



OpenShift Container Platform 4.2

迁移

从 OpenShift Container Platform 3 迁移到 4

OpenShift Container Platform 4.2 迁移

从 OpenShift Container Platform 3 迁移到 4

法律通告

Copyright © 2020 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

摘要

本文档提供了将 OpenShift Container Platform 集群从版本 3 迁移到版本 4 的信息。

目录

第 1 章 将 OPENSIFT CONTAINER PLATFORM 3 迁移到 4	3
1.1. 关于将 OPENSIFT CONTAINER PLATFORM 3 迁移到 4	3
1.2. 规划迁移	3
1.3. 将应用程序工作负载从 OPENSIFT CONTAINER PLATFORM 3.7 迁移到 4.2	7
1.4. 配置复制存储库	10
1.5. 部署集群应用程序迁移 (CAM) 工具	20
1.6. 使用 CAM WEB 控制台迁移应用程序	23
1.7. 使用 CONTROL PLANE MIGRATION ASSISTANT (CPMA) 迁移 CONTROL PLANE 设置	28
1.8. 故障排除	32
第 2 章 把 OPENSIFT CONTAINER PLATFORM 4.1 迁移到 4.2	41
2.1. 将应用程序工作负载从 OPENSIFT CONTAINER PLATFORM 4.1 迁移到 4.2	41
2.2. 配置复制存储库	43
2.3. 部署集群应用程序迁移 (CAM) 工具	53
2.4. 使用 CAM WEB 控制台迁移应用程序	55
2.5. 故障排除	61
第 3 章 在 OPENSIFT CONTAINER PLATFORM 4.2 集群间迁移	70
3.1. 将应用程序工作负载在 OPENSIFT CONTAINER PLATFORM 4.2 集群间迁移	70
3.2. 配置复制存储库	72
3.3. 部署集群应用程序迁移 (CAM) 工具	82
3.4. 使用 CAM WEB 控制台迁移应用程序	84
3.5. 故障排除	89

第 1 章 将 OPENSIFT CONTAINER PLATFORM 3 迁移到 4

1.1. 关于将 OPENSIFT CONTAINER PLATFORM 3 迁移到 4

OpenShift Container Platform 4 包含新的技术和功能，这些技术和功能可使集群具有自我管理、灵活性和自动化的特性。OpenShift Container Platform 4 集群部署和管理的方式与 OpenShift Container Platform 3 有很大不同。

要成功从 OpenShift Container Platform 3 转换到 OpenShift Container Platform 4，您必须检查以下信息：

规划系统迁移

了解 OpenShift Container Platform 版本 3 和 4 之间的不同。在迁移前，请确定您已对存储、网络、日志记录、安全及监控等方面进行了规划。

执行迁移

了解并使用以下工具执行迁移操作：

- 用于迁移应用程序负载的集群应用程序迁移 (CAM) 工具
- 用于迁移 control plane 的 Control Plane Migration Assistant (CPMA)

1.2. 规划迁移

在把系统迁移到 OpenShift Container Platform 4.2 前，需要花时间来正确地规划整个转换过程。OpenShift Container Platform 4 引入了与架构相关的更改和增强，因此您用来管理 OpenShift Container Platform 3 集群的操作可能不适用于 OpenShift Container Platform 4。



注意

本计划文档假定您要从 OpenShift Container Platform 3.11 转换到 OpenShift Container Platform 4.2。

本文档提供了有关 [OpenShift Container Platform 3 和 OpenShift Container Platform 4 之间重要的区别](#)，以及在迁移时需要注意的[重要迁移注意事项](#)。有关配置 OpenShift Container Platform 4 集群的详细信息，请查阅 OpenShift Container Platform 文档的适当章节。如需了解有关新功能和其他显著技术更改的详细信息，请参阅 [OpenShift Container Platform 4.2 发行注记](#)。

无法将现有 OpenShift Container Platform 3 集群升级到 OpenShift Container Platform 4。您必须开始新的 OpenShift Container Platform 4 安装。然后使用提供的工具来帮助迁移 control plane 设置和应用程序工作负载。

1.2.1. OpenShift Container Platform 3 和 OpenShift Container Platform 4 的比较

在 OpenShift Container Platform 3 中，管理员单独部署 Red Hat Enterprise Linux (RHEL) 主机，然后在这些主机之上安装 OpenShift Container Platform 来组成集群。管理员负责正确配置这些主机并执行更新。

在 OpenShift Container Platform 4 中，OpenShift Container Platform 集群的部署和管理方式有了显著变化。OpenShift Container Platform 4 包括新的技术和功能，如 Operators、MachineSets 和 Red Hat Enterprise Linux CoreOS (RHCOS)，它们是集群操作的核心。这个技术转换使集群能够自我管理以前需要由管理员执行的一些功能。这也确保平台的稳定性和一致性，并简化了安装和扩展。

如需更多信息，请参阅 [OpenShift Container Platform 架构](#)。

1.2.1.1. 架构的不同

不可变基础架构

OpenShift Container Platform 4 使用 Red Hat Enterprise Linux CoreOS (RHCOS)，它旨在运行容器化应用程序，并提供有效的安装、基于 Operator 的管理以及简化的升级。RHCOS 是不可变容器主机，而不是类似 RHEL 的可定制操作系统。RHCOS 使 OpenShift Container Platform 4 能够管理和自动化底层容器主机的部署。RHCOS 是 OpenShift Container Platform 的一部分，这意味着所有版本都在容器中运行，并且都使用 OpenShift Container Platform 部署。

在 OpenShift Container Platform 4 中，control plane 节点必须运行 RHCOS，以确保为 control plane 维护全堆栈的自动化。这使得更新和升级的过程比在 OpenShift Container Platform 3 中要容易。

如需更多信息，请参阅 [Red Hat Enterprise Linux CoreOS](#)。

Operator

Operator 是一种打包、部署和管理 Kubernetes 应用程序的方法。Operator 可简化运行另一部分软件的操作复杂性。它们会检测您的环境，并使用当前状态实时做出决定。高级 Operator 旨在自动升级并对失败做出适当的响应。

如需更多信息，请参阅 [Understanding Operators](#)。

1.2.1.2. 安装和更新的不同

安装过程

对于安装 OpenShift Container Platform 3.11，需要准备 Red Hat Enterprise Linux (RHEL) 主机，设置集群所需的所有配置值，然后运行 Ansible playbook 来安装和设置集群。

对于 OpenShift Container Platform 4.2，需要使用 OpenShift 安装程序创建集群所需的最小资源集合。集群运行后，您可以使用 Operator 来进一步配置集群并安装新服务。首次启动后，RHCOS 系统由 OpenShift Container Platform 集群中运行的 Machine Config Operator (MCO) 进行管理。

如需更多信息，请参阅 [安装过程](#)。

如果要将 RHEL worker 机器添加到 OpenShift Container Platform 4.2 集群中，您可以使用 Ansible playbook 在集群运行后加入 RHEL worker 机器。如需更多信息，请参阅 [在 OpenShift Container Platform 集群中添加 RHEL 计算机器](#)。

基础架构选项

对于 OpenShift Container Platform 3.11，需要在自己准备好的且被自己维护的基础架构上安装集群。对于 OpenShift Container Platform 4，除了可以在您自己提供的基础架构上安装集群外，还提供了一个在 OpenShift Container Platform 安装程序置备和集群维护的基础架构上部署集群的选项。

如需更多信息，请参阅 [OpenShift Container Platform 安装概述](#)。

升级集群

在 OpenShift Container Platform 3.11 中，您可以运行 Ansible playbook 来升级集群。在 OpenShift Container Platform 4.2 中，集群管理自己的更新，包括集群节点上的 Red Hat Enterprise Linux CoreOS (RHCOS) 的更新。您可以使用 Web 控制台或使用 OpenShift CLI 的 `oc adm upgrade` 命令轻松升级集群，Operator 会自动升级其自身。如果您的 OpenShift Container Platform 4.2 集群有 Red Hat Enterprise Linux worker 机器，那么您仍需要运行 Ansible playbook 来升级这些 worker 机器。

详细信息请参阅 [Updating a cluster](#)。

1.2.2. 迁移考虑

查看可能会影响 OpenShift Container Platform 3.11 转换到 OpenShift Container Platform 4 的更改和其他注意事项。

1.2.2.1. 存储注意事项

在从 OpenShift Container Platform 3.11 转换到 OpenShift Container Platform 4.2 时，请考虑以下与存储相关的变化。

本地卷持久性存储

只有在 OpenShift Container Platform 4.2 中使用 Local Storage Operator 才支持本地存储。不支持使用 OpenShift Container Platform 3.11 的 local provisioner 方法。

如需更多信息，请参阅 [使用本地卷的持久性存储](#)。

FlexVolume 持久性存储

FlexVolume 插件位置已与 OpenShift Container Platform 3.11 中的不同。它在 OpenShift Container Platform 4.2 中的新位置为 `/etc/kubernetes/kubelet-plugins/volume/exec`。不再支持可附加的 FlexVolume 插件。

如需更多信息，请参阅 [使用 FlexVolume 的持久性存储](#)。

使用容器存储接口 (CSI) 的持久性存储

使用 Container Storage Interface (CSI) 的持久性存储在 OpenShift Container Platform 3.11 中是一个 [技术预览](#) 功能。OpenShift Container Platform 4.2 完全支持 CSI 版本 1.1.0，但不附带任何 CSI 驱动程序。您必须安装自己的驱动程序。

使用容器存储接口 (CSI) 的持久性存储

OpenShift Container Storage

Red Hat OpenShift Container Storage 3 可以与 OpenShift Container Platform 3.11 一起使用，使用 Red Hat Gluster Storage 作为后端存储。

Red Hat OpenShift Container Storage 4 可以与 OpenShift Container Platform 4 一起使用，使用 Red Hat Ceph Storage 作为后备存储。

如需更多信息，请参阅 [使用 Red Hat OpenShift Container Storage 做为持久性存储](#) 和 [系统互操作性文档](#)。

可用的持久性存储选项

OpenShift Container Platform 4.2 中更改了对 OpenShift Container Platform 3.11 的以下持久性存储选项的支持：

- GlusterFS 不再被支持。
- CephFS 作为独立产品不再被支持。
- Ceph RBD 作为独立产品不再被支持。
- iSCSI 现在是技术预览。

如果您在 OpenShift Container Platform 3.11 中使用了其中之一，则需要选择不同的持久性存储选项以便在 OpenShift Container Platform 4.2 中提供全面支持。

如需更多信息，请参阅 [了解持久性存储](#)。

1.2.2.2. 网络注意事项

在从 OpenShift Container Platform 3.11 转换到 OpenShift Container Platform 4.2 时，请考虑以下与网络相关的变化。

网络隔离模式

虽然用户经常切换为使用 **ovn-multitenant**，但是 OpenShift Container Platform 3.11 的默认网络隔离模式是 **ovs-subnet**。OpenShift Container Platform 4.2 的默认网络隔离模式是 NetworkPolicy。

如果您的 OpenShift Container Platform 3.11 集群使用了 **ovs-subnet** 或 **ovs-multitenant** 模式，则建议在 OpenShift Container Platform 4.2 集群中将模式切换为 NetworkPolicy。NetworkPolicy 支持上游，更灵活，同时还提供 **ovs-multitenant** 的功能。如果您要在 OpenShift Container Platform 4.2 中使用 NetworkPolicy 时仍然希望维持 **ovs-multitenant** 的行为，请按照以下步骤[使用 NetworkPolicy 配置多租户隔离](#)。

如需更多信息，请参阅 [关于网络策略](#)。

加密主机间的网络数据

在 OpenShift Container Platform 3.11 中，您可以使用 IPsec 来加密主机间的网络流量。OpenShift Container Platform 4.2 不支持 IPsec。建议使用 Red Hat OpenShift Service Mesh 在服务间启用 mutual TLS。

如需更多信息，请参阅[了解 Red Hat OpenShift Service Mesh](#)。

1.2.2.3. 日志记录注意事项

在从 OpenShift Container Platform 3.11 转换到 OpenShift Container Platform 4.2 时，请考虑以下与日志相关的变化。

部署集群日志记录

OpenShift Container Platform 4 通过使用集群日志记录自定义资源，为集群日志记录提供了一个简单的部署机制。部署后，集群日志记录体验与 OpenShift Container Platform 3.11 相同。

如需更多信息，请参阅[关于部署和配置集群日志记录](#)。

聚合日志数据

您无法将 OpenShift Container Platform 3.11 的聚合日志记录数据转换到新的 OpenShift Container Platform 4 集群中。

如需更多信息，请参阅[关于集群日志记录](#)。

1.2.2.4. 安全考虑

在从 OpenShift Container Platform 3.11 转换到 OpenShift Container Platform 4.2 时，请考虑以下与安全相关的变化。

对发现端点的未验证访问

在 OpenShift Container Platform 3.11 中，未经身份验证的用户可以访问发现端点（例如：`/api/*` 和 `/apis/*`）。为了安全起见，OpenShift Container Platform 4.2 不再允许对发现端点进行未经身份验证的访问。如果确实需要允许未经身份验证的访问，可根据需要配置 RBAC 设置，但请务必考虑安全性影响，因为这可能会使内部集群组件暴露给外部网络。

用户身份供应商

为 OpenShift Container Platform 4 配置身份供应商包括以下显著的更改：

- OpenShift Container Platform 4.2 中的请求标头身份提供程序需要 mutual TLS，而这在 OpenShift Container Platform 3.11 中不需要。

- OpenShift Container Platform 4.2 中简化了 OpenID Connect 身份提供程序的配置。现在，它从供应商的 `/.well-known/OpenID-configuration` 端点获取数据。而之前需要在 OpenShift Container Platform 3.11 中指定。

如需更多信息，请参阅 [Understanding identity provider configuration](#)。

1.2.2.5. 监控注意事项

在从 OpenShift Container Platform 3.11 转换到 OpenShift Container Platform 4.2 时，请考虑以下与监控相关的变化。

监控基础架构可用性的警报

OpenShift Container Platform 3.11 中，触发的确保监控结构可用的默认警报称为 **DeadMansSwitch**。在 OpenShift Container Platform 4 中，它被重新命名为 **Watchdog**。如果您在 OpenShift Container Platform 3.11 中使用带有此警报设置的 PagerDuty 集成，则需要在 OpenShift Container Platform 4 中使用带有 **Watchdog** 警报设置的 PagerDuty 集成。

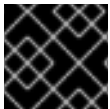
如需更多信息，请参阅 [应用自定义 Alertmanager 配置](#)。

1.3. 将应用程序工作负载从 OPENSIFT CONTAINER PLATFORM 3.7 迁移到 4.2

您可以使用集群应用程序迁移 (CAM) 工具将应用程序工作负载从 OpenShift Container Platform 3.7（及更高版本）迁移到 OpenShift Container Platform 4.2。使用 CAM 工具，您可以控制迁移并最小化应用程序的停机时间。

CAM 工具的 Web 控制台和 API，基于 Kubernetes 自定义资源，您可以按照命名空间迁移有状态应用程序工作负载。

另外，您还可以使用 [Control Plane Migration Assistant \(CPMA\)](#) 来帮助迁移 control plane 设置。



重要

在开始迁移前，请查看[规划迁移](#)中的信息。

1.3.1. 迁移先决条件

- 源集群必须是 OpenShift Container Platform 3.7、3.9、3.10 或 3.11。
- 必须安装 **podman**。
- 需要在所有集群中都有 **cluster-admin** 权限。
- 源和目标集群必须有对复制存储库的不受限制的网络访问权限。
- 安装 Migration controller 的集群必须具有对其他集群的不受限制的访问权限。
- 如果应用程序使用 **openshift** 命名空间中的镜像，则目标集群中必须有所需的镜像版本。如果没有所需的镜像，您必须更新 **imagestreamtags** 的引用以使用与应用程序兼容的可用版本。如果无法更新 **imagestreamtags**，您可以手动将相关的镜像上传到应用程序命名空间中，并更新应用程序以引用它们。

以下 **imagestreamtags** 已从 OpenShift Container Platform 4.2 中 *删除*：

- **dotnet:1.0, dotnet:1.1, dotnet:2.0**

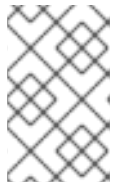
- **dotnet-runtime:2.0**
- **mariadb:10.1**
- **mongodb:2.4、mongodb:2.6**
- **mysql:5.5、mysql:5.6**
- **nginx:1.8**
- **nodejs:0.10、nodejs:4、nodejs:6**
- **perl:5.16、perl:5.20**
- **php:5.5、php:5.6**
- **postgresql:9.2、postgresql:9.4、postgresql:9.5**
- **python:3.3、python:3.4**
- **ruby:2.0、ruby:2.2**

1.3.2. 了解集群应用程序迁移工具

集群应用程序迁移 (CAM) 工具可让您使用 CAM web 控制台或 Kubernetes API 将 OpenShift Container Platform 源集群中的 Kubernetes 资源、持久性卷数据和内部容器镜像迁移到 OpenShift Container Platform 4.2 目标集群。

使用 CAM web 控制台迁移应用程序涉及以下步骤：

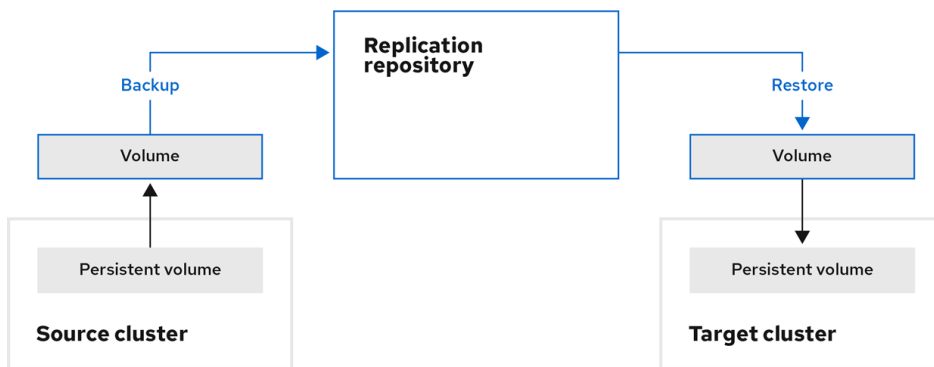
1. 在所有集群中安装 Cluster Application Migration Operator



注意

Cluster Application Migration Operator 默认在目标集群上安装 CAM 工具（CAM web 控制台和迁移控制器）。您可以在 [OpenShift Container Platform 3](#) 和 [远程集群中安装 CAM 工具](#)。

2. 配置复制存储库，这是 CAM 工具用来迁移数据的中间对象存储
3. 在 CAM web 控制台中添加源集群
4. 在 CAM web 控制台中添加复制存储库
5. 创建迁移计划，包含以下数据迁移选项之一：
 - **Copy**：CAM 工具将数据从源集群复制到复制存储库，再从复制存储库把数据复制到目标集群。



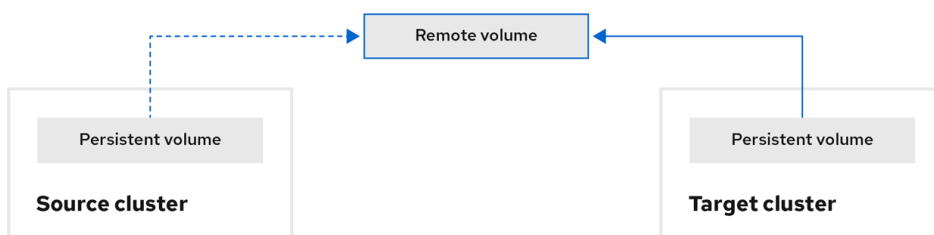
OpenShift_45_1019

- **Move** : CAM 工具从源集群中卸载一个远程卷（例如 NFS），在目标集群上创建一个指向这个远程卷的 PV 资源，然后在目标集群中挂载远程卷。在目标集群中运行的应用程序使用源集群使用的同一远程卷。远程卷必须可以被源集群和目标集群访问。



