性卷，使用 EXT4 文件系统创建的持久性卷，请将 CDI 中的文件系统开销增加到大于 10%。CDI 假设的默认开销不会完全包括 root 分区保留的位置。如果您没有增加 CDI 中的文件系统开销，您的迁移可能会失败。

注意

当从 OpenStack 迁移或运行从 RHV 迁移到 MTV 的 OCP 集群时，迁移会分配没有 CDI 的持久性卷。在这些情况下，您可能需要调整文件系统开销。

如果配置的文件系统开销（默认值为 10%）太低，则磁盘传输会因为缺少空间而失败。在这种情况下，您要提高文件系统开销。

然而，在某些情况下，您可能希望减少文件系统开销来减少存储消耗。

您可以通过更改 **forklift-controller** CR 的 **spec** 部分中的 **controller_filesystem_overhead** 值来更改文件系统开销，如 配置 MTV Operator 所述。

2.3. 网络先决条件

以下先决条件适用于所有迁移：

- 在迁移之前或期间不得更改 IP 地址、VLAN 和其他网络配置设置。在迁移过程中保留虚拟机的 MAC 地址。
- 源环境、OpenShift Virtualization 集群和复制存储库之间的网络连接必须可靠且不会中断。
- 如果要映射多个源和目标网络，您必须为每个额外目标网络创建一个 网络附加定义。

2.3.1. 端口

防火墙必须启用以下端口的流量：

表 2.2. 从 VMware vSphere 迁移所需的网络端口

端口	协议	源	目的地	目的
443	TCP	OpenShift 节点	VMware vCenter	VMware 供应商清单 磁盘传输身份验证

端口	协议	源	目的地	目的
443	TCP	OpenShift 节点	VMware ESXi 主机	磁盘传输身份验证
902	TCP	OpenShift 节点	VMware ESXi 主机	磁盘传输数据复制

表 2.3. 从 Red Hat Virtualization 迁移所需的网络端口

端口	协议	源	目的地	目的
443	TCP	OpenShift 节点	RHV Engine	RHV 供应商清单 磁盘传输身份验证
443	TCP	OpenShift 节点	RHV 主机	磁盘传输身份验证
54322	TCP	OpenShift 节点	RHV 主机	磁盘传输数据复制

2.4. 源虚拟机先决条件

以下先决条件适用于所有迁移：

- 必须卸载 ISO/CDROM 磁盘。
- 每个 NIC 必须包含一个 IPv4 和/或一个 IPv6 地址。
- 虚拟机操作系统必须经过认证并支持作为 带有 OpenShift Virtualization 的客户机操作系统使用。
- 虚拟机名称只能包含小写字母 (**a-z**)、数字 (**0-9**) 或连字符 (**-**)，最多 253 个字符。第一个和最后一个字符必须为字母数字。名称不得包含大写字母、空格、句点 (**.**) 或特殊字符。
- 虚拟机名称不能与 OpenShift Virtualization 环境中的虚拟机的名称重复。

注意

Migration Toolkit for Virtualization 会自动为不遵循规则的虚拟机分配新名称。

Migration Toolkit for Virtualization 在自动生成新虚拟机名称时进行以下更改：

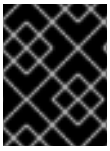
- 排除的字符被删除。
- 大写字母切换到小写字母。
- 所有下划线 (_) 都被更改为短划线 (**-**)。

此功能允许迁移平稳进行，即使某人输入了不符合规则的虚拟机名称。

2.5. RED HAT VIRTUALIZATION 的先决条件

以下先决条件适用于 Red Hat Virtualization 迁移：

- 要创建源供应商，必须至少为其分配 **UserRole** 和 **ReadOnlyAdmin** 角色。这些是最低所需的权限，但任何其他管理员或超级用户权限也可以正常工作。



重要

您必须保留 **UserRole** 和 **ReadOnlyAdmin** 角色，直到源供应商的虚拟机已迁移为止。否则，迁移将失败。

- 迁移虚拟机：
  - 您必须有以下之一：
    - RHV admin 权限。这些权限允许您迁移系统中的任何虚拟机。
    - 要迁移的每个虚拟机上的 **DiskCreator** 和 **UserVmManager** 权限。
  - 您必须使用兼容版本的 Red Hat Virtualization。
  - 您必须具有 Manager CA 证书，除非它被第三方证书替代，在这种情况下，指定 Manager Apache CA 证书。
    您可以在浏览器中导航到 https://<engine_host>/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA 来获取 Manager CA 证书。
  - 如果您要迁移使用直接 LUN 磁盘的虚拟机，请确保虚拟机在 OpenShift Virtualization 目标集群中的节点可以访问后端存储。

注意

- 与从源供应商 *复制到目标* 供应商的磁盘镜像不同，LUN *会从* 源供应商中的 *虚拟机分离*，然后附加到目标供应商中创建的虚拟机(VM)。
- 如果需要回退到源供应商，则 LUN 不会在迁移过程中从源供应商中删除。但是，在将 LUN 重新附加到源供应商中的虚拟机之前，请确保目标环境中的虚拟机不会同时使用 LUN，这可能会导致数据崩溃。

2.6. OPENSTACK 的先决条件

以下先决条件适用于 OpenStack 迁移：

- 您必须使用 兼容版本的 OpenStack。

2.6.1. 使用 OpenStack 源供应商迁移的其他验证方法

MTV 版本 2.6 及更新的版本除了标准用户名和密码凭证集外，还支持使用 OpenStack 源供应商迁移的以下验证方法：

- 令牌身份验证
- 应用程序凭证身份验证

您可以使用这些方法通过 CLI 将虚拟机与 OpenStack 源供应商迁移，这与迁移其他虚拟机的方式相同，除非您如何准备 **Secret** 清单。

2.6.1.1. 将令牌身份验证与 OpenStack 源提供程序搭配使用

在创建 OpenStack 源供应商时，您可以使用令牌身份验证，而不是用户名和密码身份验证。

MTV 支持以下令牌身份验证类型：

- 使用用户 ID 的令牌
- 使用用户名令牌

对于每种类型的令牌身份验证，您需要使用 OpenStack 中的数据来创建 **Secret** 清单。

先决条件

具有 OpenStack 帐户。

流程

1. 在 OpenStack web 控制台的仪表板中，点 **Project > API Access**。
2. 扩展 **Download OpenStack RC 文件** 并点击 **OpenStack RC 文件**。
   下载的文件，这里称为 < **openstack_rc_file** >，包括用于令牌身份验证的以下字段：

   ```
   OS_AUTH_URL
   OS_PROJECT_ID
   OS_PROJECT_NAME
   OS_DOMAIN_NAME
   OS_USERNAME
   ```

3. 要获取令牌身份验证所需的数据，请运行以下命令：

   ```
   $ openstack token issue
   ```

   此处称为 < **openstack_ token _output&** gt; 的输出，包括使用具有 **用户 ID** 的令牌进行身份验证的令牌、userID 和 **projectID**。

4. 创建类似如下的 **Secret** 清单：
   - 对于使用用户 ID 的令牌进行身份验证：

     ```
     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
     ```

- 对于使用具有用户名的令牌进行身份验证：

```
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. 根据迁移虚拟机的流程，从第 2 步开始，"为源供应商创建 **提供程序** 清单"开始，继续 迁移您的虚拟机。

2.6.1.2. 将应用凭据身份验证与 OpenStack 源提供程序搭配使用

在创建 OpenStack 源供应商时，您可以使用应用程序凭据身份验证而不是用户名和密码身份验证。

MTV 支持以下应用程序凭证身份验证类型：

- 应用程序凭证 ID
- 应用程序凭证名称

对于每种类型的应用程序凭据身份验证，您需要使用 OpenStack 中的数据来创建 **Secret** 清单。

先决条件

您有一个 OpenStack 帐户。

流程

1. 在 OpenStack web 控制台的仪表板中，点 **Project** > **API Access**。
2. 扩展 **Download OpenStack RC 文件** 并点击 **OpenStack RC 文件**。
   下载的文件，这里称为 < **openstack_rc_file&** gt;，包括用于应用程序凭证身份验证的以下字段：

```
OS_AUTH_URL
OS_PROJECT_ID
OS_PROJECT_NAME
OS_DOMAIN_NAME
OS_USERNAME
```

3. 要获取应用程序凭证身份验证所需的数据，请运行以下命令：

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

此处的输出称为 < **openstack_credential_output>**，包括：

- 使用应用程序凭证 ID 进行身份验证的 **id** 和 **secret**
- 使用应用程序凭证 **名称** 进行身份验证所需的名称和 **secret**

4. 创建类似如下的 **Secret** 清单：

- 使用应用程序凭证 ID 进行身份验证：

```
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
```

- 使用应用程序凭证名称进行身份验证：

```
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. 根据迁移虚拟机的流程，从第 2 步开始，"为源供应商创建 **提供程序** 清单"开始，继续 迁移您的虚拟机。

2.7. VMWARE 的先决条件

强烈建议您创建 VDDK 镜像来加快迁移。如需更多信息，请参阅创建 VDDK 镜像。

以下先决条件适用于 VMware 迁移：

- 您必须使用兼容版本的 VMware vSphere 。
- 您必须以至少最小组的 VMware 权限的用户身份登录。
- 要使用预迁移 hook 访问虚拟机，必须在源虚拟机上安装 VMware 工具。
- 虚拟机操作系统必须经过认证并支持作为 带有 OpenShift Virtualization 的客户机操作系统使用以及 使用 **virt-v2v**转换为 KVM。
- 如果您正运行 warm 迁移，则必须在虚拟机和 VM 磁盘中启用 已更改的块跟踪 (CBT)。
- 如果您要从同一迁移计划中的 ESXi 主机迁移超过 10 个虚拟机，您必须增加主机的 NFC 服务内存。
- 强烈建议您禁用休眠功能，因为 Migration Toolkit for Virtualization (MTV)不支持迁移休眠虚拟机。

重要

如果出现电源中断，则禁用休眠的虚拟机可能会丢失数据。但是，如果没有禁用休眠，迁移将失败

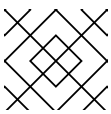
注意

MTV 和 OpenShift Virtualization 都不支持从 VMWare 迁移虚拟机转换。

VMware 权限

要使用 Migration Toolkit for Virtualization (MTV) 将虚拟机迁移到 OpenShift Virtualization 需要以下最小 VMware 权限集合。

表 2.4. VMware 权限

特权	描述
Virtual machine.Interaction 权限：	
Virtual machine.Interaction.Power Off	允许关闭一个已启动的虚拟机。此操作关闭客户端操作系统。
Virtual machine.Interaction.Power On	允许打开已关闭的虚拟机并恢复暂停的虚拟机。
Virtual machine.Provisioning 权限：  **注意** 需要所有 **Virtual machine.Provisioning** 权限。	
Virtual machine.Provisioning.Allow disk access	允许在虚拟机上打开磁盘以实现随机读写访问。主要用于远程磁盘挂载。

特权	描述
Virtual machine.Provisioning.Allow file access	允许操作与虚拟机相关的文件，包括 VMX、磁盘、日志和 NVRAM。
Virtual machine.Provisioning.Allow read-only disk access	允许在虚拟机上打开一个磁盘以实现随机读取访问。主要用于远程磁盘挂载。
Virtual machine.Provisioning.Allow virtual machine download	允许对与虚拟机关联的文件读取操作，包括 VMX、磁盘、日志和 NVRAM。
Virtual machine.Provisioning.Allow virtual machine files upload	允许对与虚拟机关联的文件写操作，包括 VMX、磁盘、日志和 NVRAM。
Virtual machine.Provisioning.Clone template	允许克隆模板。
Virtual machine.Provisioning.Clone virtual machine	允许克隆现有虚拟机并分配资源。
Virtual machine.Provisioning.Create template from virtual machine	允许从虚拟机创建新模板。
Virtual machine.Provisioning.Customize guest	允许自定义虚拟机的客户机操作系统，而无需移动虚拟机。
Virtual machine.Provisioning.Deploy template	允许从模板部署虚拟机。
Virtual machine.Provisioning.Mark as template	允许将现有的关机虚拟机标记为模板。
Virtual machine.Provisioning.Mark as virtual machine	允许将现有模板标记为虚拟机。
Virtual machine.Provisioning.Modify customization specification	允许创建、修改或删除自定义规格。
Virtual machine.Provisioning.Promote disks	允许对虚拟机磁盘提升操作。
Virtual machine.Provisioning.Read customization specifications	允许读取自定义规格。
Virtual machine.Snapshot management 权限：	
Virtual machine.Snapshot management.Create snapshot	允许从虚拟机当前状态创建快照。

特权	描述
Virtual machine.Snapshot management.Remove Snapshot	允许从快照历史记录中移除快照。

2.7.1. 创建 VDDK 镜像

Migration Toolkit for Virtualization (MTV)使用 VMware Virtual Disk Development Kit (VDDK) SDK 来加快从 VMware vSphere 传输虚拟磁盘。因此，强烈建议创建 VDDK 镜像（可选）。

要使用这个功能，您可以下载 VMware Virtual Disk Development Kit (VDDK)，构建 VDDK 镜像，并将 VDDK 镜像推送到您的镜像 registry。

VDDK 软件包包含符号链接，因此创建 VDDK 镜像的步骤必须在保留符号链接(symlinks)的文件系统上执行。

注意

在公共 registry 中存储 VDDK 镜像可能会违反 VMware 许可证条款。

先决条件

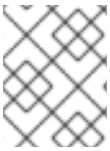
- Red Hat OpenShift 镜像 registry。
- 已安装 **podman**。
- 您正在使用一个保留符号链接(symlinks)的文件系统。
- 如果使用外部 registry，OpenShift Virtualization 必须能够访问它。

