解决方案验证和使用情形
NetApp Solutions

NetApp
April 12, 2024

目录

解决方案验证和使用情形：采用 NetApp 的 Red Hat OpenShift

此页面上提供的示例包括解决方案验证以及采用 NetApp 的 Red Hat OpenShift 的用例。

- "部署具有永久性存储的 Jenkins CI/CD 管道"
- "在使用 NetApp 的 Red Hat OpenShift 上配置多租户"
- "借助 NetApp ONTAP 实现 Red Hat OpenShift 虚拟化"
- "借助 NetApp 在 Red Hat OpenShift 上为 Kubernetes 提供高级集群管理"

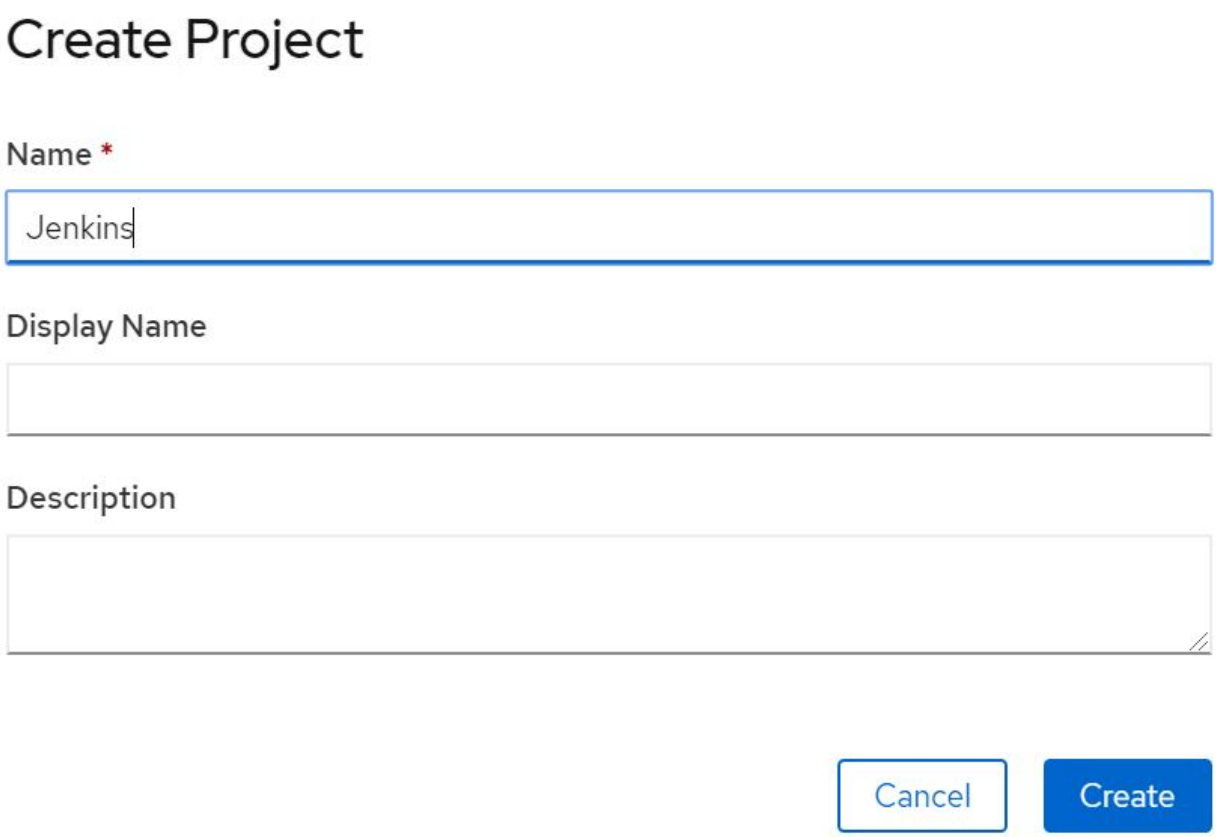
部署采用永久性存储的 Jenkins CI/CD 管道：采用 NetApp 的 Red Hat OpenShift

本节介绍了与 Jenkins 部署持续集成 / 持续交付或部署（ CI/CD ）管道以验证解决方案运行的步骤。

创建 Jenkins 部署所需的资源

要创建部署 Jenkins 应用程序所需的资源，请完成以下步骤：

1. 创建一个名为 Jenkins 的新项目。



2. 在此示例中，我们使用永久性存储部署了 Jenkins 。要支持 Jenkins 构建，请创建 PVC 。导航到 "Storage">"Persistent Volume Claim " ，然后单击 "Create Persistent Volume Claim " 。选择已创建的存储类，确保永久性卷声明名称是 Jenkins ，选择适当的大小和访问模式，然后单击创建。

Project: jenkins

Create Persistent Volume Claim

Edit YAML

Storage Class

SC basic

Storage class for the new claim.

Persistent Volume Claim Name *

jenkins

A unique name for the storage claim within the project.

Access Mode *

Single User (RWO) Shared Access (RWX) Read Only (ROX)

Permissions to the mounted drive.

Size *

100 GiB

Desired storage capacity.

Use label selectors to request storage

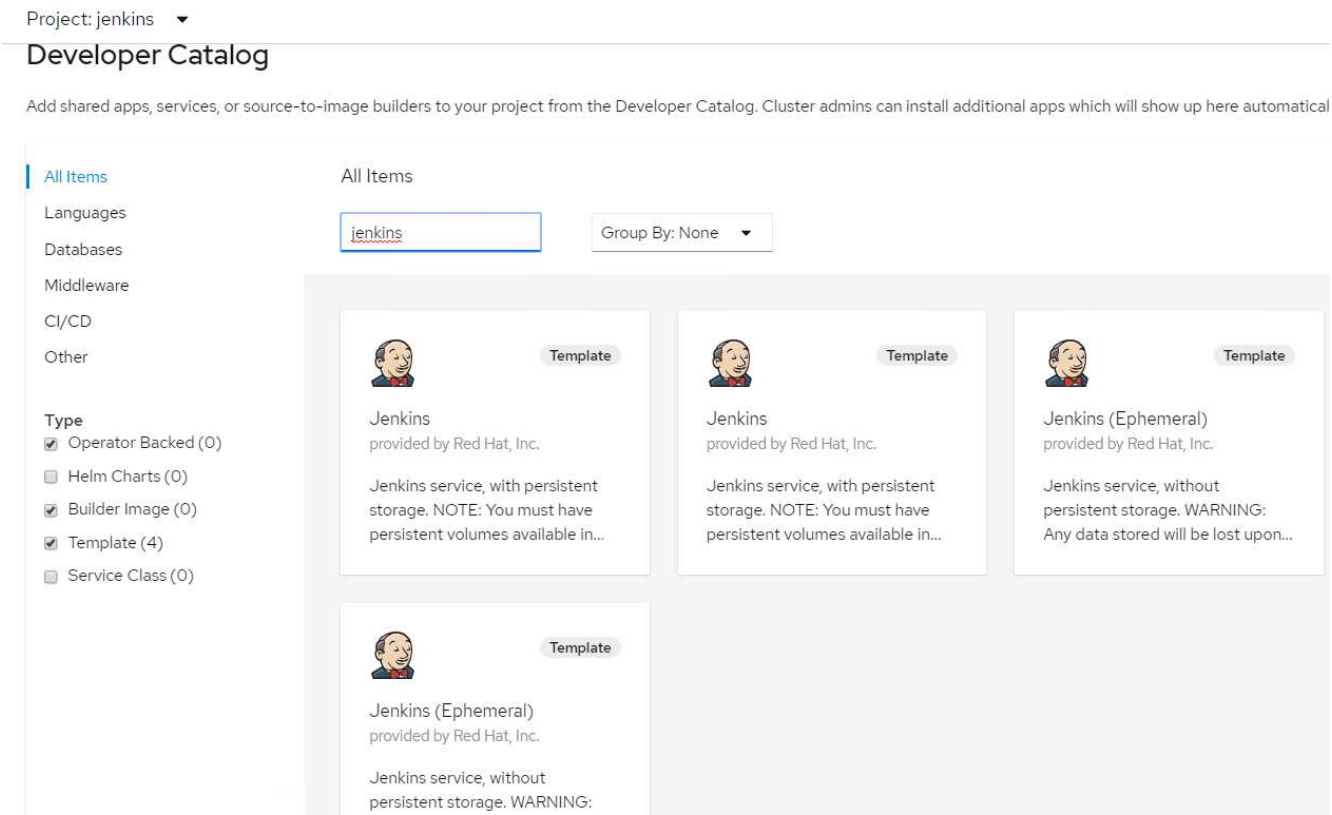
Use label selectors to define how storage is created.

Create Cancel

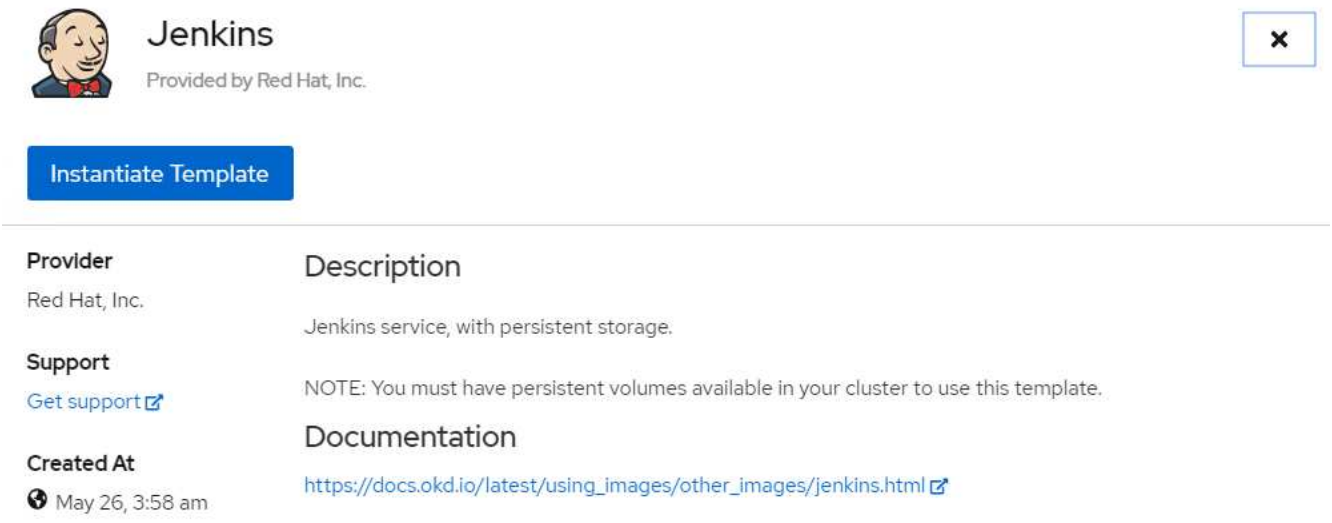
使用永久性存储部署 Jenkins

要使用永久性存储部署 Jenkins ，请完成以下步骤：

1. 在左上角，将角色从管理员更改为开发人员。单击 +Add ，然后从目录中选择。在 Filter by Keyword 栏中，搜索 Jenkins 。选择 Jenkins Service with Persistent Storage 。



2. 单击 实例化模板 。



3. 默认情况下，系统会填充 Jenkins 应用程序的详细信息。根据您的要求，修改参数并单击创建。此过程将创建支持 OpenShift 上的 Jenkins 所需的所有资源。

Instantiate Template

Namespace *

PR jenkins

Jenkins Service Name

jenkins

The name of the OpenShift Service exposed for the Jenkins container.

Jenkins JNLP Service Name

jenkins-jnlp

The name of the service used for master/slave communication.

Enable OAuth in Jenkins

true

Whether to enable OAuth OpenShift integration. If false, the static account 'admin' will be initialized with the password 'password'.

Memory Limit

1Gi

Maximum amount of memory the container can use.

Volume Capacity *

50Gi

Volume space available for data, e.g. 512Mi, 2Gi.

Jenkins ImageStream Namespace

openshift

The OpenShift Namespace where the Jenkins ImageStream resides.

Disable memory intensive administrative monitors

false

Whether to perform memory intensive, possibly slow, synchronization with the Jenkins Update Center on start. If true, the Jenkins core update monitor and site warnings monitor are disabled.

Jenkins ImageStreamTag

jenkins:2

Name of the ImageStreamTag to be used for the Jenkins image.

Fatal Error Log File

false

When a fatal error occurs, an error log is created with information and the state obtained at the time of the fatal error.

Allows use of Jenkins Update Center repository with invalid SSL certificate

false

Whether to allow use of a Jenkins Update Center that uses invalid certificate (self-signed, unknown CA). If any value other than 'false', certificate check is bypassed. By default, certificate check is enforced.

Create Cancel

Jenkins

INSTANT-APP JENKINS

View documentation Get support

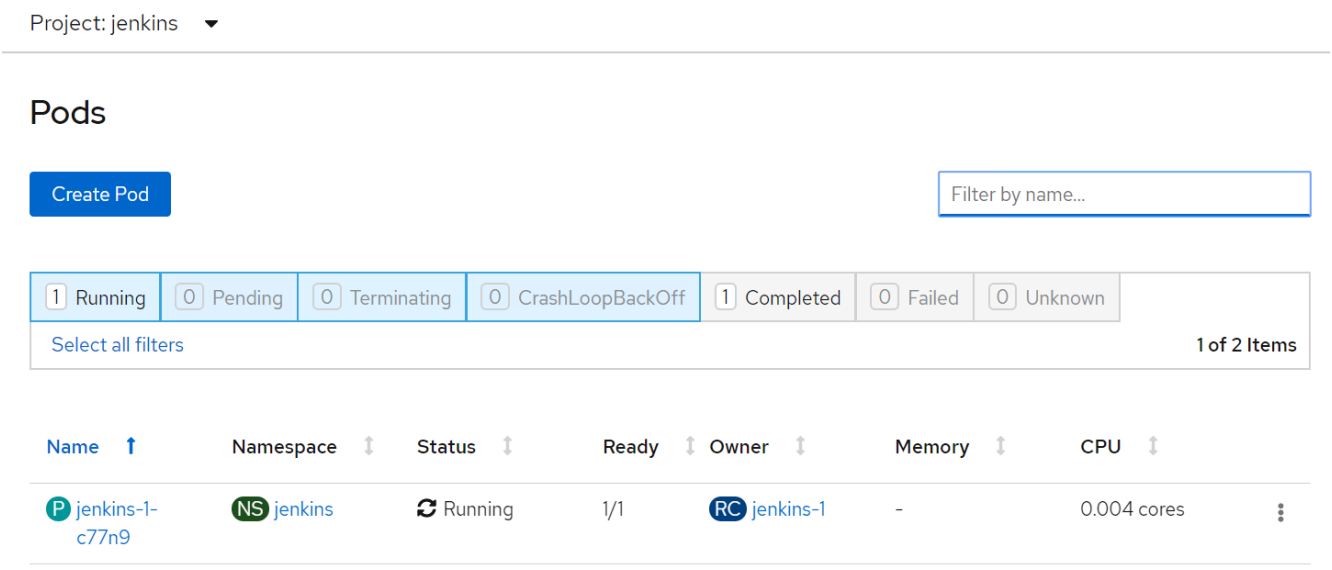
Jenkins service, with persistent storage.

NOTE: You must have persistent volumes available in your cluster to use this template.

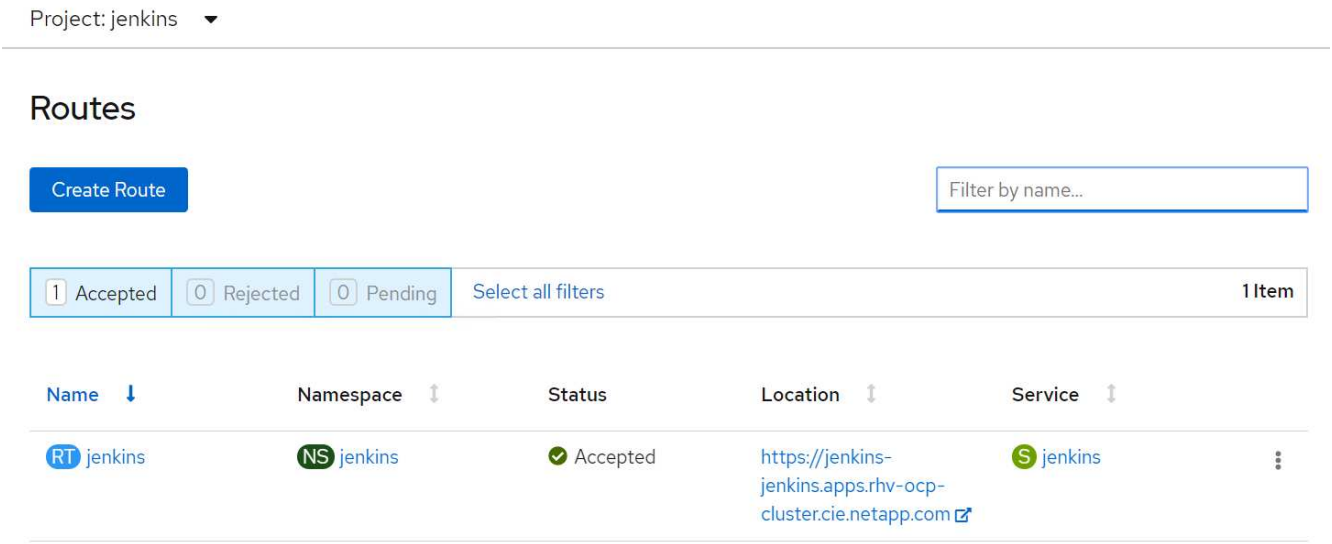
The following resources will be created:

- DeploymentConfig
- PersistentVolumeClaim
- RoleBinding
- Route
- Service
- ServiceAccount