注意

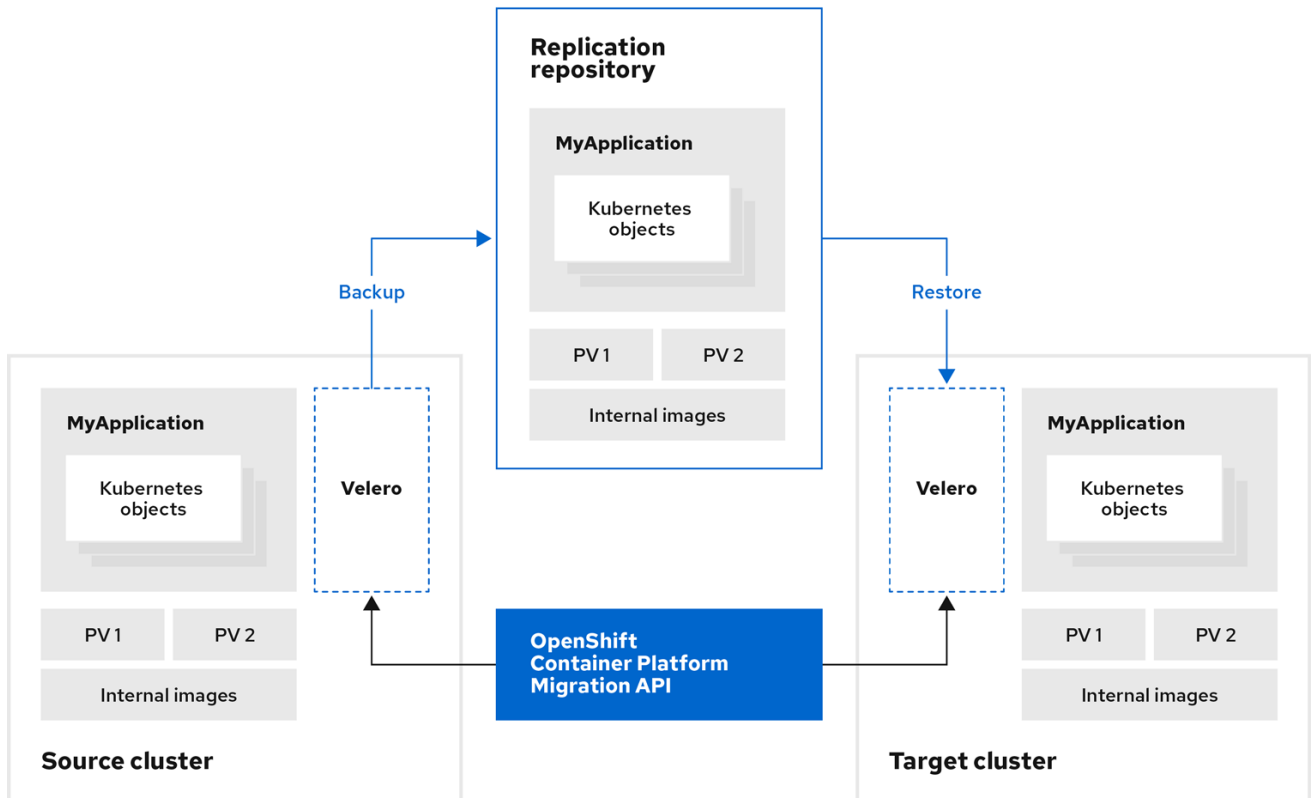
虽然复制存储库没有出现在此图表中，但实际迁移过程需要它。



OpenShift_45_1019

6. 运行迁移计划，使用以下选项之一：

- **Stage** （可选）在不停止应用程序的情况下将数据复制到目标集群。Stage 可以多次运行，以便在迁移前将大多数数据复制到目标。这样可最小化实际迁移时间和应用程序停机时间。
- **Migrate** 在源集群中停止应用程序，并在目标集群中重新创建其资源。您可以选择在不停止应用程序的情况下迁移工作负载。



OpenShift_45_1019

1.4. 配置复制存储库

您必须将对象存储配置为用作复制存储库。集群应用程序迁移工具将数据从源集群复制到复制存储库，然后使用文件系统或者快照数据复制方法从复制存储库复制到目标集群。

支持以下存储供应商：

- 通用 S3 对象存储，例如 Minio 或 Ceph S3
- 多云对象网关 (MCG)
- Amazon Web Services (AWS) S3
- Google Cloud Provider (GCP)
- Microsoft Azure

1.4.1. 了解用于迁移的数据复制方法

CAM 工具支持文件系统和快照数据复制方法，用于将数据从源集群迁移到目标集群。您可以选择适合于您的环境并受您的存储供应商支持的方法。

1.4.1.1. 文件系统复制方法

CAM 工具将数据文件从源集群复制到复制存储库，并从那里复制到目标集群。

表 1.1. 文件系统复制方法概述

优点	限制：
<ul style="list-style-type: none"> ● 集群可以有不同的存储类 ● 所有 S3 存储供应商均支持 	<ul style="list-style-type: none"> ● 比快照复制方法慢

1.4.1.2. 快照复制方法

CAM 工具将源集群的数据快照复制到云供应商的对象存储，后者配置为复制存储库。数据在目标集群上恢复。

AWS、Google Cloud Provider 和 Microsoft Azure 支持快照复制方法。

表 1.2. 快照复制方法概述

优点	限制：
<ul style="list-style-type: none"> ● 比文件系统复制方法快 	<ul style="list-style-type: none"> ● 云供应商必须支持快照。 ● 集群必须位于相同的云供应商。 ● 集群必须位于同一位置或区域。 ● 集群必须具有相同的存储类。 ● 存储类必须与快照兼容。



重要

将多云对象网关 (MCG) 配置为迁移的复制存储库只是技术预览功能。红帽产品服务等级协议 (SLA) 不支持技术预览功能，并且这些功能可能并不完善。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的详情，请参阅 <https://access.redhat.com/support/offerings/techpreview/>。

1.4.2. 配置 MCG 存储桶做为复制存储库

您可以安装 OpenShift Container Storage Operator，并将一个 Multi-Cloud Object Gateway (MCG) 存储桶配置为复制存储库。

1.4.2.1. 安装 OpenShift Container Storage Operator

您可以从 OperatorHub 安装 OpenShift Container Storage Operator。

流程

1. 在 OpenShift Container Platform web 控制台中，点 **Administration** → **Namespaces**。

2. 点 **Create Namespace**。
3. 在 **Name** 字段中输入 **openshift-storage**，点 **Create**。
4. 点 **Operators** → **OperatorHub**。
5. 使用 **Filter by keyword**（本例中为 **OCS**）来查找 **OpenShift Container Storage Operator**。
6. 选择 **OpenShift Container Storage Operator** 并点 **Install**。
7. 在 **Create Operator Subscription** 页面中，选择 **openshift-storage** 命名空间。
8. 指定您的更新频道和批准策略。
9. 点 **Subscribe**。
在 **Installed Operators** 页面中，**OpenShift Container Storage Operator** 会出现在 **openshift-storage** 项目中，状态为 **Succeeded**。

1.4.2.2. 创建 Multi-Cloud Object Gateway 存储桶

您可以创建 Multi-Cloud Object Gateway (MCG) 存储桶的自定义资源 (CR)。

流程

1. 登录到 OpenShift Container Platform 集群：

```
$ oc login
```

2. 使用以下内容创建 **NooBaa** CR 配置文件，**noobaa.yml**：

```
apiVersion: noobaa.io/v1alpha1
kind: NooBaa
metadata:
  name: noobaa
  namespace: openshift-storage
spec:
  dbResources:
    requests:
      cpu: 0.5 1
      memory: 1Gi
  coreResources:
    requests:
      cpu: 0.5 2
      memory: 1Gi
```

1 **2** 对于非常小的集群，您可以将 **cpu** 的值改为 **0.1**。

3. 创建 **NooBaa** 对象：

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

4. 使用以下内容创建 **BackingStore** CR 配置文件，**bs.yml**：

```
apiVersion: noobaa.io/v1alpha1
```



```

kind: BackingStore
metadata:
  finalizers:
  - noobaa.io/finalizer
  labels:
    app: noobaa
  name: mcg-pv-pool-bs
  namespace: openshift-storage
spec:
  pvPool:
    numVolumes: 3 ❶
    resources:
      requests:
        storage: 50Gi ❷
    storageClass: gp2 ❸
  type: pv-pool

```

- ❶ 指定 PV 池中的卷数量。
- ❷ 指定卷的大小。
- ❸ 指定存储类。

5. 创建 **BackingStore** 对象：

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

6. 使用以下内容创建 **BucketClass** CR 配置文件，**bc.yml**：

```

apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: noobaa
  name: mcg-pv-pool-bc
  namespace: openshift-storage
spec:
  placementPolicy:
    tiers:
    - backingStores:
      - mcg-pv-pool-bs
    placement: Spread

```

7. 创建 **BucketClass** 对象：

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

8. 使用以下内容创建 **ObjectBucketClaim** CR 配置文件，**obc.yml**：

```

apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: migstorage

```

```

namespace: openshift-storage
spec:
  bucketName: migstorage ❶
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: mcg-pv-pool-bc

```

- ❶ 记录下在 CAM web 控制台中添加为复制存储库的存储桶的名称。

9. 创建 **ObjectBucketClaim** 对象：

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

10. 监控资源创建过程以验证 **ObjectBucketClaim** 的状态变为 **Bound**：

```
$ watch -n 30 'oc get -n openshift-storage objectbucketclaim migstorage -o yaml'
```

这个过程可能需要五到十分钟。

11. 获取并记录以下值，当您复制存储库添加到 CAM web 控制台时需要这些值：

- S3 端点：

```
$ oc get route -n openshift-storage s3
```

- S3 provider access key:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{
.data.AWS_ACCESS_KEY_ID }}' | base64 -d
```

- S3 provider secret access key:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{
.data.AWS_SECRET_ACCESS_KEY }}' | base64 -d
```

1.4.3. 将 AWS S3 存储桶配置为复制存储库

您可以将 AWS S3 存储桶配置为复制存储库。

先决条件

- AWS S3 存储桶必须可以被源和目标集群访问。
- 您必须安装了 [AWS CLI](#)。
- 如果您使用快照复制方法：
 - 您必须有权访问 EC2 Elastic Block Storage (EBS)。
 - 源和目标集群必须位于同一区域。
 - 源和目标集群必须具有相同的存储类。

- 存储类必须与快照兼容。

流程

1. 创建 AWS S3 存储桶：

```
$ aws s3api create-bucket \
  --bucket <bucket_name> \ 1
  --region <bucket_region> 2
```

- 1 指定 S3 存储桶名称。
- 2 指定 S3 存储桶区域，例如 **us-east-1**。

2. 创建 IAM 用户 **velero**：

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

3. 创建 EC2 EBS 快照策略：

```
$ cat > velero-ec2-snapshot-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:DescribeSnapshots",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:CreateSnapshot",
        "ec2>DeleteSnapshot"
      ],
      "Resource": "*"
    }
  ]
}
EOF
```

4. 为一个或所有 S3 存储桶创建 AWS S3 访问策略：

```
$ cat > velero-s3-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3>DeleteObject",
        "s3:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
      "arn:aws:s3:::<bucket_name>/*" ❶
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket",
      "s3:GetBucketLocation",
      "s3:ListBucketMultipartUploads"
    ],
    "Resource": [
      "arn:aws:s3:::<bucket_name>" ❷
    ]
  }
]
}
EOF

```

- ❶ ❷ 要授予对单一 S3 存储桶的访问权限，请指定存储桶名称。要授予对所有 AWS S3 存储桶的访问权限，请指定 * 而不是存储桶名称：

```

"Resource": [
  "arn:aws:s3::*"
]

```

5. 将 EC2 EBS 策略附加到 **velero**：

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-ebs \
  --policy-document file://velero-ec2-snapshot-policy.json

```

6. 将 AWS S3 策略附加到 **velero**：

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-s3 \
  --policy-document file://velero-s3-policy.json

```

7. 为 **velero** 创建访问密钥：

```

$ aws iam create-access-key --user-name velero
{
  "AccessKey": {
    "UserName": "velero",
    "Status": "Active",
    "CreateDate": "2017-07-31T22:24:41.576Z",
    "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>, ❶
    "AccessKeyId": <AWS_ACCESS_KEY_ID> ❷
  }
}

```

- 1 2 记录 `AWS_SECRET_ACCESS_KEY` 和 `AWS_ACCESS_KEY_ID` 以将 AWS 存储库添加到 CAM Web 控制台。

1.4.4. 将 Google Cloud Provider 存储桶配置为复制存储库

您可以将 Google Cloud Provider (GCP) 存储桶配置为复制存储库。

先决条件

- AWS S3 存储桶必须可以被源和目标集群访问。
- 您必须安装了 `gsutil`。
- 如果您使用快照复制方法：
 - 源和目标集群必须位于同一区域。
 - 源和目标集群必须具有相同的存储类。
 - 存储类必须与快照兼容。

流程

1. 运行 `gsutil init` 以登录：

```
$ gsutil init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

To continue, you must login. Would you like to login (Y/n)?
```

2. 设置 `BUCKET` 变量：

```
$ BUCKET=<bucket_name> 1
```

- 1 指定存储桶名称。

3. 创建存储桶：

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

4. 将 `PROJECT_ID` 变量设置为您的活跃项目：

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

5. 创建 `velero` 服务帐户：

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

6. 将 `SERVICE_ACCOUNT_EMAIL` 变量设置为服务帐户的电子邮件地址：

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

7. 向服务帐户授予权限：

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

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

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

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

8. 将服务帐户的密钥保存到当前目录中的 **credentials-velero** 文件中：

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

1.4.5. 将 Microsoft Azure Blob 存储容器配置为复制存储库

您可以将 Microsoft Azure Blob 存储容器配置为复制存储库。

先决条件

- 您必须具有 [Azure 存储帐户](#)。
- 您必须安装了 [Azure CLI](#)。
- Azure Blob 存储容器必须可以被源和目标集群访问。
- 如果您使用快照复制方法：
 - 源和目标集群必须位于同一区域。
 - 源和目标集群必须具有相同的存储类。
 - 存储类必须与快照兼容。

流程

1. 设置 **AZURE_RESOURCE_GROUP** 变量 :

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

2. 创建 Azure 资源组 :

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

- 1 指定位置。

3. 设置 **AZURE_STORAGE_ACCOUNT_ID** 变量 :

```
$ AZURE_STORAGE_ACCOUNT_ID=velerobackups
```

4. 创建 Azure 存储帐户 :

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

5. 设置 **BLOB_CONTAINER** 变量 :

```
$ BLOB_CONTAINER=velero
```

6. 创建 Azure Blob 存储容器 :

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

7. 为 **velero** 创建服务主体和凭证 :

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

8. 在 **credentials-velero** 文件中保存服务主体的凭证 :

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

```
AZURE_RESOURCE_GROUP=${AZURE_RESOURCE_GROUP}
AZURE_CLOUD_NAME=AzurePublicCloud
EOF
```

1.5. 部署集群应用程序迁移 (CAM) 工具

部署集群应用程序迁移 (CAM) 工具需要在 [OpenShift Container Platform 3 源](#) 和 [OpenShift Container Platform 4.2 目标集群](#) 上安装 CAM Operator。并在 [OpenShift Container Platform 3 源集群](#) 中配置跨原始资源共享。

Cluster Application Migration Operator 默认在目标集群上安装 CAM 工具 (CAM web 控制台和迁移控制器)。您可以在 [OpenShift Container Platform 3 和远程集群中安装 CAM 工具](#)。

1.5.1. 在 OpenShift Container Platform 3 源集群上安装 Cluster Application Migration Operator

您可以手动在 OpenShift Container Platform 3 源集群上安装 Cluster Application Migration Operator，因为 OpenShift Container Platform 3 不支持 Operator Lifecycle Manager。

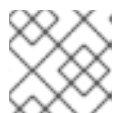
先决条件

- 必须安装 **podman**。
- 您的 OpenShift Container Platform 3 集群需要被配置为从 registry.redhat.io 拉取镜像。为了拉取镜像，您需要 [创建一个 imagestreamsecret](#)，并把它复制到集群中的每个节点。

流程

1. 使用您的红帽客户门户网站账户登陆到 registry.redhat.io：

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



注意

如果系统是为无根 Podman 容器配置的，则此过程不需要 **sudo**。

2. 下载 **operator.yml** 文件：

```
$ sudo podman cp $(sudo podman create registry.redhat.io/rhcam-1-2/openshift-migration-rhel7-operator:v1.2):/operator.yml ./
```

3. 下载 **controller-3.yml** 文件：

```
$ sudo podman cp $(sudo podman create registry.redhat.io/rhcam-1-2/openshift-migration-rhel7-operator:v1.2):/controller-3.yml ./
```

4. 登录您的 OpenShift Container Platform 3 集群。
5. 验证集群可以在 registry.redhat.io 中进行身份验证：

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


6. 创建 Cluster Application Migration Operator CR 对象：

```
$ oc create -f operator.yml
namespace/openshift-migration created
rolebinding.rbac.authorization.k8s.io/system:deployers created
serviceaccount/migration-operator created
customresourcedefinition.apiextensions.k8s.io/migrationcontrollers.migration.openshift.io
created
role.rbac.authorization.k8s.io/migration-operator created
rolebinding.rbac.authorization.k8s.io/migration-operator created
clusterrolebinding.rbac.authorization.k8s.io/migration-operator created
deployment.apps/migration-operator created
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-builders" already exists 1
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-pullers" already exists 2
```

- 1 2 您可以忽略 **Error from server (AlreadyExists)** 信息。它们是由 Cluster Application Migration Operator 为早期版本的 OpenShift Container Platform 3 创建资源造成的，这些资源在以后的版本中已提供。

7. 创建 Migration controller CR 对象：

```
$ oc create -f controller-3.yml
```

8. 确认 Velero 和 Restic Pod 正在运行：

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

1.5.2. 在 OpenShift Container Platform 4.2 目标集群上安装 Cluster Application Migration Operator

您可以使用 OLM 在 OpenShift Container Platform 4.2 目标集群上安装 Cluster Application Migration Operator。

默认情况下，Cluster Application Migration Operator 会在目标集群上安装 CAM 工具：

流程

1. 在 OpenShift Container Platform Web 控制台中，点击 **Operators → OperatorHub**。
2. 使用 **Filter by keyword** 项（在这里是 **Migration**）找到 **Cluster Application Migration Operator**。
3. 选择 **Cluster Application Migration Operator** 并点 **Install**。
4. 在 **Create Operator Subscription** 页面中，选择 **openshift-migration** 命名空间，并指定批准策略。
5. 点 **Subscribe**。
在 **Installed Operators** 页中，**Cluster Application Migration Operator** 会出现在 **openshift-migration** 项目中，其状态为 **InstallSucceeded**。

6. 在 **Provided APIs** 中点 **View 12 more...**
7. 点 **Create New → MigrationController**。
8. 点击 **Create**。
9. 点 **Workloads → Pod** 来验证 Controller Manager、Migration UI、Restic 和 Velero Pod 是否正在运行。

1.5.3. 在 OpenShift Container Platform 3 源集群中配置跨原始资源共享

您必须在 OpenShift Container Platform 3 源集群中配置跨原始资源共享，以启用源集群 API 服务器和 CAM 工具间的通信。

流程

1. 登录到已安装 CAM 工具的集群。
2. 获取 CORS 配置的值：

```
$ oc get -n openshift-migration route/migration -o go-template='{(?i)/{{ .spec.host }}(:|\z){{
println }}' | sed 's,\.,\\.g'
```

3. 登陆到 OpenShift Container Platform 3 源集群。
4. 将 CORS 配置值添加到 **/etc/origin/master/master-config.yaml** 配置文件的 **corsAllowedOrigins** 字段中：

```
corsAllowedOrigins:
- (?i)//migration-openshift-migration\.apps\.cluster\.com(:|\z) ❶
- (?i)//openshift\.default\.svc(:|\z)
- (?i)//kubernetes\.default(:|\z)
```

- ❶ 指定您的 CORS 配置。

5. 重启 API 服务器和控制器管理器以应用更改：

- 在 OpenShift Container Platform 3.7 和 3.9 中，这些组件作为独立主机进程运行，由 **systemd** 管理，并通过运行以下命令来重新启动：

```
$ systemctl restart atomic-openshift-master-api atomic-openshift-master-controllers
```

- 在 OpenShift Container Platform 3.10 和 3.11 中，这些组件在由 kubelet 管理的静态 Pod 中运行，并通过运行以下命令来重新启动：

```
$ /usr/local/bin/master-restart api
$ /usr/local/bin/master-restart controllers
```

6. 验证配置：

```
$ curl -v -k -X OPTIONS \
"<cluster_url>/apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migclusters" \ ❶
```

```
-H "Access-Control-Request-Method: GET" \  
-H "Access-Control-Request-Headers: authorization" \  
-H "Origin: https://<CAM_web_console_url>" 2
```

- 1 指定您配置的 CORS 集群的 URL。
- 2 指定 CAM web 控制台的 URL。URL 基于 CORS 配置值，例如 <https://migration-openshift-migration.apps.cluster>。

输出结果类似如下：

```
< HTTP/2 204  
< access-control-allow-credentials: true  
< access-control-allow-headers: Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Authorization, X-Requested-With, If-Modified-Since  
< access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, PATCH  
< access-control-allow-origin: https://migration-openshift-migration.apps.cluster  
< access-control-expose-headers: Date  
< cache-control: no-store
```

1.6. 使用 CAM WEB 控制台迁移应用程序

1.6.1. 启动 CAM web 控制台

您可以在浏览器中启动 CAM web 控制台。

流程

1. 登录到已安装 CAM 工具的 OpenShift Container Platform 集群。
2. 运行以下命令来获取 CAM web 控制台 URL:

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

3. 启动浏览器并进入 CAM web 控制台。



注意

如果在安装 CAM Operator 后尝试立即访问 CAM web 控制台，则该控制台可能无法加载，因为 Operator 仍然在配置集群并正在启用跨原始资源共享。等待几分钟后重试。