流程

1. 创建并导航到临时目录：

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

2. 在浏览器中，进入 VMware VDDK 版本 8 下载页面。
3. 选择版本 8.0.1，再点 **Download**。



注意

要迁移到 OpenShift Virtualization 4.12，请从 VMware VDDK 版本 7 下载 VDDK 版本 7.0.3.2。

1. 将 VDDK 归档文件保存到临时目录中。
2. 提取 VDDK 归档：

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

3. 创建 **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
```

4. 构建 VDDK 镜像：

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

5. 将 VDDK 镜像推送到 registry：

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

6. 确保镜像可以被 OpenShift Virtualization 环境访问。

2.7.2. 增加 ESXi 主机的 NFC 服务内存

如果您要从同一迁移计划中的 ESXi 主机迁移超过 10 个虚拟机，您必须增加主机的 NFC 服务内存。否则，迁移将失败，因为 NFC 服务内存仅限于 10 个并行连接。

流程

1. 以 root 用户身份登录 ESXi 主机。
2. 在 **/etc/vmware/hostd/config.xml** 中将 **maxMemory** 改为 **1000000000**:

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

3. 重启 **hostd** ：

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

   您不需要重启主机。

2.8. 开放虚拟设备(OVA)先决条件

以下先决条件适用于开放虚拟设备(OVA)文件迁移：

- 所有 OVA 文件都由 VMware vSphere 创建。

注意

不是由 VMware vSphere 创建但与 vSphere 兼容的 OVA 文件迁移可能会成功。但是 MTV 不支持迁移此类文件。MTV 仅支持由 VMware vSphere 创建的 OVA 文件。

- OVA 文件位于以下结构之一 NFS 共享目录下的一个或多个文件夹中：
  - 在包含所有虚拟机信息的一个或多个压缩的 Open Virtualization Format (OVF)软件包中。每个压缩的软件包的文件名 **都必须具有 .ova** 扩展。多个压缩的软件包可以存储在同一文件夹中。

    当使用此结构时，MTV 会扫描根文件夹以及压缩的软件包的第一级子文件夹。

    例如，如果 NFS 共享是 **/nfs**，则：
    文件夹 **/nfs** 被扫描。
    文件夹 **/nfs/subfolder1** 被扫描。
    但是 **/nfs/subfolder1/subfolder2** 不会被扫描。

  - 在提取的 OVF 软件包中。
    当使用此结构时，MTV 会扫描根文件夹、第一级子文件夹和用于提取的 OVF 软件包的第二个子文件夹。但是，文件夹中只能有一个 **.ovf** 文件。否则，迁移将失败。

    例如，如果 NFS 共享是 **/nfs**，则
    OVF 文件 **/nfs/vm.ovf** 被扫描。
    OVF 文件 **/nfs/subfolder1/vm.ovf** 被扫描。
    OVF 文件 **/nfs/subfolder1/subfolder2/vm.ovf** 被扫描。
    但是，OVF 文件 **/nfs/subfolder1/subfolder2/subfolder3/vm.ovf** 不会被扫描。

2.9. 软件兼容性指南

您必须安装兼容软件版本。

表 2.5. 兼容软件版本

Migration Toolkit for Virtualization	Red Hat OpenShift	OpenShift Virtualization	VMware vSphere	Red Hat Virtualization	OpenStack
2.6.2	4.14 或更高版本	4.14 或更高版本	6.5 或更高版本	4.4 SP1 或更高版本	16.1 或更高版本

从 RED HAT VIRTUALIZATION 4.3 迁移

MTV 2.6 仅使用 Red Hat Virtualization (RHV) 4.4 SP1 测试。从 Red Hat Virtualization (RHV) 4.3 的迁移没有使用 MTV 2.6 测试。虽然不支持，但可以从 RHV 4.3 进行的基本迁移应该可以正常工作。

因为 RHV 4.3 缺少了 RHV 4.4 for MTV 中引入的改进，新的功能没有在 RHV 4.3 中测试，与从 RHV 4.4 迁移的迁移可能无法正常工作，但一些功能可能会缺失。

因此，建议在迁移到 OpenShift Virtualization 前将 RHV 升级到上面支持的版本。

但是，从 RHV 4.3.11 迁移使用 MTV 2.3 测试，在许多使用 MTV 2.6 的环境中可能会工作。在这种情况下，我们建议将 Red Hat Virtualization Manager (RHVM)升级到前面提到的支持的版本，然后再迁移到 OpenShift Virtualization。

2.9.1. OpenShift Operator 生命周期

有关红帽提供的用于 OpenShift Container Platform 的软件维护生命周期分类的更多信息，请参阅 OpenShift Operator 生命周期。

第 3 章 安装和配置 MTV OPERATOR

您可以使用 Red Hat OpenShift Web 控制台或命令行界面(CLI)安装 MTV Operator。

在 Migration Toolkit for Virtualization (MTV)版本 2.4 及更高版本中，MTV Operator 包括 Red Hat OpenShift Web 控制台的 MTV 插件。

使用 Red Hat OpenShift Web 控制台或 CLI 安装 MTV Operator 后，您可以配置 Operator。

3.1. 使用 RED HAT OPENSHIFT WEB 控制台安装 MTV OPERATOR

您可以使用 Red Hat OpenShift Web 控制台安装 MTV Operator。

先决条件

- 安装了 Red Hat OpenShift 4.14 或更高版本。
- 在 OpenShift 迁移目标集群上安装的 OpenShift Virtualization Operator。
- 您必须以具有 **cluster-admin** 权限的用户身份登录。

流程

1. 在 Red Hat OpenShift Web 控制台中，点 **Operators → OperatorHub**。
2. 使用 **Filter by keyword** 字段搜索 **mtv-operator**。
3. 点 **Migration Toolkit for Virtualization Operator**，然后点 **Install**。
4. 当按钮处于活跃状态时，点 **Create ForkliftController**。
5. 点 **Create**。
   您的 ForkliftController 会出现在显示的列表中。
6. 点 **Workloads → Pods** 来验证 MTV pod 是否正在运行。
7. 点 **Operators → Installed Operators** 来验证 **Migration Toolkit for Virtualization Operator**是否出现在 **openshift-mtv** 项目中，其状态为 **Succeeded**。
   当插件就绪时，系统会提示您重新加载页面。**Migration** 菜单项会自动添加到导航栏中，显示在 Red Hat OpenShift Web 控制台左侧。

3.2. 使用命令行界面安装 MTV OPERATOR

您可以使用命令行界面 (CLI) 安装 MTV Operator。

先决条件

- 安装了 Red Hat OpenShift 4.14 或更高版本。
- 在 OpenShift 迁移目标集群上安装的 OpenShift Virtualization Operator。
- 您必须以具有 **cluster-admin** 权限的用户身份登录。

流程

1. 创建 openshift-mtv 项目：

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

2. 创建名为 **migration** 的 **OperatorGroup** CR：

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

3. 为 Operator 创建 **Subscription** CR：

```
$ cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: mtv-operator
  namespace: openshift-mtv
spec:
  channel: release-v2.6
  installPlanApproval: Automatic
  name: mtv-operator
  source: redhat-operators
  sourceNamespace: openshift-marketplace
  startingCSV: "mtv-operator.v2.6.2"
EOF
```

4. 创建一个 **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. 验证 MTV pod 是否正在运行：

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

输出示例

```
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. 配置 MTV OPERATOR

您可以通过修改 **ForkliftController** CR，或在 Overview 页面的 **Settings** 部分中，或在 **Overview** 页面的 Settings 部分来配置 MTV Operator 的所有设置，除非另有说明。

- 每个计划可同时迁移的最大虚拟机(VM)数量。
- 在自动删除前，**必须保留收集** 报告的时间。
- 分配给主控制器容器的 CPU 限制。
- 分配给主控制器容器的内存限值。
- 在启动温迁移前请求新快照的时间间隔。
- 系统在温迁移期间检查快照创建或删除状态的频率。
- 当 **storageclass** 是 **文件系统**时，作为文件系统开销分配的持久性卷中的空间百分比（仅限**ForkliftController** CR）。
- 修复了持久块卷中分配的额外空间量。此设置适用于任何基于块的 **storageclass** （仅适用于**ForkliftController** CR）。
- 操作系统的配置映射，用于 vSphere 源供应商（仅限**ForkliftController** CR）。
- 操作系统的配置映射，用于 Red Hat Virtualization (RHV)源供应商（仅限**ForkliftController** CR）。

使用用户界面配置这些设置的步骤请参考 配置 MTV 设置。修改 **ForkliftController** CR 配置这些设置的步骤如下。

流程

- 通过添加标签和值来更改 **ForkliftController** CR 的 **spec** 部分中的参数值，如下所示：

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

(1) 下表中显示了您可以使用 CLI 配置的标签，以及每个标签及其默认值的描述。

表 3.1. MTV Operator 标签

标签	描述	默认值
controller_max_vm_inflight	每个计划可同时迁移的最大虚拟机数量。	**20**
must_gather_api_cleanup_max_age	保留 **必须收集** 报告的时间（以小时为单位），然后再自动删除报告。	**-1** （禁用）
controller_container_limits_cpu	分配给主控制器容器的 CPU 限制。	**500m**
controller_container_limits_memory	分配给主控制器容器的内存限值。	**800Mi**
controller_precopy_interval	在启动温迁移前请求新快照的时间间隔（以分钟为单位）。	**60**
controller_snapshot_status_check_rate_seconds	系统在温迁移期间检查快照创建或删除状态的频率（以秒为单位）。	**10**
controller_filesystem_overhead	当 **storageclass** 是 **文件系统** 时，作为文件系统开销分配的持久性卷中的空间百分比。 仅限 **ForkliftController** CR。	**10**
controller_block_overhead	修复了持久块卷中分配的额外空间量。此设置适用于任何基于块的 **storageclass**。当数据（如加密标头）写入持久性卷之外，还可以使用它。 仅限 **ForkliftController** CR。	**0**

标签	描述	默认值
vsphere_osmap_configmap_name	vSphere 源供应商的配置映射。此配置映射将传入虚拟机的操作系统映射到 OpenShift Virtualization 首选项名称。此配置映射需要位于部署 MTV Operator 的命名空间。 要查看 OpenShift Virtualization 环境中的首选项列表，请打开 OpenShift Web 控制台并点 **Virtualization → Preferences**。 当此标签具有默认值 **forklift-vsphere-osmap** 时，**您可以在配置映射中添加值。**要覆盖或删除值，请指定与 **forklift-vsphere-osmap** 不同的配置映射。 仅限 **ForkliftController** CR。	**forklift-vsphere-osmap**
ovirt_osmap_configmap_name	RHV 源提供程序的配置映射。此配置映射将传入虚拟机的操作系统映射到 OpenShift Virtualization 首选项名称。此配置映射需要位于部署 MTV Operator 的命名空间。 要查看 OpenShift Virtualization 环境中的首选项列表，请打开 OpenShift Web 控制台并点 **Virtualization → Preferences**。 当此标签具有默认值 **forklift-ovirt-osmap** 时，**您可以在配置映射中添加值。**要覆盖或删除值，请指定与 **forklift-ovirt-osmap** 不同的配置映射。 仅限 **ForkliftController** CR。	**forklift-ovirt-osmap**

第 4 章 使用 RED HAT OPENSHIFT WEB 控制台迁移虚拟机

您可以使用 Red Hat OpenShift web 控制台迁移虚拟机：

1. 创建源供应商
2. 创建目的地供应商
3. 创建迁移计划

重要

您必须确保 满足所有先决条件。

仅 VMware：您必须拥有最少的 VMware 权限集合。

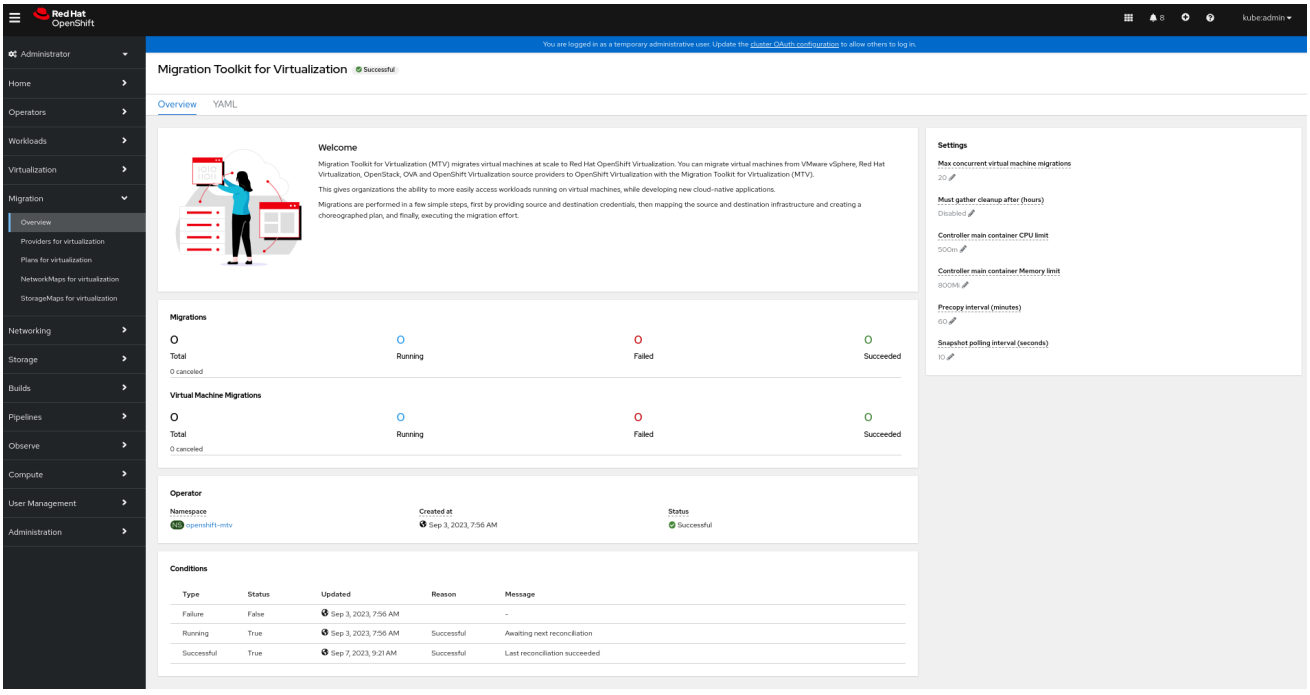
仅 VMware：创建 VMware Virtual Disk Development Kit (VDDK) 镜像将提高迁移速度。

4.1. MTV 用户界面

Migration Toolkit for Virtualization (MTV)用户界面集成到 OpenShift Web 控制台中。

在左侧面板中，您可以选择与迁移进度组件相关的页面，例如：迁移的供应商，或者如果您是管理员，您可以选择 **Overview**，其中包含有关迁移的信息，并可让您配置 MTV 设置。

图 4.1. MTV 扩展接口



在与组件相关的页面中，您可以点击 Projects 列表（位于页面的左上部分中），并查看您可以使用哪些项目（命名空间）。

- **如果是管理员，您可以查看所有项目。**
- **如果您是非管理员用户，只能看到您有权使用的项目。**

4.2. MTV OVERVIEW 页

Migration Toolkit for Virtualization (MTV) Overview 页显示有关迁移的系统范围信息，以及您可以更改的 Settings 列表。

如果您有 Administrator 权限，点 Red Hat OpenShift web 控制台中的 Migration → Overview 来访问 Overview 页面。

Overview 页有 3 个标签页：

- 概述
- YAML
- 指标

4.2.1. 概述标签

Overview 选项卡可让您看到：

- Operator：部署 MTV Operator 的命名空间以及 Operator 的状态
- Pods： MTV Operator 部署的每个 pod 的名称、状态和创建时间
- conditions: MTV Operator 的状态：
  - 失败：最后失败。false 表示自部署以来没有失败。
  - Running: Operator 当前是否正在运行并等待下一个协调。
  - 成功 ：last successful 协调。

4.2.2. YAML 标签页

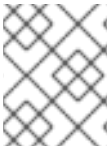
定义 MTV Operator 操作的自定义资源 ForkliftController。您可以从此标签页修改自定义资源。

4.2.3. Metrics 标签页

Metrics 选项卡可让您看到：

- Migration：使用 MTV 执行的迁移数量：
  - 总计
  - Running
  - Failed
  - Succeeded
  - 已取消
- 虚拟机迁移： 使用 MTV 迁移的虚拟机数量：
  - 总计
  - Running
  - Failed

- Succeeded
- 已取消



注意

因为一个迁移可能涉及很多虚拟机，使用 MTV 执行的迁移数量可能会与使用 MTV 迁移的虚拟机数量有很大不同。

- 显示在最后 7 天使用 MTV 执行运行、失败和成功迁移的数量
- 显示在最后 7 天使用 MTV 执行的运行、失败和成功虚拟机迁移的数量

4.3. 配置 MTV 设置

如果您有 Administrator 权限，您可以访问 Overview 页面并更改其中的以下设置：

表 4.1. MTV 设置

设置	描述	默认值
最大并发虚拟机迁移	每个计划可同时迁移的最大虚拟机数量	20
必须在之后收集清理（小时）	保留报告的持续时间 **必须收集** 报告，然后才能自动删除	Disabled
控制器主容器 CPU 限制	分配给主控制器容器的 CPU 限制	500 M
控制器主容器内存限制	分配给主控制器容器的内存限值	800 Mi
预复制内部（分钟）	在启动温迁移前请求新快照的时间间隔	60
快照轮询间隔（秒）	系统在温迁移过程中检查快照创建或删除状态的频率	10

流程

1. 在 Red Hat OpenShift Web 控制台中，点 Migration → Overview。Settings 列表位于页面的右侧。
2. 在 Settings 列表中，点您要更改的设置的 Edit 图标。
3. 从列表中选择一个设置。
4. 点击 Save。

4.4. 添加供应商

您可以使用 Red Hat OpenShift web 控制台为虚拟机迁移添加源供应商和目的地供应商。

4.4.1. 添加源供应商

您可以使用 MTV 从以下源供应商迁移虚拟机：

- VMware vSphere
- Red Hat Virtualization
- OpenStack
- 由 VMware vSphere 创建的开放虚拟设备(OVA)
- OpenShift Virtualization