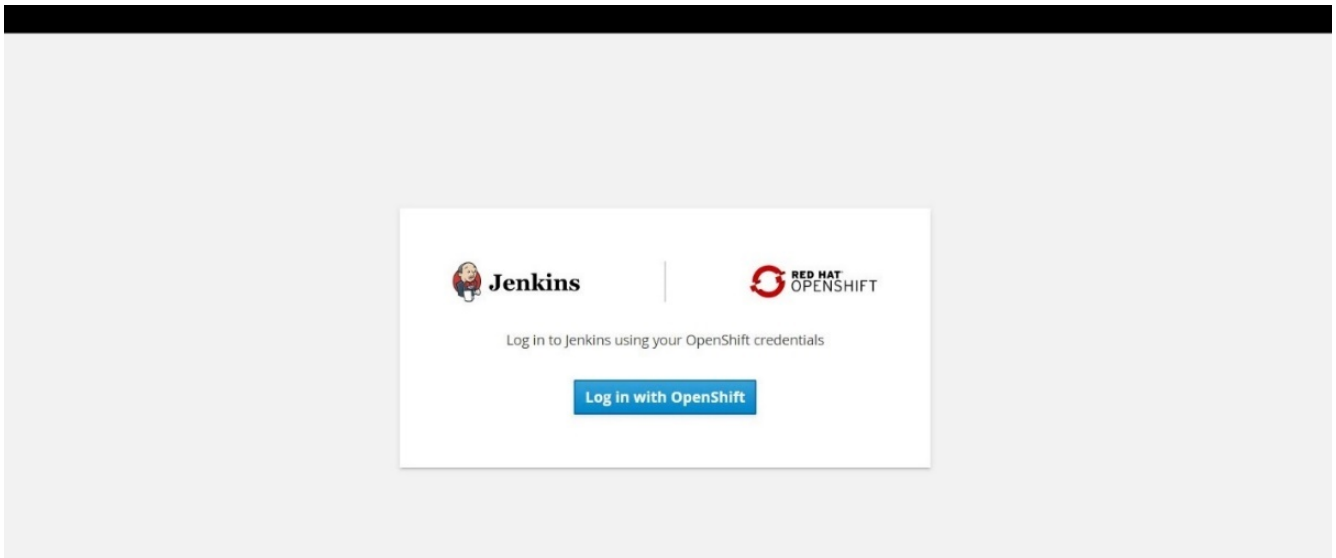
4. Jenkins Pod 大约需要 10 到 12 分钟才能进入就绪状态。



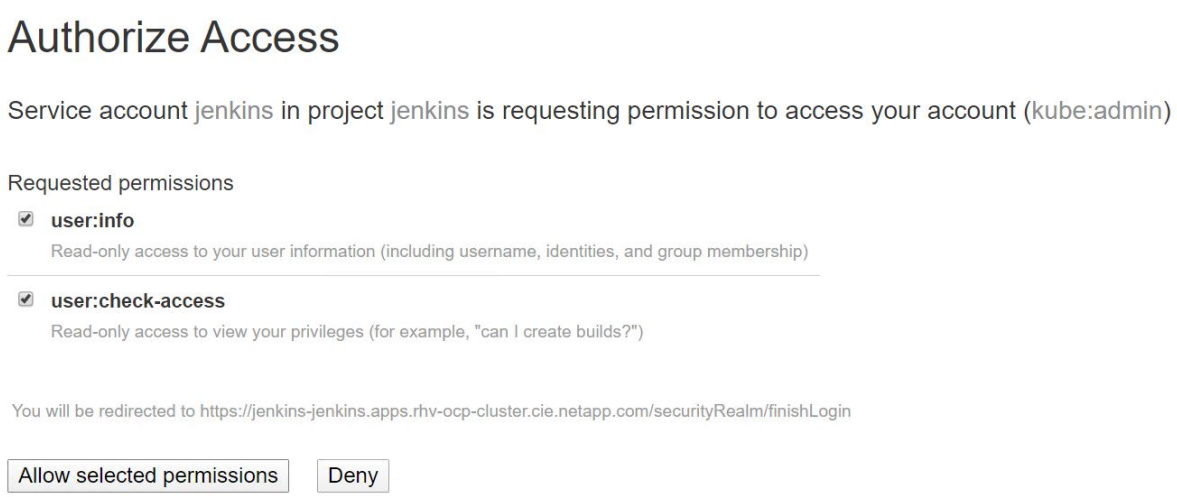
5. 实例化 Pod 后，导航到 "Networking" （网络） >"routes" （路由）。要打开 Jenkins 网页，请单击为 Jenkins 路由提供的 URL 。



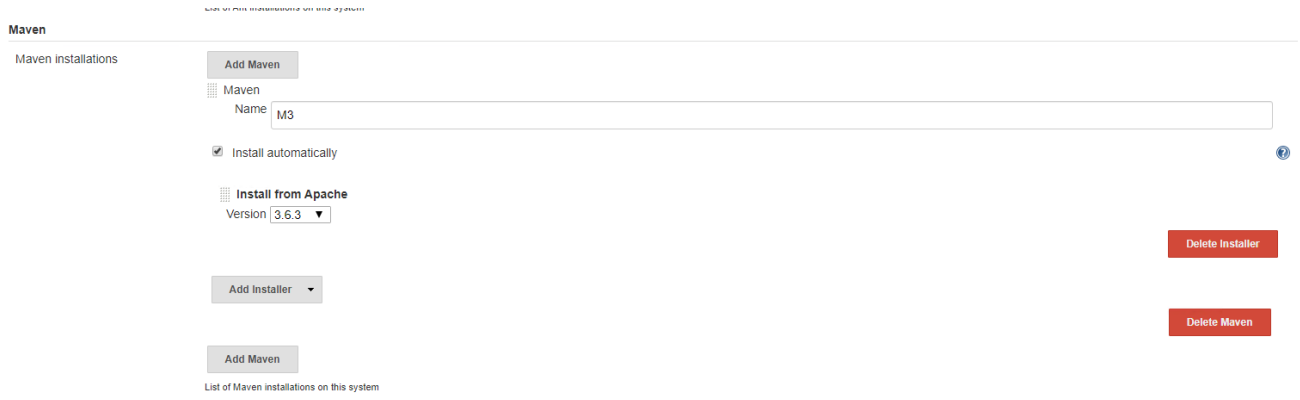
6. 由于在创建 Jenkins 应用程序时使用了 OpenShift OAuth ，因此请单击使用 OpenShift 登录。



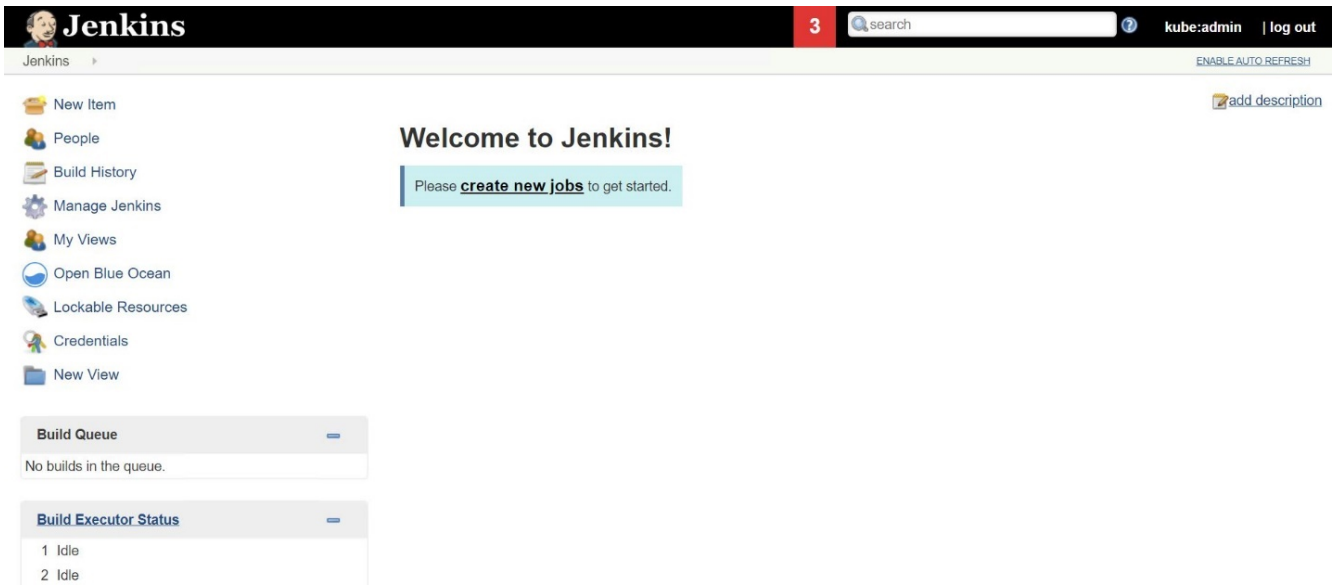
7. 授权 Jenkins 服务帐户访问 OpenShift 用户。



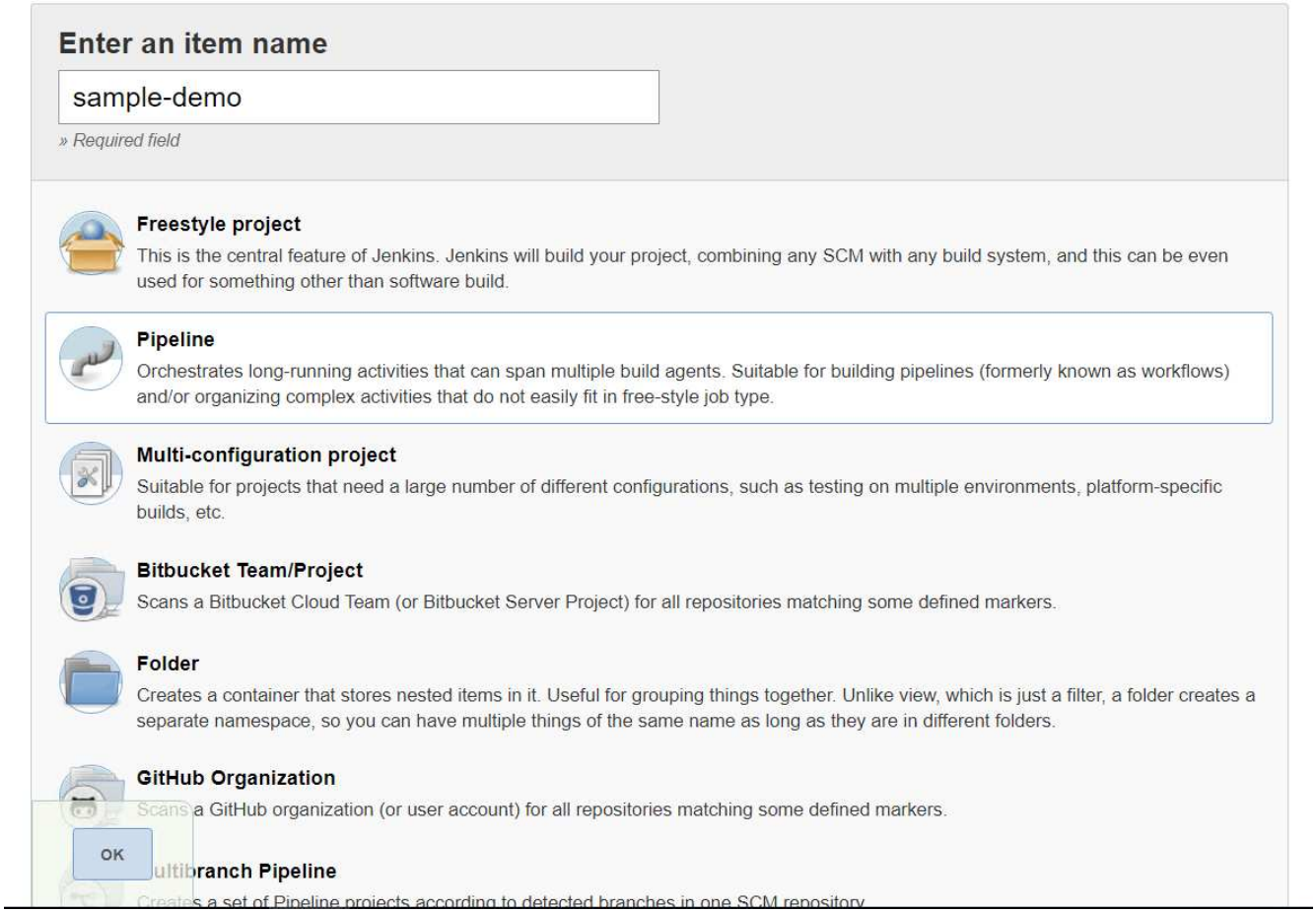
8. 此时将显示 Jenkins 欢迎页面。由于我们使用的是 Maven 内部版本，因此请先完成 Maven 安装。导航到 Manage Jenkins > Global Tool Configuration ，然后在 Maven 子标题中单击 Add Maven 。输入您选择的名称，并确保已选中自动安装选项。单击保存。



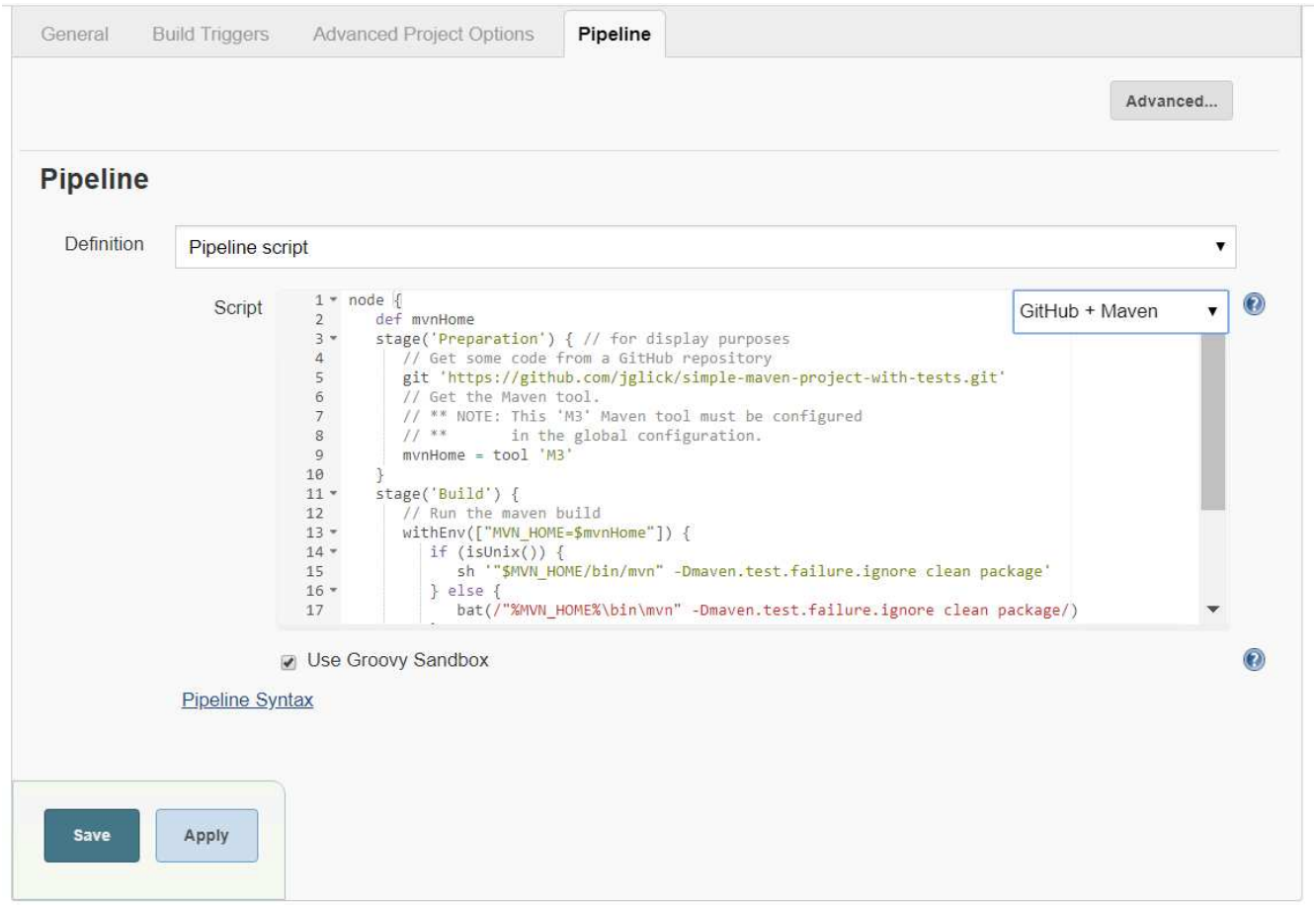
9. 现在，您可以创建一个管道来演示 CI/CD 工作流。在主页上，单击左侧菜单中的创建新作业或新建项目。



10. 在创建项目页面上，输入所选名称，选择管道，然后单击确定。



11. 选择管道选项卡。从试用样本管道下拉菜单中，选择 Github + Maven 。代码将自动填充。单击保存。



12. 单击 Build now ，在准备，构建和测试阶段触发开发。完成整个构建过程并显示构建结果可能需要几分钟的时间。

13. 只要代码发生任何更改，就可以重新构建管道来修补新版本的软件，从而实现持续集成和持续交付。单击 Recent Changes 以跟踪与先前版本相比的更改。

使用 NetApp ONTAP 在 Red Hat OpenShift 上配置多租户

在使用 NetApp 的 Red Hat OpenShift 上配置多租户

许多在容器上运行多个应用程序或工作负载的组织往往会为每个应用程序或工作负载部署一个 Red Hat OpenShift 集群。这样，他们就可以对应用程序或工作负载实施严格的隔离，优化性能并减少安全漏洞。但是，为每个应用程序部署一个单独的 Red Hat OpenShift 集群会产生自己的一系列问题。它增加了单独监控和管理每个集群所需的运营开销，由于为不同应用程序配置了专用资源而增加了成本，并妨碍了高效的可扩展性。

要解决这些问题，可以考虑在一个 Red Hat OpenShift 集群中运行所有应用程序或工作负载。但是，在这种架构中，资源隔离和应用程序安全漏洞是主要挑战之一。一个工作负载中的任何安全漏洞都可能自然溢出到另一个工作负载，从而增加影响区域。此外，一个应用程序的任何突然不受控制的资源利用率都会影响另一个应用程序的性能，因为默认情况下没有资源分配策略。

因此，企业需要寻找在这两种环境中都能获得最佳性能的解决方案，例如，允许他们在一个集群中运行所有工作负载，同时为每个工作负载提供专用集群的优势。

其中一个有效的解决方案是在 Red Hat OpenShift 上配置多租户。多租户是一种架构，允许多个租户在同一集群上共存，并正确隔离资源，安全性等。在这种情况下，可以将租户视为集群资源的一部分，这些资源配置为供特定用户组专用使用。在 Red Hat OpenShift 集群上配置多租户具有以下优势：