4. 如果您使用自签名的 CA 证书，则会提示您接受源集群 API 服务器的 CA 证书。网页会引导您接受剩余证书的过程。
5. 使用 OpenShift Container Platform 的用户名和密码进行登陆。

1.6.2. 把集群添加到 CAM web 控制台中

您可以将源集群添加到 CAM web 控制台。

- 您必须配置用于迁移数据的对象存储桶。

流程

1. 登录到 CAM web 控制台。
2. 在 **Replication repositories** 部分，点 **Add repository**。
3. 选择 **Storage provider type** 并填写以下字段：
 - **AWS** 适用于 S3、MCSG 和通用 S3 供应商：
 - **Replication repository name**：指定 CAM web 控制台中的复制存储库。
 - **S3 bucket name**：指定您创建的 S3 存储桶的名称。
 - **S3 bucket region**：指定 S3 存储桶区域。AWS S3 **必填**。Optional 用于其他 S3 供应商。
 - **S3 端点**：指定 S3 服务的 URL，而不是存储桶，例如：**https://<s3-storage.apps.cluster.com>**。通用 S3 供应商**必填**。您必须使用 **https://** 前缀。
 - **S3 provider access key**：为 AWS 指定 **<AWS_SECRET_ACCESS_KEY>**，或者为 MCG 指定 S3 供应商访问密钥。
 - **S3 provider secret access key**：为 AWS 指定 **<AWS_ACCESS_KEY_ID>**，或者为 MCG 指定 S3 供应商 secret 访问密钥。
 - **Require SSL verification**：如果您使用的是通用 S3 供应商，则清除此复选框。
 - **GCP**：
 - **Replication repository name**：指定 CAM web 控制台中的复制存储库。
 - **GCP bucket name**：指定 GCP 存储桶的名称。
 - **GCP credential JSON blob**：在 **credentials-velero** 文件中指定字符串。
 - **Azure**：
 - **Replication repository name**：指定 CAM web 控制台中的复制存储库。
 - **Azure resource group**：指定 Azure Blob 存储的资源组。
 - **Azure storage account name**：指定 Azure Blob 存储帐户名称
 - **Azure credentials - INI file contents**：在 **credentials-velero** 文件中指定字符串。
4. 点 **Add repository** 并等待连接验证。
5. 点 **Close**。
新存储库会出现在 **Replication repositories** 部分。

1.6.4. 为大型迁移修改迁移计划限制

您可以更改大型迁移的迁移计划限制。



重要

您需要首先在自己的环境对所做的更改进行测试，以避免迁移失败。

单个迁移计划有以下默认限制：

- 10 个命名空间
如果超过这个限制，CAM web 控制台会显示一个 **Namespace limit exceeded** 错误，您将无法创建迁移计划。
- 100 个 Pod
如果超过 Pod 限制，CAM web 控制台会显示类似以下示例的警告信息: **Plan has been validated with warning condition(s).查看警告信息。 pod limit: 100 exceeded, found: 104.**
- 100 个持久性卷（PV）
如果超过持久性卷限制，则 CAM web 控制台会显示类似的警告信息。

流程

1. 编辑迁移控制器 CR：

```
$ oc get migrationcontroller -n openshift-migration
NAME AGE
migration-controller 5d19h

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

2. 更新以下参数：

```
[...]
migration_controller: true

# This configuration is loaded into mig-controller, and should be set on the
# cluster where `migration_controller: true`
mig_pv_limit: 100
mig_pod_limit: 100
mig_namespace_limit: 10
[...]
```

1.6.5. 在 CAM web 控制台中创建迁移计划

您可以在 CAM web 控制台中创建迁移计划。

先决条件

- CAM web 控制台必须包含以下内容：
 - 源集群
 - 目标集群，它会在 CAM 工具安装过程中自动添加
 - 复制软件仓库
- 如果要使用快照复制数据，则源和目标集群必须在同一云供应商（AWS、GCP 或 Azure）以及同一区域运行。

流程

1. 登录到 CAM web 控制台。
2. 在 **Plans** 部分，点 **Add Plan**。
3. 输入 **Plan name** 并点 **Next**。
Plan name 最多可包含 253 个小写字母数字字符 (**a-z, 0-9**)。它不能包含空格或下划线 (**_**)。
4. 选一个 **Source cluster**。
5. 选一个 **Target cluster**。
6. 选一个 **Replication repository**。
7. 选择要迁移的项目并点 **Next**。
8. 选择 **Copy** 或 **Move PV** :
 - **Copy** 将源集群的 PV 中的数据复制到复制存储库中，然后在目标集群中新创建的具有类似特征的 PV 上恢复它。
 - **Move** 从源集群中卸载一个远程卷（例如 NFS），在目标集群上创建一个指向这个远程卷的 PV 资源，然后在目标集群中挂载远程卷。在目标集群中运行的应用程序使用源集群使用的同一远程卷。远程卷必须可以被源集群和目标集群访问。
9. 点 **Next**。
10. 为 PV 选择 **Copy method** :
 - **Snapshot** 使用云供应商的快照功能备份和恢复磁盘。它比 **Filesystem** 快得多。



注意

存储和集群必须位于同一区域，存储类必须兼容。

- **Filesystem** 将源磁盘中的数据文件复制到新创建的目标磁盘。
11. 为 PV 选择一个 **Storage class**。
 如果选择了 **Filesystem** 复制方法，您可以在迁移过程中更改存储类，例如：从 Red Hat Gluster Storage 或 NFS 存储改为 Red Hat Ceph Storage。
 12. 点 **Finish**。
 13. 点 **Close**。
 迁移计划会出现在 **Plans** 部分。

1.6.6. 在 CAM web 控制台中运行迁移计划

您可以使用在 CAM web 控制台中创建的迁移计划来 stage 或迁移应用程序和数据。

先决条件

CAM web 控制台必须包含以下内容：

- 源集群

- 目标集群，它会在 CAM 工具安装过程中自动添加
- 复制软件仓库
- 有效的迁移计划

流程

1. 登录到 OpenShift Container Platform 4 集群上的 CAM web 控制台。
2. 选择迁移计划。
3. 点 **Stage** 以在不停止应用程序的情况下，将数据从源集群复制到目标集群。
您可以多次运行 **Stage** 以减少实际迁移时间。
4. 当准备好迁移应用程序工作负载时，点 **Migrate**。
Migrate 在源集群中停止应用程序工作负载，并在目标集群中重新创建其资源。
5. 另外，还可以在 **Migrate** 窗口中选择 **Do not stop applications on the source cluster during migration**。
6. 点 **Migrate**。
7. 迁移完成后，在 OpenShift Container Platform 4.2 web 控制台中确认已成功迁移了应用程序：
 - a. 点 **Home** → **Projects**。
 - b. 点迁移的项目查看其状态。
 - c. 在 **Routes** 部分，点击 **Location** 验证应用程序是否正常运行。
 - d. 点 **Workloads** → **Pods** 来验证 Pod 在迁移的命名空间中运行。
 - e. 点 **Storage** → **Persistent volumes** 确认正确置备了被迁移的持久性卷。

1.7. 使用 CONTROL PLANE MIGRATION ASSISTANT (CPMA) 迁移 CONTROL PLANE 设置

1.7.1. 了解 Control Plane Migration Assistant

Control Plane Migration Assistant (CPMA) 是一个基于 CLI 的工具，它可帮助您将 control plane 从 OpenShift Container Platform 3.7（或更新版本）迁移到 OpenShift Container Platform 4.2。CPMA 处理 OpenShift Container Platform 3 配置文件并生成自定义资源 (CR) 清单文件，这些文件由 OpenShift Container Platform 4.2 Operator 使用。

因为 OpenShift Container Platform 3 和 4 的配置差别很大，所以不是所有的参数都会被处理。CPMA 可生成报告，描述功能是否被全面支持、部分支持或根本不支持。

配置字段

CPMA 使用 Kubernetes 和 OpenShift Container Platform API 来访问 OpenShift Container Platform 3 集群中的以下配置文件：

- master 配置文件（默认为 `/etc/origin/master/master-config.yaml`）
- CRI-O 配置文件（默认为 `/etc/crio/crio.conf`）

- etcd 配置文件（默认为 `/etc/etcd/etcd.conf`）
- 镜像 registry 文件（默认为 `/etc/containers/registries.conf`）
- 依赖性配置文件：
 - 密码文件（例如: `HTPasswd`）
 - `ConfigMaps`
 - `secret`

CR 清单

CPMA 为以下配置生成 CR 清单：

- API 服务器 CA 证书: `100_CPMA-cluster-config-APISecret.yaml`



注意

如果您使用的是未签名的 API 服务器 CA 证书，则必须手动将证书添加到目标集群中。

- CRI-O: `100_CPMA-crio-config.yaml`
- 集群资源配额: `100_CPMA-cluster-quota-resource-x.yaml`
- 项目资源配额: `100_CPMA-resource-quota-x.yaml`
- 可移植镜像 registry (`/etc/registries/registries.conf`) 和可移植镜像策略 (`etc/origin/master/master-config.yaml`) : `100_CPMA-cluster-config-image.yaml`
- OAuth 提供程序 : `100_CPMA-cluster-config-oauth.yaml`
- 项目配置: `100_CPMA-cluster-config-project.yaml`
- 调度程序 : `100_CPMA-cluster-config-scheduler.yaml`
- SDN: `100_CPMA-cluster-config-sdn.yaml`

1.7.2. 安装 Control Plane Migration Assistant

您可以从红帽客户门户网站下载 Control Plane Migration Assistant (CPMA) 二进制文件，并在 Linux、MacOSX 或者 Windows 操作系统中安装它。

流程

1. 在 [Red Hat 客户门户网站](#) 中，导航至 `Downloads → Red Hat OpenShift Container Platform`。
2. 在 `Download Red Hat OpenShift Container Platform` 页面中，从 `Product Variant` 列表中选择 `Red Hat OpenShift Container Platform`。
3. 从 `Version` 列表中选择 `CPMA 1.0 for RHEL 7`。这个二进制文件适用于 RHEL 7 和 RHEL 8。
4. 点 `Download Now` 为 Linux 或 MacOSX 下载 `cpma`，或为 Windows 下载 `cpma.exe`。

- 对于 Linux 或 MacOSX，把文件保存在由 **\$PATH** 定义的目录中；对于 Windows，把文件保存在由 **%PATH%** 定义的目录中。
- 对于 Linux，把文件设置为可执行：

```
$ sudo chmod +x cpma
```

1.7.3. 使用 Control Plane Migration Assistant

Control Plane Migration Assistant (CPMA) 生成 CR 清单，由 OpenShift Container Platform 4.2 Operator 使用，并产生包括哪些 OpenShift Container Platform 3 的功能被完全支持、部分支持或根本不支持的报告。

CPMA 可在远程模式下运行，使用 SSH 从源集群中检索配置文件，也可以使用本地模式，使用源集群配置文件的本地副本获取配置文件。

先决条件

- 源集群必须是 OpenShift Container Platform 3.7 或更高版本。
- 必须将源集群更新至最新的同步版本。
- 必须在源集群中运行环境健康检查来确定没有诊断错误或警告。
- CPMA 二进制文件必须是可执行文件。
- 必须具有源集群的 **cluster-admin** 权限。

流程

- 登陆到 OpenShift Container Platform 3 集群：

```
$ oc login https://<master1.example.com> 1
```

- 1 OpenShift Container Platform 3 主（master）节点。您必须登录到集群以接收 Kubernetes 和 OpenShift Container Platform API 的令牌。

- 运行 CPMA。根据每个提示输入，如下例所示：

```
$ cpma --manifests=false 1
? Do you wish to save configuration for future use? true
? What will be the source for OCP3 config files? Remote host 2
? Path to crio config file /etc/crio/crio.conf
? Path to etcd config file /etc/etcd/etcd.conf
? Path to master config file /etc/origin/master/master-config.yaml
? Path to node config file /etc/origin/node/node-config.yaml
? Path to registries config file /etc/containers/registries.conf
? Do wish to find source cluster using KUBECONFIG or prompt it? KUBECONFIG
? Select cluster obtained from KUBECONFIG contexts master1-example-com:443
? Select master node master1.example.com
? SSH login root 3
? SSH Port 22
? Path to private SSH key /home/user/.ssh/openshift_key
? Path to application data, skip to use current directory .
```

```

INFO[29 Aug 19 00:07 UTC] Starting manifest and report generation
INFO[29 Aug 19 00:07 UTC] Transform:Starting for - API
INFO[29 Aug 19 00:07 UTC] APITransform::Extract
INFO[29 Aug 19 00:07 UTC] APITransform::Transform:Reports
INFO[29 Aug 19 00:07 UTC] Transform:Starting for - Cluster
INFO[29 Aug 19 00:08 UTC] ClusterTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportQuotas
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportPVs
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportNamespaces
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportNodes
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportRBAC
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportStorageClasses
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - Crio
INFO[29 Aug 19 00:08 UTC] CrioTransform::Extract
WARN[29 Aug 19 00:08 UTC] Skipping Crio: No configuration file available
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - Docker
INFO[29 Aug 19 00:08 UTC] DockerTransform::Extract
INFO[29 Aug 19 00:08 UTC] DockerTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - ETCD
INFO[29 Aug 19 00:08 UTC] ETCDTransform::Extract
INFO[29 Aug 19 00:08 UTC] ETCDTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - OAuth
INFO[29 Aug 19 00:08 UTC] OAuthTransform::Extract
INFO[29 Aug 19 00:08 UTC] OAuthTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - SDN
INFO[29 Aug 19 00:08 UTC] SDNTransform::Extract
INFO[29 Aug 19 00:08 UTC] SDNTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - Image
INFO[29 Aug 19 00:08 UTC] ImageTransform::Extract
INFO[29 Aug 19 00:08 UTC] ImageTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - Project
INFO[29 Aug 19 00:08 UTC] ProjectTransform::Extract
INFO[29 Aug 19 00:08 UTC] ProjectTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Flushing reports to disk
INFO[29 Aug 19 00:08 UTC] Report:Added: report.json
INFO[29 Aug 19 00:08 UTC] Report:Added: report.html
INFO[29 Aug 19 00:08 UTC] Successfully finished transformations

```

- 1 **--manifests=false** : 不生成 CR 清单
- 2 **Remote host** : 远程模式
- 3 **SSH login** : 为了访问配置文件, SSH 用户必须在 OpenShift Container Platform 3 集群上具有 **sudo** 权限。

如果您没有指定输出目录, CPMA 会在当前目录中创建以下文件和目录:

- **cpma.yaml** 文件: 运行 CPMA 时提供的配置选项
- **master1.example.com/**: master 节点中的配置文件
- **report.json**: JSON 格式的报告
- **report.html**: HTML 格式的报告

3. 在浏览器中打开 **report.html** 文件来查看 CPMA 报告。

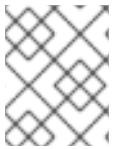
4. 如果生成 CR 清单，将 CR 清单应用到 OpenShift Container Platform 4.2 集群，如下例所示：

```
$ oc apply -f 100_CPMA-cluster-config-secret-htpasswd-secret.yaml
```

1.8. 故障排除

您可以查看迁移自定义资源 (CR)，并下载日志来排除迁移失败的问题。

如果应用程序在迁移失败时停止，您必须手动回滚，以防止数据崩溃。

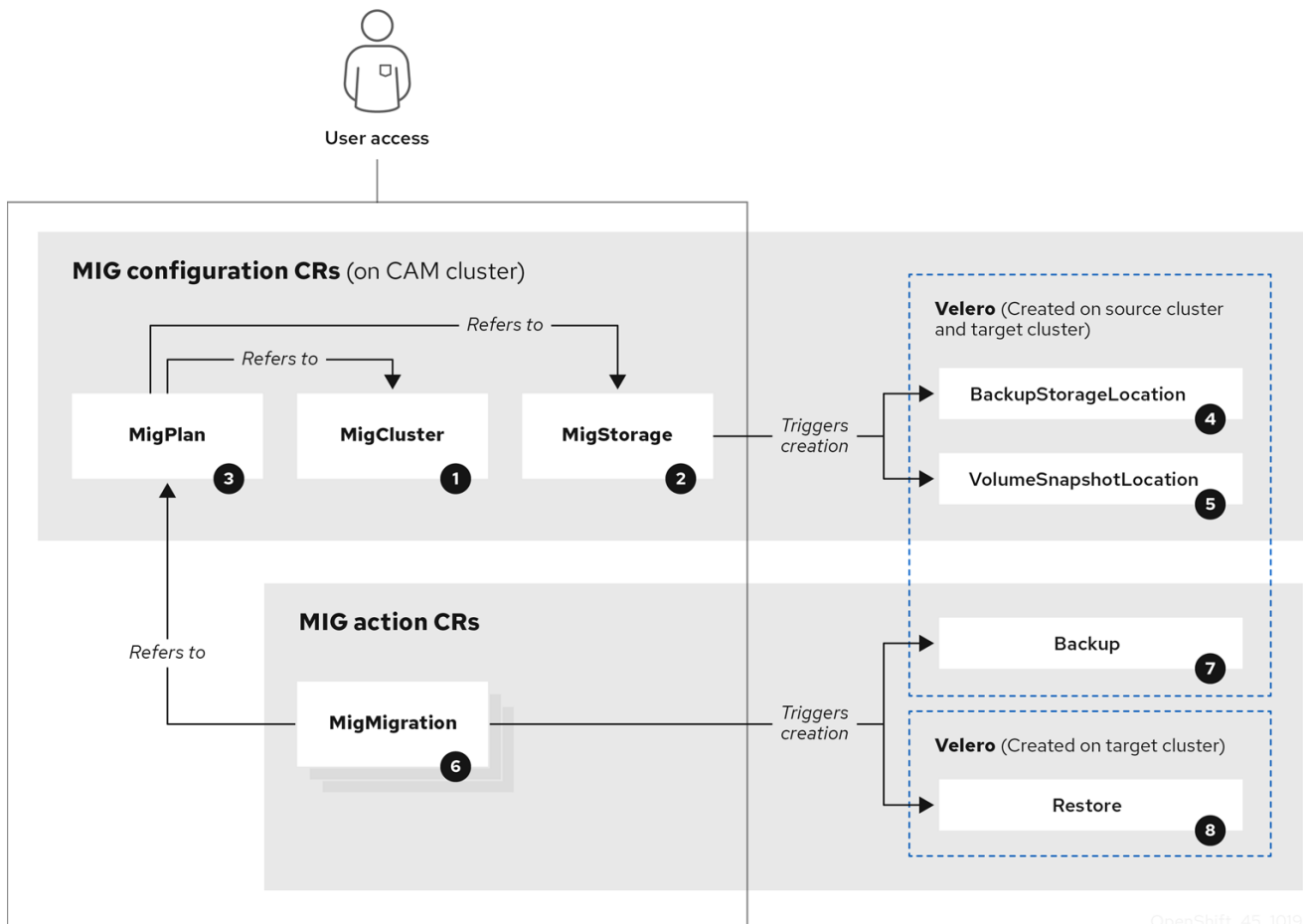


注意

如果应用程序在迁移过程中没有停止，则不需要手动回滚，因为原始应用程序仍然在源集群中运行。

1.8.1. 查看迁移自定义资源 (CR)

集群应用程序迁移 (CAM) 工具会创建以下 CR 用于迁移：



OpenShift_45_1019

- 1 MigCluster (配置, CAM 集群) : 集群定义
- 2 MigStorage (配置, CAM 集群) : 存储定义
- 3 MigPlan (配置, CAM 集群) : 迁移计划

MigPlan CR 描述了要迁移的源和目标集群、存储库和命名空间。它与 0 个、1 个或多个 MigMigration CR 关联。



注意

删除 MigPlan CR 会删除关联的 MigMigration CR。

4 `BackupStorageLocation` (配置, CAM 集群) : Velero 备份对象的位置

5 `VolumeSnapshotLocation` (配置, CAM 集群) : Velero 卷快照的位置

6 `MigMigration` (操作, CAM 集群) : Migration, 在迁移期间创建

在每次进行 stage 或迁移数据时都会创建一个 MigMigration CR。每个 MigMigration CR 都会与一个 MigPlan CR 关联。

7 `Backup` (操作, 源集群) : 当运行迁移计划时, MigMigration CR 在每个源集群上创建两个 Velero 备份 CR :

- 备份 CR #1 用于 Kubernetes 对象
- 备份 CR #2 用于 PV 数据

8 `Restore` (操作, 目标集群) : 在运行迁移计划时, MigMigration CR 在目标集群上创建两个 Velero 恢复 CR :

- 恢复 CR #1 (使用备份 CR #2) 用于 PV 数据
- 恢复 CR #2 (使用备份 CR #1) 用于 Kubernetes 对象

流程

1. 获取 CR 名称 :

```
$ oc get <cr> -n openshift-migration 1
NAME                               AGE
88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s
```

1 指定您要查看的迁移 CR。

2. 查看 CR :

```
$ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
```

输出结果类似以下示例。

MigMigration 示例

```
$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
Name:      88435fe0-c9f8-11e9-85e6-5d593ce65e10
Namespace: openshift-migration
```