您可以使用 Red Hat OpenShift Web 控制台添加源供应商。

4.4.1.1. 添加 VMware vSphere 源供应商

您可以从 VMware vCenter 或 VMWare ESX/ESXi 服务器迁移 VMware vSphere 虚拟机。在 MTV 版本 2.6 及更高版本中，您可以通过指定 ESX/ESXi 服务器来直接从 ESX/ESXi 服务器迁移，而无需通过 vCenter 将 SDK 端点指定到 ESX/ESXi 服务器。

重要

使用 VMware vSphere 源供应商迁移 EMS 强制被禁用，以便启用从 Migration Toolkit for Virtualization 支持的 vSphere 版本进行迁移，但不遵循 2023 年 FIPS 要求。因此，用户是否应该考虑从 vSphere 源供应商迁移是否存在其符合 FIPS 的风险。支持的 vSphere 版本在 软件兼容性指南 中指定。

先决条件

- 强烈建议您在所有集群可访问的安全 registry 中创建 VMware Virtual Disk Development Kit (VDDK)镜像。VDDK 镜像可加快迁移。如需更多信息，请参阅创建 VDDK 镜像。

流程

1. 在 Red Hat OpenShift web 控制台中，点 Migration → Providers for virtualization。
2. 单击 Create Provider。
3. 点 vSphere。
4. 指定以下字段：

供应商详情

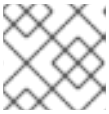
- Provider resource name：源提供程序的名称。
- 端点类型 ：选择 vSphere 供应商端点类型。选项：vCenter 或 ESXi。您可以从 vCenter 迁移虚拟机，这是不由 vCenter 管理的 ESX/ESXi 服务器，或者从由 vCenter 管理的 ESX/ESXi 服务器迁移，但不会通过 vCenter 进行管理。
- URL ：挂载源虚拟机的 vCenter 的 SDK 端点的 URL。确保 URL 包含sdk 路径，通常为 /sdk。例如： https://vCenter-host-example.com/sdk。如果指定了 FQDN 的证书，则此字段的值需要与证书中的 FQDN 匹配。

- VDDK init 镜像：**VDDKInitImage** 路径。强烈建议您创建 VDDK init 镜像来加快迁移。如需更多信息，请参阅创建 VDDK 镜像。

供应商详情

- 用户名 ： vCenter 用户或 ESXi 用户。例如：user@vsphere.local。
- 密码 ： vCenter 用户密码或 ESXi 用户密码。

1. 选择以下选项之一来验证 CA 证书：
   - 使用自定义 CA 证书 ：验证自定义 CA 证书后迁移。
   - 使用系统 CA 证书 ：验证系统 CA 证书后迁移。
   - 跳过证书验证 ：在没有验证 CA 证书的情况下迁移。
     a. 要使用自定义 CA 证书，请保留 Skip certificate validation switch 切换到 left，并将 CA 证书拖到 *文本框中*，*或者* 浏览它并点击 Select。
     b. 要使用系统 CA 证书，请保留 Skip certificate validation 开关切换到左侧，并将 CA 证书 文本框留空。
     c. 要跳过证书验证，请将 Skip certificate validation 开关切换到右侧。
2. 可选：一个sk MTV 用来从供应商的 API 端点 URL 获取自定义 CA 证书。
   a. 从 URL 单击 Fetch 证书。此时会打开 Verify certificate 窗口。
   b. 如果详情正确，请选择 I trust the authenticity of this certificate复选框，然后点 Confirm。如果没有，请单击 Cancel，然后手动输入正确的证书信息。
   确认后，将使用 CA 证书来验证与 API 端点的后续通信。
3. 点 Create provider 添加并保存该提供程序。
   供应商会出现在提供程序列表中。



注意

可能需要过几分钟后，供应商才会处于 Ready 状态。

4.4.1.1.1. 为 VMware 源供应商选择迁移网络

您可以在 Red Hat OpenShift Web 控制台中为源供应商选择迁移网络，以降低源环境的风险并提高性能。

将默认网络用于迁移可能会导致性能降低，因为网络可能没有足够的带宽。这种情形可以对源平台产生负面影响，因为磁盘传输操作可能会使网络饱和。

先决条件

- 迁移网络必须具有足够的吞吐量，最小 10 Gbps 的速度用于磁盘传输。
- OpenShift Virtualization 节点必须可通过默认网关访问迁移网络。

注意

源虚拟磁盘由连接到目标命名空间的 pod 网络的 pod 复制。

- 迁移网络必须启用巨型帧。

流程

1. 在 Red Hat OpenShift web 控制台中，点 Migration → Providers for virtualization。
2. 点供应商旁边的 Hosts 列中的主机编号，以查看主机列表。
3. 选择一个或多个主机并点击 Select migration network。
4. 指定以下字段：
   - 网络 ：网络名称
   - ESXi 主机 admin 用户名 ：例如 root
   - ESXi host admin password: Password
5. 点击 Save。
6. 验证每个主机的状态是否为 Ready。
   如果主机状态为 Ready，则在迁移网络上可能无法访问主机，或者凭证可能不正确。您可以修改主机配置并保存更改。

4.4.1.2. 添加 Red Hat Virtualization 源供应商

您可以使用 Red Hat OpenShift Web 控制台添加 Red Hat Virtualization 源供应商。

先决条件

- Manager CA 证书，除非它被第三方证书替代，在这种情况下，指定 Manager Apache CA 证书

流程

1. 在 Red Hat OpenShift web 控制台中，点 Migration → Providers for virtualization。
2. 单击 Create Provider。
3. 点 Red Hat Virtualization
4. 指定以下字段：
   - Provider resource name：源提供程序的名称。
   - URL ：挂载源虚拟机的 Red Hat Virtualization Manager (RHVM)的 API 端点的 URL。确保 URL 包含组成 RHVM API 服务器的路径，通常为 /ovirt-engine/api。例如： https://rhv-host-example.com/ovirt-engine/api。
   - 用户名 ：用户名。
   - 密码 ：密码.
5. 选择以下选项之一来验证 CA 证书：
   - 使用自定义 CA 证书 ：验证自定义 CA 证书后迁移。
   - 使用系统 CA 证书 ：验证系统 CA 证书后迁移。

- **跳过证书验证 ：在没有验证 CA 证书的情况下迁移。**

  a. **要使用自定义 CA 证书，请保留 Skip certificate validation switch 切换到 left，并将 CA 证书拖到 *文本框中，或者* 浏览它并点击 Select。**

  b. **要使用系统 CA 证书，请保留 Skip certificate validation 开关切换到左侧，并将 CA 证书文本框留空。**

  c. **要跳过证书验证，请将 Skip certificate validation 开关切换到右侧。**

6. **可选：一个sk MTV 用来从供应商的 API 端点 URL 获取自定义 CA 证书。**

   a. **从 URL 单击 Fetch 证书。此时会打开 Verify certificate 窗口。**

   b. **如果详情正确，请选择 I trust the authenticity of this certificate复选框，然后点 Confirm。如果没有，请单击 Cancel，然后手动输入正确的证书信息。**
   确认后，将使用 CA 证书来验证与 API 端点的后续通信。

7. **点 Create provider 添加并保存该提供程序。**
   供应商会出现在提供程序列表中。

4.4.1.3. 添加 OpenStack 源供应商

您可以使用 Red Hat OpenShift Web 控制台添加 OpenStack 源供应商。

重要

从 OpenStack 供应商迁移基于镜像的虚拟机时，会为附加到源虚拟机的镜像创建快照，快照中的数据将复制到目标虚拟机上。这意味着，在创建快照时，目标虚拟机的状态与源虚拟机的状态相同。

流程

1. **在 Red Hat OpenShift web 控制台中，点 Migration → Providers for virtualization。**
2. **单击 Create Provider。**
3. **单击 OpenStack。**
4. **指定以下字段：**
   - **Provider resource name ：源提供程序的名称。**
   - **URL ：OpenStack 身份(Keystone)端点的 URL。例如： http://controller:5000/v3。**
   - **身份验证类型 ：选择以下身份验证方法之一，并提供与您选择的信息。例如，如果您选择 Application credential ID 作为身份验证类型，应用程序凭证 ID 和 Application credential secret 字段将变为活跃，您需要提供 ID 和 secret。**
     - **应用程序凭证 ID**
       - **应用程序凭证 ID: OpenStack 应用凭证 ID**
       - **应用程序凭证 secret ：OpenStack https://github.com/kubev2v/forklift-documentation/pull/402pplication 凭证 Secret**
     - **应用程序凭证名称**

- 应用程序凭证名称 ：OpenStack 应用程序凭证名称
- 应用程序凭证 secret ： ： OpenStack 应用程序凭证Secret
- 用户名 ：OpenStack 用户名
- 域 ：OpenStack 域名

- 使用用户 ID 的令牌
  - 令牌 ：OpenStack 令牌
  - 用户 ID ：OpenStack 用户 ID
  - 项目 ID ：OpenStack 项目 ID
- 使用用户名的令牌
  - 令牌 ：OpenStack 令牌
  - 用户名 ：OpenStack 用户名
  - 项目 ：OpenStack 项目
  - 域名 ：OpenStack 域名
- 密码
  - 用户名 ：OpenStack 用户名
  - 密码 ：OpenStack 密码
  - 项目 ：OpenStack 项目
  - 域 ：OpenStack 域名

5. 选择以下选项之一来验证 CA 证书：
   - 使用自定义 CA 证书 ：验证自定义 CA 证书后迁移。
   - 使用系统 CA 证书 ：验证系统 CA 证书后迁移。
   - 跳过证书验证 ：在没有验证 CA 证书的情况下迁移。
     a. 要使用自定义 CA 证书，请保留 Skip certificate validation switch 切换到 left，并将 CA 证书拖到 *文本框中，或者* 浏览它并点击 Select。
     b. 要使用系统 CA 证书，请保留 Skip certificate validation 开关切换到左侧，并将 CA 证书文本框留空。
     c. 要跳过证书验证，请将 Skip certificate validation 开关切换到右侧。
6. 可选：一个sk MTV 用来从供应商的 API 端点 URL 获取自定义 CA 证书。
   a. 从 URL 单击 Fetch 证书。此时会打开 Verify certificate 窗口。
   b. 如果详情正确，请选择 I trust the authenticity of this certificate复选框，然后点 Confirm。如果没有，请单击 Cancel，然后手动输入正确的证书信息。
   确认后，将使用 CA 证书来验证与 API 端点的后续通信。

7. 点 Create provider 添加并保存该提供程序。
   供应商会出现在提供程序列表中。

4.4.1.4. 添加开放虚拟设备(OVA)源供应商

您可以使用 Red Hat OpenShift Web 控制台添加由 VMware vSphere 创建的 Open Virtual Appliance (OVA)文件作为源供应商。

流程

1. 在 Red Hat OpenShift web 控制台中，点 Migration → Providers for virtualization。
2. 单击 Create Provider。
3. 单击 Open Virtual Appliance (OVA)。
4. 指定以下字段：
   - Provider resource name：源供应商的名称
   - URL ：提供 OVA 的 NFS 文件共享的 URL
5. 点 Create provider 添加并保存该提供程序。
   供应商会出现在提供程序列表中。

注意

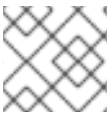
可能会出现错误消息，说明发生了错误。您可以忽略此消息。

4.4.1.5. 添加 Red Hat OpenShift Virtualization 源供应商

您可以使用 Red Hat OpenShift Virtualization 供应商作为源供应商和目的地供应商。

具体来说，自动添加为 OpenShift Virtualization 供应商的主机集群可用作源供应商和目标供应商。

您可以将虚拟机从 deployed on 的集群迁移到另一个集群，或者从远程集群迁移到部署了 MTV 的集群。



注意

源供应商的 Red Hat OpenShift 集群版本必须是 4.13 或更高版本。

流程

1. 在 Red Hat OpenShift web 控制台中，点 Migration → Providers for virtualization。
2. 单击 Create Provider。
3. 点击 OpenShift Virtualization。
4. 指定以下字段：
   - Provider resource name：源供应商的名称
   - URL ：API 服务器端点的 URL
   - 服务帐户 bearer 令牌 ：具有 cluster-admin 特权的服务帐户的令牌

如果 URL 和 Service account bearer 令牌都留空，则使用本地 OpenShift 集群。

5. 选择以下选项之一来验证 CA 证书：
   - 使用自定义 CA 证书 ：验证自定义 CA 证书后迁移。
   - 使用系统 CA 证书 ：验证系统 CA 证书后迁移。
   - 跳过证书验证 ：在没有验证 CA 证书的情况下迁移。
     a. 要使用自定义 CA 证书，请保留 Skip certificate validation switch 切换到 left，并将 CA 证书拖到 *文本框中，或者* 浏览它并点击 Select。
     b. 要使用系统 CA 证书，请保留 Skip certificate validation 开关切换到左侧，并将 CA 证书文本框留空。
     c. 要跳过证书验证，请将 Skip certificate validation 开关切换到右侧。
6. 可选：一个sk MTV 用来从供应商的 API 端点 URL 获取自定义 CA 证书。
   a. 从 URL 单击 Fetch 证书。此时会打开 Verify certificate 窗口。
   b. 如果详情正确，请选择 I trust the authenticity of this certificate复选框，然后点 Confirm。如果没有，请单击 Cancel，然后手动输入正确的证书信息。
   确认后，将使用 CA 证书来验证与 API 端点的后续通信。
7. 点 Create provider 添加并保存该提供程序。
   供应商会出现在提供程序列表中。

4.4.2. 添加目的地供应商

您可以使用 Red Hat OpenShift Web 控制台添加 OpenShift Virtualization 目的地供应商。

4.4.2.1. 添加 OpenShift Virtualization 目的地供应商

您可以使用 Red Hat OpenShift Virtualization 供应商作为源供应商和目的地供应商。

具体来说，自动添加为 OpenShift Virtualization 供应商的主机集群可用作源供应商和目标供应商。

除了默认的 OpenShift Virtualization 目的地供应商，您还可以在 Red Hat OpenShift web 控制台中添加另一个 OpenShift Virtualization 目的地供应商，这是您安装 MTV 的集群。

您可以将虚拟机从 deployed on 的集群迁移到另一个集群，或者从远程集群迁移到部署了 MTV 的集群。

先决条件

- 您必须具有具有 **cluster-admin** 权限的 OpenShift Virtualization服务帐户令牌。

流程

1. 在 Red Hat OpenShift web 控制台中，点 Migration → Providers for virtualization。
2. 单击 Create Provider。
3. 点击 OpenShift Virtualization。
4. 指定以下字段：

- Provider resource name：源供应商的名称
- URL ：API 服务器端点的 URL
- 服务帐户 bearer 令牌 ：具有 cluster-admin 特权的服务帐户的令牌
  如果 URL 和 Service account bearer 令牌都留空，则使用本地 OpenShift 集群。

5. 选择以下选项之一来验证 CA 证书：
   - 使用自定义 CA 证书 ：验证自定义 CA 证书后迁移。
   - 使用系统 CA 证书 ：验证系统 CA 证书后迁移。
   - 跳过证书验证 ：在没有验证 CA 证书的情况下迁移。
     a. 要使用自定义 CA 证书，请保留 Skip certificate validation switch 切换到 left，并将 CA 证书拖到 *文本框中，或者* 浏览它并点击 Select。
     b. 要使用系统 CA 证书，请保留 Skip certificate validation 开关切换到左侧，并将 CA 证书文本框留空。
     c. 要跳过证书验证，请将 Skip certificate validation 开关切换到右侧。
6. 可选：一个sk MTV 用来从供应商的 API 端点 URL 获取自定义 CA 证书。
   a. 从 URL 单击 Fetch 证书。此时会打开 Verify certificate 窗口。
   b. 如果详情正确，请选择 I trust the authenticity of this certificate复选框，然后点 Confirm。如果没有，请单击 Cancel，然后手动输入正确的证书信息。
   确认后，将使用 CA 证书来验证与 API 端点的后续通信。
7. 点 Create provider 添加并保存该提供程序。
   供应商会出现在提供程序列表中。