- 通过共享集群资源，降低资本支出和运营支出
- 降低运营和管理开销
- 保护工作负载免受安全违规交叉影响
- 保护工作负载，防止因资源争用而导致性能意外下降

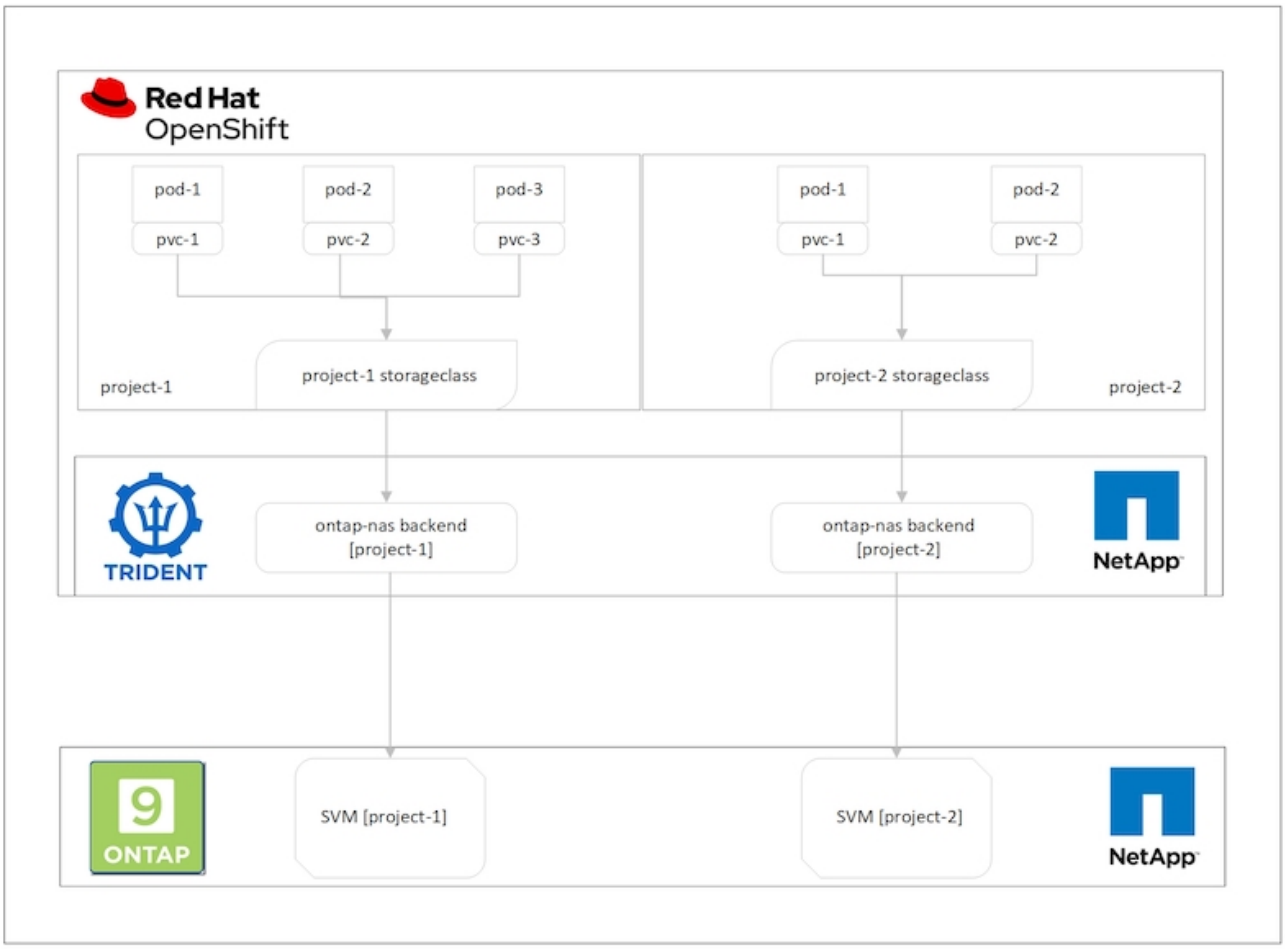
对于完全实现的多租户 OpenShift 集群，必须为属于不同资源分段的集群资源配置配额和限制：计算，存储，网络连接，安全性等。虽然我们会介绍此解决方案中所有资源分段的某些方面， 我们重点介绍最佳实践，通过在由 NetApp ONTAP 提供支持的 Astra Trident 动态分配的存储资源上配置多租户，隔离并保护同一 Red Hat OpenShift 集群上多个工作负载提供或使用的数据。

架构

虽然默认情况下，由 NetApp ONTAP 提供支持的 Red Hat OpenShift 和 Astra Trident 不会在工作负载之间提供隔离，但它们提供了广泛的功能，可用于配置多租户。为了更好地了解如何在采用 NetApp ONTAP 支持的 Astra Trident 的 Red Hat OpenShift 集群上设计多租户解决方案，让我们考虑一个包含一系列要求的示例，并概述其配置。

假设一个组织在一个 Red Hat OpenShift 集群上运行两个工作负载，作为两个不同团队正在处理的两个项目的一部分。这些工作负载的数据驻留在由 NetApp ONTAP NAS 后端的 Astra Trident 动态配置的 PVC 上。该组织需要为这两个工作负载设计多租户解决方案，并隔离用于这些项目的资源，以确保保持安全性和性能，主要侧重于为这些应用程序提供服务的数据。

下图展示了由 NetApp ONTAP 提供支持的带有 Astra Trident 的 Red Hat OpenShift 集群上的多租户解决方案。



技术要求

1. NetApp ONTAP 存储集群
2. Red Hat OpenShift 集群
3. Astra Trident

Red Hat OpenShift —集群资源

从 Red Hat OpenShift 集群的角度来看，要开始使用的顶级资源是项目。OpenShift 项目可以视为将整个 OpenShift 集群划分为多个虚拟集群的集群资源。因此，项目级别的隔离为配置多租户提供了基础。

下一步是在集群中配置 RBAC 。最佳做法是，将处理单个项目或工作负载的所有开发人员配置到身份提供程序（ IdP ）中的单个用户组中。Red Hat OpenShift 允许 IdP 集成和用户组同步，从而允许将 IdP 中的用户和组导入到集群中。这样可以帮助集群管理员将项目专用集群资源的访问权限隔离给一个或多个处理该项目的用户组，从而限制对任何集群资源的未授权访问。要了解有关 IdP 与 Red Hat OpenShift 集成的详细信息，请参见相关文档 "此处"。

NetApp ONTAP

必须隔离用作 Red Hat OpenShift 集群永久性存储提供程序的共享存储，以确保在存储上为每个项目创建的卷在主机上显示为它们，就像在单独的存储上创建一样。为此，请在 NetApp ONTAP 上创建与项目或工作负载数量相同的 SVM （ Storage Virtual Machine ），并将每个 SVM 专用于一个工作负载。

Astra Trident

在 NetApp ONTAP 上为不同项目创建不同的 SVM 之后，必须将每个 SVM 映射到不同的 Trident 后端。Trident 上的后端配置会将永久性存储分配给 OpenShift 集群资源，并且需要将 SVM 的详细信息映射到。此驱动程序至少应为后端的协议驱动程序。或者，您也可以通过它定义如何在存储上配置卷，并设置卷大小或聚合使用量等限制。有关 Trident 后端定义的详细信息，请参见 "此处"。

Red Hat OpenShift —存储资源

配置 Trident 后端后，下一步是配置 StorageClasses 。配置与后端相同数量的存储类，为每个存储类提供访问权限，以便仅在一个后端启动卷。在定义存储类时，我们可以使用 storagePools 参数将 StorageClass 映射到特定的 Trident 后端。可以找到用于定义存储类的详细信息 "此处"。因此，从 StorageClass 到 Trident 后端存在一对一映射，这种映射可指向一个 SVM 。这样可以确保通过分配给该项目的 StorageClass 处理的所有存储请求仅由专用于该项目的 SVM 处理。

由于存储类不是命名空间资源，我们如何确保拒绝另一命名空间或项目中的 Pod 向一个项目的存储类声明？问题解答将使用 ResourceQuotas 。ResourceQuotas 是控制每个项目资源总使用量的对象。它可以限制项目中的对象可以使用的资源数量以及总资源量。使用 ResourceQuotas 几乎可以限制项目中的所有资源，而高效地使用此功能可以帮助组织降低因过度配置或过度消耗资源而导致的成本和中断。请参见文档 "此处" 有关详细信息 …

对于这种使用情形，我们需要限制特定项目中的 Pod 从非专用于其项目的存储类中申请存储。为此，我们需要通过将 ` <storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` 设置为 0 来限制其他存储类的永久性卷请求。此外，集群管理员必须确保项目中的开发人员不应有权修改 ResourceQuotas 。

Configuration

对于任何多租户解决方案，任何用户都无法访问比所需更多的集群资源。因此，要在多租户配置中配置的整个资源集将在集群管理员，存储管理员和处理每个项目的开发人员之间进行划分。

下表概括了不同用户要执行的不同任务：

Role	任务
* 集群管理 *	为不同的应用程序或工作负载创建项目
	为 storage-admin 创建 ClusterRoles 和 RoleBindings
	为分配对特定项目的访问权限的开发人员创建角色和角色绑定
	[可选] 配置项目以在特定节点上计划 Pod
* 存储管理 *	在 NetApp ONTAP 上创建 SVM
	创建 Trident 后端
	创建 StorageClasses
	创建存储 ResourceQuotas
* 开发人员 *	验证对已分配项目中的 PVC 或 Pod 的创建或修补访问权限
	验证对在其他项目中创建或修补 PVC 或 Pod 的访问权限
	验证对查看或编辑项目， ResourceQuotas 和 StorageClasses 的访问权限

Configuration

前提条件

- NetApp ONTAP 集群。
- Red Hat OpenShift 集群
- 集群上安装的 Trident 。
- 安装了 tridentctl 和 oc 工具并将其添加到 $path 中的管理工作站。
- 对 ONTAP 的管理员访问权限。
- 对 OpenShift 集群的集群管理员访问。
- 集群已与身份提供程序集成。
- 身份提供程序经过配置，可以有效区分不同团队中的用户。

配置： **cluster-admin** 任务

Red Hat OpenShift cluster-admin 执行以下任务：

1. 以 cluster-admin 身份登录到 Red Hat OpenShift 集群。
2. 创建两个与不同项目对应的项目。

```
oc create namespace project-1
oc create namespace project-2
```

3. 为 project-1 创建开发人员角色。

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-1
  name: developer-project-1
rules:
  - verbs:
      - '*'
    apiGroups:
      - apps
      - batch
      - autoscaling
      - extensions
      - networking.k8s.io
      - policy
      - apps.openshift.io
      - build.openshift.io
```

```
      - image.openshift.io
      - ingress.operator.openshift.io
      - route.openshift.io
      - snapshot.storage.k8s.io
      - template.openshift.io
    resources:
      - '*'
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - bindings
      - configmaps
      - endpoints
      - events
      - persistentvolumeclaims
      - pods
      - pods/log
      - pods/attach
      - podtemplates
      - replicationcontrollers
      - services
      - limitranges
      - namespaces
      - componentstatuses
      - nodes
  - verbs:
      - '*'
    apiGroups:
      - trident.netapp.io
    resources:
      - tridentsnapshots
EOF
```

本节中提供的角色定义只是一个示例。必须根据最终用户要求定义开发人员角色。

1. 同样，为 project-2 创建开发人员角色。
2. 所有 OpenShift 和 NetApp 存储资源通常由存储管理员管理。存储管理员的访问由安装 Trident 时创建的 Trident 操作员角色控制。此外，存储管理员还需要访问 ResourceQuotas 来控制存储的使用方式。
3. 在集群中的所有项目中创建一个用于管理 ResourceQuotas 的角色，以将其连接到存储管理员：

```
cat << EOF | oc create -f -
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: resource-quotas-role
rules:
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - resourcequotas
  - verbs:
      - '*'
    apiGroups:
      - quota.openshift.io
    resources:
      - '*'
EOF
```

4. 确保集群与组织的身份提供程序集成，并且用户组与集群组同步。以下示例显示身份提供程序已与集群集成并与用户组同步。

```
$ oc get groups
NAME                          USERS
ocp-netapp-storage-admins     ocp-netapp-storage-admin
ocp-project-1                 ocp-project-1-user
ocp-project-2                 ocp-project-2-user
```

1. 为存储管理员配置 ClusterRoleBindings 。

```
cat << EOF | oc create -f -
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-trident-operator
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-operator
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-resource-quotas-cr
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: resource-quotas-role
EOF
```

对于存储管理员，必须绑定两个角色： Trident 操作员和资源配额。

1. 为开发人员创建 RoleBindings ，将开发人员项目 1 角色绑定到项目 1 中的相应组（ OCP-project-1 ）。

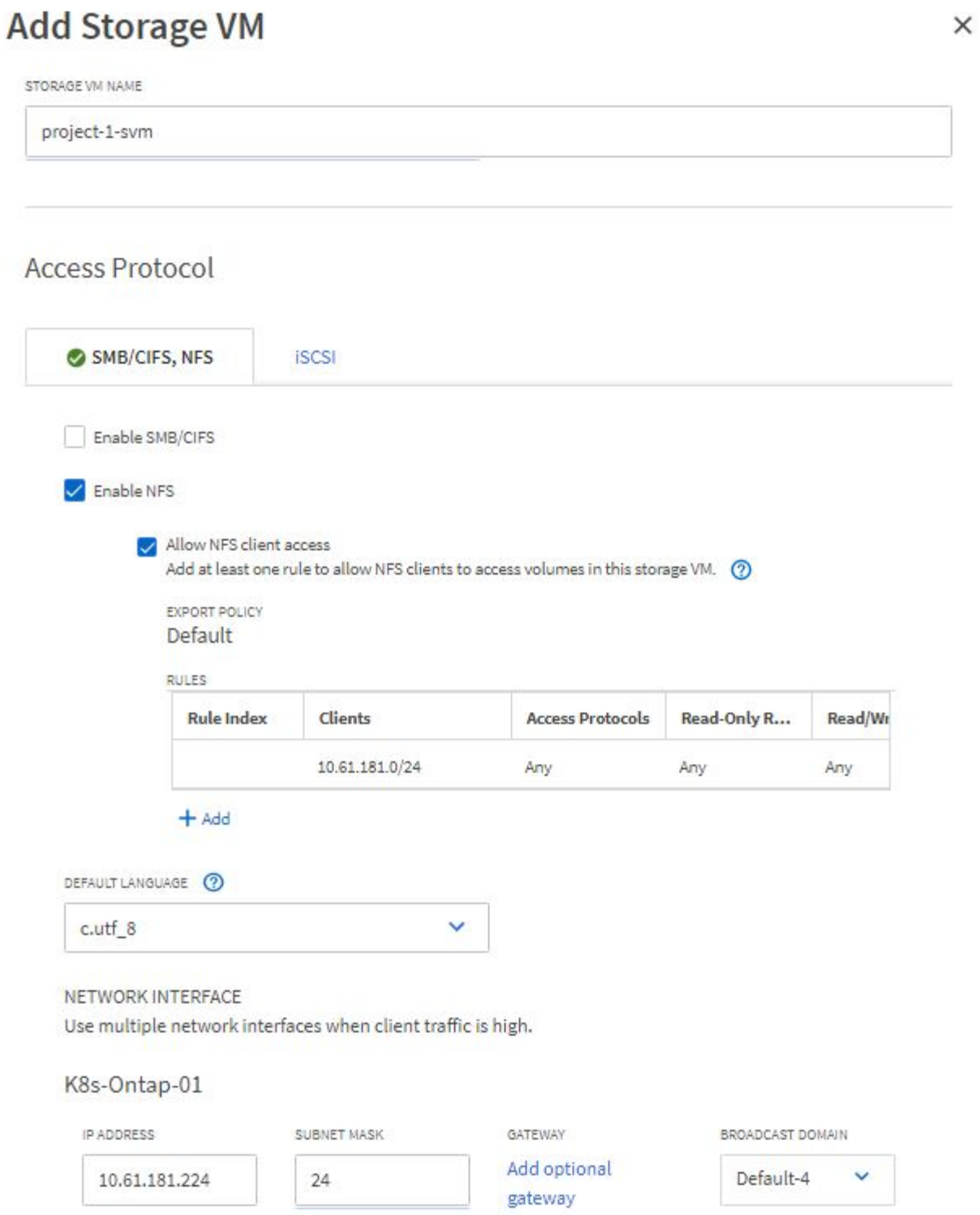
```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-1
EOF
```

2. 同样，为开发人员创建 RoleBindings ，将开发人员角色绑定到 project-2 中的相应用户组。

配置： **storage-admin** 任务

存储管理员必须配置以下资源：

1. 以管理员身份登录到 NetApp ONTAP 集群。
2. 导航到存储 > Storage VM ，然后单击添加。通过提供所需的详细信息，创建两个 SVM ，一个用于 project-1 ，另一个用于 project-2 。此外，还可以创建 vsadmin 帐户来管理 SVM 及其资源。



1. 以存储管理员身份登录到 Red Hat OpenShift 集群。
2. 为 project-1 创建后端，并将其映射到专用于该项目的 SVM 。NetApp 建议使用 SVM 的 vsadmin 帐户将后端连接到 SVM ，而不是使用 ONTAP 集群管理员。

```
cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_1",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.224",
    "svm": "project-1-svm",
    "username": "vsadmin",
    "password": "NetApp123"
}
EOF
```

在此示例中，我们使用的是 ontap-NAS 驱动程序。根据使用情形创建后端时，请使用相应的驱动程序。

我们假定 Trident 已安装在 Trident 项目中。

1. 同样，为 project-2 创建 Trident 后端，并将其映射到专用于 project-2 的 SVM 。
2. 接下来，创建存储类。为 project-1 创建存储类，并通过设置 storagePools 参数将其配置为使用后端专用于 project-1 的存储池。

```
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-1-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_1:.*"
EOF
```

3. 同样，为 project-2 创建一个存储类，并将其配置为使用专用于 project-2 的后端存储池。
4. 创建 ResourceQuota 以限制 project-1 中的资源，从而从专用于其他项目的存储库请求存储。

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-1-sc-rq
  namespace: project-1
spec:
  hard:
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

5. 同样，也可以创建 ResourceQuota 来限制项目 2 中的资源，以便从专用于其他项目的存储库请求存储。

验证

要验证在上述步骤中配置的多租户架构，请完成以下步骤：

验证在分配的项目中创建 **PVC** 或 **Pod** 的访问权限

1. 以项目 1 中的 OCP-project-1-user 和开发人员身份登录。
2. 检查访问权限以创建新项目。