```

Labels:      <none>
Annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
API Version: migration.openshift.io/v1alpha1
Kind:        MigMigration
Metadata:
  Creation Timestamp: 2019-08-29T01:01:29Z
  Generation:        20
  Resource Version:  88179
  Self Link:         /apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
  UID:               8886de4c-c9f8-11e9-95ad-0205fe66cbb6
Spec:
  Mig Plan Ref:
    Name:      socks-shop-mig-plan
    Namespace: openshift-migration
  Quiesce Pods: true
  Stage:       false
Status:
  Conditions:
    Category:      Advisory
    Durable:       true
    Last Transition Time: 2019-08-29T01:03:40Z
    Message:       The migration has completed successfully.
    Reason:        Completed
    Status:        True
    Type:          Succeeded
  Phase:         Completed
  Start Timestamp: 2019-08-29T01:01:29Z
Events:         <none>

```

Velero 备份 CR #2 示例 (PV 数据)

```

apiVersion: velero.io/v1
kind: Backup
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.105.179:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-
0205fe66cbb6
  creationTimestamp: "2019-08-29T01:03:15Z"
  generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
  generation: 1
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    velero.io/storage-location: myrepo-vpzq9
  name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  namespace: openshift-migration
  resourceVersion: "87313"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-
5d593ce65e10-59gb7
  uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:

```

```

excludedNamespaces: []
excludedResources: []
hooks:
  resources: []
includeClusterResources: null
includedNamespaces:
- sock-shop
includedResources:
- persistentvolumes
- persistentvolumeclaims
- namespaces
- imagestreams
- imagestreamtags
- secrets
- configmaps
- pods
labelSelector:
  matchLabels:
    migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
storageLocation: myrepo-vpzq9
ttl: 720h0m0s
volumeSnapshotLocations:
- myrepo-wv6fx
status:
  completionTimestamp: "2019-08-29T01:02:36Z"
  errors: 0
  expiration: "2019-09-28T01:02:35Z"
  phase: Completed
  startTimestamp: "2019-08-29T01:02:35Z"
  validationErrors: null
  version: 1
  volumeSnapshotsAttempted: 0
  volumeSnapshotsCompleted: 0
  warnings: 0

```

Velero 恢复 CR #2 示例 (Kubernetes 资源)

```

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
  generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
    migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
  name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  namespace: openshift-migration
  resourceVersion: "82329"

```

```

selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-
d129df7f3b96-gb8nx
uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
  - restores.velero.io
  - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
  namespaceMapping: null
  restorePVs: true
status:
  errors: 0
  failureReason: ""
  phase: Completed
  validationErrors: null
  warnings: 15

```

1.8.2. 下载迁移日志

您可以在 CAM web 控制台中下载 Velero、Restic 和 Migration controller 日志，以排除出现故障的迁移问题。

流程

1. 登录到 CAM 控制台。
2. 点击 **Plans** 查看迁移计划列表。
3. 点一个迁移计划的 **Options** 菜单  并选择 **Logs**。
4. 点 **Download Logs** 为所有集群下载迁移控制器、Velero 和 Restic 的日志。
5. 要下载特定的日志：
 - a. 指定日志选项：
 - **Cluster**：选择源、目标或 CAM 主机集群。
 - **Log source**：选择 **Velero**、**Restic** 或 **Controller**。
 - **Pod source**：选择 Pod 名称，例如：**controller-manager-78c469849c-v6wcf**
此时会显示所选日志。

您可以通过更改您的选择来清除日志选择设置。
 - b. 点 **Download Selected** 下载所选日志。

另外，您可以使用 CLI 访问日志，如下例所示：


```
$ oc get pods -n openshift-migration | grep controller
controller-manager-78c469849c-v6wcf      1/1   Running   0      4h49m

$ oc logs controller-manager-78c469849c-v6wcf -f -n openshift-migration
```

1.8.3. Restic 超时错误

如果因为 Restic 超时造成迁移失败，以下出错信息会出现在 Velero 日志中：

```
level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all
PodVolumeBackups to complete"
error.file="/go/src/github.com/heptio/velero/pkg/restic/backupper.go:165"
error.function="github.com/heptio/velero/pkg/restic.(*backupper).BackupPodVolumes" group=v1
```

`restic_timeout` 的默认值为一小时。您可以为大型迁移增加这个值，请注意，高的值可能会延迟返回出错信息。

流程

1. 在 OpenShift Container Platform web 控制台中导航至 **Operators** → **Installed Operators**。
2. 点 **Cluster Application Migration Operator**。
3. 在 **MigrationController** 标签页中点 **migration-controller**。
4. 在 **YAML** 标签页中，更新以下参数值：

```
spec:
  restic_timeout: 1h 1
```

1 有效单元是 **h**（小时）、**m**（分钟）和 **s**（秒），例如 **3h30m15s**。

5. 点 **Save**。

1.8.4. 手动回滚迁移

如果您的应用程序在迁移失败时停止，您必须手动回滚，以防止 PV 中的数据被破坏。

如果应用程序在迁移过程中没有停止，则不需要进行手动回滚，因为原始应用程序仍然在源集群中运行。

流程

1. 在目标集群中，切换到迁移的项目：

```
$ oc project <project>
```

2. 获取部署的资源：

```
$ oc get all
```

3. 删除部署的资源以确保应用程序没有在目标集群中运行，并访问 PVC 上的数据：

```
$ oc delete <resource_type>
```

- 要停止 DaemonSet 而不删除它，在 YAML 文件中更新 **nodeSelector**：

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: hello-daemonset
spec:
  selector:
    matchLabels:
      name: hello-daemonset
  template:
    metadata:
      labels:
        name: hello-daemonset
    spec:
      nodeSelector:
        role: worker ❶
```

- ❶ 指定一个没有存在于任何节点上的 **nodeSelector** 值。

- 更新每个 PV 的重新声明策略，以便删除不必要的数据库。在迁移过程中，绑定 PV 的重新声明策略是 **reclaim**，以确保应用程序从源集群中被删除时不会丢失数据。您可以在回滚过程中删除这些 PV。

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain ❶
  ...
status:
  ...
```

- ❶ 指定 **Recycle** 或 **Delete**。

- 在源集群中，切换到迁移的项目并获取其部署的资源：

```
$ oc project <project>
$ oc get all
```

- 启动每个部署资源的一个或多个副本：

```
$ oc scale --replicas=1 <resource_type>/<resource_name>
```

- 如果在操作中被更改了，把 DaemonSet 的 **nodeSelector** 改回其原始值。

1.8.5. 为客户支持问题单收集数据

如果创建一个客户支持问题单，您可以使用 **openshift-migration-must-gather-rhel8** 镜像的 **must-gather** 工具来收集与您的集群相关的信息，并把这些信息上传到[红帽客户门户网站](#)。

openshift-migration-must-gather-rhel8 镜像会收集默认的 **must-gather** 镜像不收集的日志和 CR 数据。

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/rhcam-1-2/openshift-migration-must-gather-rhel8
```

must-gather 工具程序收集集群数据，并把它保存在 **must-gather.local.<uid>** 目录中。

3. 从 **must-gather** 数据中删除验证密钥和其他敏感信息。
4. 创建一个包含 **must-gather.local.<uid>** 目录内容的归档文件：

```
$ tar cvaf must-gather.tar.gz must-gather.local.<uid>/
```

在[红帽客户门户](#)中，为您的问题单附上这个压缩文件。

1.8.6. 已知问题

这个版本有以下已知问题：

- 在迁移过程中，CAM 工具会保留以下命名空间注解：
 - **openshift.io/sa.scc.mcs**
 - **openshift.io/sa.scc.supplemental-groups**
 - **openshift.io/sa.scc.uid-range**
这些注解会保留 UID 范围，确保容器在目标集群中保留其文件系统权限。这可能会存在一定的风险。因为迁移的 UID 可能已存在于目标集群的现有或将来的命名空间中。
([BZ#1748440](#))
- 当在 CAM web 控制台添加 S3 端点时，只有 AWS 支持 **https://**。对于其他 S3 供应商，请使用 **http://**。
- 如果一个 AWS 存储桶被添加到 CAM web 控制台，然后将其删除，则其状态会保持为 **True**，这是因为 MigStorage CR 没有被更新。(BZ#1738564)
- 如果迁移控制器在目标集群以外的集群中运行，迁移将失败。**EnsureCloudSecretPropagated** 阶段会跳过，并给出一个日志警告。(BZ#1757571)
- 目前，集群范围的资源，包括集群角色绑定和安全上下文约束，还没有由 CAM 处理。如果应用程序需要集群范围的资源，则必须在目标集群中手动创建它们。(BZ#1759804)
- 创建迁移计划时会显示不正确的源集群存储类。(BZ#1777869)

- 如果 CAM web 控制台中的一个集群变得无法访问，它会阻止尝试关闭打开的迁移计划。
([BZ#1758269](#))
- 如果迁移失败，则迁移计划不会为静默的 pod 保留自定义 PV 设置。您必须手动回滚，删除迁移计划，并使用 PV 设置创建新的迁移计划。([BZ#1784899](#))

第 2 章 把 OPENSIFT CONTAINER PLATFORM 4.1 迁移到 4.2

2.1. 将应用程序工作负载从 OPENSIFT CONTAINER PLATFORM 4.1 迁移到 4.2

您可以使用 Cluster Application Migration (CAM) 工具将应用程序工作负载从 OpenShift Container Platform 4.1 迁移到 4.2。使用 CAM 工具，您可以控制迁移并最小化应用程序的停机时间。

CAM 工具的 web 控制台和 API，基于 Kubernetes 自定义资源，您可以按照命名空间迁移有状态及无状态的应用程序工作负载。

2.1.1. 迁移先决条件

- 需要在所有集群中都有 **cluster-admin** 权限。
- 源和目标集群必须有对复制存储库的不受限制的网络访问权限。
- 安装 Migration controller 的集群必须具有对其他集群的不受限制的访问权限。
- 如果应用程序使用 **openshift** 命名空间中的镜像，则目标集群中必须有所需的镜像版本。如果没有所需的镜像，您必须更新 **imagestreamtags** 的引用以使用与应用程序兼容的可用版本。如果无法更新 **imagestreamtags**，您可以手动将相关的镜像上传到应用程序命名空间中，并更新应用程序以引用它们。

以下 **imagestreamtags** 已从 OpenShift Container Platform 4.2 中 *删除*：

- **dotnet:1.0, dotnet:1.1, dotnet:2.0**
- **dotnet-runtime:2.0**
- **mariadb:10.1**
- **mongodb:2.4, mongodb:2.6**
- **mysql:5.5, mysql:5.6**
- **nginx:1.8**
- **nodejs:0.10, nodejs:4, nodejs:6**
- **perl:5.16, perl:5.20**
- **php:5.5, php:5.6**
- **postgresql:9.2, postgresql:9.4, postgresql:9.5**
- **python:3.3, python:3.4**
- **ruby:2.0, ruby:2.2**

2.1.2. 了解集群应用程序迁移工具

集群应用程序迁移 (CAM) 工具可让您使用 CAM web 控制台或 Kubernetes API 将 OpenShift Container Platform 源集群中的 Kubernetes 资源、持久性卷数据和内部容器镜像迁移到 OpenShift Container Platform 4.2 目标集群。

使用 CAM web 控制台迁移应用程序涉及以下步骤：

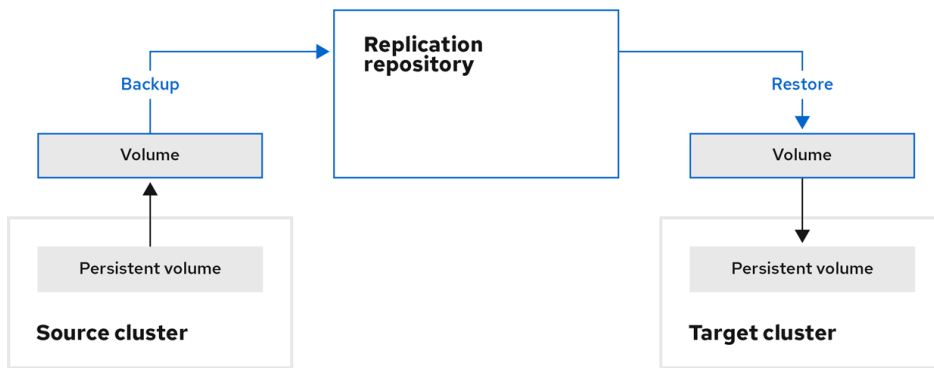
1. 在所有集群中安装 Cluster Application Migration Operator



注意

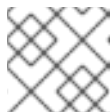
Cluster Application Migration Operator 默认在目标集群上安装 CAM 工具（CAM web 控制台和迁移控制器）。您可以在 [OpenShift Container Platform 3 和远程集群中安装 CAM 工具](#)。

2. 配置复制存储库，这是 CAM 工具用来迁移数据的中间对象存储
3. 在 CAM web 控制台添加源集群
4. 在 CAM web 控制台添加复制存储库
5. 创建迁移计划，包含以下数据迁移选项之一：
 - **Copy**：CAM 工具将数据从源集群复制到复制存储库，再从复制存储库把数据复制到目标集群。



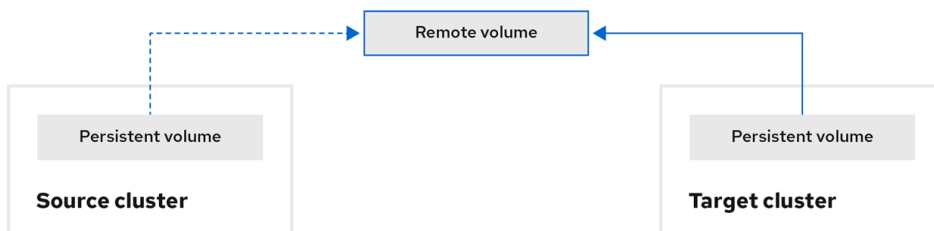
OpenShift_45_1019

- **Move**：CAM 工具从源集群中卸载一个远程卷（例如 NFS），在目标集群上创建一个指向这个远程卷的 PV 资源，然后在目标集群中挂载远程卷。在目标集群中运行的应用程序使用源集群使用的同一远程卷。远程卷必须可以被源集群和目标集群访问。



注意

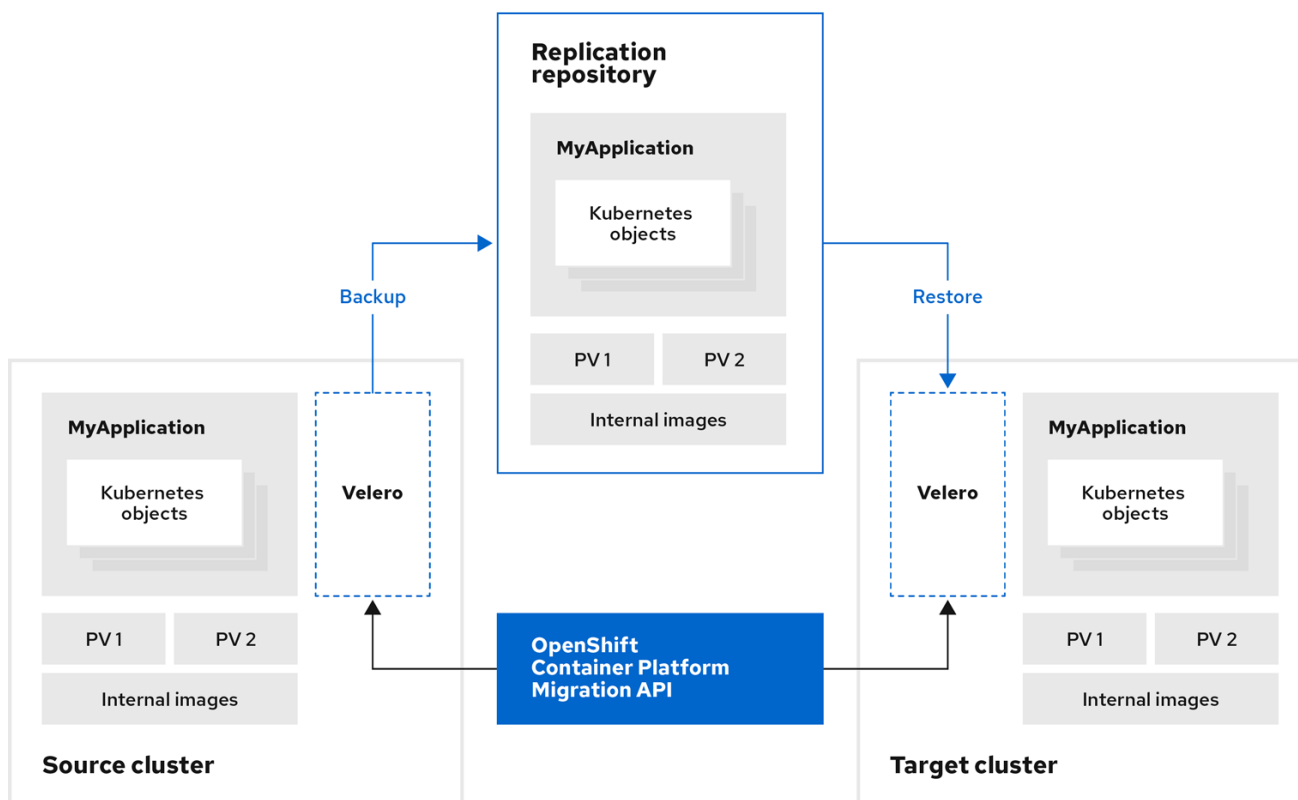
虽然复制存储库没有出现在此图表中，但实际迁移过程需要它。



OpenShift_45_1019

6. 运行迁移计划，使用以下选项之一：
 - **Stage**（可选）在不停止应用程序的情况下将数据复制到目标集群。Stage 可以多次运行，以便在迁移前将大多数数据复制到目标。这样可最小化实际迁移时间和应用程序停机时间。

- **Migrate** 在源集群中停止应用程序，并在目标集群中重新创建其资源。您可以选择在不停止应用程序的情况下迁移工作负载。



OpenShift_45_1019

2.2. 配置复制存储库

您必须将对象存储配置为用作复制存储库。集群应用程序迁移工具将数据从源集群复制到复制存储库，然后使用[文件系统](#)或者[快照](#)数据复制方法从复制存储库复制到目标集群。

支持以下存储供应商：

- 通用 S3 对象存储，例如 Minio 或 Ceph S3
- 多云对象网关 (MCG)
- Amazon Web Services (AWS) S3
- Google Cloud Provider (GCP)
- Microsoft Azure

2.2.1. 了解用于迁移的数据复制方法

CAM 工具支持[文件系统](#)和[快照](#)数据复制方法，用于将数据从源集群迁移到目标集群。您可以选择适合于您的环境并受您的存储供应商支持的方法。

2.2.1.1. 文件系统复制方法

CAM 工具将数据文件从源集群复制到复制存储库，并从那里复制到目标集群。

表 2.1. 文件系统复制方法概述

优点	限制：
<ul style="list-style-type: none"> ● 集群可以有不同的存储类 ● 所有 S3 存储供应商均支持 	<ul style="list-style-type: none"> ● 比快照复制方法慢

2.2.1.2. 快照复制方法

CAM 工具将源集群的数据快照复制到云供应商的对象存储，后者配置为复制存储库。数据在目标集群上恢复。

AWS、Google Cloud Provider 和 Microsoft Azure 支持快照复制方法。

表 2.2. 快照复制方法概述

优点	限制：
<ul style="list-style-type: none"> ● 比文件系统复制方法快 	<ul style="list-style-type: none"> ● 云供应商必须支持快照。 ● 集群必须位于相同的云供应商。 ● 集群必须位于同一位置或区域。 ● 集群必须具有相同的存储类。 ● 存储类必须与快照兼容。



重要

将多云对象网关 (MCG) 配置为迁移的复制存储库只是技术预览功能。红帽产品服务等级协议 (SLA) 不支持技术预览功能，并且这些功能可能并不完善。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的详情，请参阅 <https://access.redhat.com/support/offerings/techpreview/>。

2.2.2. 配置 MCG 存储桶做为复制存储库

您可以安装 OpenShift Container Storage Operator，并将一个 Multi-Cloud Object Gateway (MCG) 存储桶配置为复制存储库。

2.2.2.1. 安装 OpenShift Container Storage Operator

您可以从 OperatorHub 安装 OpenShift Container Storage Operator。

流程

1. 在 OpenShift Container Platform web 控制台中，点 **Administration** → **Namespaces**。

2. 点 **Create Namespace**。
3. 在 **Name** 字段中输入 **openshift-storage**，点 **Create**。
4. 点 **Operators** → **OperatorHub**。
5. 使用 **Filter by keyword**（本例中为 **OCS**）来查找 **OpenShift Container Storage Operator**。
6. 选择 **OpenShift Container Storage Operator** 并点 **Install**。
7. 在 **Create Operator Subscription** 页面中，选择 **openshift-storage** 命名空间。
8. 指定您的更新频道和批准策略。
9. 点 **Subscribe**。
在 **Installed Operators** 页面中，**OpenShift Container Storage Operator** 会出现在 **openshift-storage** 项目中，状态为 **Succeeded**。