4.4.2.2. 为 OpenShift Virtualization 供应商选择迁移网络

您可以在 Red Hat OpenShift Web 控制台中为 OpenShift Virtualization 供应商选择默认迁移网络，以提高性能。默认迁移网络用于将磁盘传输到其配置的命名空间。

如果您没有选择迁移网络，则默认迁移网络为 pod 网络，这可能不是磁盘传输的最佳选择。

注意

您可以在创建迁移计划时选择不同的网络来覆盖供应商的默认迁移网络。

流程

1. 在 Red Hat OpenShift web 控制台中，点 Migration → Providers for virtualization。
2. 在供应商右侧，从 Options 菜单 ⋮ 中选择 Select migration network。
3. 从可用网络列表中选择网络，然后点 Select。

4.5. 创建迁移计划

您可以使用 Red Hat OpenShift web 控制台指定源供应商、您要迁移的虚拟机(VM)和其他计划详情来创建迁移计划。

为方便起见，基于源供应商和虚拟机创建迁移计划的两个步骤是，一开始于供应商 虚拟化 页面 的供应商，另一个是从 计划 进行虚拟化页面 开始。

4.5.1. 在供应商上为虚拟化创建并运行迁移计划

流程

1. 在 Red Hat OpenShift Web 控制台中，点 Providers for virtualization。
2. 在适当的源供应商行中，单击 VMs. +。此时会打开 Virtual Machines 选项卡。
3. 选择您要迁移的虚拟机，然后点 Create migration plan。
   Create migration plan 窗格打开。它显示目标供应商和命名空间、网络映射和存储映射的源供应商名称和建议。
4. 输入 Plan 名称。
5. 对可编辑的项目进行所需的更改。
6. 点 Add mapping 来编辑推荐的网络映射或存储映射，或者添加一个或多个附加映射。
7. 点 Create migration plan。
   MTV 验证迁移计划，并打开 Plan details 页面，指示计划是否准备就绪以供使用或包含错误。在所有情况下，都会列出计划的详情，您可以编辑您在上页中填写的项目。如果您进行任何更改，MTV 会再次验证计划。
8. 如果计划有效，
   a. 现在，您可以通过点 Start migration 来运行计划。
   b. 您可以在 计划中选择虚拟化 页面并遵循运行迁移计划 中的步骤来稍后运行计划。

4.5.2. 为虚拟化创建并运行迁移计划

流程

1. 在 Red Hat OpenShift Web 控制台中，点 Plans for virtualization，然后点 Create Plan。
   Create migration plan 向导会打开 Select source provider 接口。
2. 选择您要迁移的虚拟机的源供应商。
   此时会打开 Select Virtual Machines 接口。
3. 选择您要迁移的虚拟机，然后点 Next。
   Create migration plan 窗格打开。它显示目标供应商和命名空间、网络映射和存储映射的源供应商名称和建议。
4. 输入 Plan 名称。
5. 对可编辑的项目进行所需的更改。
6. 点 Add mapping 来编辑推荐的网络映射或存储映射，或者添加一个或多个附加映射。
7. 点 Create migration plan。

MTV 验证迁移计划，并打开 Plan details 页面，指示计划是否准备就绪以供使用或包含错误。在所有情况下，都会列出计划的详情，您可以编辑您在上页中填写的项目。如果您进行任何更改，MTV 会再次验证计划。

8. 如果计划有效，

   a. 现在，您可以通过点 Start migration 来运行计划。

   b. 您可以在 计划中选择虚拟化 页面并遵循运行迁移计划 中的步骤来稍后运行计划。

4.6. 运行迁移计划

您可以运行迁移计划，并在 Red Hat OpenShift Web 控制台中查看其进度。

先决条件

- 有效的迁移计划。

流程

1. 在 Red Hat OpenShift web 控制台中，点 Migration → Plans for virtualization。
   Plans 列表显示源和目标供应商、正在迁移的虚拟机数量、状态和每个计划的描述。
2. 点迁移计划旁边的 Start 来启用迁移。
3. 在打开的确认窗口中点 Start。
   Migration details by VM屏幕将打开，显示迁移的进度

   仅限 warm 迁移：

   - precopy 阶段会启动。
   - 点 Cutover 完成迁移。
4. 如果迁移失败：

   a. 点 Get logs 来检索迁移日志。

   b. 在打开的确认窗口中点 Get logs。

   c. 等待 Get logs 变为 Download logs，然后单击 按钮来下载日志。
5. 点迁移的 Status （无论是失败还是成功）查看迁移的详情。
   Migration details by VM屏幕将打开，显示迁移的开始和结束时间、复制的数据量以及要迁移的每个虚拟机的进度管道。
6. 扩展单个虚拟机以查看其步骤以及每个步骤所经过的时间和状态。

4.7. 迁移计划选项

在 Red Hat OpenShift Web 控制台的 Virtualization 页中，点迁移计划旁的 Options 菜单 ⋮ 访问以下选项：

- 获取日志 ：检索迁移的日志。当您点 Get logs 时，会打开一个确认窗口。在窗口中点击 Get logs 后，等待 Get logs 变为 Download logs，然后点按钮下载日志。

- 编辑 ：编辑迁移计划的详情。您无法在迁移计划正在运行或成功完成后编辑迁移计划。
- 副本 ：创建一个与现有计划相同的虚拟机 (VM)、参数、映射和 hook 的新迁移计划。您可以将此功能用于以下任务：
  - 将虚拟机迁移到不同的命名空间。
  - 编辑归档的迁移计划。
  - 编辑具有不同状态的迁移计划，如 failed、canceled、running、critical 或 ready。
- 归档 ：删除迁移计划的日志、历史记录和元数据。计划不能编辑或重启。只能查看它。

注意

Archive 选项不可逆。但是，您可以复制归档的计划。

- 删除 ：永久删除迁移计划。您不能删除正在运行的迁移计划。

注意

Delete 选项不可逆。

删除迁移计划不会删除 **importer** pod、**conversion** pod、配置映射、secret、失败的虚拟机和数据卷等临时资源。(BZ#2018974) 您必须在删除迁移计划前归档迁移计划来清理临时资源。

- 查看详情 ：显示迁移计划的详细信息。
- 重启 ：重启失败或取消的迁移计划。
- 取消调度的 cutover ：取消计划的过渡计划。

4.8. 取消迁移

在使用 Red Hat OpenShift web 控制台进行迁移计划时，您可以取消一些或所有虚拟机(VM)迁移。

流程

1. 在 Red Hat OpenShift Web 控制台中，点 Plans for virtualization。
2. 点正在运行的迁移计划的名称查看迁移详情。
3. 选择一个或多个虚拟机，点 Cancel。
4. 点 Yes, cancel 确认取消。
   在 Migration details by VM 列表中，取消的虚拟机的状态为 Canceled。未迁移且迁移的虚拟机不受影响。

您可以通过点 Migration Plan 页的迁移计划旁的 Restart 重启迁移。

第 5 章 从命令行迁移虚拟机

您可以使用命令行将虚拟机迁移到 OpenShift Virtualization。

重要

您必须确保 满足所有先决条件。

5.1. 非管理员用于迁移计划组件所需的权限

如果您是管理员，您可以处理迁移计划的所有组件（如供应商、网络映射和迁移计划）。

默认情况下，非管理员用户对迁移计划及其组件具有有限的能力。作为管理员，您可以修改其角色，以允许他们完全访问所有组件，或者授予他们有限的权限。

例如，管理员可以为迁移计划分配以下一个或多个集群角色的非管理员用户：

表 5.1. 迁移计划角色及其权限示例

角色	描述
plans.forklift.konveyor.io-v1beta1-view	可以查看迁移计划，但不能创建、删除或修改它们
plans.forklift.konveyor.io-v1beta1-edit	可以创建、删除或修改(**编辑权限的所有部分**)单独的迁移计划
plans.forklift.konveyor.io-v1beta1-admin	所有 **编辑权限**，以及删除整个迁移计划集合的功能

请注意，预定义的集群角色包含一个资源（例如，计划）、API 组（例如 forklift.konveyor.io-v1beta1）和一个操作（如 view、edit）。

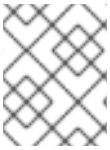
作为更全面的示例，您可以为每个命名空间授予以下权限集：

- 为他们有权访问的命名空间创建和修改存储映射、网络映射和迁移计划
- 将管理员创建的提供程序附加到存储映射、网络映射和迁移计划
- 无法创建供应商或更改系统设置

表 5.2. 非 adminstrators 与迁移计划组件（但不创建供应商）所需的权限示例

Actions	API 组	资源
get, **list**, **watch**, **create**, **update**, **patch**, **delete**	**forklift.konveyor.io**	**plans**
get, **list**, **watch**, **create**, **update**, **patch**, **delete**	**forklift.konveyor.io**	**迁移**
get, **list**, **watch**, **create**, **update**, **patch**, **delete**	**forklift.konveyor.io**	**hooks**

Actions	API 组	资源
get, list, watch	forklift.konveyor.io	providers
get, list, watch, create, update, patch, delete	forklift.konveyor.io	networkmaps
get, list, watch, create, update, patch, delete	forklift.konveyor.io	storagemaps
get, list, watch	forklift.konveyor.io	forkliftcontrollers
创建、补丁、删除	空字符串	secrets



注意

非管理员用户需要具有 创建 权限，它们是网络映射 的编辑 角色的一部分，而存储映射则用于创建迁移计划，即使将模板用于网络映射或存储映射。

5.2. 迁移虚拟机

您可以通过创建 MTV 自定义资源(CR)从命令行(CLI)迁移虚拟机。CR 和迁移步骤因源供应商而异。

重要

您必须为集群范围的 CR 指定一个名称。

您必须为命名空间范围 CR 指定名称和命名空间。

要迁移到与迁移计划不同的 OpenShift 集群，您必须具有具有 cluster-admin 权限的 OpenShift Virtualization 服务帐户令牌。

5.2.1. 从 VMware vSphere 源供应商迁移

您可以使用 CLI 从 VMware vSphere 源供应商迁移。

流程

1. 为源供应商凭证创建 Secret 清单：

```
$ 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: vsphere
    createdForResourceType: providers
type: Opaque
stringData:
  user: <user> 2
  password: <password> 3
  insecureSkipVerify: <"true"/"false"> 4
  cacert: | 5
    <ca_certificate>
  url: <api_end_point> 6
EOF
```

1. **ownerReferences 部分是可选的。**
2. **指定 vCenter 用户或 ESX/ESXi 用户。**
3. **指定 vCenter 用户的密码或 ESX/ESXi 用户。**
4. **指定 "true" 来跳过证书验证，指定 "false" 来验证证书。如果没有指定，则默认为 "false "。跳过证书验证会进行不安全的迁移，然后不需要证书。不安全的迁移意味着传输的数据通过不安全的连接发送，并可能会公开敏感数据。**
5. **如果没有设置此字段 *并跳过禁用了证书验证*，MTV 会尝试使用系统 CA。**
6. **指定 vCenter 的 API 端点 URL 或 ESX/ESX，例如 https://<vCenter_host>/sdk。**

2. 为**源供应商创建 Provider 清单：**
   +

```
$ cat << EOF | {oc} apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Provider
metadata:
  name: <source_provider>
  namespace: <namespace>
spec:
  type: vsphere
  url: <api_end_point> 1
  settings:
    vddkInitImage: <VDDK_image> 2
    sdkEndpoint: vcenter 3
  secret:
    name: <secret> 4
    namespace: <namespace>
EOF
```

1. **指定 API 端点的 URL，例如 https://<vCenter_host>/sdk。**
2. **可选，但强烈建议创建 VDDK 镜像来加快迁移速度。按照 OpenShift 文档指定您创建的 VDDK 镜像。**
3. 选项：**vcenter 或 esxi.**

4 指定供应商 Secret CR 的名称。

3. 创建 主机清单 ：

```
$ 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 指定 VMware vSphere Provider CR 的名称。

2 指定 VMware vSphere 主机的 Managed Object Reference (MoRef)。

3 指定 VMware vSphere 迁移网络的 IP 地址。

4. 创建 NetworkMap 清单来映射源和目标网络：

```
$  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>
        name: <source_network_name>
    - destination:
        name: <network_attachment_definition> 3
        namespace: <network_attachment_definition_namespace> 4
        type: multus
      source:
        id: <source_network_id>
        name: <source_network_name>
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
```

```
    name: <destination_provider>
    namespace: <namespace>
EOF
```

1. 允许的值是 pod 和 multus。
2. 您可以使用 id *或* name 参数来指定源网络。对于 id，指定 VMware vSphere 网络受管对象参考(MoRef)。
3. 为每个额外 OpenShift Virtualization 网络指定网络附加定义。
4. 仅在 类型为 multus 时才需要。指定 OpenShift Virtualization 网络附加定义的命名空间。

5. 创建 StorageMap 清单来映射源和目标存储：

```
$ 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)
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

1. 允许的值有 ReadWriteOnce 和 ReadWriteMany。
2. 指定 VMware vSphere 数据存储 MoRef。例如，f2737930-b567-451a-9ceb-2887f6207009。

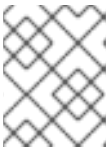
6. 可选：在 Plan CR 中指定的阶段创建一个 Hook 清单以在虚拟机中运行自定义代码：

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

```
LS0tCi0gbmFtZTogTWFpbgogIGhvc3RzOiBsb2NhbGhvc3QKICB0YXNrczoKICAtIG5hbWU6IExv

YWQgUGxhbgogICAgaW5jbHVkZV92YXJzOgogICAgICBmaWxlOiAiL3RtC9ob29rL3BsYW4ueW1s

IgogICAgICBuYW1lOiBwbGFuCiAgLSBuYW1lOiBMb2FkIFdvcmtsb2FkCiAgICBpbmNsdWRlX3Zh

cnM6CiAgICAgIGZpbGU6ICIvdG1wL2hvb2svd29ya2xvYWQueW1sIgogICAgICBuYW1lOiB3b3Jr

  bG9hZAoK
EOF
```

其中：

Playbook 指的是一个可选的 Base64 编码的 Ansible playbook。如果指定 playbook，image 必须是 hook-runner。



注意

您可以使用默认 hook-runner 镜像或指定自定义镜像。如果指定自定义镜像，则不需要指定 playbook。

7. 为迁移创建 Plan 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Plan
metadata:
  name: <plan> 1
  namespace: <namespace>
spec:
  warm: false 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>
      hooks: 10
        - hook:
```

```
      namespace: <namespace>
      name: <hook> 11
    step: <step> 12
EOF
```

1. 指定 Plan CR 的名称。
2. 指定迁移是 warm - true - 或 cold -false。如果您指定了 warm 迁移，且没有为 Migration 清单中的 cutover 参数指定一个值，则只有 precopy 阶段将运行。
3. 每个计划仅指定一个网络映射和一个存储映射。
4. 指定网络映射，即使要迁移的虚拟机没有分配给网络。在这种情况下，映射可以为空。
5. 指定 NetworkMap CR 的名称。
6. 即使要迁移的虚拟机没有使用磁盘镜像分配，请指定存储映射。在这种情况下，映射可以为空。
7. 指定 StorageMap CR 的名称。
8. 您可以使用 id *或* name 参数来指定源虚拟机。
9. 指定 VMware vSphere VM MoRef。
10. 可选：为虚拟机指定最多两个 hook。每个 hook 必须在不同的迁移步骤中运行。
11. 指定 Hook CR 的名称。
12. 迁移完成后，允许的值是 PreHook，在迁移计划启动或 PostHook 之前。

8. 创建运行 Plan CR 的 Migration 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Migration
metadata:
  name: <name_of_migration_cr>
  namespace: <namespace>
spec:
  plan:
    name: <name_of_plan_cr>
    namespace: <namespace>
  cutover: <optional_cutover_time>
EOF
```