```
oc create ns sub-project-1
```

3. 使用分配给 project-1 的 storageclass 在 project-1 中创建 PVC 。

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

4. 检查与 PVC 关联的 PV 。

```
oc get pv
```

5. 验证 PV 及其卷是否已在 NetApp ONTAP 上专用于 project-1 的 SVM 中创建。

```
volume show -vserver project-1-svm
```

6. 在 project-1 中创建 POD ，然后挂载上一步创建的 PVC 。

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  volumes:
    - name: test-pvc-project-1
      persistentVolumeClaim:
       claimName: test-pvc-project-1
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: test-pvc-project-1
EOF
```

7. 检查 POD 是否正在运行以及是否已挂载卷。

```
oc describe pods test-pvc-pod -n project-1
```

验证在其他项目中创建 **PVC** 或 **Pod** 的访问权限，或者使用专用于另一项目的资源

1. 以项目 1 中的 OCP-project-1-user 和开发人员身份登录。
2. 使用分配给 project-2 的 storageclass 在 project-1 中创建 PVC 。

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1-sc-2
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-2-sc
EOF
```

3. 在 project-2 中创建 PVC 。

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-2-sc-1
  namespace: project-2
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

4. 请确保未创建 PVC `test-vpa-project-1-sc-2` 和 `test-vpa-project-2-sc-1` 。

```
oc get pvc -n project-1
oc get pvc -n project-2
```

5. 在 project-2 中创建 POD 。

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
EOF
```

验证对查看和编辑项目， **ResourceQuotas** 和 **StorageClasses** 的访问权限

1. 以项目 1 中的 OCP-project-1-user 和开发人员身份登录。
2. 检查访问权限以创建新项目。

```
oc create ns sub-project-1
```

3. 验证对查看项目的访问权限。

```
oc get ns
```

4. 检查用户是否可以在 project-1 中查看或编辑 ResourceQuotas 。

```
oc get resourcequotas -n project-1
oc edit resourcequotas project-1-sc-rq -n project-1
```

5. 验证用户是否有权查看存储器。

```
oc get sc
```

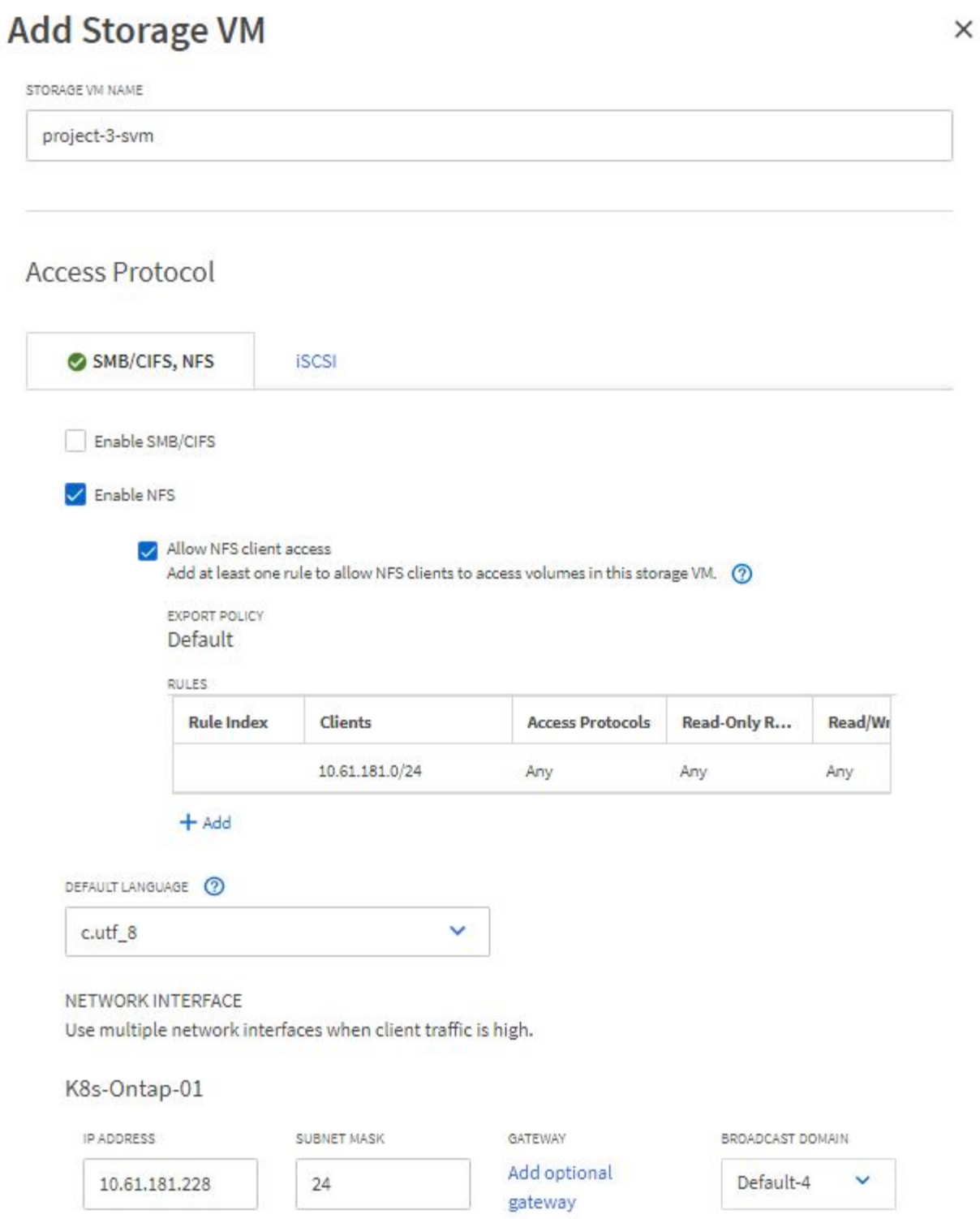
6. 检查访问权限以描述存储器。
7. 验证用户的访问权限以编辑存储器库。

```
oc edit sc project-1-sc
```

扩展：添加更多项目

在多租户配置中，使用存储资源添加新项目需要进行额外配置，以确保不会违反多租户要求。要在多租户集群中添加更多项目，请完成以下步骤：

1. 以存储管理员身份登录到 NetApp ONTAP 集群。
2. 导航到 `Storage → Storage VM`，然后单击 `Add`。创建一个专用于 project-3 的新 SVM 。此外，还可以创建 vsadmin 帐户来管理 SVM 及其资源。



1. 以集群管理员身份登录到 Red Hat OpenShift 集群。
2. 创建新项目。

```
oc create ns project-3
```

3. 确保已在 IdP 上为 project-3 创建用户组并与 OpenShift 集群同步。

```
oc get groups
```

4. 为 project-3 创建开发人员角色。

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-3
  name: developer-project-3
rules:
  - verbs:
      - '*'
    apiGroups:
      - apps
      - batch
      - autoscaling
      - extensions
      - networking.k8s.io
      - policy
      - apps.openshift.io
      - build.openshift.io
      - image.openshift.io
      - ingress.operator.openshift.io
      - route.openshift.io
      - snapshot.storage.k8s.io
      - template.openshift.io
    resources:
      - '*'
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - bindings
      - configmaps
      - endpoints
      - events
      - persistentvolumeclaims
      - pods
      - pods/log
      - pods/attach
      - podtemplates
```

```
      - replicationcontrollers
      - services
      - limitranges
      - namespaces
      - componentstatuses
      - nodes
  - verbs:
      - '*'
    apiGroups:
      - trident.netapp.io
    resources:
      - tridentsnapshots
EOF
```

本节中提供的角色定义只是一个示例。必须根据最终用户要求定义开发人员角色。

1. 在 project-3 中为开发人员创建 RoleBinding. 将开发人员项目 3 角色绑定到 project-3 中的相应组（ OCP-project-3 ）。

```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-3-developer
  namespace: project-3
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-3
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-3
EOF
```

2. 以存储管理员身份登录到 Red Hat OpenShift 集群
3. 创建 Trident 后端并将其映射到专用于 project-3 的 SVM 。NetApp 建议使用 SVM 的 vsadmin 帐户将后端连接到 SVM ，而不是使用 ONTAP 集群管理员。

```
cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_3",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.228",
    "svm": "project-3-svm",
    "username": "vsadmin",
    "password": "NetApp!23"
}
EOF
```

在此示例中，我们使用的是 ontap-NAS 驱动程序。使用相应的驱动程序根据使用情形创建后端。

我们假定 Trident 已安装在 Trident 项目中。

1. 为 project-3 创建存储类，并将其配置为使用专用于 project-3 的后端存储池。