2.2.2.2. 创建 Multi-Cloud Object Gateway 存储桶

您可以创建 Multi-Cloud Object Gateway (MCG) 存储桶的自定义资源 (CR)。

流程

1. 登录到 OpenShift Container Platform 集群：

```
$ oc login
```

2. 使用以下内容创建 **NooBaa** CR 配置文件，**noobaa.yml**：

```
apiVersion: noobaa.io/v1alpha1
kind: NooBaa
metadata:
  name: noobaa
  namespace: openshift-storage
spec:
  dbResources:
    requests:
      cpu: 0.5 1
      memory: 1Gi
  coreResources:
    requests:
      cpu: 0.5 2
      memory: 1Gi
```

1 **2** 对于非常小的集群，您可以将 **cpu** 的值改为 **0.1**。

3. 创建 **NooBaa** 对象：

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

4. 使用以下内容创建 **BackingStore** CR 配置文件，**bs.yml**：

```
apiVersion: noobaa.io/v1alpha1
```

```

kind: BackingStore
metadata:
  finalizers:
  - noobaa.io/finalizer
  labels:
    app: noobaa
  name: mcg-pv-pool-bs
  namespace: openshift-storage
spec:
  pvPool:
    numVolumes: 3 ❶
    resources:
      requests:
        storage: 50Gi ❷
    storageClass: gp2 ❸
  type: pv-pool

```

- ❶ 指定 PV 池中的卷数量。
- ❷ 指定卷的大小。
- ❸ 指定存储类。

5. 创建 **BackingStore** 对象：

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

6. 使用以下内容创建 **BucketClass** CR 配置文件，**bc.yml**：

```

apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: noobaa
  name: mcg-pv-pool-bc
  namespace: openshift-storage
spec:
  placementPolicy:
    tiers:
    - backingStores:
      - mcg-pv-pool-bs
    placement: Spread

```

7. 创建 **BucketClass** 对象：

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

8. 使用以下内容创建 **ObjectBucketClaim** CR 配置文件，**obc.yml**：

```

apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: migstorage

```

```
namespace: openshift-storage
spec:
  bucketName: migstorage ❶
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: mcg-pv-pool-bc
```

- ❶ 记录下在 CAM web 控制台中添加为复制存储库的存储桶的名称。

9. 创建 **ObjectBucketClaim** 对象：

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

10. 监控资源创建过程以验证 **ObjectBucketClaim** 的状态变为 **Bound**：

```
$ watch -n 30 'oc get -n openshift-storage objectbucketclaim migstorage -o yaml'
```

这个过程可能需要五到十分钟。

11. 获取并记录以下值，当您复制存储库添加到 CAM web 控制台时需要这些值：

- S3 端点：

```
$ oc get route -n openshift-storage s3
```

- S3 provider access key:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{
.data.AWS_ACCESS_KEY_ID }}' | base64 -d
```

- S3 provider secret access key:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{
.data.AWS_SECRET_ACCESS_KEY }}' | base64 -d
```

2.2.3. 将 AWS S3 存储桶配置为复制存储库

您可以将 AWS S3 存储桶配置为复制存储库。

先决条件

- AWS S3 存储桶必须可以被源和目标集群访问。
- 您必须安装了 [AWS CLI](#)。
- 如果您使用快照复制方法：
 - 您必须有权访问 EC2 Elastic Block Storage (EBS)。
 - 源和目标集群必须位于同一区域。
 - 源和目标集群必须具有相同的存储类。

- 存储类必须与快照兼容。

流程

1. 创建 AWS S3 存储桶：

```
$ aws s3api create-bucket \  
  --bucket <bucket_name> \ 1  
  --region <bucket_region> 2
```

- 1** 指定 S3 存储桶名称。
- 2** 指定 S3 存储桶区域，例如 **us-east-1**。

2. 创建 IAM 用户 **velero**：

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

3. 创建 EC2 EBS 快照策略：

```
$ cat > velero-ec2-snapshot-policy.json <<EOF  
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "ec2:DescribeVolumes",  
        "ec2:DescribeSnapshots",  
        "ec2:CreateTags",  
        "ec2:CreateVolume",  
        "ec2:CreateSnapshot",  
        "ec2>DeleteSnapshot"  
      ],  
      "Resource": "*"   
    }  
  ]  
}  
EOF
```

4. 为一个或所有 S3 存储桶创建 AWS S3 访问策略：

```
$ cat > velero-s3-policy.json <<EOF  
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetObject",  
        "s3>DeleteObject",  
        "s3:PutObject",  
        "s3:AbortMultipartUpload",  
        "s3:ListMultipartUploadParts"  
      ]  
    }  
  ]  
}
```

```

    ],
    "Resource": [
      "arn:aws:s3:::<bucket_name>/*" ❶
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket",
      "s3:GetBucketLocation",
      "s3:ListBucketMultipartUploads"
    ],
    "Resource": [
      "arn:aws:s3:::<bucket_name>" ❷
    ]
  }
]
}
EOF

```

- ❶ ❷ 要授予对单一 S3 存储桶的访问权限，请指定存储桶名称。要授予对所有 AWS S3 存储桶的访问权限，请指定 * 而不是存储桶名称：

```

"Resource": [
  "arn:aws:s3:::*"
]

```

5. 将 EC2 EBS 策略附加到 **velero**：

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-ebs \
  --policy-document file://velero-ec2-snapshot-policy.json

```

6. 将 AWS S3 策略附加到 **velero**：

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-s3 \
  --policy-document file://velero-s3-policy.json

```

7. 为 **velero** 创建访问密钥：

```

$ aws iam create-access-key --user-name velero
{
  "AccessKey": {
    "UserName": "velero",
    "Status": "Active",
    "CreateDate": "2017-07-31T22:24:41.576Z",
    "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>, ❶
    "AccessKeyId": <AWS_ACCESS_KEY_ID> ❷
  }
}

```

- 1 2 记录 **AWS_SECRET_ACCESS_KEY** 和 **AWS_ACCESS_KEY_ID** 以将 AWS 存储库添加到 CAM Web 控制台。

2.2.4. 将 Google Cloud Provider 存储桶配置为复制存储库

您可以将 Google Cloud Provider (GCP) 存储桶配置为复制存储库。

先决条件

- AWS S3 存储桶必须可以被源和目标集群访问。
- 您必须安装了 **gsutil**。
- 如果您使用快照复制方法：
 - 源和目标集群必须位于同一区域。
 - 源和目标集群必须具有相同的存储类。
 - 存储类必须与快照兼容。

流程

1. 运行 **gsutil init** 以登录：

```
$ gsutil init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

To continue, you must login. Would you like to login (Y/n)?
```

2. 设置 **BUCKET** 变量：

```
$ BUCKET=<bucket_name> 1
```

- 1 指定存储桶名称。

3. 创建存储桶：

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

4. 将 **PROJECT_ID** 变量设置为您的活跃项目：

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

5. 创建 **velero** 服务帐户：

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

6. 将 **SERVICE_ACCOUNT_EMAIL** 变量设置为服务帐户的电子邮件地址：

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

7. 向服务帐户授予权限：

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

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

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

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

8. 将服务帐户的密钥保存到当前目录中的 **credentials-velero** 文件中：

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

2.2.5. 将 Microsoft Azure Blob 存储容器配置为复制存储库

您可以将 Microsoft Azure Blob 存储容器配置为复制存储库。

先决条件

- 您必须具有 [Azure 存储帐户](#)。
- 您必须安装了 [Azure CLI](#)。
- Azure Blob 存储容器必须可以被源和目标集群访问。
- 如果您使用快照复制方法：
 - 源和目标集群必须位于同一区域。
 - 源和目标集群必须具有相同的存储类。
 - 存储类必须与快照兼容。

流程

1. 设置 **AZURE_RESOURCE_GROUP** 变量：

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

2. 创建 Azure 资源组：

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

- 1 指定位置。

3. 设置 **AZURE_STORAGE_ACCOUNT_ID** 变量：

```
$ AZURE_STORAGE_ACCOUNT_ID=velerobackups
```

4. 创建 Azure 存储帐户：

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

5. 设置 **BLOB_CONTAINER** 变量：

```
$ BLOB_CONTAINER=velero
```

6. 创建 Azure Blob 存储容器：

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

7. 为 **velero** 创建服务主体和凭证：

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

8. 在 **credentials-velero** 文件中保存服务主体的凭证：

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



```
AZURE_RESOURCE_GROUP=${AZURE_RESOURCE_GROUP}
AZURE_CLOUD_NAME=AzurePublicCloud
EOF
```

2.3. 部署集群应用程序迁移 (CAM) 工具

部署集群应用程序迁移 (CAM) 工具需要在 [OpenShift Container Platform 4.1 源](#) 和 [OpenShift Container Platform 4.2 目标集群](#) 上安装 CAM Operator，并在 [OpenShift Container Platform 4.1 源集群](#) 中配置跨原始资源共享。

Cluster Application Migration Operator 默认在目标集群上安装 CAM 工具（CAM web 控制台和迁移控制器）。您可以在 [OpenShift Container Platform 3](#) 和 [远程集群中安装 CAM 工具](#)。

2.3.1. 在 OpenShift Container Platform 4.1 源集群上安装 Cluster Application Migration Operator

您可以使用 OLM 在 OpenShift Container Platform 4.1 源集群上安装 Cluster Application Migration Operator。

流程

1. 在 OpenShift Container Platform Web 控制台中，点击 **Catalog** → **OperatorHub**。
2. 使用 **Filter by keyword** 项（在这里是 **Migration**）找到 **Cluster Application Migration Operator**。
3. 选择 **Cluster Application Migration Operator** 并点 **Install**。
4. 在 **Create Operator Subscription** 页面中，选择 **openshift-migration** 命名空间，并指定批准策略。
5. 点 **Subscribe**。
在 **Installed Operators** 页中，**Cluster Application Migration Operator** 会出现在 **openshift-migration** 项目中，其状态为 **InstallSucceeded**。
6. 在 **Provided APIs** 中点 **View 12 more...**
7. 点 **Create New** → **MigrationController**。
8. 更新 **migration_controller** 和 **migration_ui** 参数，并在 **spec** 中添加 **deprecated_cors_configuration** 参数：

```
spec:
  [...]
  migration_controller: false
  migration_ui: false
  [...]
  deprecated_cors_configuration: true
```

9. 点击 **Create**。
10. 点 **Workloads** → **Pod** 来验证 Restic 和 Velero Pod 是否正在运行。

2.3.2. 在 OpenShift Container Platform 4.2 目标集群上安装 Cluster Application Migration Operator

您可以使用 OLM 在 OpenShift Container Platform 4.2 目标集群上安装 Cluster Application Migration Operator。

默认情况下，Cluster Application Migration Operator 会在目标集群上安装 CAM 工具：

流程

1. 在 OpenShift Container Platform Web 控制台中，点击 **Operators** → **OperatorHub**。
2. 使用 **Filter by keyword** 项（在这里是 **Migration**）找到 **Cluster Application Migration Operator**。
3. 选择 **Cluster Application Migration Operator** 并点 **Install**。
4. 在 **Create Operator Subscription** 页面中，选择 **openshift-migration** 命名空间，并指定批准策略。
5. 点 **Subscribe**。
在 **Installed Operators** 页中，**Cluster Application Migration Operator** 会出现在 **openshift-migration** 项目中，其状态为 **InstallSucceeded**。
6. 在 **Provided APIs** 中点 **View 12 more...**
7. 点 **Create New** → **MigrationController**。
8. 点击 **Create**。
9. 点 **Workloads** → **Pod** 来验证 Controller Manager、Migration UI、Restic 和 Velero Pod 是否正在运行。

2.3.3. 在 OpenShift Container Platform 4.1 源集群中配置跨原始资源共享

您必须在 OpenShift Container Platform 4.1 源集群中配置跨原始资源共享，以启用源集群 API 服务器和 CAM 工具之间的通信。

流程

1. 登录到已安装 CAM 工具的集群。
2. 获取 CORS 配置的值：

```
$ oc get -n openshift-migration route/migration -o go-template='{(?i)/{{ .spec.host }}(:|z){{ println }}' | sed 's,\.,\\.,g'
```

3. 登录到源集群。
4. 编辑 OAuth 服务器 CR：

```
$ oc edit authentication.operator cluster
```

5. 将 CORS 配置值添加到 **spec** 的 **unsupportedConfigOverrides** 下的 **corsAllowedOrigins** 中：

```
spec:
  unsupportedConfigOverrides:
    corsAllowedOrigins:
      - (?i)//migration-openshift-migration\.apps\.cluster\.com(:|\z) ❶
```

❶ 指定您的 CORS 配置值。

6. 保存文件以使改变生效。

7. 编辑 Kubernetes API 服务器 CR :

```
$ oc edit kubeapiserver.operator cluster
```

8. 将 CORS 配置值添加到 **spec** 的 **unsupportedConfigOverrides** 下的 **corsAllowedOrigins** 中 :

```
spec:
  unsupportedConfigOverrides:
    corsAllowedOrigins:
      - (?i)//migration-openshift-migration\.apps\.cluster\.com(:|\z) ❶
```

❶ 指定您的 CORS 配置值。

9. 保存文件以使改变生效。

10. 验证配置 :

```
$ curl -v -k -X OPTIONS \
"<cluster_url>/apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migclusters" \ ❶
-H "Access-Control-Request-Method: GET" \
-H "Access-Control-Request-Headers: authorization" \
-H "Origin: https://<CAM_web_console_url>" ❷
```

❶ 指定您配置的 CORS 集群的 URL。

❷ 指定 CAM web 控制台的 URL。URL 基于 CORS 配置值，例如 **https://migration-openshift-migration.apps.cluster**。

输出结果类似如下 :

```
< HTTP/2 204
< access-control-allow-credentials: true
< access-control-allow-headers: Content-Type, Content-Length, Accept-Encoding, X-CSRF-
Token, Authorization, X-Requested-With, If-Modified-Since
< access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, PATCH
< access-control-allow-origin: https://migration-openshift-migration.apps.cluster
< access-control-expose-headers: Date
< cache-control: no-store
```

2.4. 使用 CAM WEB 控制台迁移应用程序


```
RG2J2rkqmPm6vr7K-
cm7ibD1IBpdQJCcVDuoHYsFgV4mp9vgOfn9osSDp2TGikwNz4Az95e81xnjVUmzh-
NjDsEpw71DH92iHV_xt2sTwtzftS49LpPW2LjrV0evtNBP_t_RfskdArt5VSv25eORI7zScqfe1CiM
kcVbf2UqACQjo3LbkpfN26HAioO2oH0ECPiRzT0Xyh-KwFutJLS9Xgghyw-
LD9kPKcE_xbbJ9Y4Rqajh7WdPYuB0Jd9DPVrslmzK-F6cgHHYoZEv0SvLQi-
PO0rpDrcjOEEQQ
```

3. 登录到 CAM web 控制台。
4. 在 **Clusters** 部分，点 **Add cluster**。
5. 填写以下字段：
 - **Cluster name**：可包括小写字母 (**a-z**) 和数字 (**0-9**)。不能包含空格或国际字符。
 - **URL**：集群 API 服务器的 URL，如 **https://<master1.example.com>:8443**。
 - **Service account token**：从源集群获取的字符串。
 - **Azure cluster**：可选。如果要使用 Azure 快照复制数据，请选择此项。
 - **Azure resource group**：如果选中了 **Azure cluster**，则会出现此字段。
6. 点 **Add cluster**。
集群会出现在 **Clusters** 部分。

2.4.3. 在 CAM web 控制台中添加复制程序库

您可以将对象存储桶作为复制存储库添加到 CAM web 控制台。

先决条件

- 您必须配置用于迁移数据的对象存储桶。

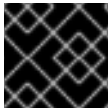
流程

1. 登录到 CAM web 控制台。
2. 在 **Replication repositories** 部分，点 **Add repository**。
3. 选择 **Storage provider type** 并填写以下字段：
 - **AWS** 适用于 S3、MCSG 和通用 S3 供应商：
 - **Replication repository name**：指定 CAM web 控制台中的复制存储库。
 - **S3 bucket name**：指定您创建的 S3 存储桶的名称。
 - **S3 bucket region**：指定 S3 存储桶区域。AWS S3 **必填**。Optional 用于其他 S3 供应商。
 - **S3 端点**：指定 S3 服务的 URL，而不是存储桶，例如：**https://<s3-storage.apps.cluster.com>**。通用 S3 供应商 **必填**。您必须使用 **https://** 前缀。
 - **S3 provider access key**：为 AWS 指定 **<AWS_SECRET_ACCESS_KEY>**，或者为 MCG 指定 S3 供应商访问密钥。

- **S3 provider secret access key**: 为 AWS 指定 `<AWS_ACCESS_KEY_ID>`，或者为 MCG 指定 S3 供应商 secret 访问密钥。
 - **Require SSL verification**：如果您使用的是通用 S3 供应商，则清除此复选框。
 - **GCP**：
 - **Replication repository name**：指定 CAM web 控制台中的复制存储库。
 - **GCP bucket name**：指定 GCP 存储桶的名称。
 - **GCP credential JSON blob**：在 `credentials-velero` 文件中指定字符串。
 - **Azure**：
 - **Replication repository name**：指定 CAM web 控制台中的复制存储库。
 - **Azure resource group**：指定 Azure Blob 存储的资源组。
 - **Azure storage account name**：指定 Azure Blob 存储帐户名称
 - **Azure credentials - INI file contents**: 在 `credentials-velero` 文件中指定字符串。
4. 点 **Add repository** 并等待连接验证。
 5. 点 **Close**。
新存储库会出现在 **Replication repositories** 部分。

2.4.4. 为大型迁移修改迁移计划限制

您可以更改大型迁移的迁移计划限制。



重要

您需要首先在自己的环境对所做的更改进行测试，以避免迁移失败。

单个迁移计划有以下默认限制：

- 10 个命名空间
如果超过这个限制，CAM web 控制台会显示一个 **Namespace limit exceeded** 错误，您将无法创建迁移计划。
- 100 个 Pod
如果超过 Pod 限制，CAM web 控制台会显示类似以下示例的警告信息: **Plan has been validated with warning condition(s).查看警告信息。 pod limit: 100 exceeded, found: 104.**
- 100 个持久性卷 (PV)
如果超过持久性卷限制，则 CAM web 控制台会显示类似的警告信息。

流程

1. 编辑迁移控制器 CR：

```
$ oc get migrationcontroller -n openshift-migration
NAME AGE
migration-controller 5d19h
```

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

2. 更新以下参数：

```
[...]
migration_controller: true

# This configuration is loaded into mig-controller, and should be set on the
# cluster where `migration_controller: true`
mig_pv_limit: 100
mig_pod_limit: 100
mig_namespace_limit: 10
[...]
```

2.4.5. 在 CAM web 控制台中创建迁移计划

您可以在 CAM web 控制台中创建迁移计划。

先决条件

- CAM web 控制台必须包含以下内容：
 - 源集群
 - 目标集群，它会在 CAM 工具安装过程中自动添加
 - 复制软件仓库
- 如果要使用快照复制数据，则源和目标集群必须在同一云供应商（AWS、GCP 或 Azure）以及同一区域运行。

流程

1. 登录到 CAM web 控制台。
2. 在 **Plans** 部分，点 **Add Plan**。
3. 输入 **Plan name** 并点 **Next**。
Plan name 最多可包含 253 个小写字母数字字符（**a-z**，**0-9**）。它不能包含空格或下划线（**_**）。
4. 选一个 **Source cluster**。
5. 选一个 **Target cluster**。
6. 选一个 **Replication repository**。
7. 选择要迁移的项目并点 **Next**。
8. 选择 **Copy** 或 **Move PV**：
 - **Copy** 将源集群的 PV 中的数据复制到复制存储库中，然后在目标集群中新建的具有类似特征的 PV 上恢复它。

... 从源集群上复制整个程序卷（例如，PVC）... 在目标集群上创建... 指定这个程序卷转

- **Move** 从源集群中卸载一个远程卷（例如 NFS），在目标集群上创建一个指向这个远程卷的 PV 资源，然后在目标集群中挂载远程卷。在目标集群中运行的应用程序使用源集群使用的同一远程卷。远程卷必须可以被源集群和目标集群访问。

9. 点 **Next**。

10. 为 PV 选择 **Copy method**：

- **Snapshot** 使用云供应商的快照功能备份和恢复磁盘。它比 **Filesystem** 快得多。



注意

存储和集群必须位于同一区域，存储类必须兼容。

- **Filesystem** 将源磁盘中的数据文件复制到新创建的目标磁盘。

11. 为 PV 选择一个 **Storage class**。

如果选择了 **Filesystem** 复制方法，您可以在迁移过程中更改存储类，例如：从 Red Hat Gluster Storage 或 NFS 存储改为 Red Hat Ceph Storage。