注意

如果您指定了截止时间，请使用 ISO 8601 格式的 UTC 时间偏移，例如 2024-04-04T01:23:45.678+09:00。

5.2.2. 从 Red Hat Virtualization 源供应商迁移

您可以使用 CLI 从 Red Hat Virtualization (RHV)源供应商迁移。

先决条件

如果您要迁移使用直接 LUN 磁盘的虚拟机，请确保虚拟机在 OpenShift Virtualization 目标集群中的节点可以访问后端存储。

注意

- 与从源供应商 *复制到目标* 供应商的磁盘镜像不同，LUN会*从* 源供应商中的 *虚拟机分离*，然后附加到目标供应商中创建的虚拟机(VM)。
- 如果需要回退到源供应商，则 LUN 不会在迁移过程中从源供应商中删除。但是，在将 LUN 重新附加到源供应商中的虚拟机之前，请确保目标环境中的虚拟机不会同时使用 LUN，这可能会导致数据崩溃。

流程

1. 为源供应商凭证创建 Secret 清单：

```
$ 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: ovirt
    createdForResourceType: providers
type: Opaque
stringData:
  user: <user> 2
  password: <password> 3
  insecureSkipVerify: <"true"/"false"> 4
  cacert: | 5
    <ca_certificate>
  url: <api_end_point> 6
EOF
```

1. ownerReferences 部分是可选的。
2. 指定 RHV Manager 用户。
3. 指定用户密码。
4. 指定 "true" 来跳过证书验证，指定 "false" 来验证证书。如果没有指定，则默认为 "false"。跳过证书验证会进行不安全的迁移，然后不需要证书。不安全的迁移意味着传输的数据通过不安全的连接发送，并可能会公开敏感数据。
5. 输入 Manager CA 证书，除非它被第三方证书替代，否则请输入 Manager Apache CA 证书。您可以在 https://<engine_host>/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA 中检索 Manager CA 证书。

6 指定 API 端点 URL，例如 https://<engine_host>/ovirt-engine/api。

2. 为源供应商创建 Provider 清单：

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

1 指定 API 端点的 URL，例如 https://<engine_host>/ovirt-engine/api。

2 指定供应商 Secret CR 的名称。

3. 创建 NetworkMap 清单来映射源和目标网络：

```
$  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>
        name: <source_network_name>
    - destination:
        name: <network_attachment_definition> 3
        namespace: <network_attachment_definition_namespace> 4
        type: multus
      source:
        id: <source_network_id>
        name: <source_network_name>
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

1. 允许的值是 pod 和 multus。
2. 您可以使用 id 或 name 参数来指定源网络。对于 id，指定 RHV 网络通用唯一 ID (UUID)。
3. 为每个额外 OpenShift Virtualization 网络指定网络附加定义。
4. 仅在 类型为 multus 时才需要。指定 OpenShift Virtualization 网络附加定义的命名空间。

4. 创建 StorageMap 清单来映射源和目标存储：

```
$ 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_storage_domain> 2
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

1. 允许的值有 ReadWriteOnce 和 ReadWriteMany。
2. 指定 RHV 存储域 UUID。例如，f2737930-b567-451a-9ceb-2887f6207009。

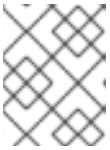
5. 可选：在 Plan CR 中指定的阶段创建一个 Hook 清单以在虚拟机中运行自定义代码：

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

LS0tCi0gbmFtZTogTWFpbgogIGhvc3RzOiBsb2NhbGhvc3QKICB0YXNrczoKICAtIG5hbWU6IExv

YWQgUGxhbgogICAgaW5jbHVkZV92YXJzOgogICAgICBmaWxlOiAiL3RtC9ob29rL3BsYW4ueW1s
```

```
IgogICAgICBuYW1lOiBwbGFuCiAgLSBuYW1lOiBMb2FkIFdvcmtsb2FkCiAgICBpbmNsdWRlX3Zh

cnM6CiAgICAgICAgZpbGU6ICIvdG1wL2hvb2svd29ya2xvYWQueW1sIgogICAgICBuYW1lOiB3b3Jr
    bG9hZAoK
EOF
```

其中：

Playbook 指的是一个可选的 Base64 编码的 Ansible playbook。如果指定 playbook，image 必须是 hook-runner。



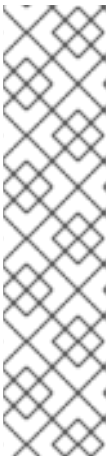
注意

您可以使用默认 hook-runner 镜像或指定自定义镜像。如果指定自定义镜像，则不需要指定 playbook。

6. **为迁移创建 Plan 清单：**

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Plan
metadata:
  name: <plan> 1
  namespace: <namespace>
  preserveClusterCpuModel: true 2
spec:
  warm: false 3
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
  map: 4
    network: 5
      name: <network_map> 6
      namespace: <namespace>
    storage: 7
      name: <storage_map> 8
      namespace: <namespace>
  targetNamespace: <target_namespace>
  vms: 9
    - id: <source_vm> 10
    - name: <source_vm>
      hooks: 11
        - hook:
            namespace: <namespace>
            name: <hook> 12
          step: <step> 13
EOF
```

1. 指定 Plan CR 的名称。
2. 请参见以下备注。
3. 指定迁移是温迁移还是冷迁移。如果您指定了 warm 迁移，且没有为 Migration 清单中的 cutover 参数指定一个值，则只有 precopy 阶段将运行。
4. 每个计划仅指定一个网络映射和一个存储映射。
5. 指定网络映射，即使要迁移的虚拟机没有分配给网络。在这种情况下，映射可以为空。
6. 指定 NetworkMap CR 的名称。
7. 即使要迁移的虚拟机没有使用磁盘镜像分配，请指定存储映射。在这种情况下，映射可以为空。
8. 指定 StorageMap CR 的名称。
9. 您可以使用 id *或* name 参数来指定源虚拟机。
10. 指定 RHV VM UUID。
11. 可选：为虚拟机指定最多两个 hook。每个 hook 必须在不同的迁移步骤中运行。
12. 指定 Hook CR 的名称。
13. 迁移完成后，允许的值是 PreHook，在迁移计划启动或 PostHook 之前。



注意

- 如果使用自定义 CPU 模型设置了迁移的机器，它将在目标集群中使用该 CPU 模型设置，而不考虑 preserveClusterCpuModel 的设置。
- *如果没有使用* 自定义 CPU 模型设置迁移的机器：
  - 如果将 preserveClusterCpuModel 设置为 'true'，则 MTV 会根据集群的配置在 RHV 中运行时检查虚拟机的 CPU 模型，然后使用该 CPU 模型设置迁移的虚拟机。
  - 如果将 preserveClusterCpuModel 设置为 'false'，则 MTV 不会设置 CPU 类型，并使用目标集群的默认 CPU 模型设置虚拟机。

7. 创建运行 Plan CR 的Migration 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Migration
metadata:
  name: <name_of_migration_cr>
  namespace: <namespace>
spec:
  plan:
    name: <name_of_plan_cr>
```

```
  namespace: <namespace>
 cutover: <optional_cutover_time>
EOF
```

注意

如果您指定了截止时间，请使用 ISO 8601 格式的 UTC 时间偏移，例如 2024-04-04T01:23:45.678+09:00。

5.2.3. 从 OpenStack 源供应商迁移

您可以使用 CLI 从 OpenStack 源供应商迁移。

流程

1. 为**源供应商凭证创建 Secret 清单：**

```
$ 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: openstack
  createdForResourceType: providers
type: Opaque
stringData:
 user: <user> (2)
 password: <password> (3)
 insecureSkipVerify: <"true"/"false"> (4)
 domainName: <domain_name>
 projectName: <project_name>
 regionName: <region_name>
 cacert: | (5)
  <ca_certificate>
 url: <api_end_point> (6)
EOF
```

(1) **ownerReferences 部分是可选的。**

(2) **指定 OpenStack 用户。**

(3) **指定用户 OpenStack 密码。**

(4) **指定 "true" 来跳过证书验证，指定 "false" 来验证证书。如果没有指定，则默认为 "false"。跳过证书验证会进行不安全的迁移，然后不需要证书。不安全的迁移意味着传输的数据通过不安全的连接发送，并可能会公开敏感数据。**

5 如果没有设置此字段 *并跳过禁用了证书验证*，MTV 会尝试使用系统 CA。

6 指定 API 端点 URL，例如 https://<identity_service>/v3。

2. 为源供应商创建 Provider 清单：

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

1 指定 API 端点的 URL。

2 指定供应商 Secret CR 的名称。

3. 创建 NetworkMap 清单来映射源和目标网络：

```
$  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>
        name: <source_network_name>
    - destination:
        name: <network_attachment_definition> 3
        namespace: <network_attachment_definition_namespace> 4
        type: multus
      source:
        id: <source_network_id>
        name: <source_network_name>
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
```

```
    name: <destination_provider>
    namespace: <namespace>
EOF
```

1. 允许的值是 pod 和 multus。
2. 您可以使用 id 或 name 参数来指定源网络。对于 id，指定 OpenStack 网络 UUID。
3. 为每个额外 OpenShift Virtualization 网络指定网络附加定义。
4. 仅在 类型为 multus 时才需要。指定 OpenShift Virtualization 网络附加定义的命名空间。

4. 创建 StorageMap 清单来映射源和目标存储：

```
$ 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_volume_type> 2
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

1. 允许的值有 ReadWriteOnce 和 ReadWriteMany。
2. 指定 OpenStack volume_type UUID。例如，f2737930-b567-451a-9ceb-2887f6207009。

5. 可选：在 Plan CR 中指定的阶段创建一个 Hook 清单以在虚拟机中运行自定义代码：

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

LS0tCi0gbmFtZTogTWFpbgogIGhvc3RzOiBsb2NhbGhvc3QKICB0YXNrczoKICAtIG5hbWU6IExv
```

```
YWQgUGxhbgogICAgaW5jbHVkZV92YXJzOgogICAgICBmaWxlOiAiL3RtcC9ob29rL3BsYW4ueW1s

IgogICAgICBuYW1lOiBwbGFuCiAgLSBuYW1lOiBMb2FkIFdvcmtsb2FkCiAgICBpbmNsdWRlX3Zh

cnM6CiAgICAgIGZpbGU6ICIvdG1wL2hvb2svd29ya2xvYWQueW1sIgogICAgICBuYW1lOiB3b3Jr
  bG9hZAoK
EOF
```

其中：

Playbook 指的是一个可选的 Base64 编码的 Ansible playbook。如果指定 playbook，image 必须是 hook-runner。

注意

您可以使用默认 hook-runner 镜像或指定自定义镜像。如果指定自定义镜像，则不需要指定 playbook。

6. 为迁移创建 Plan 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Plan
metadata:
  name: <plan> 1
  namespace: <namespace>
spec:
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
  map: 2
    network: 3
      name: <network_map> 4
      namespace: <namespace>
    storage: 5
      name: <storage_map> 6
      namespace: <namespace>
  targetNamespace: <target_namespace>
  vms: 7
    - id: <source_vm> 8
    - name: <source_vm>
      hooks: 9
        - hook:
            namespace: <namespace>
```

```
      name: <hook> 10
    step: <step> 11
EOF
```

1. 指定 Plan CR 的名称。
2. 每个计划仅指定一个网络映射和一个存储映射。
3. 指定网络映射，即使要迁移的虚拟机没有分配给网络。在这种情况下，映射可以为空。
4. 指定 NetworkMap CR 的名称。
5. 指定存储映射，即使要迁移的虚拟机没有使用磁盘镜像分配。在这种情况下，映射可以为空。
6. 指定 StorageMap CR 的名称。
7. 您可以使用 id *或* name 参数来指定源虚拟机。
8. 指定 OpenStack 虚拟机 UUID。
9. 可选：为虚拟机指定最多两个 hook。每个 hook 必须在不同的迁移步骤中运行。
10. 指定 Hook CR 的名称。
11. 迁移完成后，允许的值是 PreHook，在迁移计划启动或 PostHook 之前。

7. 创建运行 Plan CR 的Migration 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Migration
metadata:
  name: <name_of_migration_cr>
  namespace: <namespace>
spec:
  plan:
    name: <name_of_plan_cr>
    namespace: <namespace>
  cutover: <optional_cutover_time>
EOF
```

注意

如果您指定了截止时间，请使用 ISO 8601 格式的 UTC 时间偏移，例如 2024-04-04T01:23:45.678+09:00。

5.2.4. 从开放虚拟设备(OVA)源供应商迁移

您可以使用 CLI 从 VMware vSphere 创建的开放虚拟设备(OVA)文件迁移为源供应商。

流程

1. 为源供应商凭证创建 Secret 清单：

```
$ 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: ova
    createdForResourceType: providers
type: Opaque
stringData:
  url: <nfs_server:/nfs_path> 2
EOF
```

1 ownerReferences 部分是可选的。

2 其中： nfs_server 是创建共享的服务器的 IP 或主机名，nfs_path 是存储 OVA 文件在服务器上的路径。

2. 为源供应商创建 Provider 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Provider
metadata:
  name: <source_provider>
  namespace: <namespace>
spec:
  type: ova
  url:  <nfs_server:/nfs_path> 1
  secret:
    name: <secret> 2
    namespace: <namespace>
EOF
```

1 其中： nfs_server 是创建共享的服务器的 IP 或主机名，nfs_path 是存储 OVA 文件在服务器上的路径。

2 指定供应商 Secret CR 的名称。

3. 创建 NetworkMap 清单来映射源和目标网络：

```
$  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:
        id: <source_network_id> (2)
    - destination:
        name: <network_attachment_definition> (3)
        namespace: <network_attachment_definition_namespace> (4)
        type: multus
      source:
        id: <source_network_id>
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

(1) 允许的值是 pod 和 multus。

(2) 指定 OVA 网络通用唯一 ID (UUID)。

(3) 为每个额外 OpenShift Virtualization 网络指定网络附加定义。

(4) 仅在 类型为 multus 时才需要。指定 OpenShift Virtualization 网络附加定义的命名空间。

4. 创建 StorageMap 清单来映射源和目标存储：

```
$ 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:
        name:  Dummy storage for source provider <provider_name> (2)
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

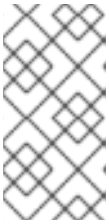
1. 允许的值有 ReadWriteOnce 和 ReadWriteMany。
2. 对于 OVA，StorageMap 只能将单个存储（来自 OVA 的所有磁盘都与之关联）映射到目的地上的存储类。因此，存储在 UI 中被称为 "Dummy storage for source provider <provider_name>"。在 YAML 中，按上面显示的编写短语，没有引号，并将 <provider_name> 替换为供应商的实际名称。

5. 可选：在 Plan CR 中指定的阶段创建一个 Hook 清单以在虚拟机中运行自定义代码：

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

LS0tCi0gbmFtZTogTWFpbgogIGhvc3RzOiBsb2NhbGhvc3QKICB0YXNrczoKICAtIG5hbWU6IExv

YWQgUGxhbgogICAgaW5jbHVkZV92YXJzOgogICAgICBmaWxlOiAiL3RtC9ob29rL3BsYW4ueW1s

IgogICAgICBuYW1lOiBwbGFuCiAgLSBuYW1lOiBMb2FkIFdvcmtsb2FkCiAgICBpbmNsdWRlX3Zh

cnM6CiAgICAgIGZpbGU6ICIvdG1wL2hvb2svd29ya2xvYWQueW1sIgogICAgICBuYW1lOiB3b3Jr
    bG9hZAoK
EOF
```

其中：

Playbook 指的是一个可选的 Base64 编码的 Ansible playbook。如果指定 playbook，image 必须是 hook-runner。



注意

您可以使用默认 hook-runner 镜像或指定自定义镜像。如果指定自定义镜像，则不需要指定 playbook。

6. 为迁移创建 Plan 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Plan
metadata:
  name: <plan> 1
  namespace: <namespace>
spec:
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
  map: 2
    network: 3
      name: <network_map> 4
      namespace: <namespace>
    storage: 5
      name: <storage_map> 6
      namespace: <namespace>
  targetNamespace: <target_namespace>
  vms: 7
    - id: <source_vm> 8
    - name: <source_vm>
      hooks: 9
        - hook:
            namespace: <namespace>
            name: <hook> 10
          step: <step> 11
EOF
```

1. 指定 Plan CR 的名称。

2. 每个计划仅指定一个网络映射和一个存储映射。

3. 指定网络映射，即使要迁移的虚拟机没有分配给网络。在这种情况下，映射可以为空。

4.