```
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-3-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_3:.*"
EOF
```

2. 创建 ResourceQuota 以限制项目 3 中的资源，从而从专用于其他项目的存储库请求存储。

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-3-sc-rq
  namespace: project-3
spec:
  hard:
    project-1-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

3. 在其他项目中修补 ResourceQuotas ，以限制这些项目中的资源从专用于项目 3 的存储库访问存储。

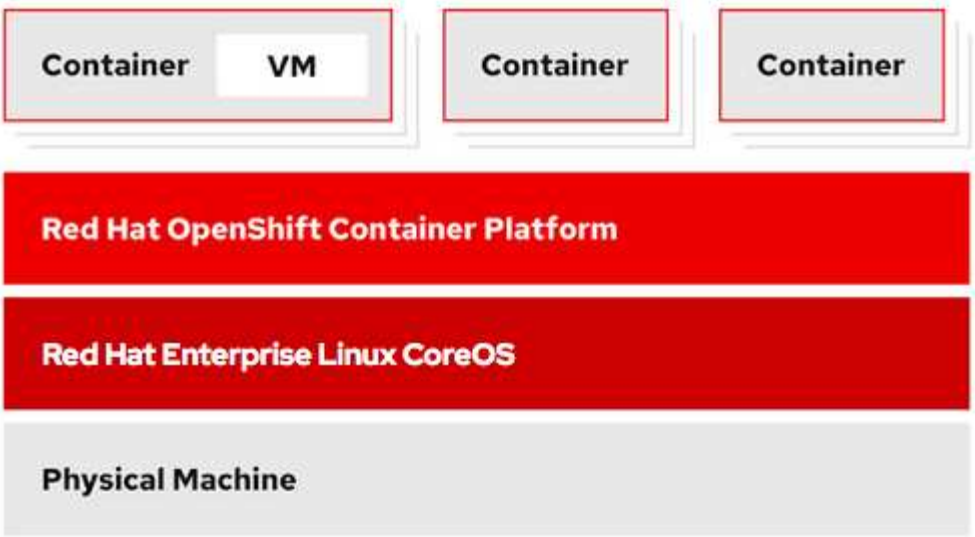
```
oc patch resourcequotas project-1-sc-rq -n project-1 --patch
'{"spec":{"hard":{ "project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
oc patch resourcequotas project-2-sc-rq -n project-2 --patch
'{"spec":{"hard":{ "project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
```

借助 NetApp ONTAP 实现 Red Hat OpenShift 虚拟化

借助 NetApp ONTAP 实现 Red Hat OpenShift 虚拟化

根据具体使用情形，容器和虚拟机（ VM ）均可用作不同类型应用程序的最佳平台。因此，许多组织在容器上运行部分工作负载，而在 VM 上运行部分工作负载。通常，这会导致企业面临额外的挑战，需要管理不同的平台：虚拟机管理程序和应用程序容器编排程序。

为了应对这一挑战， Red Hat 从 OpenShift 4.6 版开始引入了 OpenShift 虚拟化（以前称为容器原生虚拟化）。通过 OpenShift 虚拟化功能，您可以在同一 OpenShift 容器平台安装中运行和管理虚拟机以及容器，从而提供混合管理功能，通过操作员自动部署和管理 VM 。除了使用 OpenShift 虚拟化在 OpenShift 中创建 VM 之外，Red Hat 还支持从 VMware vSphere ， Red Hat 虚拟化和 Red Hat OpenStack Platform 部署导入 VM 。



在由 NetApp ONTAP 提供支持的 Astra Trident 的协助下， OpenShift 虚拟化还支持某些功能，例如实时 VM 迁移， VM 磁盘克隆， VM 快照等。这些工作流的示例将在本文档后面的相应章节中进行讨论。

要了解有关 Red Hat OpenShift 虚拟化的详细信息，请参见相关文档 "此处"。

部署OpenShift虚拟化

使用 **NetApp ONTAP** 部署 **Red Hat OpenShift** 虚拟化

前提条件

- Red Hat OpenShift 集群（ 4.6 版之后的版本），安装在具有 RHCOS 工作节点的裸机基础架构上
- OpenShift 集群必须通过安装程序配置的基础架构（ IPI ）进行安装
- 部署计算机运行状况检查以保持虚拟机的 HA
- NetApp ONTAP 集群
- 安装在 OpenShift 集群上的 Astra Trident
- 在 ONTAP 集群上配置了 SVM 的 Trident 后端
- 一种在 OpenShift 集群上配置的存储类，其中使用 Astra Trident 作为配置程序
- 对 Red Hat OpenShift 集群的集群管理员访问
- 对 NetApp ONTAP 集群的管理员访问权限
- 安装了 tridentctl 和 oc 工具并将其添加到 $path 中的管理工作站

由于 OpenShift 虚拟化由 OpenShift 集群上安装的操作员管理，因此会增加内存， CPU 和存储的开销，在规划集群的硬件要求时必须考虑这些开销。请参见文档 "此处" 有关详细信息：

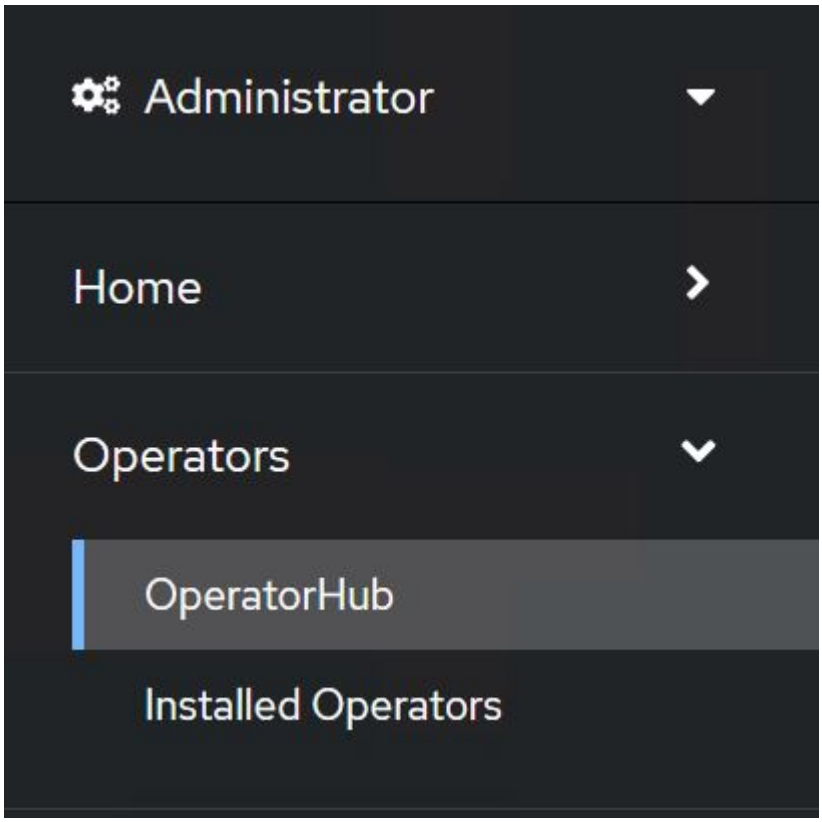
或者，您也可以通过配置节点放置规则来指定一组 OpenShift 集群节点，以托管 OpenShift 虚拟化操作员，控制器和 VM 。要为 OpenShift 虚拟化配置节点放置规则，请按照文档进行操作 "此处"。

对于支持 OpenShift 虚拟化的存储， NetApp 建议使用一个专用 StorageClass ，以便从特定 Trident 后端请求存储，而该后端又由专用 SVM 提供支持。这样就可以在 OpenShift 集群上为基于 VM 的工作负载提供的数据方面保持多租户级别。

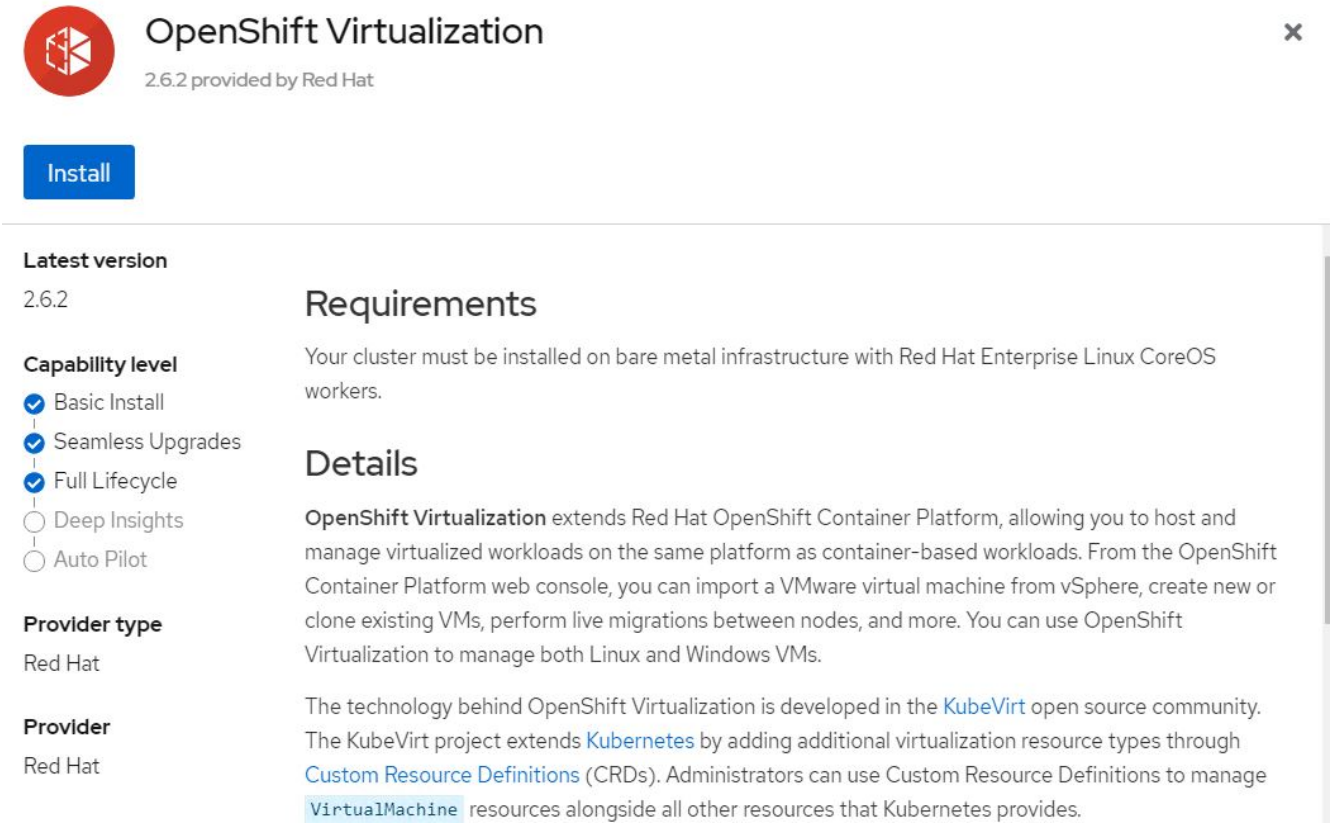
使用 **NetApp ONTAP** 部署 **Red Hat OpenShift** 虚拟化

要安装 OpenShift 虚拟化，请完成以下步骤：

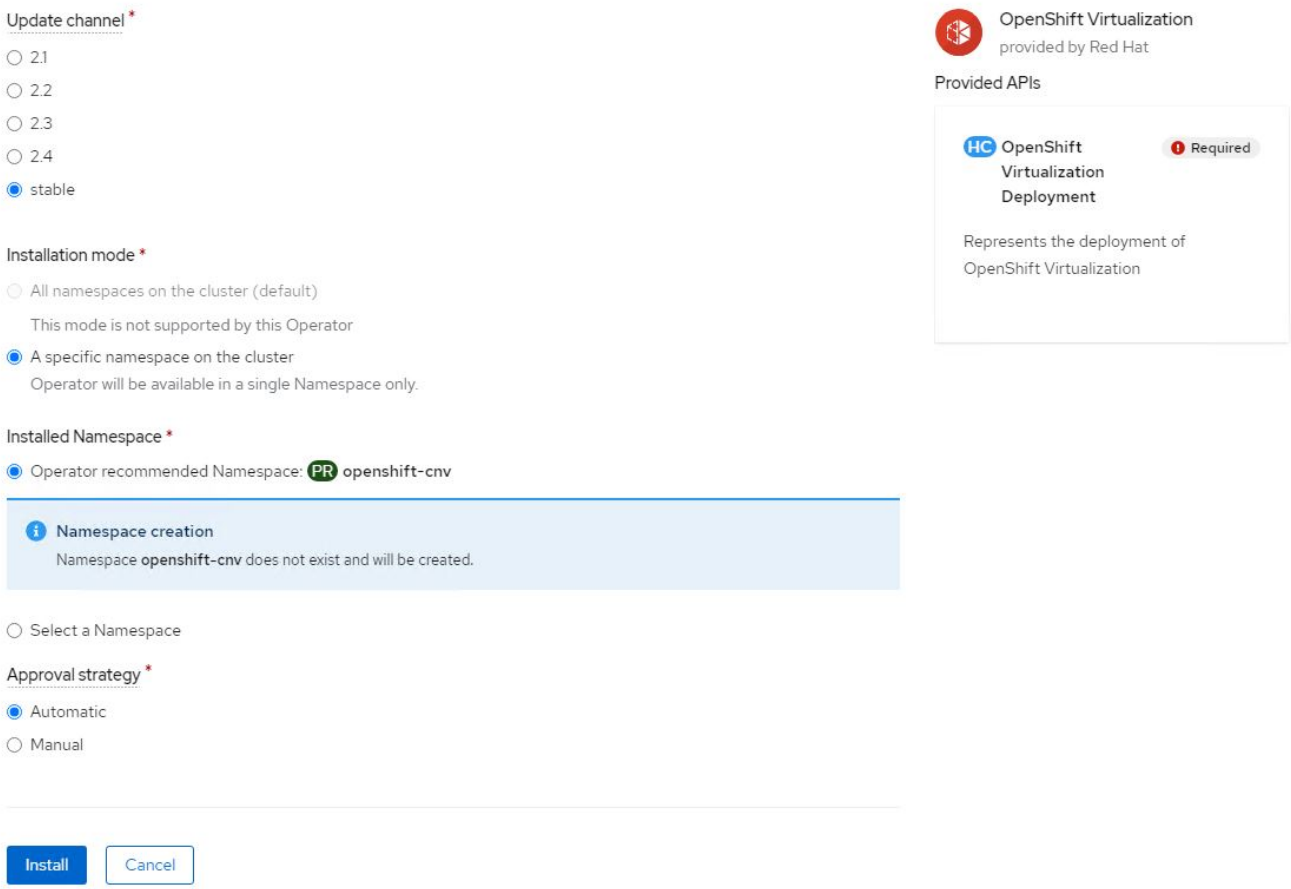
1. 使用 cluster-admin 访问权限登录到 Red Hat OpenShift 裸机集群。
2. 从 "Perspective" 下拉列表中选择 "Administrator" 。
3. 导航到 Operators > OperatorHub 并搜索 OpenShift 虚拟化。



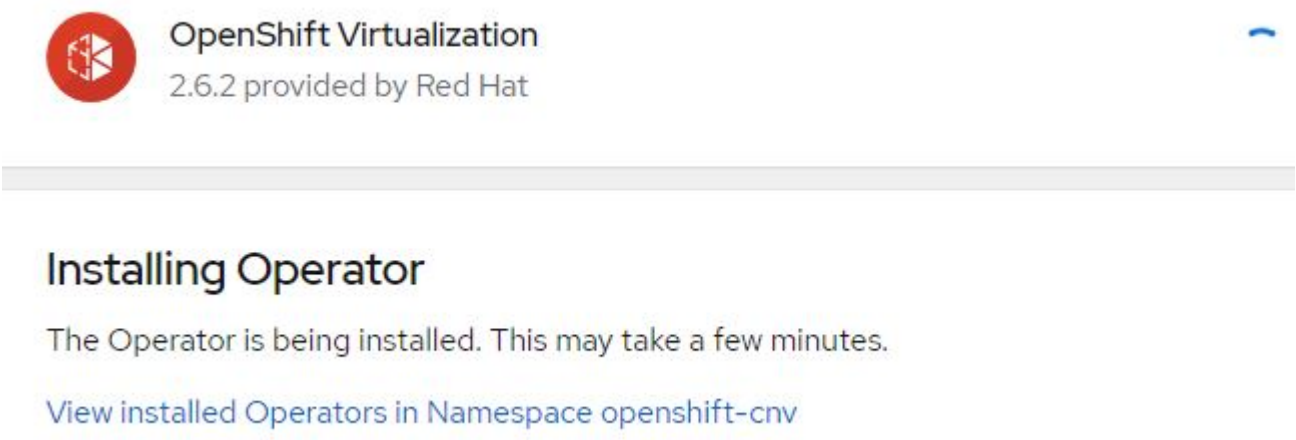
4. 选择 OpenShift 虚拟化磁贴，然后单击安装。



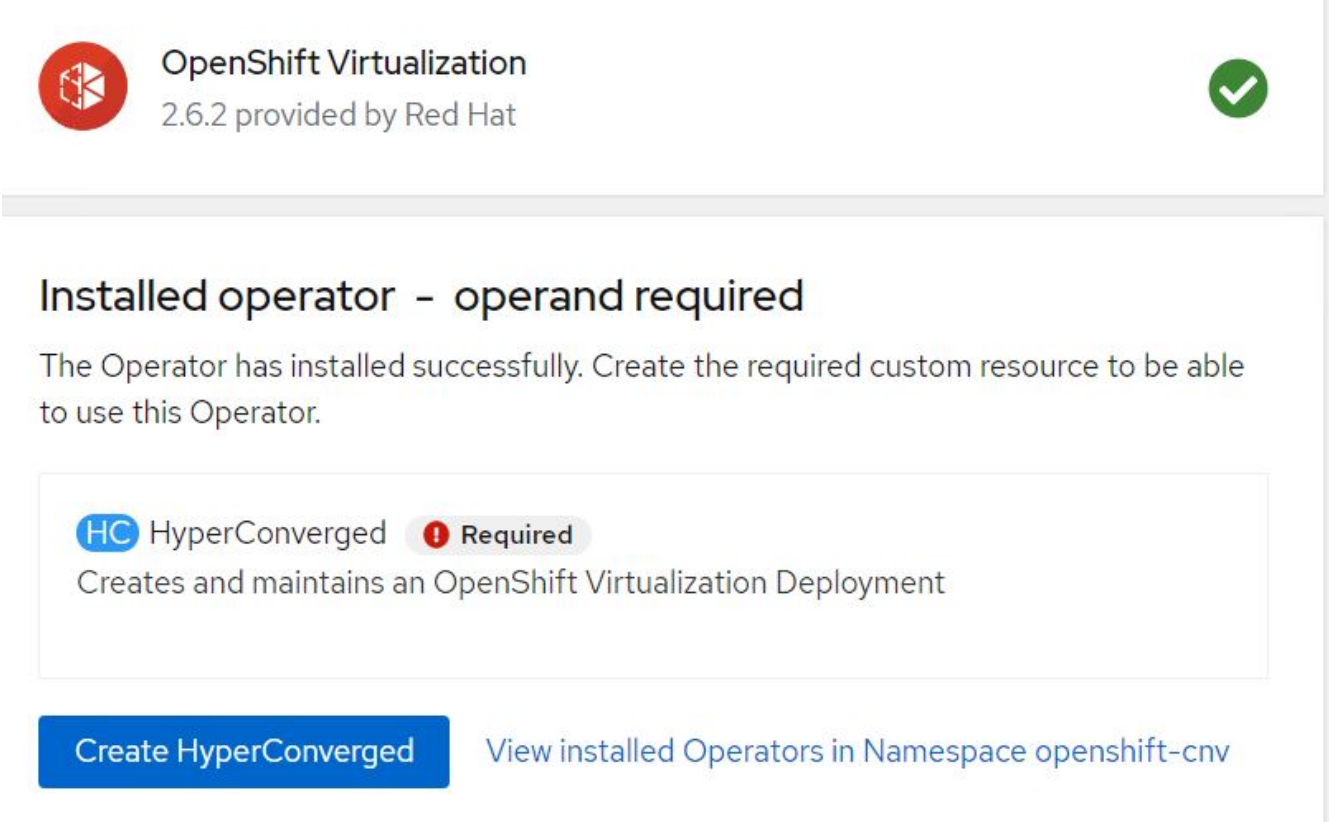
5. 在 Install Operator 屏幕上，保留所有默认参数，然后单击 Install 。



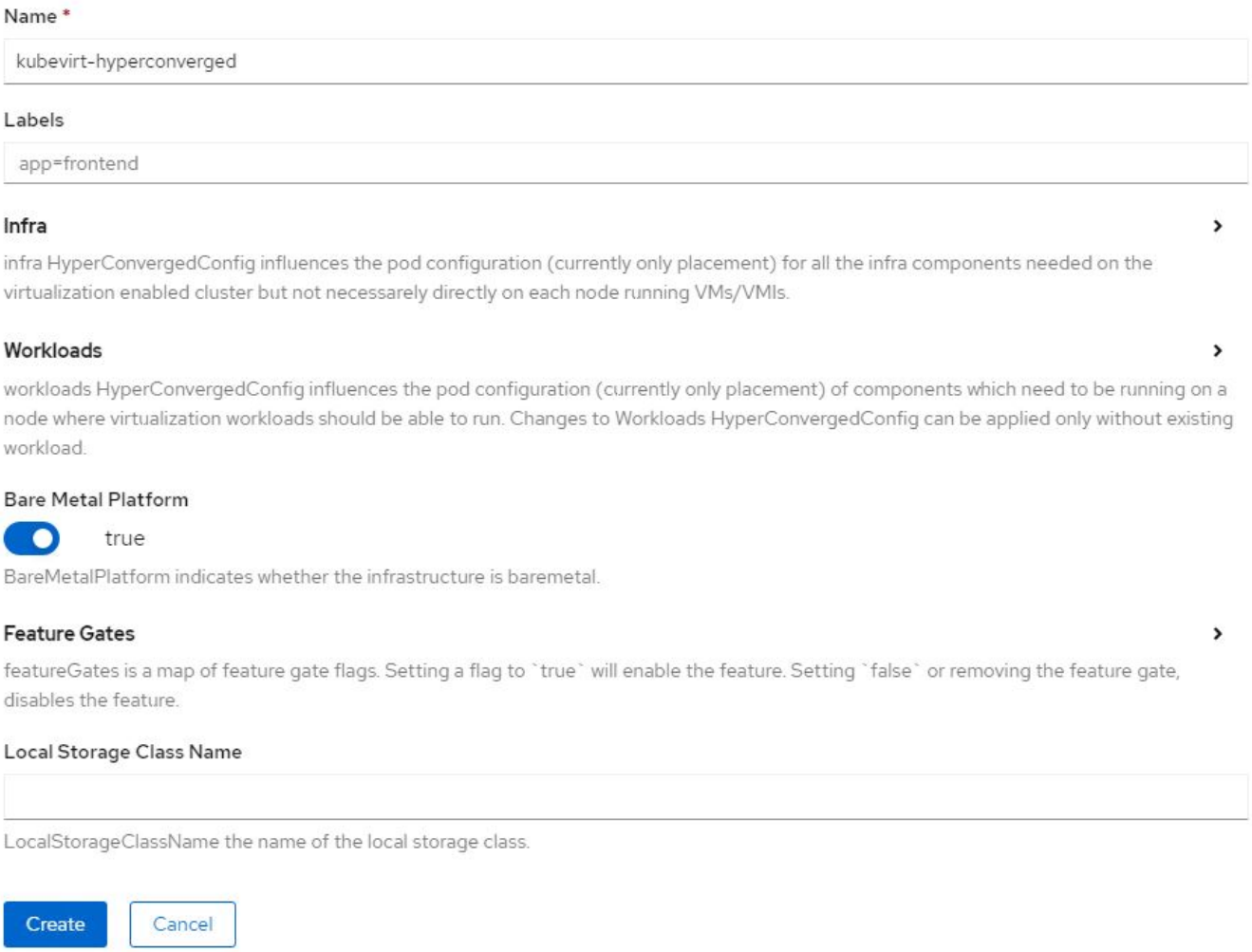
6. 等待操作员安装完成。



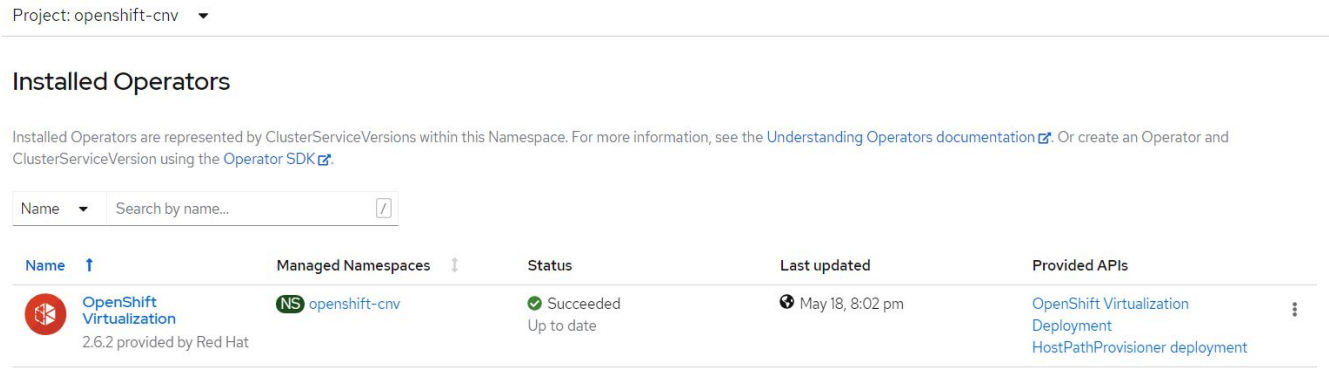
7. 安装操作员后，单击 Create HyperConverged 。



8. 在 Create HyperConverged 屏幕上，单击 Create ，接受所有默认参数。此步骤将开始安装 OpenShift 虚拟化。



9. 在 OpenShift-cnv 命名空间中的所有 Pod 都变为 running 状态且 OpenShift 虚拟化操作员处于 succeeded 状态后，操作员便可随时使用了。现在，可以在 OpenShift 集群上创建 VM 。



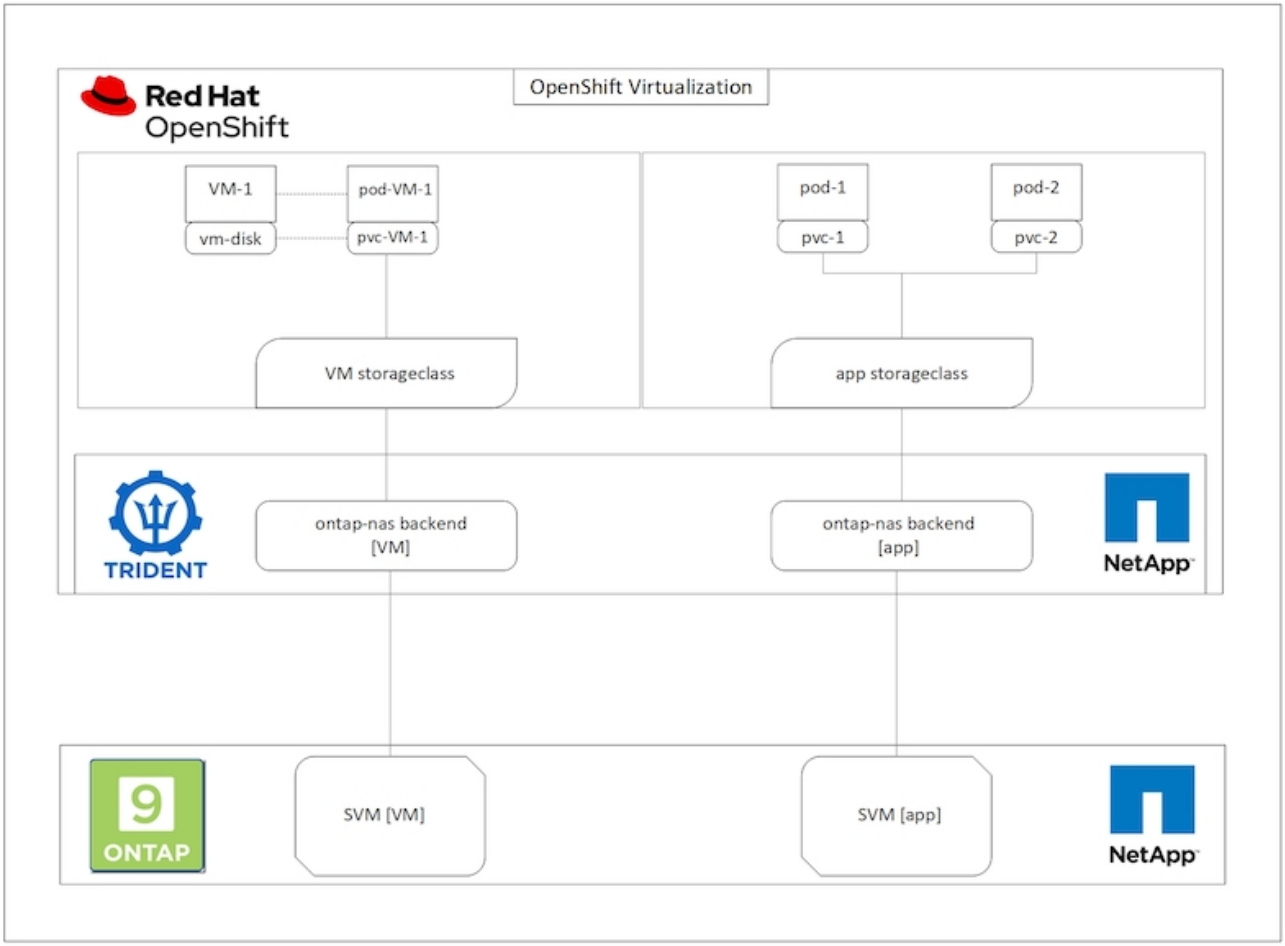
工作流

工作流：使用 **NetApp ONTAP** 实现 **Red Hat OpenShift** 虚拟化

创建虚拟机

VM 是有状态部署，需要使用卷来托管操作系统和数据。使用 CNV 时，由于 VM 作为 Pod 运行，因此 VM 由 NetApp ONTAP 上通过 Trident 托管的 PV 提供支持。这些卷作为磁盘连接并存储整个文件系统，包括虚拟机的

启动源。



要在 OpenShift 集群上创建虚拟机，请完成以下步骤：

1. 导航到工作负载 > 虚拟化 > 虚拟机，然后单击创建 > 使用向导。
2. 选择所需的操作系统，然后单击下一步。
3. 如果选定操作系统未配置启动源，则必须对其进行配置。对于启动源，选择是要从 URL 还是从注册表导入操作系统映像，并提供相应的详细信息。展开高级并选择 Trident 支持的 StorageClass 。然后单击下一步。