12. 点 **Finish**。

13. 点 **Close**。

迁移计划会出现在 **Plans** 部分。

2.4.6. 在 CAM web 控制台中运行迁移计划

您可以使用在 CAM web 控制台中创建的迁移计划来 stage 或迁移应用程序和数据。

先决条件

CAM web 控制台必须包含以下内容：

- 源集群
- 目标集群，它会在 CAM 工具安装过程中自动添加
- 复制软件仓库
- 有效的迁移计划

流程

1. 登录到 OpenShift Container Platform 4 集群上的 CAM web 控制台。
2. 选择迁移计划。
3. 点 **Stage** 以在不停止应用程序的情况下，将数据从源集群复制到目标集群。
您可以多次运行 **Stage** 以减少实际迁移时间。
4. 当准备好迁移应用程序工作负载时，点 **Migrate**。
Migrate 在源集群中停止应用程序工作负载，并在目标集群中重新创建其资源。
5. 另外，还可以在 **Migrate** 窗口中选择 **Do not stop applications on the source cluster during migration**。

6. 点 Migrate。

7. 迁移完成后，在 OpenShift Container Platform 4.2 web 控制台中确认已成功迁移了应用程序：

- a. 点 Home → Projects。
- b. 点迁移的项目查看其状态。
- c. 在 Routes 部分，点击 Location 验证应用程序是否正常运行。
- d. 点 Workloads → Pods 来验证 Pod 在迁移的命名空间中运行。
- e. 点 Storage → Persistent volumes 确认正确置备了被迁移的持久性卷。

2.5. 故障排除

您可以查看迁移自定义资源 (CR)，并下载日志来排除迁移失败的问题。

如果应用程序在迁移失败时停止，您必须手动回滚，以防止数据崩溃。

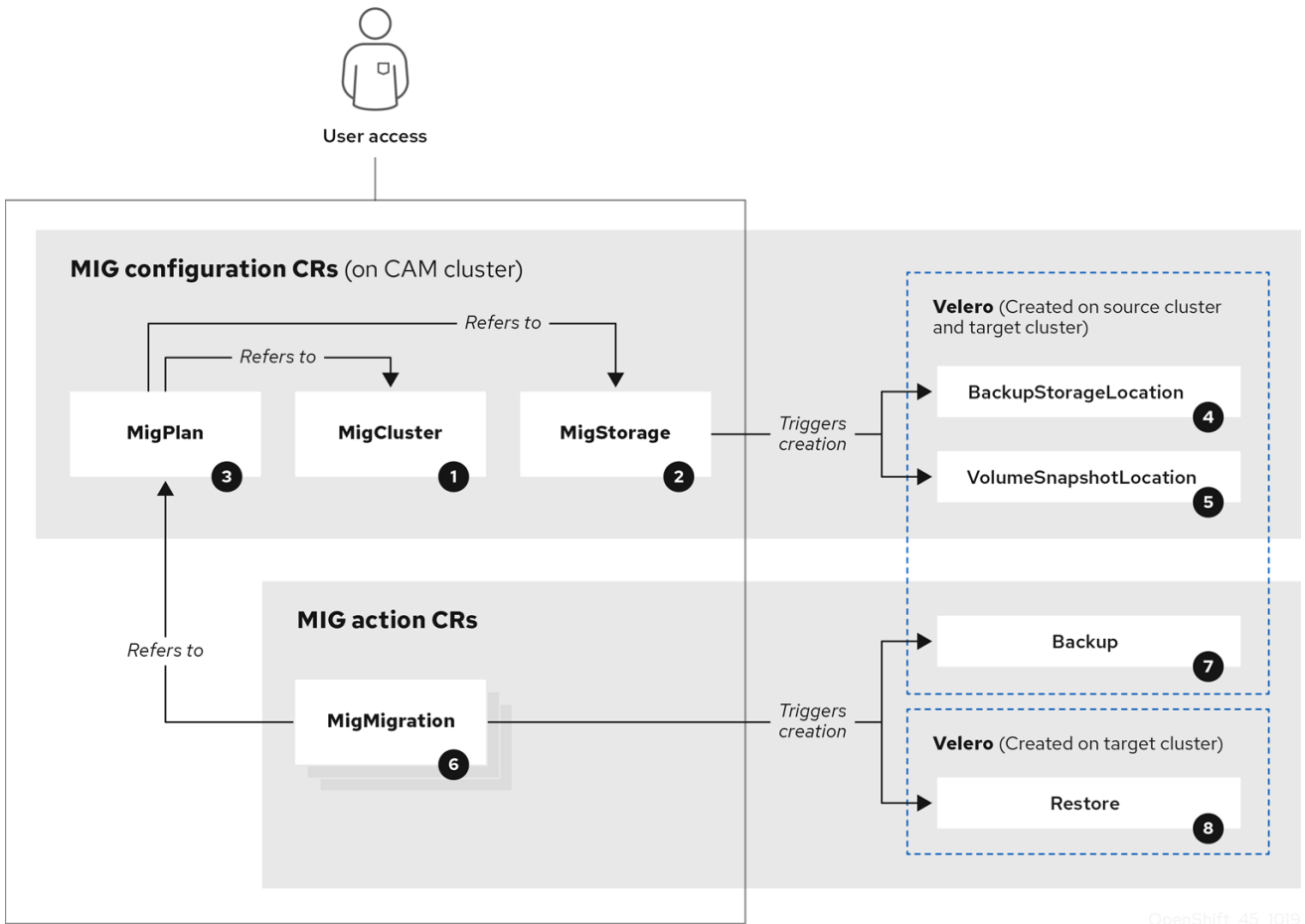


注意

如果应用程序在迁移过程中没有停止，则不需要手动回滚，因为原始应用程序仍然在源集群中运行。

2.5.1. 查看迁移自定义资源 (CR)

集群应用程序迁移 (CAM) 工具会创建以下 CR 用于迁移：



OpenShift_45_1019

- 1 MigCluster (配置, CAM 集群) : 集群定义
- 2 MigStorage (配置, CAM 集群) : 存储定义
- 3 MigPlan (配置, CAM 集群) : 迁移计划

MigPlan CR 描述了要迁移的源和目标集群、存储库和命名空间。它与 0 个、1 个 或多个 MigMigration CR 关联。



注意

删除 MigPlan CR 会删除关联的 MigMigration CR。

- 4 BackupStorageLocation (配置, CAM 集群) : Velero 备份对象的位置
- 5 VolumeSnapshotLocation (配置, CAM 集群) : Velero 卷快照的位置
- 6 MigMigration (操作, CAM 集群) : Migration, 在迁移期间创建

在每次进行 stage 或迁移数据时都会创建一个 MigMigration CR。每个 MigMigration CR 都会与一个 MigPlan CR 关联。

7 Backup (操作, 源集群) : 当运行迁移计划时, MigMigration CR 在每个源集群上创建两个 Velero 备份 CR :

- 备份 CR #1 用于 Kubernetes 对象
- 备份 CR #2 用于 PV 数据

8 Restore (操作, 目标集群) : 在运行迁移计划时, MigMigration CR 在目标集群上创建两个 Velero 恢复 CR :

- 恢复 CR #1 (使用备份 CR #2) 用于 PV 数据
- 恢复 CR #2 (使用备份 CR #1) 用于 Kubernetes 对象

流程

1. 获取 CR 名称 :

```
$ oc get <cr> -n openshift-migration 1
NAME                               AGE
88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s
```

1 指定您要查看的迁移 CR。

2. 查看 CR :

```
$ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
```

输出结果类似以下示例。

MigMigration 示例

```
$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
Name:      88435fe0-c9f8-11e9-85e6-5d593ce65e10
Namespace: openshift-migration
Labels:    <none>
Annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
API Version: migration.openshift.io/v1alpha1
Kind:      MigMigration
Metadata:
  Creation Timestamp: 2019-08-29T01:01:29Z
  Generation:        20
  Resource Version:  88179
  Self Link:         /apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
  UID:               8886de4c-c9f8-11e9-95ad-0205fe66cbb6
Spec:
  Mig Plan Ref:
    Name:      socks-shop-mig-plan
    Namespace: openshift-migration
  Quiesce Pods: true
  Stage:      false
Status:
```

```

Conditions:
  Category:      Advisory
  Durable:       true
  Last Transition Time: 2019-08-29T01:03:40Z
  Message:       The migration has completed successfully.
  Reason:        Completed
  Status:        True
  Type:          Succeeded
  Phase:         Completed
  Start Timestamp: 2019-08-29T01:01:29Z
  Events:        <none>

```

Velero 备份 CR #2 示例 (PV 数据)

```

apiVersion: velero.io/v1
kind: Backup
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.105.179:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-0205fe66cbb6
  creationTimestamp: "2019-08-29T01:03:15Z"
  generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
  generation: 1
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    velero.io/storage-location: myrepo-vpzq9
  name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  namespace: openshift-migration
  resourceVersion: "87313"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:
  excludedNamespaces: []
  excludedResources: []
  hooks:
    resources: []
  includeClusterResources: null
  includedNamespaces:
  - sock-shop
  includedResources:
  - persistentvolumes
  - persistentvolumeclaims
  - namespaces
  - imagestreams
  - imagestreamtags
  - secrets
  - configmaps
  - pods
  labelSelector:
    matchLabels:

```

```

migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
storageLocation: myrepo-vpzq9
ttl: 720h0m0s
volumeSnapshotLocations:
- myrepo-wv6fx
status:
completionTimestamp: "2019-08-29T01:02:36Z"
errors: 0
expiration: "2019-09-28T01:02:35Z"
phase: Completed
startTimestamp: "2019-08-29T01:02:35Z"
validationErrors: null
version: 1
volumeSnapshotsAttempted: 0
volumeSnapshotsCompleted: 0
warnings: 0

```

Velero 恢复 CR #2 示例 (Kubernetes 资源)

```

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
  generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
    migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
  name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  namespace: openshift-migration
  resourceVersion: "82329"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
  - restores.velero.io
  - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
  namespaceMapping: null
  restorePVs: true
status:


```

```
errors: 0
failureReason: ""
phase: Completed
validationErrors: null
warnings: 15
```

2.5.2. 下载迁移日志

您可以在 CAM web 控制台中下载 Velero、Restic 和 Migration controller 日志，以排除出现故障的迁移问题。

流程

1. 登录到 CAM 控制台。
2. 点击 **Plans** 查看迁移计划列表。
3. 点一个迁移计划的 **Options** 菜单  并选择 **Logs**。
4. 点 **Download Logs** 为所有集群下载迁移控制器、Velero 和 Restic 的日志。
5. 要下载特定的日志：

a. 指定日志选项：

- **Cluster**：选择源、目标或 CAM 主机集群。
- **Log source**：选择 **Velero**、**Restic** 或 **Controller**。
- **Pod source**：选择 Pod 名称，例如：**controller-manager-78c469849c-v6wcf**
此时会显示所选日志。

您可以通过更改您的选择来清除日志选择设置。

b. 点 **Download Selected** 下载所选日志。

另外，您可以使用 CLI 访问日志，如下例所示：

```
$ oc get pods -n openshift-migration | grep controller
controller-manager-78c469849c-v6wcf      1/1   Running   0      4h49m
$ oc logs controller-manager-78c469849c-v6wcf -f -n openshift-migration
```

2.5.3. Restic 超时错误

如果因为 Restic 超时造成迁移失败，以下出错信息会出现在 Velero 日志中：

```
level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all
PodVolumeBackups to complete"
error.file="/go/src/github.com/heptio/velero/pkg/restic/backupper.go:165"
error.function="github.com/heptio/velero/pkg/restic.(*backupper).BackupPodVolumes" group=v1
```

`restic_timeout` 的默认值为一小时。您可以为大型迁移增加这个值，请注意，高的值可能会延迟返回出错信息。

流程

1. 在 OpenShift Container Platform web 控制台中导航至 **Operators** → **Installed Operators**。
2. 点 **Cluster Application Migration Operator**。
3. 在 **MigrationController** 标签页中点 **migration-controller**。
4. 在 **YAML** 标签页中，更新以下参数值：

```
spec:
  restic_timeout: 1h 1
```

1 有效单元是 **h**（小时）、**m**（分钟）和 **s**（秒），例如 **3h30m15s**。

5. 点 **Save**。

2.5.4. 手动回滚迁移

如果您的应用程序在迁移失败时停止，您必须手动回滚，以防止 PV 中的数据被破坏。

如果应用程序在迁移过程中没有停止，则不需要进行手动回滚，因为原始应用程序仍然在源集群中运行。

流程

1. 在目标集群中，切换到迁移的项目：

```
$ oc project <project>
```

2. 获取部署的资源：

```
$ oc get all
```

3. 删除部署的资源以确保应用程序没有在目标集群中运行，并访问 PVC 上的数据：

```
$ oc delete <resource_type>
```

4. 要停止 DaemonSet 而不删除它，在 YAML 文件中更新 **nodeSelector**：

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: hello-daemonset
spec:
  selector:
    matchLabels:
      name: hello-daemonset
  template:
    metadata:
      labels:
```

```

name: hello-daemonset
spec:
  nodeSelector:
    role: worker ❶

```

- ❶ 指定一个没有存在于任何节点上的 **nodeSelector** 值。

5. 更新每个 PV 的重新声明策略，以便删除不必要的数据库。在迁移过程中，绑定 PV 的重新声明策略是 **reclaim**，以确保应用程序从源集群中被删除时不会丢失数据。您可以在回滚过程中删除这些 PV。

```

apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain ❶
  ...
status:
  ...

```

- ❶ 指定 **Recycle** 或 **Delete**。

6. 在源集群中，切换到迁移的项目并获取其部署的资源：

```

$ oc project <project>
$ oc get all

```

7. 启动每个部署资源的一个或多个副本：

```

$ oc scale --replicas=1 <resource_type>/<resource_name>

```

8. 如果在操作中被更改了，把 DaemonSet 的 **nodeSelector** 改回其原始值。

2.5.5. 为客户支持问题单收集数据

如果创建一个客户支持问题单，您可以使用 **openshift-migration-must-gather-rhel8** 镜像的 **must-gather** 工具来收集与您的集群相关的信息，并把这些信息上传到[红帽客户门户网站](#)。

openshift-migration-must-gather-rhel8 镜像会收集默认的 **must-gather** 镜像不收集的日志和 CR 数据。

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/rhcam-1-2/openshift-migration-must-gather-rhel8
```

must-gather 工具程序收集集群数据，并把它保存在 **must-gather.local.<uid>** 目录中。

3. 从 **must-gather** 数据中删除验证密钥和其他敏感信息。
4. 创建一个包含 **must-gather.local.<uid>** 目录内容的归档文件：

```
$ tar cvaf must-gather.tar.gz must-gather.local.<uid>/
```

在[红帽客户门户](#)中，为您的问题单附上这个压缩文件。

2.5.6. 已知问题

这个版本有以下已知问题：

- 在迁移过程中，CAM 工具会保留以下命名空间注解：
 - **openshift.io/sa.scc.mcs**
 - **openshift.io/sa.scc.supplemental-groups**
 - **openshift.io/sa.scc.uid-range**
 这些注解会保留 UID 范围，确保容器在目标集群中保留其文件系统权限。这可能会存在一定的风险。因为迁移的 UID 可能已存在于目标集群的现有或将来的命名空间中。
[\(BZ#1748440\)](#)
- 当在 CAM web 控制台中添加 S3 端点时，只有 AWS 支持 **https://**。对于其他 S3 供应商，请使用 **http://**。
- 如果一个 AWS 存储桶被添加到 CAM web 控制台，然后将其删除，则其状态会保持为 **True**，这是因为 MigStorage CR 没有被更新。[\(BZ#1738564\)](#)
- 如果迁移控制器在目标集群以外的集群中运行，迁移将失败。**EnsureCloudSecretPropagated** 阶段会跳过，并给出一个日志警告。[\(BZ#1757571\)](#)
- 目前，集群范围的资源，包括集群角色绑定和安全上下文约束，还没有由 CAM 处理。如果应用程序需要集群范围的资源，则必须在目标集群中手动创建它们。[\(BZ#1759804\)](#)
- 创建迁移计划时会显示不正确的源集群存储类。[\(BZ#1777869\)](#)
- 如果 CAM web 控制台中的一个集群变得无法访问，它会阻止尝试关闭打开的迁移计划。[\(BZ#1758269\)](#)
- 如果迁移失败，则迁移计划不会为静默的 pod 保留自定义 PV 设置。您必须手动回滚，删除迁移计划，并使用 PV 设置创建新的迁移计划。[\(BZ#1784899\)](#)

第 3 章 在 OPENSIFT CONTAINER PLATFORM 4.2 集群间迁移

3.1. 将应用程序工作负载在 OPENSIFT CONTAINER PLATFORM 4.2 集群间迁移

您可以使用集群应用程序迁移 (CAM) 工具将应用程序工作负载在 OpenShift Container Platform 4.2 集群间进行迁移。使用 CAM 工具，您可以控制迁移并最小化应用程序的停机时间。

CAM 工具的 web 控制台和 API，基于 Kubernetes 自定义资源，您可以按照命名空间迁移有状态及无状态的应用程序工作负载。

3.1.1. 迁移先决条件

- 需要在所有集群中都有 **cluster-admin** 权限。
- 源和目标集群必须有对复制存储库的不受限制的网络访问权限。
- 安装 Migration controller 的集群必须具有对其他集群的不受限制的访问权限。

3.1.2. 了解集群应用程序迁移工具

集群应用程序迁移 (CAM) 工具可让您使用 CAM web 控制台或 Kubernetes API 将 OpenShift Container Platform 源集群中的 Kubernetes 资源、持久性卷数据和内部容器镜像迁移到 OpenShift Container Platform 4.2 目标集群。

使用 CAM web 控制台迁移应用程序涉及以下步骤：

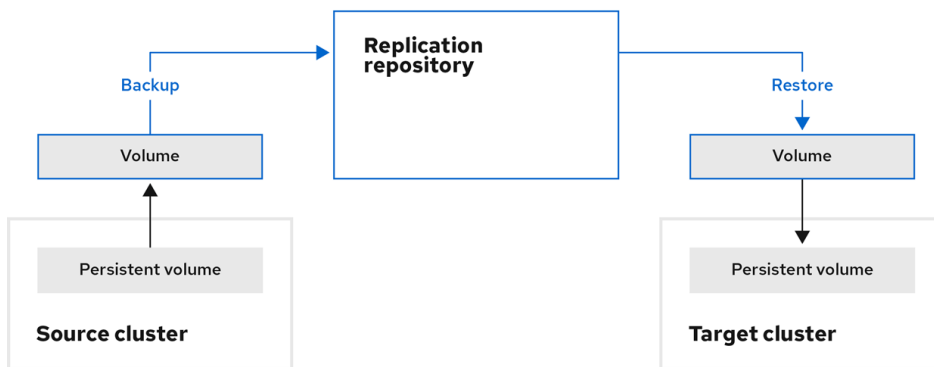
1. 在所有集群中安装 Cluster Application Migration Operator



注意

Cluster Application Migration Operator 默认在目标集群上安装 CAM 工具（CAM web 控制台和迁移控制器）。您可以在 [OpenShift Container Platform 3](#) 和 [远程集群中安装 CAM 工具](#)。

2. 配置复制存储库，这是 CAM 工具用来迁移数据的中间对象存储
3. 在 CAM web 控制台中添加源集群
4. 在 CAM web 控制台中添加复制存储库
5. 创建迁移计划，包含以下数据迁移选项之一：
 - **Copy**：CAM 工具将数据从源集群复制到复制存储库，再从复制存储库把数据复制到目标集群。



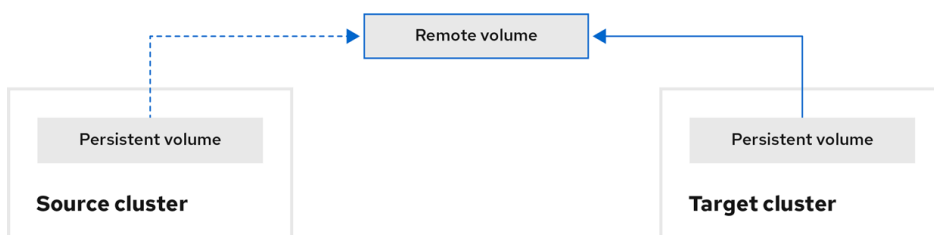
OpenShift_45_1019

- **Move** : CAM 工具从源集群中卸载一个远程卷（例如 NFS），在目标集群上创建一个指向这个远程卷的 PV 资源，然后在目标集群中挂载远程卷。在目标集群中运行的应用程序使用源集群使用的同一远程卷。远程卷必须可以被源集群和目标集群访问。