5

6

指定 StorageMap CR 的名称。

7

您可以使用 id *或* name 参数来指定源虚拟机。

8

指定 OVA 虚拟机 UUID。

9

可选：为虚拟机指定最多两个 hook。每个 hook 必须在不同的迁移步骤中运行。

10

指定 Hook CR 的名称。

11

迁移完成后，允许的值是 PreHook，在迁移计划启动或 PostHook 之前。

7. 创建运行 Plan CR 的 Migration 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Migration
metadata:
  name: <name_of_migration_cr>
  namespace: <namespace>
spec:
  plan:
    name: <name_of_plan_cr>
    namespace: <namespace>
  cutover: <optional_cutover_time>
EOF
```

注意

如果您指定了截止时间，请使用 ISO 8601 格式的 UTC 时间偏移，例如 2024-04-04T01:23:45.678+09:00。

5.2.5. 从 Red Hat OpenShift Virtualization 源供应商迁移

您可以使用 Red Hat OpenShift Virtualization 供应商作为源供应商和目标供应商。

流程

1. 为源供应商凭证创建 Secret 清单：

```
$ 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: openshift
    createdForResourceType: providers
type: Opaque
stringData:
  token: <token> 2
  password: <password> 3
  insecureSkipVerify: <"true"/"false"> 4
  cacert: | 5
    <ca_certificate>
  url: <api_end_point> 6
EOF
```

1. ownerReferences 部分是可选的。

2. 为具有 cluster-admin 权限的服务帐户指定令牌。如果 令牌和 url 都留空，则使用本地 OpenShift 集群。

3

指定用户密码。

4

指定 "true" 来跳过证书验证，指定 "false" 来验证证书。如果没有指定，则默认为 "false "。跳过证书验证会进行不安全的迁移，然后不需要证书。不安全的迁移意味着传输的数据通过不安全的连接发送，并可能会公开敏感数据。

5

如果没有设置此字段 *并跳过禁用了证书验证*，MTV 会尝试使用系统 CA。

6

指定 API 服务器端点的 URL。

2. 为源供应商创建 Provider 清单：

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

1

指定 API 服务器端点的 URL。

2

指定供应商 Secret CR 的名称。

3. 创建 NetworkMap 清单来映射源和目标网络：

```
$ 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:
        name: <network_name>
        type: pod
    - destination:
        name: <network_attachment_definition> (2)
        namespace: <network_attachment_definition_namespace> (3)
        type: multus
      source:
        name: <network_attachment_definition>
        namespace: <network_attachment_definition_namespace>
        type: multus
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

(1) 允许的值是 pod 和 multus。

(2) 为每个额外 OpenShift Virtualization 网络指定网络附加定义。使用 namespace 属性 或构建的名称来指定命名空间： <network_ namespace>/<network_name >。

(3) 仅在 类型为 multus 时才需要。指定 OpenShift Virtualization 网络附加定义的命名空间。

4. 创建 StorageMap 清单来映射源和目标存储：

```
$ 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:
        name: <storage_class>
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
EOF
```

1

允许的值有 ReadWriteOnce 和 ReadWriteMany。

5. 可选：在 Plan CR 中指定的阶段创建一个 Hook 清单以在虚拟机中运行自定义代码：

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

LS0tCi0gbmFtZTogTWFpbgogIGhvc3RzOiBsb2NhbGhvc3QKICB0YXNrczoKICAtIG5hbWU6IExv

YWQgUGxhbgogICAgaW5jbHVkZV92YXJzOgogICAgICBmaWxlOiAiL3RtC9ob29rL3BsYW4ueW1s

IgogICAgICBuYW1lOiBwbGFuCiAgLSBuYW1lOiBMb2FkIFdvcmtsb2FkCiAgICBpbmNsdWRlX3Zh

cnM6CiAgICAgIGZpbGU6ICIvdG1wL2hvb2svd29ya2xvYWQueW1sIgogICAgbmFtZTogbG9hZAoK
EOF
```

其中：

Playbook 指的是一个可选的 Base64 编码的 Ansible playbook。如果指定 playbook，image 必须是 hook-runner。

注意

您可以使用默认 hook-runner 镜像或指定自定义镜像。如果指定自定义镜像，则不需要指定 playbook。

6. 为迁移创建 Plan 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Plan
metadata:
  name: <plan> 1
  namespace: <namespace>
spec:
  provider:
    source:
      name: <source_provider>
      namespace: <namespace>
    destination:
      name: <destination_provider>
      namespace: <namespace>
  map: 2
    network: 3
      name: <network_map> 4
      namespace: <namespace>
    storage: 5
      name: <storage_map> 6
      namespace: <namespace>
  targetNamespace: <target_namespace>
  vms:
    - name: <source_vm>
      namespace: <namespace>
      hooks: 7
        - hook:
            namespace: <namespace>
            name: <hook> 8
          step: <step> 9
EOF
```

1

2

每个计划仅指定一个网络映射和一个存储映射。

3

指定网络映射，即使要迁移的虚拟机没有分配给网络。在这种情况下，映射可以为空。

4

指定 NetworkMap CR 的名称。

5

指定存储映射，即使要迁移的虚拟机没有使用磁盘镜像分配。在这种情况下，映射可以为空。

6

指定 StorageMap CR 的名称。

7

可选：为虚拟机指定最多两个 hook。每个 hook 必须在不同的迁移步骤中运行。

8

指定 Hook CR 的名称。

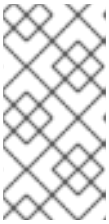
9

迁移完成后，允许的值是 PreHook，在迁移计划启动或 PostHook 之前。

7. 创建运行 Plan CR 的 Migration 清单：

```
$ cat << EOF | oc apply -f -
apiVersion: forklift.konveyor.io/v1beta1
kind: Migration
metadata:
  name: <name_of_migration_cr>
  namespace: <namespace>
spec:
  plan:
```

```
  name: <name_of_plan_cr>
  namespace: <namespace>
 cutover: <optional_cutover_time>
EOF
```



注意

如果您指定了截止时间，请使用 ISO 8601 格式的 UTC 时间偏移，例如 2024-04-04T01:23:45.678+09:00。

5.3. 取消迁移

在从命令行界面 (CLI) 进行迁移时，您可以取消整个迁移或单独的虚拟机。

取消整个迁移

- 删除 Migration CR:

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

1. 指定 Migration CR 的名称。

取消单个虚拟机的迁移

1. 将独立虚拟机添加到 Migration 清单的 spec.cancel 块中：

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

1.

id 键的值是*受管对象的引用*（VMware VM），或 *VM UUID*（RHV VM）。

2. **检索 Migration CR 以监控剩余的虚拟机的进度：**

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

第 6 章 高级迁移选项

6.1. 为 WARM 迁移更改预复制间隔

您可以通过修补 ForkliftController 自定义资源(CR)来更改快照间隔。

流程

- 对 ForkliftController CR 进行补丁：

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

1. 以分钟为单位指定 precopy 间隔。默认值为 60。

您不需要重启 forklift-controller pod。

6.2. 为 VALIDATION 服务创建自定义规则

Validation 服务使用 Open Policy Agent(OPA)策略规则来检查要迁移的每个虚拟机(VM)的适用性。Validation 服务为每个虚拟机生成 *问题* 列表，它们作为虚拟机属性存储在 Provider Inventory 服务中。Web 控制台显示供应商清单中的每个虚拟机的顾虑。

您可以创建自定义规则来扩展 Validation 服务的默认规则集。例如，您可以创建一个规则来检查虚拟机是否有多个磁盘。

6.2.1. 关于 Rego 文件

验证规则使用 Rego 编写，即 Open Policy Agent (OPA) 原生查询语言。规则作为 .rego 文件存储在 Validation pod 的 /usr/share/opa/policies/io/konveyor/forklift/<provider> 目录中。

每个验证规则都在单独的 .rego 文件中定义，以及对特定条件的测试。如果条件评估为 true，则该规则会将 {"category", "label", "assessment"} hash 添加到 concerns 中。concerns 内容将添加到虚拟机清单记录中的 concerns 键中。Web 控制台显示供应商清单中每个虚拟机的 concerns 键的内容。

以下 .rego 文件示例检查在 VMware 虚拟机的集群中启用了分布式资源调度：

drs_enabled.rego 示例

```
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 每个验证规则都在软件包中定义。软件包命名空间是 io.konveyor.forklift.vmware（VMware）和 io.konveyor.forklift.ovirt（Red Hat Virtualization）。

2 查询参数基于 Validation 服务 JSON 的 input 键。

6.2.2. 检查默认验证规则

在创建自定义规则前，您必须检查 Validation 服务的默认规则，以确保您不会创建重新定义现有默认值的规则。

示例：如果默认规则包含行 default valid_input = false，并且您创建一个包含行 default valid_input = true，则 Validation 服务将不会启动。

流程

1. 连接到 Validation pod 的终端：

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

2. 进入您的供应商的 OPA 策略目录：

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

1 指定 vmware 或 ovirt。

3. 搜索默认策略：

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

6.2.3. 检索库存服务 JSON

您可以通过向虚拟机发送一个 Inventory 服务查询来获取 Inventory 服务 JSON。输出中包含一个 "input" 键，其中包含由 Validation 服务规则查询的清单属性。

您可以根据 "input" 键中的任何属性（如 input.snapshot.kind ）创建验证规则。

流程

1. 为项目检索路由：

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

2. 检索 Inventory 服务路由：

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

3. 检索访问令牌：

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

4. 触发 HTTP GET 请求（例如，使用 Curl）：

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

5. 检索供应商的 UUID ：

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

1 供应商允许的值有 vsphere、ovirt 和 openstack。

6. 检索供应商的虚拟机：

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

7. 检索虚拟机的详情：

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

输出示例


```
{
   "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",
```

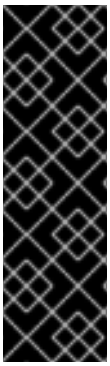
```
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch1"
        },
        {
            "key": "key-vim.host.PortGroup-VM_DHCP_Network",
            "name": "VM_DHCP_Network",
            "vSwitch": "key-vim.host.VirtualSwitch-vSwitch1"
        },
        {
            "key": "key-vim.host.PortGroup-Storage 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
        }
    }
  }
}
```


6.2.4. 创建验证规则

您可以通过将包含规则的配置映射自定义资源(CR)应用到 Validation 服务来创建验证规则。



重要

- 如果您创建与现有规则*相同*的规则，Validation 服务将使用规则执行 OR 操作。
- 如果您创建使用默认规则迭代的规则，则 Validation 服务将不会启动。

验证规则示例

验证规则基于由 Provider Inventory 服务收集的虚拟机(VM)属性。

例如，VMware API 使用此路径来检查 VMware 虚拟机是否配置了 NUMA 节点关联性：MOR:VirtualMachine.config.extraConfig["numa.nodeAffinity"]。

Provider Inventory 服务使用列表值简化了此配置并返回一个可测试属性：

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

您可以根据此属性创建一个 Rego 查询，并将其添加到 forklift-validation-config 配置映射中：

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

流程

1. 根据以下示例创建配置映射 CR：

```
$ 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 指定供应商软件包名称。允许的值是 VMware 的 io.konveyor.forklift.vmware 和 io.konveyor.forklift.ovirt。

2 指定 concerns 名称和 Rego 查询。

3 指定 concerns 名称和 flag 参数值。

4 允许的值是 Critical, Warning, 和 Information。

2. 通过将 forklift-controller 部署扩展到 0 来停止 Validation pod：

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

3. 通过将 forklift-controller 部署扩展到 1 来启动 Validation pod：

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

4. 检查 Validation pod 日志，以验证 pod 是否已启动：

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

如果自定义规则与默认规则冲突，则 Validation pod 将不会启动。

5. 删除源供应商：

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

6. 添加源供应商以应用新规则：

```
$ 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 允许的值有 ovirt、vsphere 和 openstack。

2 指定 API 端点 URL，例如 https://<vCenter_host>/sdk 用于 vSphere，https://<engine_host>/ovirt-engine/api 用于 RHV，或者 https://<identity_service>/v3 用于 OpenStack。

3 指定供应商 Secret CR 的名称。

您必须在创建自定义规则后更新规则版本，以便 Inventory 服务检测到更改并验证虚拟机。

6.2.5. 更新清单规则版本

每次更新规则时，您必须更新 inventory 规则版本，以便 Provider Inventory 服务检测到更改并触发 Validation 服务。

规则版本记录在每个供应商的 rules_version.rego 文件中。

流程

1. 检索当前的规则版本：

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

输出示例

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

2. 连接到 Validation pod 的终端：

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

3. 更新 /usr/share/opa/policies/io/konveyor/forklift/<provider>/rules_version.rego 文件中的规则版本。

4. 从 Validation pod 终端注销。

5. 验证更新的规则版本：

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

输出示例

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

6.3. 在迁移计划中添加 HOOK

您可以使用 Migration Toolkit for Virtualization API 从命令行添加 hook 迁移计划。

6.3.1. MTV 迁移计划的基于 API 的 hook

您可以使用 Migration Toolkit for Virtualization API 在命令行中添加 hook。

默认 hook 镜像

MTV hook 的默认 hook 镜像为 registry.redhat.io/rhmtc/openshift-migration-hook-runner-rhel8:v1.8.2-2。该镜像基于 Ansible Runner 镜像，它添加了 python-openshift，以提供 Ansible Kubernetes 资源和最新的 oc 二进制文件。

hook 执行

作为迁移 hook 的一部分提供的 Ansible playbook 作为 ConfigMap 挂载到 hook 容器中。hook 容器使用 konveyor-forklift 命名空间中的默认 ServiceAccount 作为作业运行。

prehook 和 PostHooks

您可以为每个虚拟机指定 hook，并作为 *PreHook* 或 *PostHook* 运行。在这种情况下，PreHook 是一个 hook，它在迁移前运行，PostHook 是迁移后运行的 hook。

添加 hook 时，您必须指定 hook CR 所在的命名空间，即 hook 的名称，并指定 hook 是否为 PreHook 或 PostHook。

重要

要让 PreHook 在虚拟机上运行，必须启动虚拟机并通过 SSH 使用。

PreHook 示例：

```
kind: Plan
apiVersion: forklift.konveyor.io/v1beta1
metadata:
  name: test
  namespace: konveyor-forklift
spec:
  vms:
    - id: vm-2861
      hooks:
        - hook:
            namespace: konveyor-forklift
            name: playbook
          step: PreHook
```

6.3.2. 使用 MTV API 将 Hook CR 添加到虚拟机迁移中

当使用 Migration Toolkit for Virtualization API 从命令行迁移虚拟机时，您可以添加 PreHook 或 PostHook Hook CR。PreHook 在迁移前会(PostHook)后运行。

注意

您可以使用 k8s 模块来检索存储在 secret 或 configMap 中的其他信息。

例如，您可以创建一个 hook CR 来在虚拟机上安装 cloud-init，并在迁移前写入文件。

流程

1. 如果需要，使用虚拟机的 SSH 私钥创建 secret。您可以使用现有密钥或生成密钥对，在虚拟机上安装公钥，并在 secret 中对私钥进行 base64 编码。

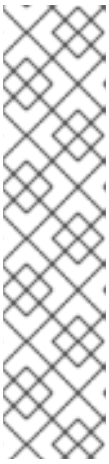
```
apiVersion: v1
```