Boot source

This template does not have a boot source. Provide a custom boot source for this **CentOS 8.0+ VM** virtual machine.

Boot source type *

Import via URL (creates PVC)

Import URL *

https://access.cdn.redhat.com/content/origin/files/sha256/58/588167f828001e57688ec4b9b31c11a59d532489f527488ebc89ac5e952...

Example: For RHEL, visit the **RHEL download page** (requires login) and copy the download link URL of the KVM guest image

☑ Mount this as a CD-ROM boot source

Persistent Volume Claim size *

5 GiB

Ensure your PVC size covers the requirements of the uncompressed image and any other space requirements. More storage can be added later.

Advanced

Storage class *

basic (default)

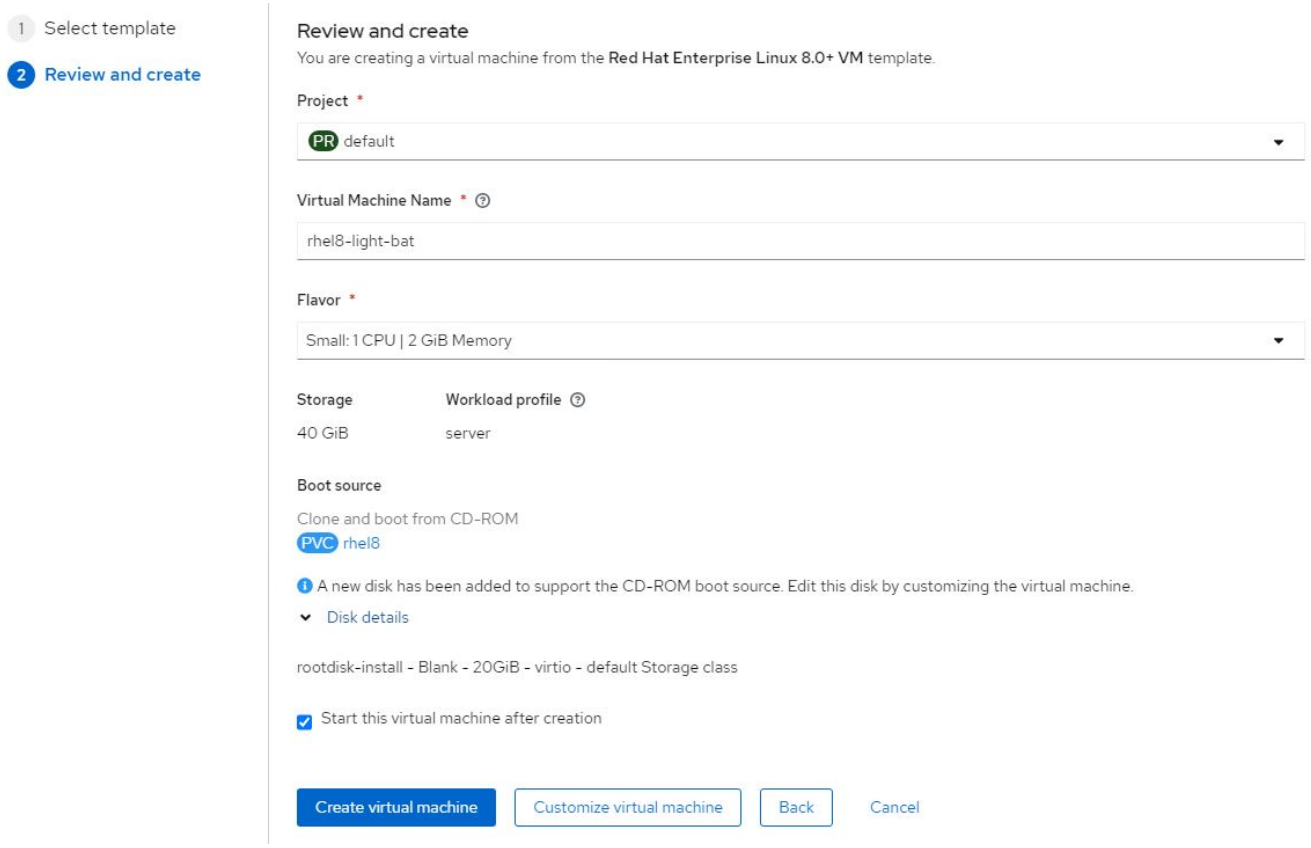
Access mode *

Single User (RWO)

Volume mode *

Filesystem

4. 如果选定操作系统已配置启动源，则可以跳过上一步。
5. 在 Review and Create 窗格中，选择要在其中创建 VM 的项目并提供 VM 详细信息。确保选择了要克隆的启动源，并使用为选定操作系统分配的相应 PVC 从 CD-ROM 启动。



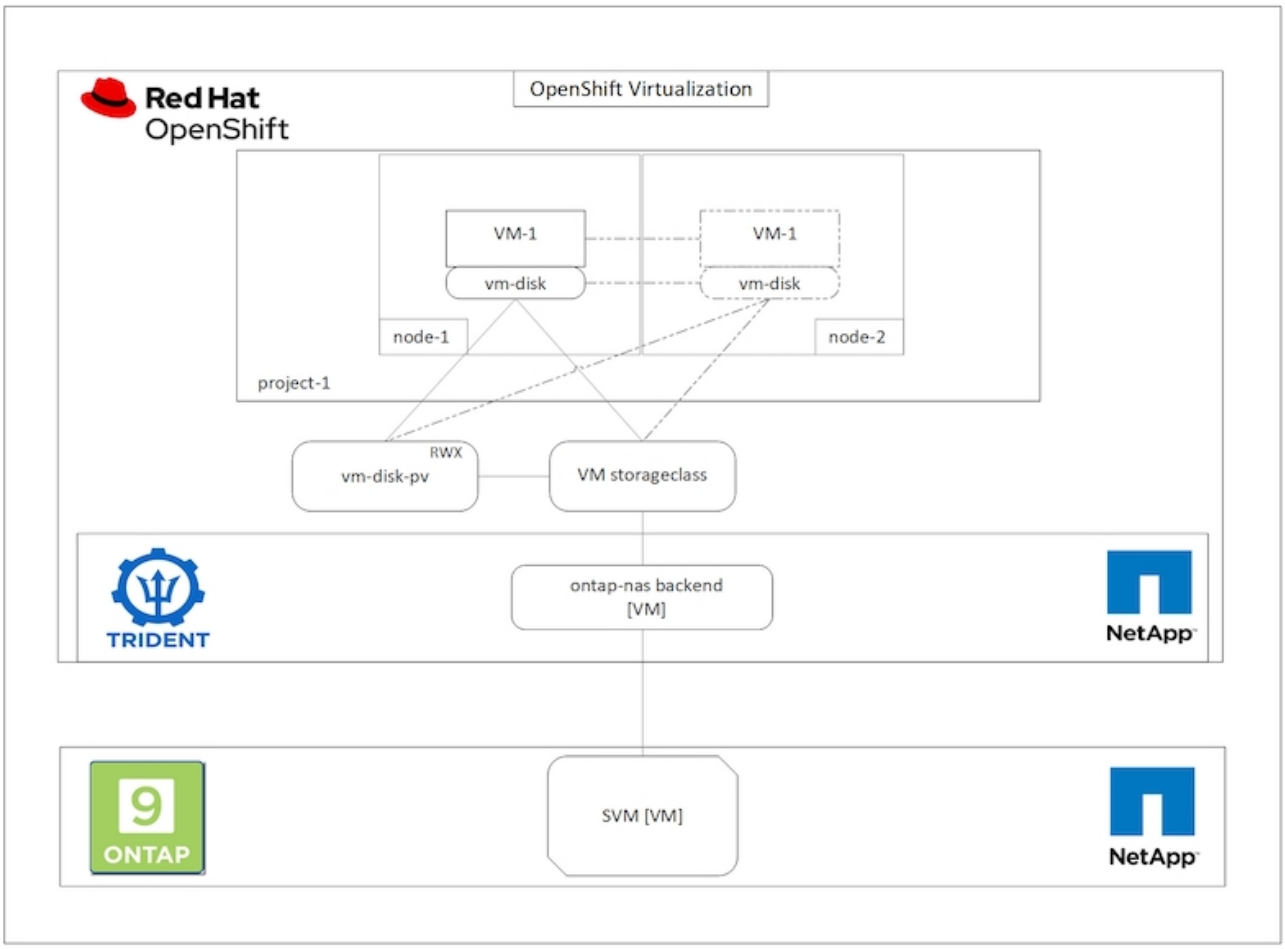
6. 如果要自定义虚拟机，请单击 "Customize Virtual Machine" 并修改所需的参数。
7. 单击 Create Virtual Machine 以创建虚拟机；此操作将在后台生成相应的 Pod 。

从 URL 或注册表为模板或操作系统配置启动源时，它会在 `OpenShift-virtual-os-images` 项目中创建一个 PVC ，并将 KVM 子映像下载到 PVC 。您必须确保模板 PVC 具有足够的已配置空间，以容纳相应操作系统的 KVM 子映像。然后，使用任何项目中的相应模板创建这些 PVC 时，这些 PVC 会克隆并作为根磁盘附加到虚拟机中。

工作流：使用 **NetApp ONTAP** 实现 **Red Hat OpenShift** 虚拟化

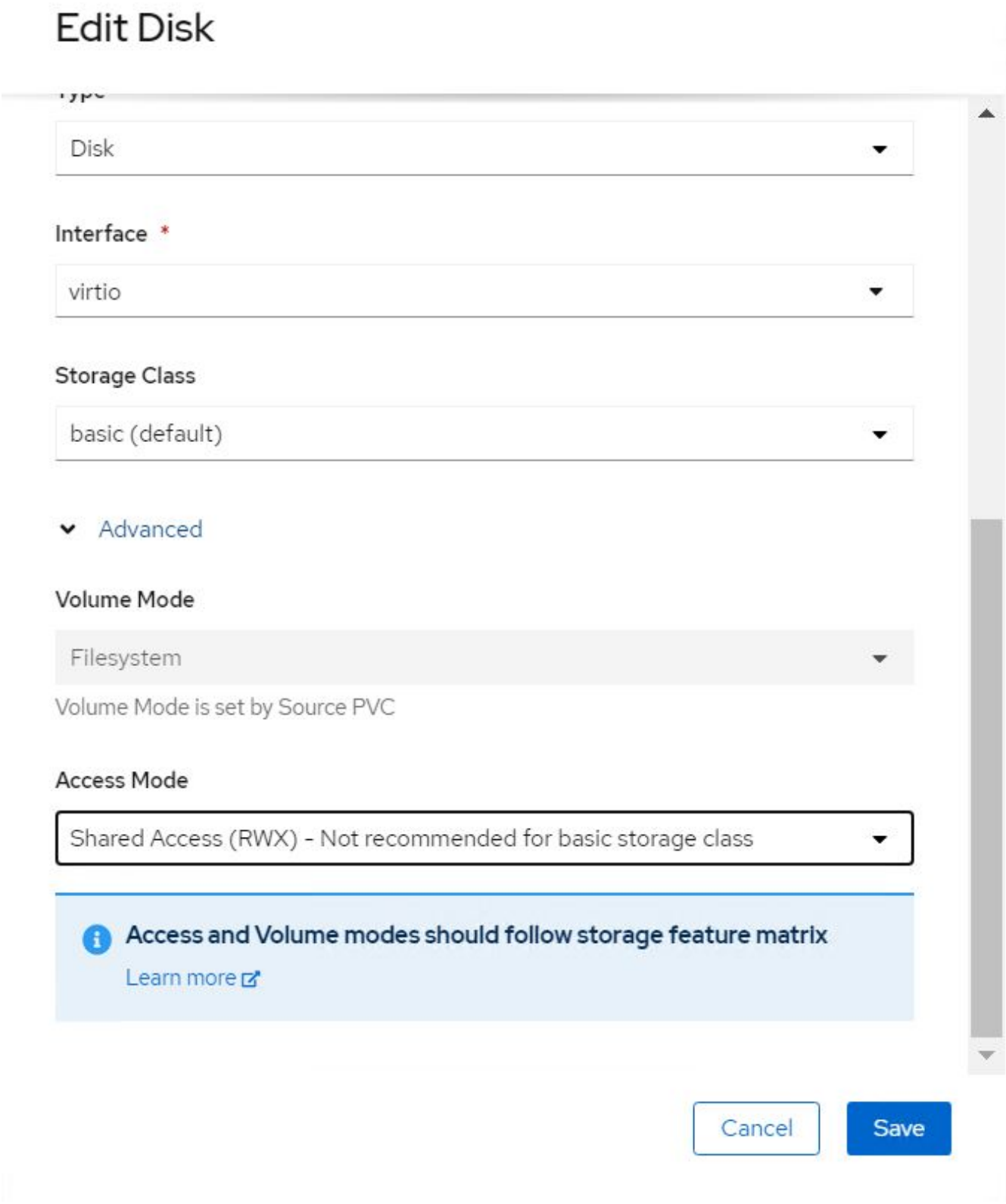
VM 实时迁移

实时迁移是指在不停机的情况下将 VM 实例从 OpenShift 集群中的一个节点迁移到另一个节点的过程。要在 OpenShift 集群中执行实时迁移， VM 必须绑定到具有共享 ReadWriteMany 访问模式的 PVC 。在启用了 NFS 协议的 NetApp ONTAP 集群上配置了 SVM 的 Astra Trident 后端支持对 PVC 的共享 ReadWriteMany 访问。因此，对于从启用了 NFS 的 SVM 中由 Trident 配置的 StorageClasses 请求具有 PVC 的 VM ，可以在不停机的情况下进行迁移。



要创建绑定到具有共享 ReadWriteMany 访问权限的 PVC 的 VM ，请执行以下操作：

1. 导航到工作负载 > 虚拟化 > 虚拟机，然后单击创建 > 使用向导。
2. 选择所需的操作系统，然后单击下一步。假设选定操作系统已配置了启动源。
3. 在 Review and Create 窗格中，选择要在其中创建 VM 的项目并提供 VM 详细信息。确保选择了要克隆的启动源，并使用为选定操作系统分配的相应 PVC 从 CD-ROM 启动。
4. 单击自定义虚拟机，然后单击存储。
5. 单击 rootdisk 旁边的省略号，并确保已选择使用 Trident 配置的 storageclass 。展开高级，然后为访问模式选择共享访问（ rwx ）。然后单击保存。



6. 单击 Review 并确认，然后单击 Create Virtual Machine 。

要手动将虚拟机迁移到 OpenShift 集群中的另一个节点，请完成以下步骤。

1. 导航到工作负载 > 虚拟化 > 虚拟机。

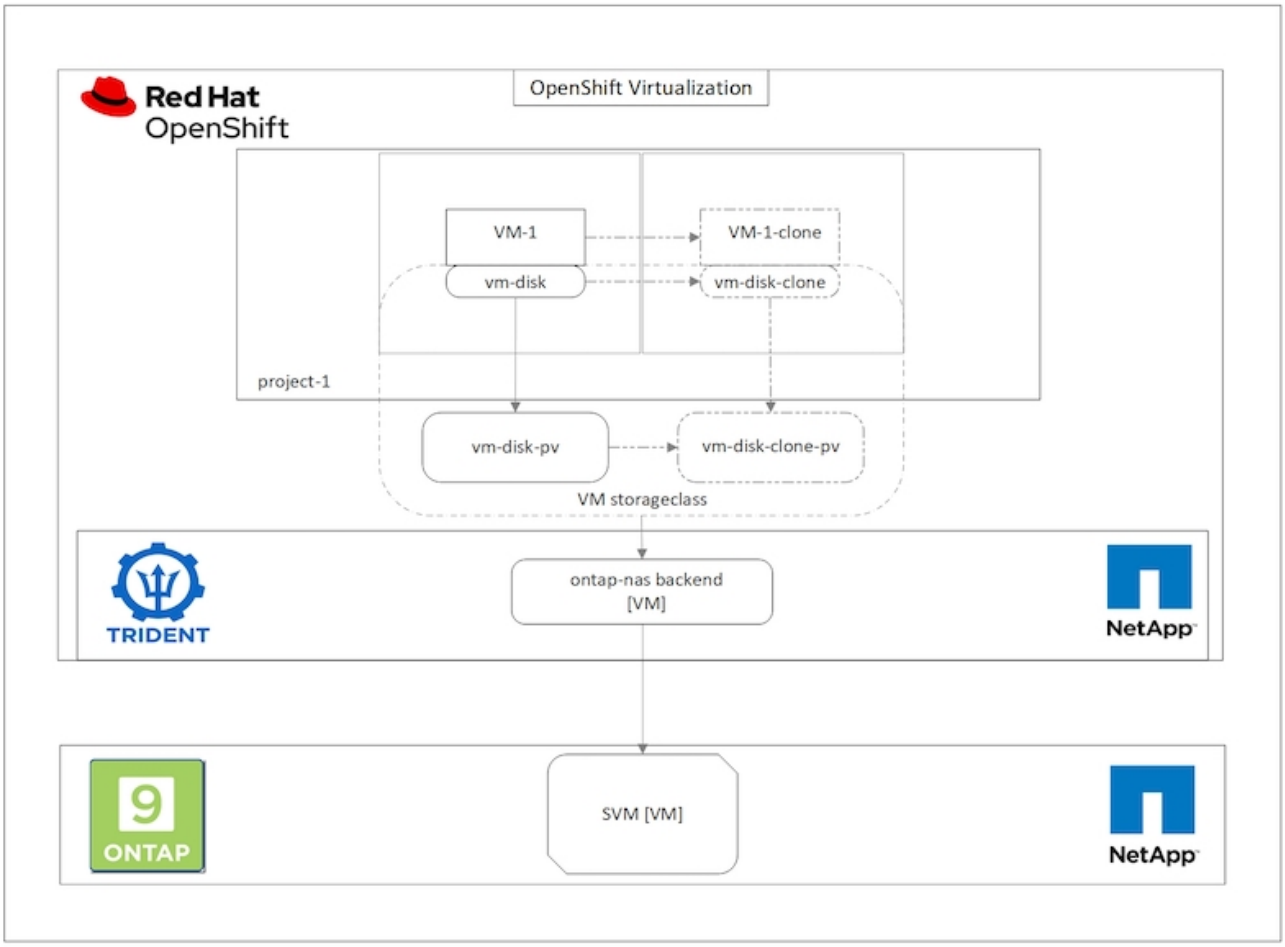
2. 对于要迁移的虚拟机，单击省略号，然后单击迁移虚拟机。
3. 当消息弹出时，单击迁移进行确认。

> 如果 evictionStrategy 设置为 LiveMigrate ，则在将原始节点置于维护模式时， OpenShift 集群中的 VM 实例会自动迁移到另一节点。

工作流：使用 **NetApp ONTAP** 实现 **Red Hat OpenShift** 虚拟化

VM 克隆

通过支持 Astra Trident 的卷 CSI 克隆功能，可以在 OpenShift 中克隆现有虚拟机。通过 CSI 卷克隆，可以使用现有 PVC 作为数据源并通过复制其 PV 来创建新的 PVC 。创建新的 PVC 后，它将作为一个单独的实体运行，并且不会与源 PVC 建立任何链接或依赖关系。



要考虑 CSI 卷克隆的某些限制：

1. 源 PVC 和目标 PVC 必须位于同一项目中。
2. 在同一存储类中支持克隆。
3. 只有当源卷和目标卷使用相同的卷模式设置时，才能执行克隆；例如，一个块卷只能克隆到另一个块卷。

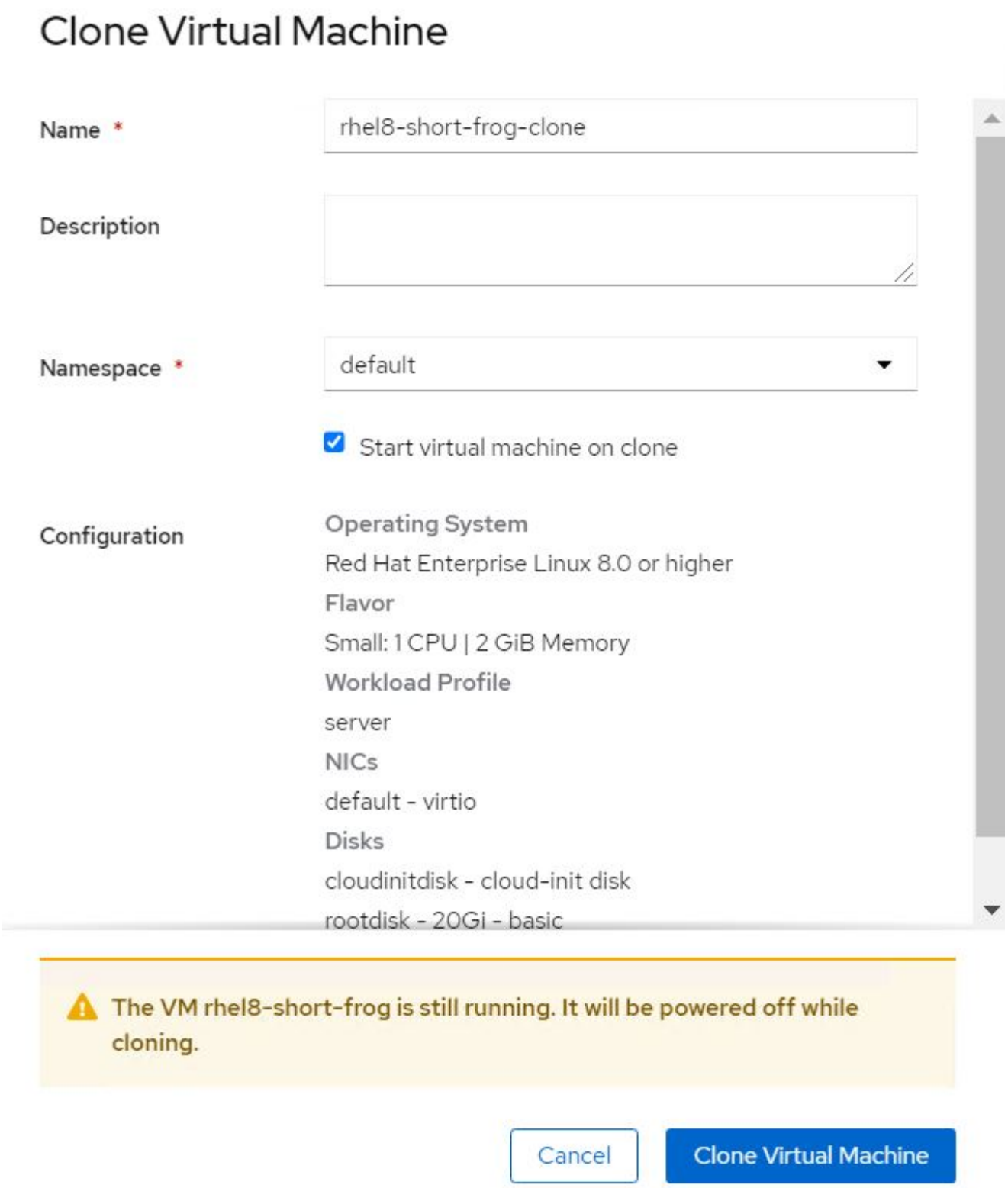
可以通过两种方式克隆 OpenShift 集群中的 VM ：

1. 关闭源 VM
2. 使源 VM 保持活动状态

关闭源 **VM**

通过关闭虚拟机克隆现有虚拟机是一项原生 OpenShift 功能，该功能在 Astra Trident 的支持下实施。要克隆虚拟机，请完成以下步骤。

1. 导航到工作负载 > 虚拟化 > 虚拟机，然后单击要克隆的虚拟机旁边的省略号。
2. 单击克隆虚拟机并提供新虚拟机的详细信息。

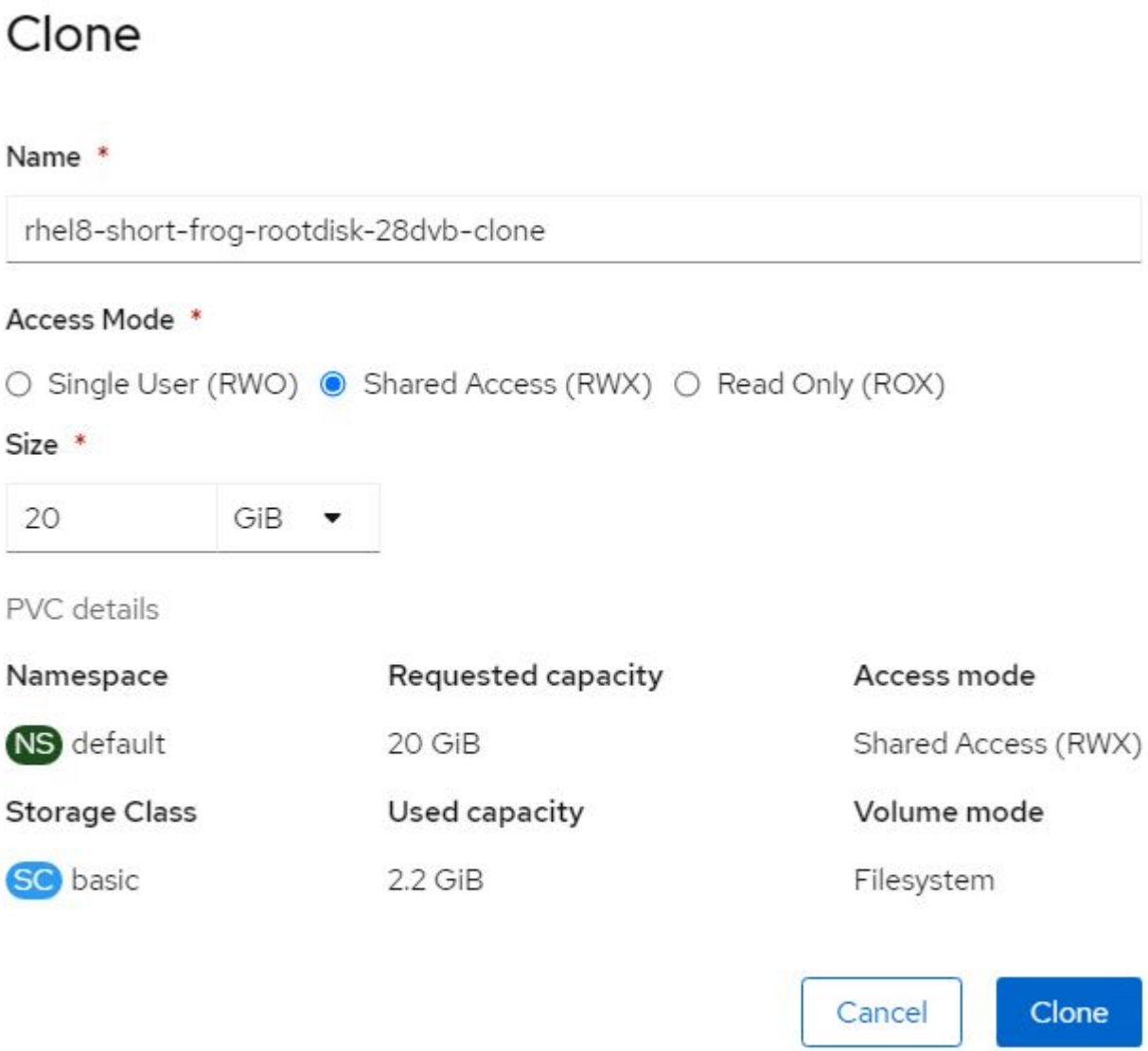


3. 单击克隆虚拟机；此操作将关闭源 VM 并启动克隆 VM 的创建。
4. 完成此步骤后，您可以访问并验证克隆的虚拟机的内容。

使源 **VM** 保持活动状态

也可以通过克隆源 VM 的现有 PVC ，然后使用克隆的 PVC 创建新 VM 来克隆现有 VM 。此方法不需要关闭源 VM 。要克隆虚拟机而不关闭它，请完成以下步骤。

1. 导航到 "Storage">"PersistentVolumeClass" ，然后单击附加到源 VM 的 PVC 旁边的省略号。
2. 单击克隆 PVC 并提供新 PVC 的详细信息。



3. 然后单击克隆。这样就会为新虚拟机创建一个 PVC 。
4. 导航到工作负载 > 虚拟化 > 虚拟机，然后单击创建 > 使用 YAML 。
5. 在规范 > 模板 > 规范 > 卷部分中，附加克隆的 PVC ，而不是容器磁盘。根据您的要求提供新虚拟机的所有其他详细信息。

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvb-clone
```