注意

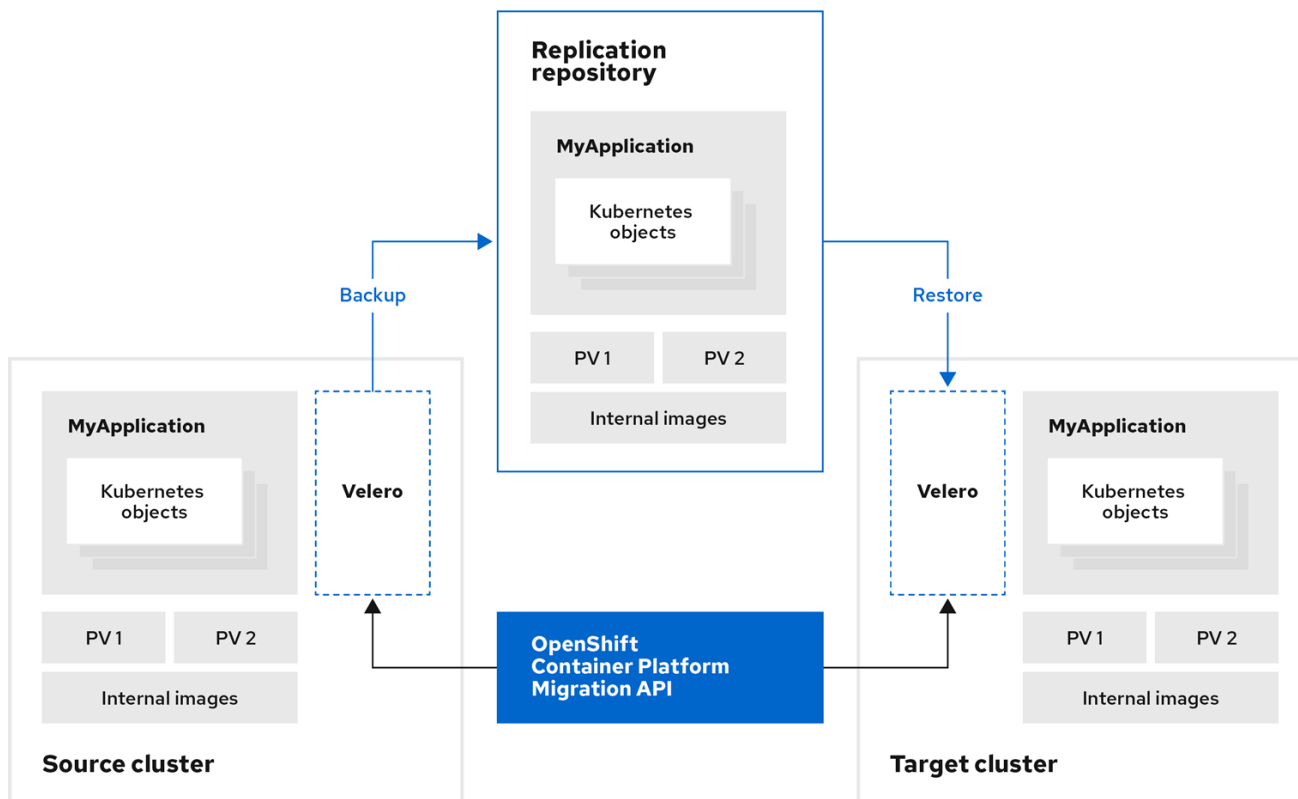
虽然复制存储库没有出现在此图表中，但实际迁移过程需要它。



OpenShift_45_1019

6. 运行迁移计划，使用以下选项之一：

- **Stage**（可选）在不停止应用程序的情况下将数据复制到目标集群。Stage 可以多次运行，以便在迁移前将大多数数据复制到目标。这样可最小化实际迁移时间和应用程序停机时间。
- **Migrate** 在源集群中停止应用程序，并在目标集群中重新创建其资源。您可以选择在不停止应用程序的情况下迁移工作负载。



OpenShift_45_1019

3.2. 配置复制存储库

您必须将对象存储配置为用作复制存储库。集群应用程序迁移工具将数据从源集群复制到复制存储库，然后使用[文件系统](#)或者[快照](#)数据复制方法从复制存储库复制到目标集群。

支持以下存储供应商：

- 通用 S3 对象存储，例如 Minio 或 Ceph S3
- 多云对象网关 (MCG)
- Amazon Web Services (AWS) S3
- Google Cloud Provider (GCP)
- Microsoft Azure

3.2.1. 了解用于迁移的数据复制方法

CAM 工具支持文件系统和快照数据复制方法，用于将数据从源集群迁移到目标集群。您可以选择适合于您的环境并受您的存储供应商支持的方法。

3.2.1.1. 文件系统复制方法

CAM 工具将数据文件从源集群复制到复制存储库，并从那里复制到目标集群。

表 3.1. 文件系统复制方法概述

优点	限制：
<ul style="list-style-type: none"> ● 集群可以有不同的存储类 ● 所有 S3 存储供应商均支持 	<ul style="list-style-type: none"> ● 比快照复制方法慢

3.2.1.2. 快照复制方法

CAM 工具将源集群的数据快照复制到云供应商的对象存储，后者配置为复制存储库。数据在目标集群上恢复。

AWS、Google Cloud Provider 和 Microsoft Azure 支持快照复制方法。

表 3.2. 快照复制方法概述

优点	限制：
<ul style="list-style-type: none"> ● 比文件系统复制方法快 	<ul style="list-style-type: none"> ● 云供应商必须支持快照。 ● 集群必须位于相同的云供应商。 ● 集群必须位于同一位置或区域。 ● 集群必须具有相同的存储类。 ● 存储类必须与快照兼容。



重要

将多云对象网关 (MCG) 配置为迁移的复制存储库只是技术预览功能。红帽产品服务等级协议 (SLA) 不支持技术预览功能，并且这些功能可能并不完善。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的详情，请参阅 <https://access.redhat.com/support/offerings/techpreview/>。

3.2.2. 配置 MCG 存储桶做为复制存储库

您可以安装 OpenShift Container Storage Operator，并将一个 Multi-Cloud Object Gateway (MCG) 存储桶配置为复制存储库。

3.2.2.1. 安装 OpenShift Container Storage Operator

您可以从 OperatorHub 安装 OpenShift Container Storage Operator。

流程

1. 在 OpenShift Container Platform web 控制台中，点 **Administration** → **Namespaces**。

2. 点 **Create Namespace**。
3. 在 **Name** 字段中输入 **openshift-storage**，点 **Create**。
4. 点 **Operators** → **OperatorHub**。
5. 使用 **Filter by keyword**（本例中为 **OCS**）来查找 **OpenShift Container Storage Operator**。
6. 选择 **OpenShift Container Storage Operator** 并点 **Install**。
7. 在 **Create Operator Subscription** 页面中，选择 **openshift-storage** 命名空间。
8. 指定您的更新频道和批准策略。
9. 点 **Subscribe**。
在 **Installed Operators** 页面中，**OpenShift Container Storage Operator** 会出现在 **openshift-storage** 项目中，状态为 **Succeeded**。

3.2.2.2. 创建 Multi-Cloud Object Gateway 存储桶

您可以创建 Multi-Cloud Object Gateway (MCG) 存储桶的自定义资源 (CR)。

流程

1. 登录到 OpenShift Container Platform 集群：

```
$ oc login
```

2. 使用以下内容创建 **NooBaa** CR 配置文件，**noobaa.yml**：

```
apiVersion: noobaa.io/v1alpha1
kind: NooBaa
metadata:
  name: noobaa
  namespace: openshift-storage
spec:
  dbResources:
    requests:
      cpu: 0.5 1
      memory: 1Gi
  coreResources:
    requests:
      cpu: 0.5 2
      memory: 1Gi
```

1 2 对于非常小的集群，您可以将 **cpu** 的值改为 **0.1**。

3. 创建 **NooBaa** 对象：

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

4. 使用以下内容创建 **BackingStore** CR 配置文件，**bs.yml**：

```
apiVersion: noobaa.io/v1alpha1
```

```

kind: BackingStore
metadata:
  finalizers:
  - noobaa.io/finalizer
  labels:
    app: noobaa
  name: mcg-pv-pool-bs
  namespace: openshift-storage
spec:
  pvPool:
    numVolumes: 3 ❶
    resources:
      requests:
        storage: 50Gi ❷
    storageClass: gp2 ❸
  type: pv-pool

```

- ❶ 指定 PV 池中的卷数量。
- ❷ 指定卷的大小。
- ❸ 指定存储类。

5. 创建 **BackingStore** 对象：

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

6. 使用以下内容创建 **BucketClass** CR 配置文件，**bc.yml**：

```

apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: noobaa
  name: mcg-pv-pool-bc
  namespace: openshift-storage
spec:
  placementPolicy:
    tiers:
    - backingStores:
      - mcg-pv-pool-bs
    placement: Spread

```

7. 创建 **BucketClass** 对象：

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

8. 使用以下内容创建 **ObjectBucketClaim** CR 配置文件，**obc.yml**：

```

apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: migstorage

```

```

namespace: openshift-storage
spec:
  bucketName: migstorage ❶
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: mcg-pv-pool-bc

```

- ❶ 记录下在 CAM web 控制台中添加为复制存储库的存储桶的名称。

9. 创建 **ObjectBucketClaim** 对象：

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

10. 监控资源创建过程以验证 **ObjectBucketClaim** 的状态变为 **Bound**：

```
$ watch -n 30 'oc get -n openshift-storage objectbucketclaim migstorage -o yaml'
```

这个过程可能需要五到十分钟。

11. 获取并记录以下值，当您复制存储库添加到 CAM web 控制台时需要这些值：

- S3 端点：

```
$ oc get route -n openshift-storage s3
```

- S3 provider access key:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{
.data.AWS_ACCESS_KEY_ID }}' | base64 -d
```

- S3 provider secret access key:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{
.data.AWS_SECRET_ACCESS_KEY }}' | base64 -d
```

3.2.3. 将 AWS S3 存储桶配置为复制存储库

您可以将 AWS S3 存储桶配置为复制存储库。

先决条件

- AWS S3 存储桶必须可以被源和目标集群访问。
- 您必须安装了 [AWS CLI](#)。
- 如果您使用快照复制方法：
 - 您必须有权访问 EC2 Elastic Block Storage (EBS)。
 - 源和目标集群必须位于同一区域。
 - 源和目标集群必须具有相同的存储类。

- 存储类必须与快照兼容。

流程

1. 创建 AWS S3 存储桶：

```
$ aws s3api create-bucket \
  --bucket <bucket_name> \ 1
  --region <bucket_region> 2
```

- 1 指定 S3 存储桶名称。
- 2 指定 S3 存储桶区域，例如 **us-east-1**。

2. 创建 IAM 用户 **velero**：

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

3. 创建 EC2 EBS 快照策略：

```
$ cat > velero-ec2-snapshot-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:DescribeSnapshots",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:CreateSnapshot",
        "ec2>DeleteSnapshot"
      ],
      "Resource": "*"
    }
  ]
}
EOF
```

4. 为一个或所有 S3 存储桶创建 AWS S3 访问策略：

```
$ cat > velero-s3-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3>DeleteObject",
        "s3:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts"
      ]
    }
  ]
}
```

```

    ],
    "Resource": [
      "arn:aws:s3:::<bucket_name>/*" ❶
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket",
      "s3:GetBucketLocation",
      "s3:ListBucketMultipartUploads"
    ],
    "Resource": [
      "arn:aws:s3:::<bucket_name>" ❷
    ]
  }
]
}
EOF

```

- ❶ ❷ 要授予对单一 S3 存储桶的访问权限，请指定存储桶名称。要授予对所有 AWS S3 存储桶的访问权限，请指定 * 而不是存储桶名称：

```

"Resource": [
  "arn:aws:s3::*"
]

```

5. 将 EC2 EBS 策略附加到 **velero**：

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-ebs \
  --policy-document file://velero-ec2-snapshot-policy.json

```

6. 将 AWS S3 策略附加到 **velero**：

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-s3 \
  --policy-document file://velero-s3-policy.json

```

7. 为 **velero** 创建访问密钥：

```

$ aws iam create-access-key --user-name velero
{
  "AccessKey": {
    "UserName": "velero",
    "Status": "Active",
    "CreateDate": "2017-07-31T22:24:41.576Z",
    "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>, ❶
    "AccessKeyId": <AWS_ACCESS_KEY_ID> ❷
  }
}

```

- 1 2 记录 **AWS_SECRET_ACCESS_KEY** 和 **AWS_ACCESS_KEY_ID** 以将 AWS 存储库添加到 CAM Web 控制台。

3.2.4. 将 Google Cloud Provider 存储桶配置为复制存储库

您可以将 Google Cloud Provider (GCP) 存储桶配置为复制存储库。

先决条件

- AWS S3 存储桶必须可以被源和目标集群访问。
- 您必须安装了 **gsutil**。
- 如果您使用快照复制方法：
 - 源和目标集群必须位于同一区域。
 - 源和目标集群必须具有相同的存储类。
 - 存储类必须与快照兼容。

流程

1. 运行 **gsutil init** 以登录：

```
$ gsutil init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

To continue, you must login. Would you like to login (Y/n)?
```

2. 设置 **BUCKET** 变量：

```
$ BUCKET=<bucket_name> 1
```

- 1 指定存储桶名称。

3. 创建存储桶：

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

4. 将 **PROJECT_ID** 变量设置为您的活跃项目：

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

5. 创建 **velero** 服务帐户：

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

6. 将 **SERVICE_ACCOUNT_EMAIL** 变量设置为服务帐户的电子邮件地址：

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

7. 向服务帐户授予权限：

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

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

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

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

8. 将服务帐户的密钥保存到当前目录中的 **credentials-velero** 文件中：

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

3.2.5. 将 Microsoft Azure Blob 存储容器配置为复制存储库

您可以将 Microsoft Azure Blob 存储容器配置为复制存储库。

先决条件

- 您必须具有 [Azure 存储帐户](#)。
- 您必须安装了 [Azure CLI](#)。
- Azure Blob 存储容器必须可以被源和目标集群访问。
- 如果您使用快照复制方法：
 - 源和目标集群必须位于同一区域。
 - 源和目标集群必须具有相同的存储类。
 - 存储类必须与快照兼容。

流程

1. 设置 **AZURE_RESOURCE_GROUP** 变量 :

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

2. 创建 Azure 资源组 :

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

- 1 指定位置。

3. 设置 **AZURE_STORAGE_ACCOUNT_ID** 变量 :

```
$ AZURE_STORAGE_ACCOUNT_ID=velerobackups
```

4. 创建 Azure 存储帐户 :

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

5. 设置 **BLOB_CONTAINER** 变量 :

```
$ BLOB_CONTAINER=velero
```

6. 创建 Azure Blob 存储容器 :

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

7. 为 **velero** 创建服务主体和凭证 :

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

8. 在 **credentials-velero** 文件中保存服务主体的凭证 :

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

```
AZURE_RESOURCE_GROUP=${AZURE_RESOURCE_GROUP}
AZURE_CLOUD_NAME=AzurePublicCloud
EOF
```

3.3. 部署集群应用程序迁移 (CAM) 工具

部署集群应用程序迁移 (CAM) 工具需要在 [OpenShift Container Platform 4.2 源](#) 和 [OpenShift Container Platform 4.2 目标集群](#) 上安装 CAM Operator。并在 [OpenShift Container Platform 4.2 源集群](#) 中配置跨原始资源共享。

Cluster Application Migration Operator 默认在目标集群上安装 CAM 工具 (CAM web 控制台和迁移控制器)。您可以在 [OpenShift Container Platform 3](#) 和 [远程集群中安装 CAM 工具](#)。

3.3.1. 在 OpenShift Container Platform 4.2 源集群上安装 Cluster Application Migration Operator

您可以使用 OLM 在 OpenShift Container Platform 4.2 源集群上安装 Cluster Application Migration Operator。

流程

1. 在 OpenShift Container Platform Web 控制台中，点击 **Operators → OperatorHub**。
2. 使用 **Filter by keyword** 项 (在这里是 **Migration**) 找到 **Cluster Application Migration Operator**。
3. 选择 **Cluster Application Migration Operator** 并点 **Install**。
4. 在 **Create Operator Subscription** 页面中，选择 **openshift-migration** 命名空间，并指定批准策略。
5. 点 **Subscribe**。
在 **Installed Operators** 页中，**Cluster Application Migration Operator** 会出现在 **openshift-migration** 项目中，其状态为 **InstallSucceeded**。
6. 在 **Provided APIs** 中点 **View 12 more...**
7. 点 **Create New → MigrationController**。
8. 在 **spec** 小节中更新 **migration_controller** 和 **migration_ui** 参数：

```
spec:
  [...]
  migration_controller: false
  migration_ui: false
  [...]
```

9. 点击 **Create**。
10. 点 **Workloads → Pod** 来验证 Restic 和 Velero Pod 是否正在运行。

3.3.2. 在 OpenShift Container Platform 4.2 目标集群上安装 Cluster Application Migration Operator

您可以使用 OLM 在 OpenShift Container Platform 4.2 目标集群上安装 Cluster Application Migration Operator。

默认情况下，Cluster Application Migration Operator 会在目标集群上安装 CAM 工具：

流程

1. 在 OpenShift Container Platform Web 控制台中，点击 **Operators** → **OperatorHub**。
2. 使用 **Filter by keyword** 项（在这里是 **Migration**）找到 **Cluster Application Migration Operator**。
3. 选择 **Cluster Application Migration Operator** 并点 **Install**。
4. 在 **Create Operator Subscription** 页面中，选择 **openshift-migration** 命名空间，并指定批准策略。
5. 点 **Subscribe**。
在 **Installed Operators** 页中，**Cluster Application Migration Operator** 会出现在 **openshift-migration** 项目中，其状态为 **InstallSucceeded**。
6. 在 **Provided APIs** 中点 **View 12 more...**
7. 点 **Create New** → **MigrationController**。
8. 点击 **Create**。
9. 点 **Workloads** → **Pod** 来验证 Controller Manager、Migration UI、Restic 和 Velero Pod 是否正在运行。

3.3.3. 在 OpenShift Container Platform 4.2 源集群中配置跨原始资源共享

您必须在 OpenShift Container Platform 4.2 源集群中配置跨原始资源共享，以启用源集群 API 服务器和 CAM 工具之间的通信。

流程

1. 登录到已安装 CAM 工具的集群。
2. 获取 CORS 配置的值：

```
$ oc get -n openshift-migration route/migration -o go-template='{?i}//{{ .spec.host }}(:|z){{
println }}' | sed 's,\. ,\.,g'
```

3. 登录到源集群。
4. 编辑 Kubernetes API 服务器 CR：

```
$ oc edit apiserver.config.openshift.io cluster
```

5. 将 CORS 配置值添加到 **spec** 小节中的 **additionalCORSAAllowedOrigins**：

```
spec:
  additionalCORSAAllowedOrigins:
    - (?i)//migration-openshift-migration\.apps\.cluster\.com(:|z) 1
```

1 指定您的 CORS 配置值。

6. 保存文件以使改变生效。

7. 验证配置：

```
$ curl -v -k -X OPTIONS \
"<cluster_url>/apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migclusters" \ 1
-H "Access-Control-Request-Method: GET" \
-H "Access-Control-Request-Headers: authorization" \
-H "Origin: https://<CAM_web_console_url>" 2
```

1 指定您配置的 CORS 集群的 URL。

2 指定 CAM web 控制台的 URL。URL 基于 CORS 配置值，例如 **https://migration-openshift-migration.apps.cluster**。

输出结果类似如下：

```
< HTTP/2 204
< access-control-allow-credentials: true
< access-control-allow-headers: Content-Type, Content-Length, Accept-Encoding, X-CSRF-
Token, Authorization, X-Requested-With, If-Modified-Since
< access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, PATCH
< access-control-allow-origin: https://migration-openshift-migration.apps.cluster
< access-control-expose-headers: Date
< cache-control: no-store
```

3.4. 使用 CAM WEB 控制台迁移应用程序

3.4.1. 启动 CAM web 控制台

您可以在浏览器中启动 CAM web 控制台。

流程

1. 登录到已安装 CAM 工具的 OpenShift Container Platform 集群。
2. 运行以下命令来获取 CAM web 控制台 URL:

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

3. 启动浏览器并进入 CAM web 控制台。



注意

如果在安装 CAM Operator 后尝试立即访问 CAM web 控制台，则该控制台可能无法加载，因为 Operator 仍然在配置集群并正在启用跨原始资源共享。等待几分钟后重试。