```
data:
  key:
VGhpcyB3YXMgZ2VuZXJhdGVkIHdpdGggc3NoLWtleWdlbiBwdXJlbHkgZm9ylHRoaXM
gZXhhbXBsZS4KSXQgaXMgbm90IHVzZWQgYW55d2hlcmUuCi0tLS0tQkVHSU4gT1BF
TlNTSCBQUklWQVRFIEtFWS0tLS0tCmlzQmxibk56YUMxclpYa3RkakVBQUFBQUJHNX
ZibVVBQUFBRWJtOXVaUUFBQUFBQUFBQUJBQUFCbHdBQUFBZHpjMmd0Y24KTmh
BQUFBQXdFQUFRQUFBWUVBMzVTTFRReDBFVjdPTWJQR0FqcEsxK2JhQURTTVFu
K1NBU2pyTGZLNWM5NGpHdzhDbnA4LwovRHErZHFBR1pxQkg2ZnAxYmVJM1BZZzV
WVDk0RVdWQ2RrTjgwY3dEcEo0Z1R0NHFUQ1gzZUYvY2x5VXQyUC9zaTNjcnQ0CjBQ
di9wVnZXU1U2TlhHaDJlZC93V0MwcGh5Z0RQOVc5SHRQSUF0OFpnZmV2ZnUwZHpra
Vl6OHNVaElWU2ZsRGpaNUFqcUcKUjV2TVVUaGlrczEvZVlCeTdiMkFFSEdzYU8xN3NF
bWNiYUlHUHZuUFVwWmQrdjkyYU1JdWZoYjhLZkFSbzZ3Ty9lSW1VbQovdDdHWFBJU
mxBMUhSV0p1U05odTQzZS9DY3ZYd3Z6RnZrdE9kYXlEQzBMTklHMkpVaURlNWd0UU
Q1WHZXc1p3MHQvbEs1CklacjFrZXZRNUJsYWNlSmViV1ZNYUQvdllpdFdhSFo4OEF1
Y0czaGh2bjkrOGNSTGhNVExiVlFSMWh2UVpBL1JtQXN3eE0KT3VJSmRaUmtxTThLZl
F4Z28zQThRNGJhQW1VbnpvM3Zwa0FWdC9uaGtlOTRaRE5rV2U2RlRhdThONStyYTJ
CZkdjZVA4VApvbjFEeTBLRlpaUlpCREVVRVc0eHdTYUVOYXQ3c2RDNnhpL1d5OURa
QUFBRm1NRFBXeDdBejFzZUFBQUFCM056YUMxeWMyCkVBQUFHQkFOK1VpMDBNZ
EJGZXpqR3p4Z0k2U3RmbTJnQTBqRUova2dFbzZ5M3l1WFBlSXhzUEFwNmZQL3c2dm
5hZ0JtYWcKUituNmRXM2lOejJJT1ZVL2VCRmxRblpEZk5lTUE2U2VJRTdlS2t3bDkzaG
YzSmNsTGRqLzdJdDNLN2VORDcvNlZiMWtsTwpqVnhvZGgzZjhGZ3RLWWNvQXovVn
ZSN1R5QUxmR1lIM3lzN3RlYzVJbU0vTEZJU0ZVbjVRNDJlUUk2aGtlYnpGRTRZcExOC
mYzbUFjdTl5Z0JCeHJHanRlN0JKbkcyaUJqNzV6MUtXWGZyL2RtakNMbjRXL0Nud0Vh
T3NEdnh5SmxKdjdleGx6eUVaUU4KUjBWaWJrallidU4zdnduTDE4TDh4YjVMVG5Xc2d3
dEN6U0J0aVZJZzN1WUxVQStWNzFyR2NOTGY1U3VTR2E5WkhyME9RWgpXbkJ5WG0
xbFRHZy83MklyVm1oMmZQQUxuQnQ0WWI1L2Z2SEVTNFRFeTIxVUVkWWlwR1FQMF
pnTE1NVERyaUNYV1VaS2pQCkNuME1ZS053UEVPRzJnSmxKODZONzZaQUZiZjU0WkI
vZUdRelpGbnVoVTJydkRlZnEydGdYeG5lai9FNko5UTh0Q2hXV1UKV1FReEZCRnVNY0
VtaERXcmU3SFF1c1l2MXN2UTJRQUFBQU1CQUFFQUFBR0JBSlZtZklNNjdDQmpXcU9
KdnFua2EvakRrUwo4TDdpSE5mekg1TnRZWVdPWmRMTlk2L0lRa1pDeFcwTWtSKzlU
K0M3QUZKZzBNV2Q5ck5PeUxJZDkxNjZoOVJsNG0xdFJjCnViZ1o2dWZCZ3hGVDlXS2
1mSEdCNm4zelh5b2pQOEFJTnR6ODVpaUVHVXFFRWtVRVdMd0RGSmdvcFllQ3l1Vm
Z2ZE92MUgKRm1WWmEwNVo0b3NQNkNENXVmc2djQ1RYQTR6VnZ5ZHVCYkxqdHN5
RjdYZjNUdjZUQ1QxU0swZHErQk1OOXRvb0RZaXpwagpzbDh6NzlybXp3eUFyWFlVcnF
UUkpsNmpwRkNrWHJLcy9LeG96MHhhbXlMY2RORk9hWE51LzlnTkpjRERsV2hPcFRq
NHk4CkpkNXBuV1Jueis1RHJLRFdhY0loUW1CMUxVd2ZLWmQwbVFxaUpzMUMxcXZV
UmlKOGExaThKUTI4bHFuWTFRRk9wbk13emcKWEpla2FndThpT1ExRFJlQkhaM0Nkc
VJUYnY3bVJZSGxramx0dXJmZGc4M3hvM0ErZ1JSR001eUVOcW5xSkplQjhJQVB5Uwp
tMFp0dGdqbHNqNTJ2K1B1NmExMHoxZndKK1VML2N6dTRKeEpOYlp6WTFlMnpLODJ
BaVl1T3JYNmx2aUEvSWFSRVcwUUFBCkFNQndVeUJpcUc5bEZCUnltL2UvU1VORVM
zdHpicUZNdTdlcy84WTV5SnAxKzR6OXUxNGtJR2ttV0Y5eE5HT3hrY3V0cWwKeHVUcn
dMbjFUaFNQTHQrTjUwTGhVdzR4ZjBhNUxqemdPbkIPU0FRbm5HY1Nxa0dTRDlMR21o
bGE2WmpydFBHY29lQ3JHdAo5M1Vvcmx5YkxNRzFFRFAxWmpKS1RaZzl6OUMwdDlT
TGd3ei9DbFhydW9UNXNQVUdKWnUrbHllZXpSTDRtcHl6OEZMcnlOCkdNci9leVM5bW
dlSjNVVkZEYjNlZ3BaK1E1SUdBRU5rZVZEcHlwMGhCZXZndGd6YWtBQUFEQkFQVXQ
1RitoMnBVby94V1YKenRkcVQvMzA4dFB5MXVMMU1lWFoydEJPQmRwSDJyd0JzdWt0
aTlySGtWZUZXQjJFdUlFUXppMzY3MGc1UGdxR1p4Vng4dQpobEE0Rkg4ZXN1NTNQc
kZqVW9EeFJhb3d3WXBFcFh5Y2pnNUE1MStwR1VQcWljWjB0YjliaWlhc3BWWXZhWW5
sdGlnVG5iClN0UExMY29nemNiL0dGcVYyaXlzc3lwTlMwKzBNRTUxcEtxWGNaS2swbi8
vVHpZWWs4TW8vZzRsQ3pmUEZQUlZrVVM5blIKWU1pQzRlcEk0TERmbVdnM0xLQ2N
1Zk85all3aWgwYlFBQUFNRUE2WEtldDhEMHNvc0puZVh5WFZGd0dyVyszNlhBVGRQT
wpMWDdjaStjYzFoOGV1eHdYQWx3aTJJNFhxSmJBVjBsVEhuVGEycXN3Uy9RQlpJUU
JWSkZlVjVyS1daZTc4R2F3d1pWTFZNCldETmNwdFFyRTFaM2pGNS9TdUVzdlVxSDE0
Tkc5RUFXWG1iUkNzeIE0Vlk3NzQrSi9sTFkvMnlDT1diNzlLYTJ5OGxvYUoKVXczWWVtSl
d3blp2R3hKNldsL3BmQ2xYN3lEVXlXUktLdGl0cWNjbmpCWVkyRE1tZURwdURDYy9Zd
DZDc3dLRmRkMkJ1UwpGZGt5cDlZY3VMaDlLZEFBQUFIR3BoYzl5dVFFRlVMVGd3TW
```

```
xVdWJXOXVkR3hsYjl0dWFXNTBjbUVCQWdNRUJRWT0KLS0tLS1FTkQgT1BFTlNTSC
BQUklWQVRFIEtFWS0tLS0tCgo=
kind: Secret
metadata:
  name: ssh-credentials
  namespace: konveyor-forklift
type: Opaque
```

2. 通过同步一个文件并为 base64 传送它来对 playbook 进行编码，例如：

```
$ cat playbook.yml | base64 -w0
```



注意

您还可以使用此处文档对 playbook 进行编码：

```
$ cat << EOF | base64 -w0
- hosts: localhost
  tasks:
  - debug:
      msg: test
EOF
```

3. 创建 Hook CR：

```
apiVersion: forklift.konveyor.io/v1beta1
kind: Hook
metadata:
  name: playbook
  namespace: konveyor-forklift
spec:
  image: registry.redhat.io/rhmtc/openshift-migration-hook-runner-rhel8:v1.8.2-2
  playbook:
LSBuYW1lOiBNYWluCiAgaG9zdHM6IGxvY2FsaG9zdAogIHRhc2tzOgogIC0gbmFtZTog
TG9hZCBQbGFuCiAgICBpbmNsdWRlX3ZhcnM6CiAgICAgIGZpbGU6IHBsYW4ueW1sCi
AgICAgIG5hbWU6IHBsYW4KCiAgLSBuYW1lOiBMb2FkIFdvcmtsb2FkCiAgICBpbmNs
dWRlX3ZhcnM6CiAgICAgIGZpbGU6IHdvcmtsb2FkLnltbAogICAgICBuYW1lOiB3b3Jrb
G9hZAoKICAtIG5hbWU6IAogICAgZ2V0ZW50OgogICAgICBkYXRhYmFzZTogcGFzc3dk
CiAgICAgIGtleTogInt7IGVfdXNlcl9pZCB9fSIKICAgICAgc3BsaXQ6ICc6JwoKICAt
IG5hbWU6IEVuc3VyZSBTU0ggZGlyZWN0b3J5IGV4aXN0cwogICAgZmlsZToKICAgICAg
cGF0aDogfi8uc3NoCiAgICAgIHN0YXRlOiBkaXJlY3RvcnkKICAgICAgbW9kZTogMDc
1MAogICAgZW52aXJvbm1lbnQ6CiAgICAgIEhPTUU6ICJ7eyBhbnNpYmxlX2ZhY3RzLm
dldGVudF9wYXNzd2RbYW5zaWJsZV91c2VyX2lkXVs0XSB9fSIKCiAgLSBBrOHNfaW5mb
zoKICAgICAgYXBpX3ZlcnNpb246IHYxCiAgICAgIGtpbmQ6IFNlY3JldAogICAgICBuYW1
lOiBzc2gtY3JlZGVudGlhbHMKICAgICAgbmFtZXNwYWNlOiBrb252ZXlvci1mb3JrbGlmd
AogICAgcmVnaXN0ZXI6IHNzaF9jcmVkZW50aWFscwoKICAtIG5hbWU6IENyZWF0ZSBT
U0gga2V5CiAgICBjb3B5OgogICAgICBkZXN0OiB+Ly5zc2gvaWRfcnNhCiAgICAgIGNvb
nRlbnQ6ICJ7eyBzc2hfY3JlZGVudGlhbHMucmVzb3VyY2VzWzBdLmRhdGEua2V5lHwg
```

```
YjY0ZGVjb2RlIH19IgogICAgICBtb2RlOiAwNjAwCgogIC0gYWRkX2hvc3Q6CiAgICAgIG5
hbWU6ICJ7eyB3b3JrbG9hZC52bS5pcGFkZHJlc3MgfX0iCiAgICAgIGFuc2libGVfdXNlcj
ogcm9vdAogICAgICBncm91cHM6IHZtcwoKLSBob3N0czogdm1zCiAgdGFza3M6CiAgLSB
uYW1lOiBJbnN0YWxsIGNsb3VkLWluaXQKICAgIGRuZjoKICAgICAgbmFtZTogY2xvdWQt
CiAgICAgIHN0YXRlOiBsYXRlc3QKCiAgLSBuYW1lOiBDcmVhdGUgVGVzdCBGaWxlCiAgI
CBjb3B5OgogICAgICBkZXN0OiAvdGVzdC50eHQKICAgICAgY29udGVudDogIkhlbGxvIF
FdvcmxkIgogICAgICBtb2RlOiAwNjQ0Cg==
  serviceAccount: forklift-controller (1)
```

(1) (1) (1)

使用 指定 serviceAccount 运行 hook，以控制对集群上资源的访问。

注意

若要解码附加的 playbook，请检索带有自定义输出的资源并将其传送到 base64。例如：

```
oc get -n konveyor-forklift hook playbook -o \
  go-template='{{ .spec.playbook }}' | base64 -d
```

此处编码的 playbook 运行如下：

```
- name: Main
  hosts: localhost
  tasks:
  - name: Load Plan
    include_vars:
      file: plan.yml
      name: plan

  - name: Load Workload
    include_vars:
      file: workload.yml
      name: workload

  - name:
    getent:
      database: passwd
      key: "{{ ansible_user_id }}"
      split: ':'

  - name: Ensure SSH directory exists
    file:
      path: ~/.ssh
      state: directory
      mode: 0750
    environment:
```

```
    HOME: "{{ ansible_facts.getent_passwd[ansible_user_id][4] }}"

- k8s_info:
    api_version: v1
    kind: Secret
    name: ssh-credentials
    namespace: konveyor-forklift
  register: ssh_credentials

- name: Create SSH key
  copy:
    dest: ~/.ssh/id_rsa
    content: "{{ ssh_credentials.resources[0].data.key | b64decode }}"
    mode: 0600

- add_host:
    name: "{{ workload.vm.ipaddress }}"
    ansible_user: root
    groups: vms

- hosts: vms
  tasks:
  - name: Install cloud-init
    dnf:
      name:
      - cloud-init
      state: latest

  - name: Create Test File
    copy:
      dest: /test.txt
      content: "Hello World"
      mode: 0644
```

4. 使用 hook 创建 Plan CR：

```
kind: Plan
apiVersion: forklift.konveyor.io/v1beta1
metadata:
  name: test
  namespace: konveyor-forklift
spec:
  map:
    network:
      namespace: "konveyor-forklift"
      name: "network"
    storage:
      namespace: "konveyor-forklift"
      name: "storage"
  provider:
    source:
      namespace: "konveyor-forklift"
      name: "boston"
    destination:
```