6. 单击创建以创建新虚拟机。
7. 成功创建 VM 后，访问并验证新 VM 是否为源 VM 的克隆。

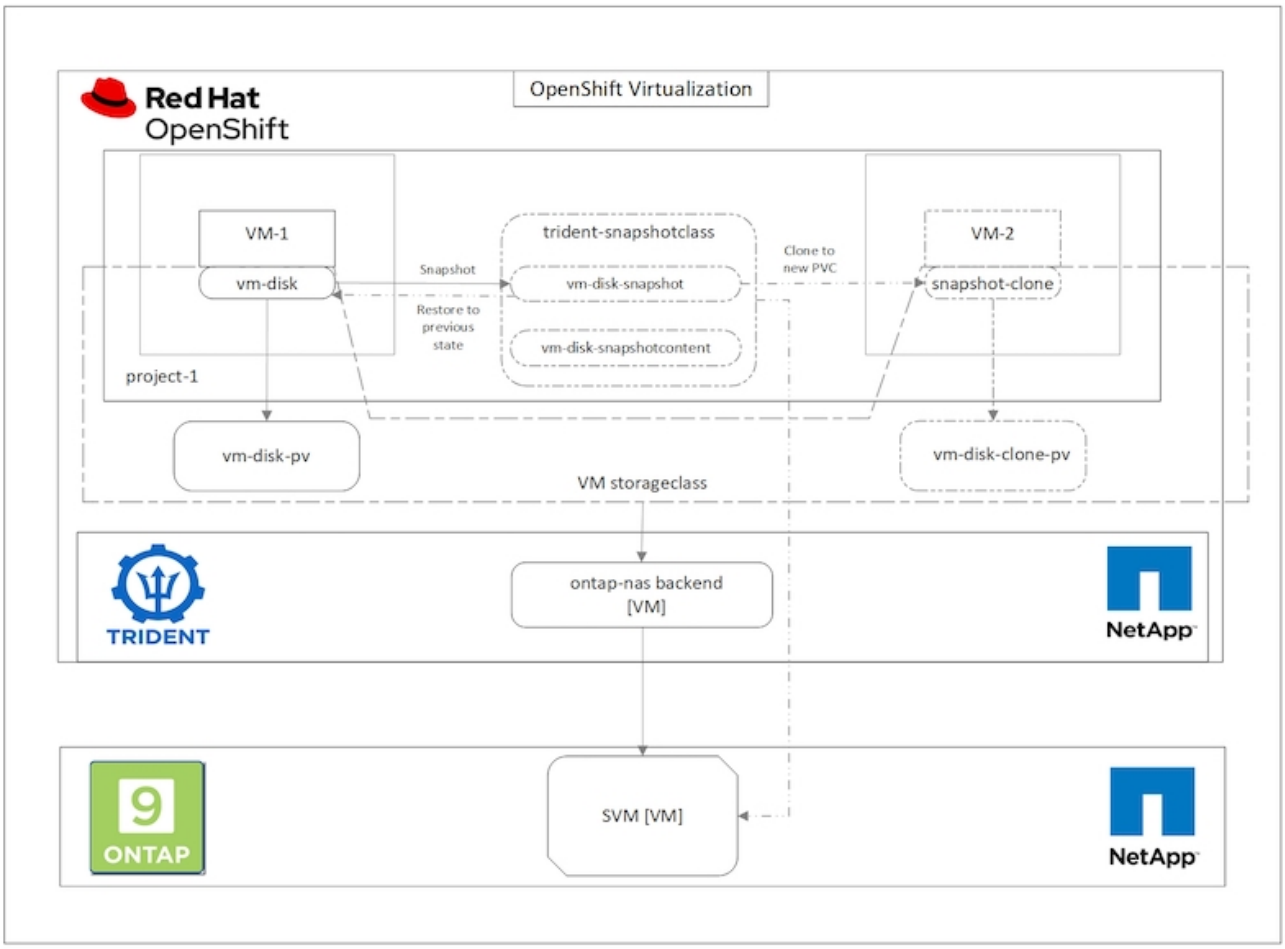
工作流：使用 **NetApp ONTAP** 实现 **Red Hat OpenShift** 虚拟化

从 **Snapshot** 创建 **VM**

借助 Astra Trident 和 Red Hat OpenShift ，用户可以在所配置的存储类上创建永久性卷的快照。通过此功能，用户可以创建卷的时间点副本，并使用该副本创建新卷或将同一卷还原到先前的状态。这样可以启用或支持从回滚到克隆再到数据还原等各种使用情形。