6. 点 **Add cluster**。
集群会出现在 **Clusters** 部分。

3.4.3. 在 CAM web 控制台中添加复制程序库

您可以将对象存储桶作为复制存储库添加到 CAM web 控制台。

先决条件

- 您必须配置用于迁移数据的对象存储桶。

流程

1. 登录到 CAM web 控制台。
2. 在 **Replication repositories** 部分，点 **Add repository**。
3. 选择 **Storage provider type** 并填写以下字段：
 - **AWS** 适用于 S3、MCSG 和通用 S3 供应商：
 - **Replication repository name**：指定 CAM web 控制台中的复制存储库。
 - **S3 bucket name**：指定您创建的 S3 存储桶的名称。
 - **S3 bucket region**：指定 S3 存储桶区域。AWS S3 **必填**。Optional 用于其他 S3 供应商。
 - **S3 端点**：指定 S3 服务的 URL，而不是存储桶，例如：**https://<s3-storage.apps.cluster.com>**。通用 S3 供应商**必填**。您必须使用 **https://** 前缀。
 - **S3 provider access key**：为 AWS 指定 **<AWS_SECRET_ACCESS_KEY>**，或者为 MCG 指定 S3 供应商访问密钥。
 - **S3 provider secret access key**：为 AWS 指定 **<AWS_ACCESS_KEY_ID>**，或者为 MCG 指定 S3 供应商 secret 访问密钥。
 - **Require SSL verification**：如果您使用的是通用 S3 供应商，则清除此复选框。
 - **GCP**：
 - **Replication repository name**：指定 CAM web 控制台中的复制存储库。
 - **GCP bucket name**：指定 GCP 存储桶的名称。
 - **GCP credential JSON blob**：在 **credentials-velero** 文件中指定字符串。
 - **Azure**：
 - **Replication repository name**：指定 CAM web 控制台中的复制存储库。
 - **Azure resource group**：指定 Azure Blob 存储的资源组。
 - **Azure storage account name**：指定 Azure Blob 存储帐户名称
 - **Azure credentials - INI file contents**：在 **credentials-velero** 文件中指定字符串。

4. 点 **Add repository** 并等待连接验证。
5. 点 **Close**。
新存储库会出现在 **Replication repositories** 部分。

3.4.4. 为大型迁移修改迁移计划限制

您可以更改大型迁移的迁移计划限制。



重要

您需要首先在自己的环境对所做的更改进行测试，以避免迁移失败。

单个迁移计划有以下默认限制：

- 10 个命名空间
如果超过这个限制，CAM web 控制台会显示一个 **Namespace limit exceeded** 错误，您将无法创建迁移计划。
- 100 个 Pod
如果超过 Pod 限制，CAM web 控制台会显示类似以下示例的警告信息: **Plan has been validated with warning condition(s).查看警告信息。 pod limit: 100 exceeded, found: 104.**
- 100 个持久性卷 (PV)
如果超过持久性卷限制，则 CAM web 控制台会显示类似的警告信息。

流程

1. 编辑迁移控制器 CR：

```
$ oc get migrationcontroller -n openshift-migration
NAME AGE
migration-controller 5d19h

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

2. 更新以下参数：

```
[...]
migration_controller: true

# This configuration is loaded into mig-controller, and should be set on the
# cluster where `migration_controller: true`
mig_pv_limit: 100
mig_pod_limit: 100
mig_namespace_limit: 10
[...]
```

3.4.5. 在 CAM web 控制台中创建迁移计划

您可以在 CAM web 控制台中创建迁移计划。

先决条件

- CAM web 控制台必须包含以下内容：
 - 源集群
 - 目标集群，它会在 CAM 工具安装过程中自动添加
 - 复制软件仓库
- 如果要使用快照复制数据，则源和目标集群必须在同一云供应商（AWS、GCP 或 Azure）以及同一区域运行。

流程

1. 登录到 CAM web 控制台。
2. 在 **Plans** 部分，点 **Add Plan**。
3. 输入 **Plan name** 并点 **Next**。
Plan name 最多可包含 253 个小写字母数字字符（**a-z, 0-9**）。它不能包含空格或下划线（**_**）。
4. 选一个 **Source cluster**。
5. 选一个 **Target cluster**。
6. 选一个 **Replication repository**。
7. 选择要迁移的项目并点 **Next**。
8. 选择 **Copy** 或 **Move PV**：
 - **Copy** 将源集群的 PV 中的数据复制到复制存储库中，然后在目标集群中新创建的具有类似特征的 PV 上恢复它。
 - **Move** 从源集群中卸载一个远程卷（例如 NFS），在目标集群上创建一个指向这个远程卷的 PV 资源，然后在目标集群中挂载远程卷。在目标集群中运行的应用程序使用源集群使用的同一远程卷。远程卷必须可以被源集群和目标集群访问。
9. 点 **Next**。
10. 为 PV 选择 **Copy method**：
 - **Snapshot** 使用云供应商的快照功能备份和恢复磁盘。它比 **Filesystem** 快得多。

注意

存储和集群必须位于同一区域，存储类必须兼容。

 - **Filesystem** 将源磁盘中的数据文件复制到新创建的目标磁盘。
11. 为 PV 选择一个 **Storage class**。
如果选择了 **Filesystem** 复制方法，您可以在迁移过程中更改存储类，例如：从 Red Hat Gluster Storage 或 NFS 存储改为 Red Hat Ceph Storage。
12. 点 **Finish**。
13. 点 **Close**。

迁移计划会出现在 **Plans** 部分。

3.4.6. 在 CAM web 控制台中运行迁移计划

您可以使用在 CAM web 控制台中创建的迁移计划来 stage 或迁移应用程序和数据。

先决条件

CAM web 控制台必须包含以下内容：

- 源集群
- 目标集群，它会在 CAM 工具安装过程中自动添加
- 复制软件仓库
- 有效的迁移计划

流程

1. 登录到 OpenShift Container Platform 4 集群上的 CAM web 控制台。
2. 选择迁移计划。
3. 点 **Stage** 以在不停止应用程序的情况下，将数据从源集群复制到目标集群。
您可以多次运行 **Stage** 以减少实际迁移时间。
4. 当准备好迁移应用程序工作负载时，点 **Migrate**。
Migrate 在源集群中停止应用程序工作负载，并在目标集群中重新创建其资源。
5. 另外，还可以在 **Migrate** 窗口中选择 **Do not stop applications on the source cluster during migration**。
6. 点 **Migrate**。
7. 迁移完成后，在 OpenShift Container Platform 4.2 web 控制台中确认已成功迁移了应用程序：
 - a. 点 **Home** → **Projects**。
 - b. 点迁移的项目查看其状态。
 - c. 在 **Routes** 部分，点击 **Location** 验证应用程序是否正常运行。
 - d. 点 **Workloads** → **Pods** 来验证 Pod 在迁移的命名空间中运行。
 - e. 点 **Storage** → **Persistent volumes** 确认正确置备了被迁移的持久性卷。

3.5. 故障排除

您可以查看迁移自定义资源 (CR)，并下载日志来排除迁移失败的问题。

如果应用程序在迁移失败时停止，您必须手动回滚，以防止数据崩溃。

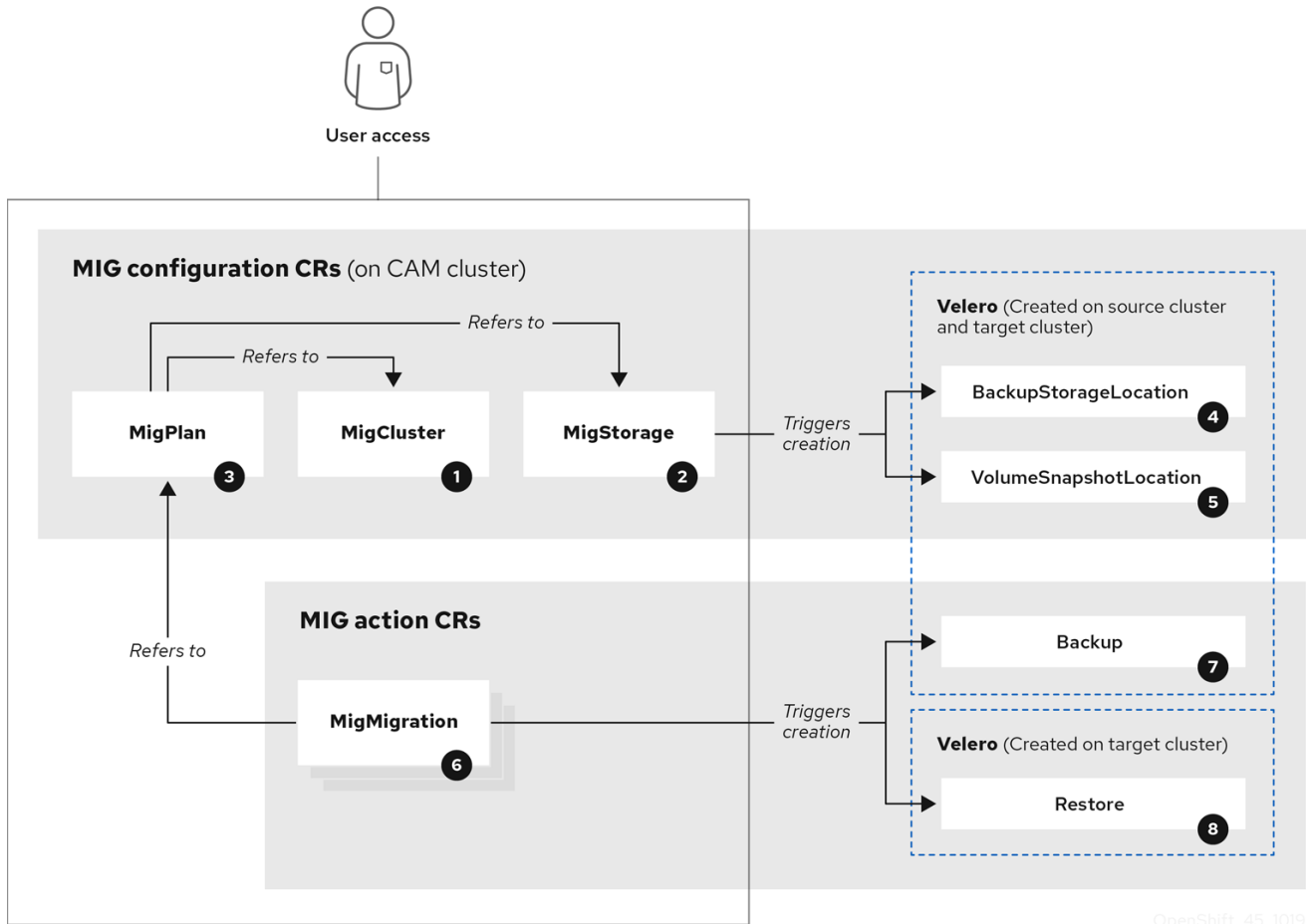


注意

如果应用程序在迁移过程中没有停止，则不需要手动回滚，因为原始应用程序仍然在源集群中运行。

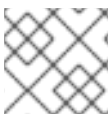
3.5.1. 查看迁移自定义资源 (CR)

集群应用程序迁移 (CAM) 工具会创建以下 CR 用于迁移：



- 1 **MigCluster** (配置, CAM 集群) : 集群定义
- 2 **MigStorage** (配置, CAM 集群) : 存储定义
- 3 **MigPlan** (配置, CAM 集群) : 迁移计划

MigPlan CR 描述了要迁移的源和目标集群、存储库和命名空间。它与 0 个、1 个或多个 MigMigration CR 关联。



注意

删除 MigPlan CR 会删除关联的 MigMigration CR。

- 4 **BackupStorageLocation** (配置, CAM 集群) : Velero 备份对象的位置

5 **VolumeSnapshotLocation** (配置, CAM 集群) : Velero 卷快照的位置

6 **MigMigration** (操作, CAM 集群) : Migration, 在迁移期间创建

在每次进行 stage 或迁移数据时都会创建一个 MigMigration CR。每个 MigMigration CR 都会与一个 MigPlan CR 关联。

7 **Backup** (操作, 源集群) : 当运行迁移计划时, MigMigration CR 在每个源集群上创建两个 Velero 备份 CR :

- 备份 CR #1 用于 Kubernetes 对象
- 备份 CR #2 用于 PV 数据

8 **Restore** (操作, 目标集群) : 在运行迁移计划时, MigMigration CR 在目标集群上创建两个 Velero 恢复 CR :

- 恢复 CR #1 (使用备份 CR #2) 用于 PV 数据
- 恢复 CR #2 (使用备份 CR #1) 用于 Kubernetes 对象

流程

1. 获取 CR 名称 :

```
$ oc get <cr> -n openshift-migration 1
NAME                                AGE
88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s
```

1 指定您要查看的迁移 CR。

2. 查看 CR :

```
$ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
```

输出结果类似以下示例。

MigMigration 示例

```
$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
Name:      88435fe0-c9f8-11e9-85e6-5d593ce65e10
Namespace: openshift-migration
Labels:    <none>
Annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
API Version: migration.openshift.io/v1alpha1
Kind:      MigMigration
Metadata:
  Creation Timestamp: 2019-08-29T01:01:29Z
  Generation:        20
  Resource Version:   88179
  Self Link:          /apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
```

```

UID:          8886de4c-c9f8-11e9-95ad-0205fe66cbb6
Spec:
  Mig Plan Ref:
    Name:      socks-shop-mig-plan
    Namespace: openshift-migration
  Quiesce Pods: true
  Stage:      false
Status:
  Conditions:
    Category:      Advisory
    Durable:       true
    Last Transition Time: 2019-08-29T01:03:40Z
    Message:       The migration has completed successfully.
    Reason:        Completed
    Status:        True
    Type:          Succeeded
  Phase:         Completed
  Start Timestamp: 2019-08-29T01:01:29Z
  Events:        <none>

```

Velero 备份 CR #2 示例 (PV 数据)

```

apiVersion: velero.io/v1
kind: Backup
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.105.179:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-0205fe66cbb6
  creationTimestamp: "2019-08-29T01:03:15Z"
  generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
  generation: 1
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    velero.io/storage-location: myrepo-vpzq9
  name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  namespace: openshift-migration
  resourceVersion: "87313"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:
  excludedNamespaces: []
  excludedResources: []
  hooks:
    resources: []
  includeClusterResources: null
  includedNamespaces:
  - sock-shop
  includedResources:
  - persistentvolumes
  - persistentvolumeclaims

```



```

- namespaces
- imagestreams
- imagestreamtags
- secrets
- configmaps
- pods
labelSelector:
  matchLabels:
    migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
storageLocation: myrepo-vpzq9
ttl: 720h0m0s
volumeSnapshotLocations:
- myrepo-wv6fx
status:
  completionTimestamp: "2019-08-29T01:02:36Z"
  errors: 0
  expiration: "2019-09-28T01:02:35Z"
  phase: Completed
  startTimestamp: "2019-08-29T01:02:35Z"
  validationErrors: null
  version: 1
  volumeSnapshotsAttempted: 0
  volumeSnapshotsCompleted: 0
  warnings: 0

```

Velero 恢复 CR #2 示例 (Kubernetes 资源)

```

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
  generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
    migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
  name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  namespace: openshift-migration
  resourceVersion: "82329"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
    - nodes
    - events
    - events.events.k8s.io

```

```

- backups.velero.io
- restores.velero.io
- resticrepositories.velero.io
includedNamespaces: null
includedResources: null
namespaceMapping: null
restorePVs: true
status:
  errors: 0
  failureReason: ""
  phase: Completed
  validationErrors: null
  warnings: 15

```

3.5.2. 下载迁移日志

您可以在 CAM web 控制台中下载 Velero、Restic 和 Migration controller 日志，以排除出现故障的迁移问题。

流程

1. 登录到 CAM 控制台。
2. 点击 **Plans** 查看迁移计划列表。
3. 点一个迁移计划的 **Options** 菜单  并选择 **Logs**。
4. 点 **Download Logs** 为所有集群下载迁移控制器、Velero 和 Restic 的日志。
5. 要下载特定的日志：
 - a. 指定日志选项：
 - **Cluster**：选择源、目标或 CAM 主机集群。
 - **Log source**：选择 **Velero**、**Restic** 或 **Controller**。
 - **Pod source**：选择 Pod 名称，例如：**controller-manager-78c469849c-v6wcf**
此时会显示所选日志。

您可以通过更改您的选择来清除日志选择设置。
 - b. 点 **Download Selected** 下载所选日志。

另外，您可以使用 CLI 访问日志，如下例所示：

```

$ oc get pods -n openshift-migration | grep controller
controller-manager-78c469849c-v6wcf      1/1   Running   0      4h49m

$ oc logs controller-manager-78c469849c-v6wcf -f -n openshift-migration

```

3.5.3. Restic 超时错误

如果因为 Restic 超时造成迁移失败，以下出错信息会出现在 Velero 日志中：

```
level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all
PodVolumeBackups to complete"
error.file="/go/src/github.com/heptio/velero/pkg/restic/backupper.go:165"
error.function="github.com/heptio/velero/pkg/restic.(*backupper).BackupPodVolumes" group=v1
```

restic_timeout 的默认值为一小时。您可以为大型迁移增加这个值，请注意，高的值可能会延迟返回出错信息。

流程

1. 在 OpenShift Container Platform web 控制台中导航至 **Operators** → **Installed Operators**。
2. 点 **Cluster Application Migration Operator**。
3. 在 **MigrationController** 标签页中点 **migration-controller**。
4. 在 **YAML** 标签页中，更新以下参数值：

```
spec:
  restic_timeout: 1h 1
```

1 有效单元是 **h**（小时）、**m**（分钟）和 **s**（秒），例如 **3h30m15s**。

5. 点 **Save**。

3.5.4. 手动回滚迁移

如果您的应用程序在迁移失败时停止，您必须手动回滚，以防止 PV 中的数据被破坏。

如果应用程序在迁移过程中没有停止，则不需要进行手动回滚，因为原始应用程序仍然在源集群中运行。

流程

1. 在目标集群中，切换到迁移的项目：

```
$ oc project <project>
```

2. 获取部署的资源：

```
$ oc get all
```

3. 删除部署的资源以确保应用程序没有在目标集群中运行，并访问 PVC 上的数据：

```
$ oc delete <resource_type>
```

4. 要停止 DaemonSet 而不删除它，在 YAML 文件中更新 **nodeSelector**：

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: hello-daemonset
spec:
```

```

selector:
  matchLabels:
    name: hello-daemonset
template:
  metadata:
    labels:
      name: hello-daemonset
  spec:
    nodeSelector:
      role: worker ❶

```

- ❶ 指定一个没有存在于任何节点上的 **nodeSelector** 值。

- 更新每个 PV 的重新声明策略，以便删除不必要的数据库。在迁移过程中，绑定 PV 的重新声明策略是 **reclaim**，以确保应用程序从源集群中被删除时不会丢失数据。您可以在回滚过程中删除这些 PV。

```

apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain ❶
  ...
status:
  ...

```

- ❶ 指定 **Recycle** 或 **Delete**。

- 在源集群中，切换到迁移的项目并获取其部署的资源：

```

$ oc project <project>
$ oc get all

```

- 启动每个部署资源的一个或多个副本：

```

$ oc scale --replicas=1 <resource_type>/<resource_name>

```

- 如果在操作中被更改了，把 DaemonSet 的 **nodeSelector** 改回其原始值。

3.5.5. 为客户支持问题单收集数据

如果创建一个客户支持问题单，您可以使用 **openshift-migration-must-gather-rhel8** 镜像的 **must-gather** 工具来收集与您的集群相关的信息，并把这些信息上传到[红帽客户门户网站](#)。

openshift-migration-must-gather-rhel8 镜像会收集默认的 **must-gather** 镜像不收集的日志和 CR 数据。

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/rhcam-1-2/openshift-migration-must-gather-rhel8
```

must-gather 工具程序收集集群数据，并把它保存在 **must-gather.local.<uid>** 目录中。

3. 从 **must-gather** 数据中删除验证密钥和其他敏感信息。
4. 创建一个包含 **must-gather.local.<uid>** 目录内容的归档文件：

```
$ tar cvaf must-gather.tar.gz must-gather.local.<uid>/
```

在[红帽客户门户](#)中，为您的问题单附上这个压缩文件。

3.5.6. 已知问题

这个版本有以下已知问题：

- 在迁移过程中，CAM 工具会保留以下命名空间注解：
 - **openshift.io/sa.scc.mcs**
 - **openshift.io/sa.scc.supplemental-groups**
 - **openshift.io/sa.scc.uid-range**
 这些注解会保留 UID 范围，确保容器在目标集群中保留其文件系统权限。这可能会存在一定的风险。因为迁移的 UID 可能已存在于目标集群的现有或将来的命名空间中。
 ([BZ#1748440](#))
- 当在 CAM web 控制台添加 S3 端点时，只有 AWS 支持 **https://**。对于其他 S3 供应商，请使用 **http://**。
- 如果一个 AWS 存储桶被添加到 CAM web 控制台，然后将其删除，则其状态会保持为 **True**，这是因为 MigStorage CR 没有被更新。[\(BZ#1738564\)](#)
- 如果迁移控制器在目标集群以外的集群中运行，迁移将失败。**EnsureCloudSecretPropagated** 阶段会跳过，并给出一个日志警告。[\(BZ#1757571\)](#)
- 目前，集群范围的资源，包括集群角色绑定和安全上下文约束，还没有由 CAM 处理。如果应用程序需要集群范围的资源，则必须在目标集群中手动创建它们。[\(BZ#1759804\)](#)
- 创建迁移计划时会显示不正确的源集群存储类。[\(BZ#1777869\)](#)
- 如果 CAM web 控制台中的一个集群变得无法访问，它会阻止尝试关闭打开的迁移计划。[\(BZ#1758269\)](#)
- 如果迁移失败，则迁移计划不会为静默的 pod 保留自定义 PV 设置。您必须手动回滚，删除迁移计划，并使用 PV 设置创建新的迁移计划。[\(BZ#1784899\)](#)