```
    namespace: "konveyor-forklift"
    name: host
targetNamespace: "konveyor-forklift"
vms:
  - id: vm-2861
    hooks:
      - hook:
          namespace: konveyor-forklift
          name: playbook
        step: PreHook (1)
```

(1) 选项是 PreHook，用于在迁移前运行 hook，并在迁移后运行 hook。

重要

要让 PreHook 在虚拟机上运行，必须启动虚拟机并通过 SSH 使用。

第 7 章 升级 MIGRATION TOOLKIT FOR VIRTUALIZATION

您可以使用 Red Hat OpenShift Web 控制台安装新版本，以此升级 MTV Operator。

流程

1. 在 Red Hat OpenShift web 控制台中，点 Operators → Installed Operators → Migration Toolkit for Virtualization Operator → Subscription。

2. 将更新频道更改为正确的发行版本。

   请参阅 Red Hat OpenShift 文档中的更改更新频道。

3. 确认 Upgrade status 从 Up to date 变为 Upgrade available。如果没有，重启 CatalogSource pod：

   a. 记录目录源，如 redhat-operators。

   b. 在命令行中检索目录源 pod:

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

   c. 删除 Pod：

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

   升级状态从Up to date 改为 Upgrade available。

   如果您在 Subscriptions 选项卡上将 Update approval 设置为 Automatic，则升级会自动启动。

4. 如果您在 Subscriptions 标签页中将 Update approval 设置为 Manual，请批准升级。

请参阅 Red Hat OpenShift 文档中的 手动批准待处理的升级 部分。

5. 如果您要从 MTV 2.2 升级并定义了 VMware 源供应商，请通过添加 VDDK init 镜像来编辑 VMware 供应商。否则，更新会将任何 VMware 提供程序的状态更改为 Critical。如需更多信息，请参阅 添加 VMSphere 源供应商。

6. 如果您在 MTV 2.2 中的 Red Hat OpenShift 目的地供应商中映射到 NFS，请编辑 NFS 存储配置集中的 AccessModes 和 VolumeMode 参数。否则，升级会导致 NFS 映射无效。如需更多信息，请参阅自定义存储配置集。

第 8 章 卸载 MIGRATION TOOLKIT FOR VIRTUALIZATION

您可以使用 Red Hat OpenShift Web 控制台或命令行界面(CLI)卸载 Virtualization (MTV)。

8.1. 使用 RED HAT OPENSHIFT WEB 控制台卸载 MTV

您可以使用 Red Hat OpenShift Web 控制台删除 openshift-mtv 项目和自定义资源定义(CRD)来卸载 Virtualization (MTV)。

先决条件

- 您必须以具有 cluster-admin 权限的用户身份登录。

流程

1. 点 Home → Projects。
2. 找到 openshift-mtv 项目。
3. 在项目右侧，从 Options 菜单 ⋮ 选择 Delete Project。
4. 在 Delete Project 窗格中，输入项目名称，再点 Delete。
5. 点 Administration → CustomResourceDefinitions。
6. 在 Search 字段中输入 forklift 在 forklift.konveyor.io 组中查找 CRD。
7. 在每个 CRD 右侧，从 Options 菜单 ⋮ 中选择 Delete CustomResourceDefinition。

8.2. 使用命令行界面卸载 MTV

您可以通过删除 openshift-mtv 项目和 forklift.konveyor.io 自定义资源定义(CRD)从命令行界面(CLI)卸载 Migration Toolkit for Virtualization(MTV)。

先决条件

- 您必须以具有 cluster-admin 权限的用户身份登录。

流程

1. 删除项目：

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

2. 删除 CRD：

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

3. 删除 OAuthClient：

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

第 9 章 故障排除

本节提供有关对常见迁移问题进行故障排除的信息。

9.1. É”™È¨Ä¿¡Æ�¯

本节论述了错误消息以及如何解决它们。

已达到温导入重试的限制

如果在 precopy 阶段一个 VMware 虚拟机 (VM) 已达到最大的改变的块跟踪（CBT）快照的数量 (28)，则会在温迁移中会显 warm import retry limit reached 错误消息。

要解决这个问题，请从虚拟机中删除一些 CBT 快照并重启迁移计划。

无法将磁盘镜像调整为所需的大小

迁移失败时会显示 Unable to resize disk image to required size 错误消息，因为目标供应商的虚拟机使用块存储上带有 EXT4 文件系统的持久性卷。出现这个问题的原因是 CDI 假设的默认开销没有完全包括根分区的保留位置。

要解决这个问题，将 CDI 中的文件系统开销增加到大于 10%。

9.2. 使用 MUST-GATHER 工具

您可以使用 must-gather 工具来收集 MTV 自定义资源 (CR) 的日志和信息。您必须将 must-gather 数据文件附加到所有客户问题单。

您可以使用过滤选项为特定命名空间、迁移计划或虚拟机 (VM) 收集数据。

注意

如果您在过滤的 must-gather 命令中指定不存在的资源，则不会创建存档文件。

先决条件

- 您必须以具有 cluster-admin 角色的用户身份登录到 OpenShift Virtualization 集群。
- 已安装 Red Hat OpenShift CLI (oc)。

收集日志和 CR 信息

1. 进入存储 must-gather 数据的目录。
2. 运行 oc adm must-gather 命令：

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.6.2
```

数据被保存为 /must-gather/must-gather.tar.gz。您可以将此文件上传到红帽客户门户网站中的支持问题单中。

3. 可选：使用以下选项运行 oc adm must-gather 命令来收集过滤的数据：
   - 命名空间：

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.6.2 \
  -- NS=<namespace> /usr/bin/targeted
```

   - 迁移计划：

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.6.2 \
  -- PLAN=<migration_plan> /usr/bin/targeted
```

   - 虚拟机：

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.6.2 \
  -- VM=<vm_id> NS=<namespace> /usr/bin/targeted 1
```

1

9.3. 架构

本节论述了 MTV 自定义资源、服务和工作流。

9.3.1. MTV 自定义资源和服务

Migration Toolkit for Virtualization (MTV)作为 Red Hat OpenShift Operator 提供。它将创建和管理以下自定义资源 (CR) 和服务。

MTV 自定义资源

- Provider CR 存储启用 MTV 连接到并与源和目标供应商交互的属性。
- NetworkMapping CR 映射源供应商的网络。
- StorageMapping CR 会映射源和目标供应商的存储。
- Plan CR 包含具有相同迁移参数和相关网络和存储映射的虚拟机列表。
- Migration CR 运行一个迁移计划。

  每个迁移计划只能有一个 Migration CR 可以在指定时间运行。您可以为单个 Plan CR 创建多个 Migration CR。

MTV 服务

- Inventory 服务执行以下操作：
  - 连接到源和目标供应商。
  - 维护本地清单以进行映射和计划。

- 存储虚拟机配置。
- 如果检测到虚拟机配置更改，则运行 Validation 服务。

- Validation 服务通过应用规则检查虚拟机是否适合迁移。
- Migration Controller 服务编配迁移。

当您创建迁移计划时，Migration Controller 服务会验证计划并添加状态标签。如果计划无法验证，计划状态为 Not ready，则计划无法用于执行迁移。如果计划通过验证，计划状态为 Ready，它可用于执行迁移。迁移成功后，Migration Controller 服务会将计划状态更改为 Completed。

- Populator Controller 服务使用卷 Populator 编配磁盘加密。
- Kubevirt Controller 和 Containerized Data Import(CDI)Controller 服务处理大多数技术操作。

9.3.2. 高级别迁移工作流

高级别工作流显示用户视图的迁移过程：

1. 您可以创建一个源供应商、目标供应商、网络映射和存储映射。
2. 您可以创建一个包含以下资源的 Plan 自定义资源(CR)：
   - 源供应商
   - 目标供应商，如果目标集群上没有安装 MTV
   - 网络映射

- 存储映射
- 一个或多个虚拟机 (VM)

3. 您可以通过创建一个引用 Plan CR 的 Migration CR 来运行迁移计划。

   如果出于某种原因无法迁移所有虚拟机，则可以为同一 Plan CR 创建多个 Migration CR，直到虚拟机迁移为止。

4. 对于 Plan CR 中的每个虚拟机，Migration Controller 服务会在 Migration CR 中记录虚拟机迁移进度。

5. 在 Plan CR 中每个虚拟机的数据传输完成后，Migration Controller 服务会创建一个 VirtualMachine CR。

   当迁移所有虚拟机时，Migration Controller 服务会将 Plan CR 的状态更新为 Completed。每个源虚拟机的电源状态在迁移后会被维护。

9.3.3. 详细的迁移工作流

您可以使用详细的迁移工作流来排除迁移失败的问题。

工作流描述了以下步骤：

温迁移或迁移到远程 OpenShift 集群：

1. 当您创建 Migration 自定义资源(CR)来运行迁移计划时，Migration Controller 服务为每个源虚拟机磁盘创建一个 DataVolume CR。

   对于每个 VM 磁盘：

2. Containerized Data Importer(CDI) 控制器服务根据 DataVolume CR 中指定的参数创建一个持久性卷声明 (PVC)。

3. 如果 StorageClass 有动态置备程序，则 StorageClass 置备程序会动态置备持久性卷 (PV)。

4. CDI Controller 服务创建一个 importer pod。

5. importer pod 将虚拟机磁盘流传输到 PV。

   虚拟机磁盘传输后：

6. 从 VMWare 导入时，Migration Controller 服务会创建一个带有附加到它的 PVC 的转换 pod。

   conversion pod 运行 virt-v2v，它会在目标虚拟机的 PVC 中安装和配置设备驱动程序。

7. Migration Controller 服务为每个源虚拟机(VM)创建一个 VirtualMachine CR，连接到 PVC。

8. 如果虚拟机在源环境中运行，Migration Controller 在虚拟机上打开，KubeVirt Controller 服务会创建一个 virt-launcher pod 和 VirtualMachineInstance CR。

   virt-launcher pod 运行 QEMU-KVM，并附加了作为 VM 磁盘的 PVC。

从 RHV 或 OpenStack 冷迁移到本地 OpenShift 集群：

1. 当您创建 迁移 自定义资源(CR)来运行迁移计划时，Migration Controller 服务为每个源虚拟机磁盘创建一个 PersistentVolumeClaim CR，并在源为 RHV 时创建一个 OvirtVolumePopulator，或者在源是 OpenStack 时创建 OpenstackVolumePopulator CR。

   对于每个 VM 磁盘：

2. Populator Controller 服务创建一个临时持久性卷声明(PVC)。

3. 如果 StorageClass 有动态置备程序，则 StorageClass 置备程序会动态置备持久性卷

(PV)。

- Migration Controller 服务创建一个 dummy pod 来绑定 所有 PVC。pod 的名称包含 pvcinit。

4. Populator Controller 服务创建一个 填充器 pod。

5. 填充器 pod 将磁盘数据传送到 PV。

   虚拟机磁盘传输后：

6. 临时 PVC 已被删除，初始 PVC 指向带有数据的 PV。

7. Migration Controller 服务为每个源虚拟机(VM)创建一个 VirtualMachine CR，连接到 PVC。

8. 如果虚拟机在源环境中运行，Migration Controller 在虚拟机上打开，KubeVirt Controller 服务会创建一个 virt-launcher pod 和 VirtualMachineInstance CR。

   virt-launcher pod 运行 QEMU-KVM，并附加了作为 VM 磁盘的 PVC。

从 VMWare 冷迁移到本地 OpenShift 集群：

1. 当您创建 Migration 自定义资源(CR)来运行迁移计划时，Migration Controller 服务为每个源虚拟机磁盘创建一个 DataVolume CR。

   对于每个 VM 磁盘：

2. Containerized Data Importer (CDI) 控制器服务根据 DataVolume CR 中指定的参数创建一个空白持久性卷声明(PVC)。

3. 如果 StorageClass 有动态置备程序，则 StorageClass 置备程序会动态置备持久性卷(PV)。

对于所有虚拟机磁盘：

1. Migration Controller 服务创建一个 dummy pod 来绑定 所有 PVC。pod 的名称包含 pvcinit。
2. Migration Controller 服务为所有 PVC 创建一个 转换 pod。
3. conversion pod 运行 virt-v2v，它将虚拟机转换为 KVM hypervisor，并将磁盘的数据传送到对应的 PV。

   虚拟机磁盘传输后：
4. Migration Controller 服务为每个源虚拟机(VM)创建一个 VirtualMachine CR，连接到 PVC。
5. 如果虚拟机在源环境中运行，Migration Controller 在虚拟机上打开，KubeVirt Controller 服务会创建一个 virt-launcher pod 和 VirtualMachineInstance CR。

   virt-launcher pod 运行 QEMU-KVM，并附加了作为 VM 磁盘的 PVC。

9.4. 日志和自定义资源

您可以下载日志和自定义资源 (CR) 信息以进行故障排除。如需更多信息，请参阅详细的迁移工作流。

9.4.1. 收集日志和自定义资源信息

您可以使用 Red Hat OpenShift Web 控制台或命令行界面(CLI)为以下目标下载日志和自定义资源 (CR) yaml 文件：

- 迁移计划：Web 控制台或 CLI。
- 虚拟机：Web 控制台或 CLI.

- 命名空间：仅限 CLI。

must-gather 工具会在存档文件中收集以下日志和 CR 文件：

- CR:
  - DataVolume CR：代表在迁移的虚拟机中挂载的磁盘。
  - VirtualMachine CR：代表一个迁移的虚拟机。
  - Plan CR：定义 VM 和存储和网络映射。
  - Job CR：可选：代表迁移前 hook、迁移后 hook 或两者。
- 日志：
  - Importer pod: Disk-to-data-volume 转换日志。importer pod 的命名格式是 importer-<migration_plan>-<vm_id><5_char_id>，例如 importer-mig-plan-ed90dfc6-9a17-4a8btnfh，其中 ed90dfc6-9a17-4a8 是经过裁剪的 RHV VM ID，btnfh 是生成的 5 个字符的 ID。
  - conversion pod：虚拟机转换日志.conversion pod 运行 virt-v2v，它会在虚拟机的 PVC 中安装和配置设备驱动程序。conversion pod 的命名格式是 <migration_plan>-<vm_id><5_char_id>。
  - virt-launcher pod：VM launcher 日志.当迁移的虚拟机被开启后，virt-launcher Pod 运行 QEMU-KVM，并附加了作为虚拟机磁盘的 PVC。
  - forklift-controller pod：针对 must-gather 命令指定的迁移计划、虚拟机或命名空间过滤日志。
  - forklift-must-gather-api pod：日志针对 must-gather 命令指定的迁移计划、虚拟机

或命名空间过滤。

- **hook-job pod：针对 hook 任务过滤日志。hook-job 命名规则为 < migration_plan>-<vm_id><5_char_id >，例如 plan2j-vm-3696-posthook-4mx85 或 plan2j-vm-3696-prehook-mwqnl。**

注意

must-gather 归档文件中不包含空的或排除的日志文件。

VMware 迁移计划的 must-gather 归档结构示例

```
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. 从 web 控制台下载日志和自定义资源信息

您可以使用 Red Hat OpenShift web 控制台下载已完成的、失败或取消迁移计划的自定义资源(CR)的日志和信息。

流程

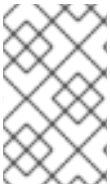
1. 在 Red Hat OpenShift web 控制台中，点 Migration → Plans for virtualization。
2. 点迁移计划名称旁的 Get logs。
3. 在 Get logs 窗口中点 Get logs。

   日志会被收集。此时会显示 Log collection complete 信息。
4. 点 Download logs 下载存档文件。
5. 要下载迁移的虚拟机的日志，请点迁移计划名称，然后点 VM 的 Get logs。

9.4.3. 使用命令行界面访问日志和自定义资源信息

您可以使用 must-gather 工具从命令行界面访问自定义资源(CR)的日志和信息。您必须将 must-gather 数据文件附加到所有客户问题单。

您可以使用过滤选项收集特定命名空间、完成、失败或取消迁移的虚拟机(VM)的数据。



注意

如果您在过滤的 must-gather 命令中指定不存在的资源，则不会创建存档文件。

先决条件

- 您必须以具有 cluster-admin 角色的用户身份登录到 OpenShift Virtualization 集群。

- 已安装 Red Hat OpenShift CLI (oc)。

流程

1. 进入要存储 must-gather 数据的目录。
2. 运行 oc adm must-gather 命令：

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.6.2
```

数据被保存为 /must-gather/must-gather.tar.gz。您可以将此文件上传到红帽客户门户网站中的支持问题单中。

3. 可选：使用以下选项运行 oc adm must-gather 命令来收集过滤的数据：

- 命名空间：

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.6.2 \
  -- NS=<namespace> /usr/bin/targeted
```

- 迁移计划：

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.6.2 \
  -- PLAN=<migration_plan> /usr/bin/targeted
```

- 虚拟机：

```
$ oc adm must-gather --image=registry.redhat.io/migration-toolkit-virtualization/mtv-must-gather-rhel8:2.6.2 \
  -- VM=<vm_name> NS=<namespace> /usr/bin/targeted 1
```

1. 您必须指定虚拟机 *名称*，而不是虚拟机 ID，因为它出现在 Plan CR 中。