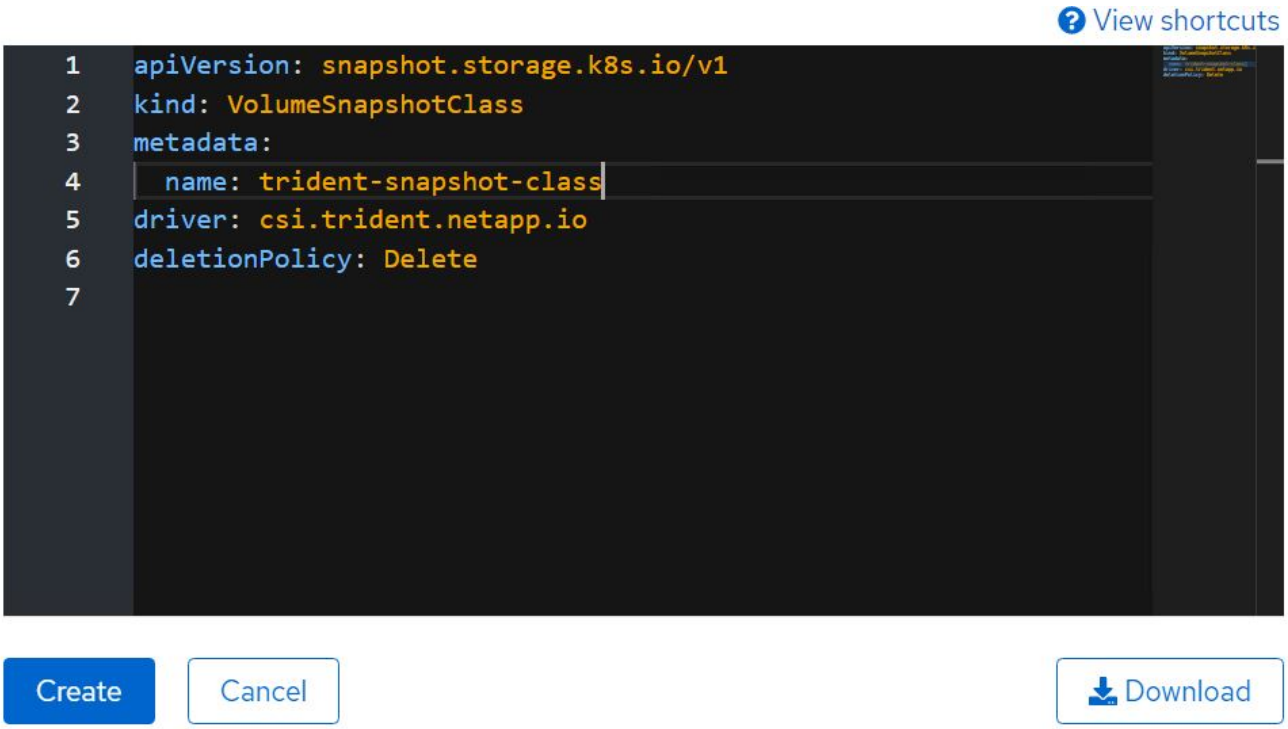
对于 OpenShift 中的 Snapshot 操作，必须定义资源 VolumeSnapshotClass ， VolumeSnapshot 和 VolumeSnapshotContent 。

- VolumeSnapshotContent 是从集群中的卷生成的实际快照。它是一种集群范围的资源，类似于用于存储的 PersistentVolume 。
- VolumeSnapshot 是指创建卷快照的请求。它类似于 PersistentVolumeClaim 。
- 管理员可以使用 VolumeSnapshotClass 为 VolumeSnapshot 指定不同的属性。通过此选项，您可以为从同一卷创建的不同快照设置不同的属性。

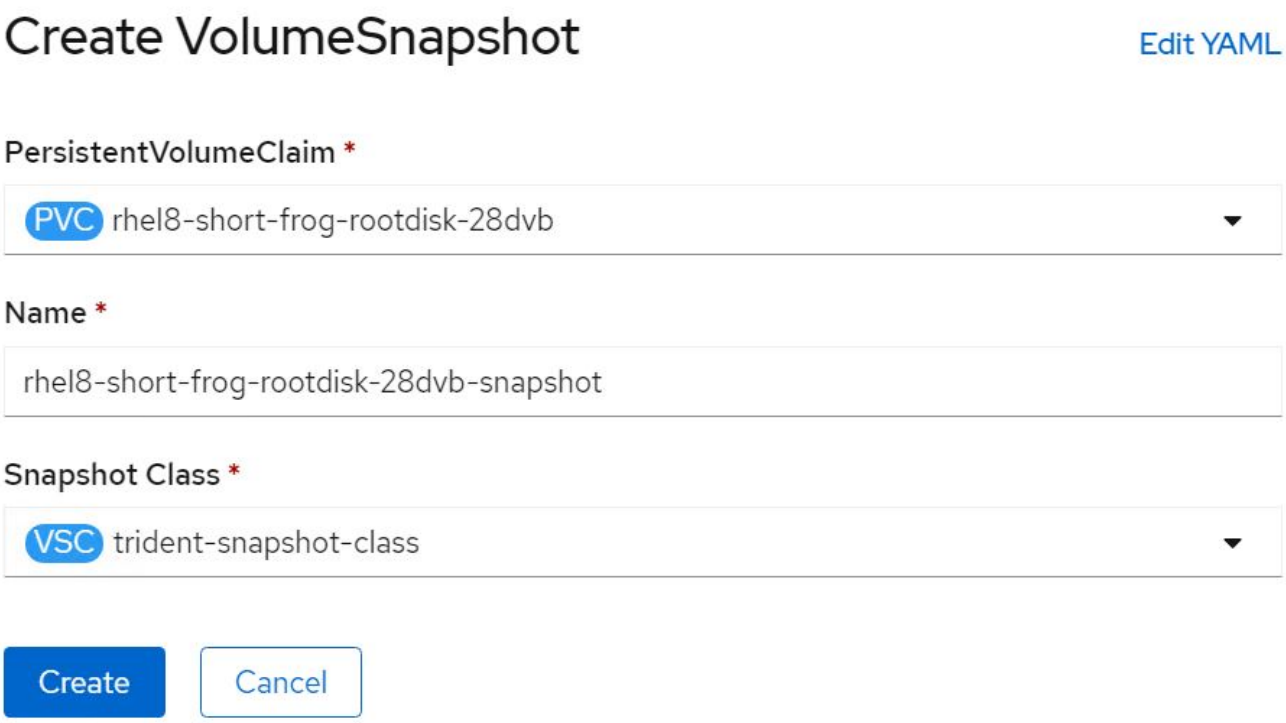


要创建虚拟机的 Snapshot ，请完成以下步骤：

1. 创建 VolumeSnapshotClass ，然后使用该类创建 VolumeSnapshot 。导航到 "Storage">"VolumeSnapshotClasss" ，然后单击 "Create VolumeSnapshotClass" 。
2. 输入 Snapshot 类的名称，输入驱动程序的 csi.trident.netapp.io ，然后单击创建。



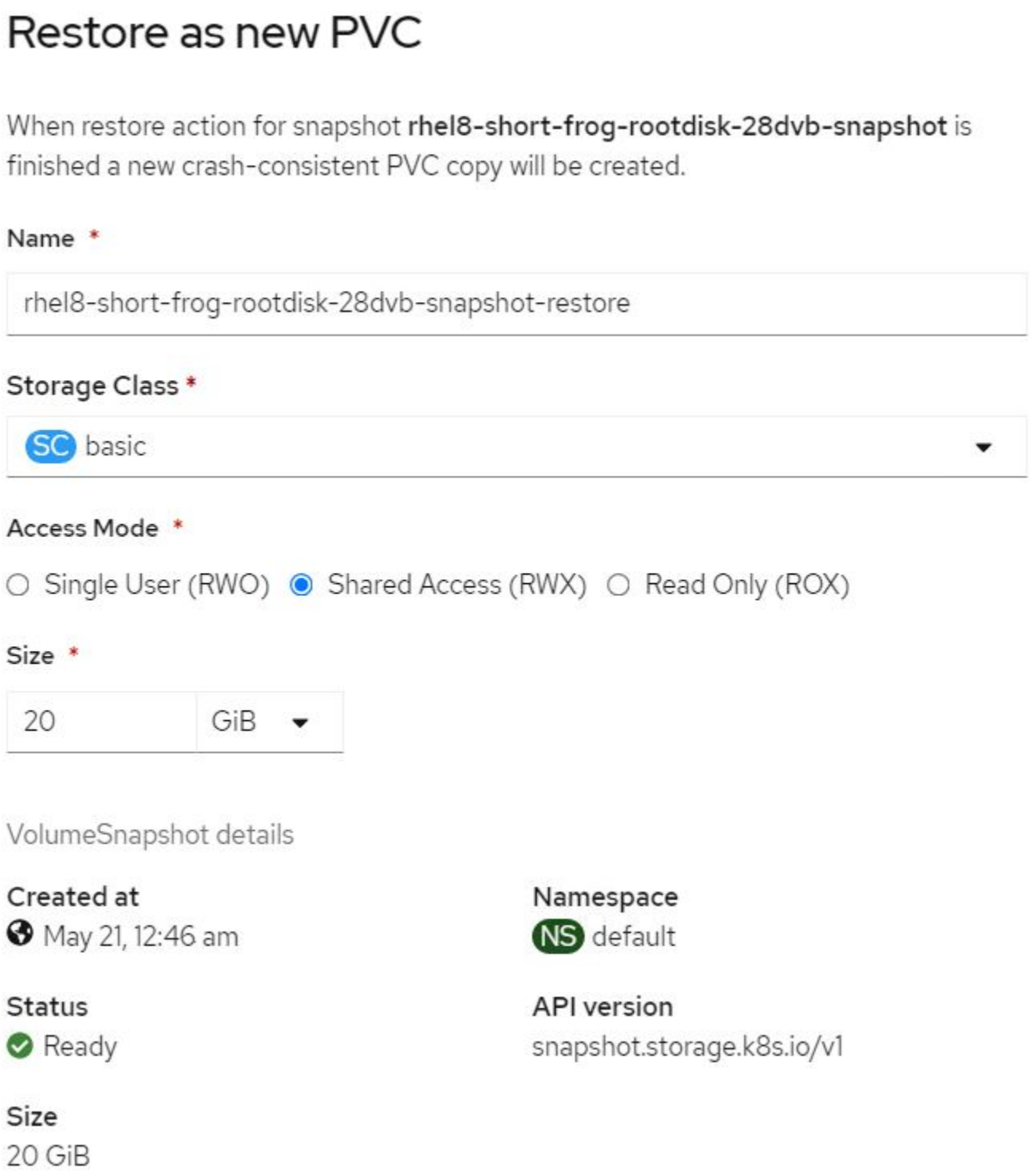
3. 确定连接到源 VM 的 PVC ，然后创建该 PVC 的 Snapshot 。导航到 `Storage > VolumeSnapshots` ，然后单击 Create VolumeSnapshots 。
4. 选择要为其创建 Snapshot 的 PVC ，输入 Snapshot 的名称或接受默认值，然后选择相应的 VolumeSnapshotClass 。然后单击创建。



5. 此时将创建 PVC 的快照。

从快照创建新虚拟机

1. 首先，将 Snapshot 还原到新的 PVC 中。导航到存储 > 卷快照，单击要还原的快照旁边的省略号，然后单击还原为新 PVC 。
2. 输入新 PVC 的详细信息，然后单击还原。这样就会创建一个新的 PVC 。



3. 接下来，使用此 PVC 创建一个新虚拟机。导航到工作负载 > 虚拟化 > 虚拟机，然后单击创建 > 使用 YAML 。

4. 在规范 > 模板 > 规范 > 卷部分中，指定从 Snapshot 创建的新 PVC ，而不是从容器磁盘创建的新 PVC 。根据您的要求提供新虚拟机的所有其他详细信息。

```
- name: rootdisk
  persistentVolumeClaim:
    claimName: rhel8-short-frog-rootdisk-28dvb-snapshot-restore
```

5. 单击创建以创建新虚拟机。
6. 成功创建虚拟机后，访问并验证新虚拟机的状态是否与创建快照时使用 PVC 创建快照的虚拟机的状态相同。

工作流：使用 **NetApp ONTAP** 实现 **Red Hat OpenShift** 虚拟化

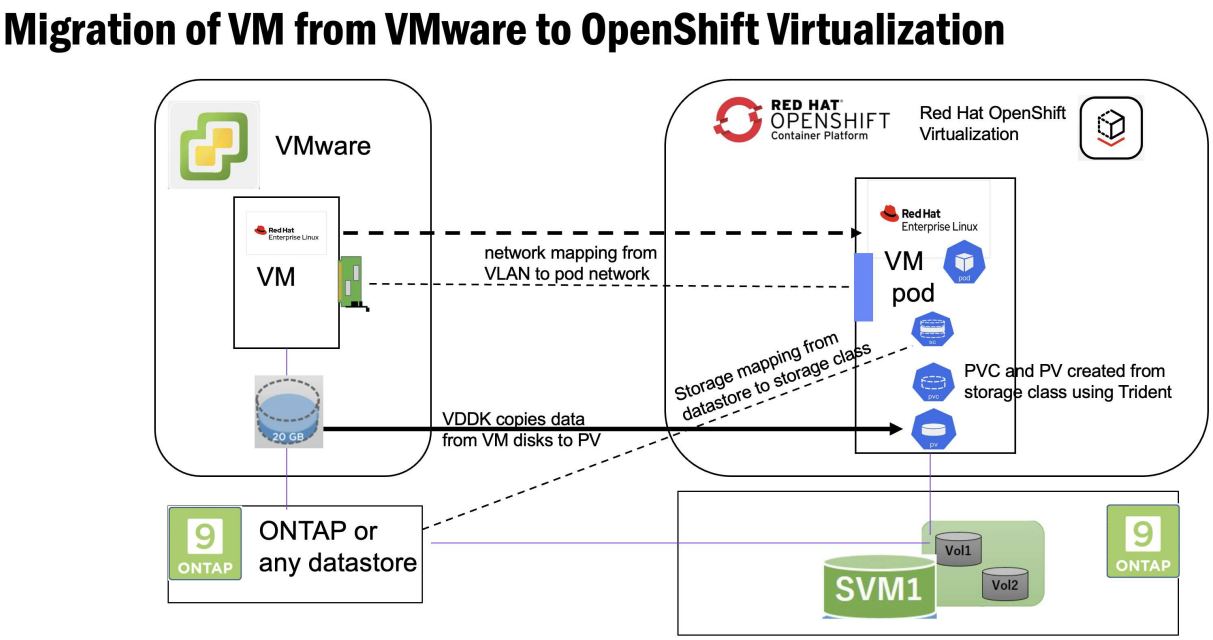
使用适用于虚拟化的迁移工具包将**VM**从**VMware**迁移到**OpenShift**虚拟化

在本节中、我们将了解如何使用虚拟化迁移工具包(Migration Toolkit for Virtualization、Mtv)将虚拟机从VMware迁移到在OpenShift容器平台上运行并使用Asta Trident与NetApp ONTAP存储集成的OpenShift虚拟化。

以下视频演示了如何使用ONTAP SAN将RHEL VM从VMware迁移到OpenShift虚拟化以实现永久性存储。

使用Red Hat VtTM通过NetApp ONTAP存储将VM迁移到OpenShift虚拟化

下图简要展示了将VM从VMware迁移到Red Hat OpenShift虚拟化的过程。



迁移示例的前提条件

在**VMware**上

- 安装了一个使用RHEL 9.3的RHEL 9 VM、并具有以下配置：

- CPU：2、内存：20 GB、硬盘：20 GB
- 用户凭据：root用户和管理员用户凭据

- 虚拟机准备就绪后、安装了PostgreSQL服务器。
  - PostgreSQL服务器已启动并启用、可在启动时启动

```
systemctl start postgresql.service`
systemctl enable postgresql.service
The above command ensures that the server can start in the VM in
OpenShift Virtualization after migration
```

  - 添加了2个数据库、其中添加了1个表和1行。请参见 "此处" 有关在RHEL上安装PostgreSQL服务器以及创建数据库和表条目的说明、请参见。

> 确保启动PostgreSQL服务器并启用服务以在启动时启动。

在**OpenShift**集群上

在安装此版本之前、已完成以下安装：

- OpenShift集群4.13.34
- "Astra三打23.10."
- 为iSCSI启用的集群节点上的多路径(对于ONONTAP SAN存储类)。请参见提供的YAML以创建一个守护进程集、以便在集群中的每个节点上启用iSCSI。
- 使用iSCSI的ONTAP SAN的三端和存储类。请参见为三元后端和存储类提供的YAML文件。
- "OpenShift 虚拟化"

要在OpenShift集群节点上安装iSCSI和多路径、请使用下面提供的YAML文件
为**iSCSI**准备群集节点

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  namespace: trident
  name: trident-iscsi-init
  labels:
    name: trident-iscsi-init
spec:
  selector:
    matchLabels:
      name: trident-iscsi-init
  template:
    metadata:
      labels:
```

```
        name: trident-iscsi-init
    spec:
      hostNetwork: true
      serviceAccount: trident-node-linux
      initContainers:
      - name: init-node
        command:
          - nsenter
          - --mount=/proc/1/ns/mnt
          - --
          - sh
          - -c
        args: ["$(STARTUP_SCRIPT)"]
        image: alpine:3.7
        env:
        - name: STARTUP_SCRIPT
          value: |
            #! /bin/bash
            sudo yum install -y lsscsi iscsi-initiator-utils sg3_utils
device-mapper-multipath
            rpm -q iscsi-initiator-utils
            sudo sed -i 's/^\(node.session.scan\).*/\1 = manual/'
/etc/iscsi/iscsid.conf
            cat /etc/iscsi/initiatorname.iscsi
            sudo mpathconf --enable --with_multipathd y --find_multipaths
n
            sudo systemctl enable --now iscsid multipathd
            sudo systemctl enable --now iscsi
        securityContext:
          privileged: true
      hostPID: true
      containers:
      - name: wait
        image: k8s.gcr.io/pause:3.1
      hostPID: true
      hostNetwork: true
      tolerations:
      - effect: NoSchedule
        key: node-role.kubernetes.io/master
  updateStrategy:
    type: RollingUpdate
```

使用以下YAML文件创建使用ONTAP SAN存储的三元后端配置
iSCSI的三端

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-san-secret
type: Opaque
stringData:
  username: <username>
  password: <password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-san
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <management LIF>
  backendName: ontap-san
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-san-secret
```

使用以下YAML文件创建要使用ONTAP SAN存储的三元存储类配置
用于**iSCSI**的三级存储类

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-san
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
allowVolumeExpansion: true
```

安装**Mst**

现在、您可以安装适用于虚拟化的迁移工具包(Migration Toolkit for Virtualization、简称为迁移工具包)。请参阅提供的说明 "此处" 有关安装的帮助。

虚拟化迁移工具包(Migration Toolkit for Virtualization、Tmb)用户界面集成到OpenShift Web控制台中。
您可以参考 "此处" 开始使用用户界面执行各种任务。

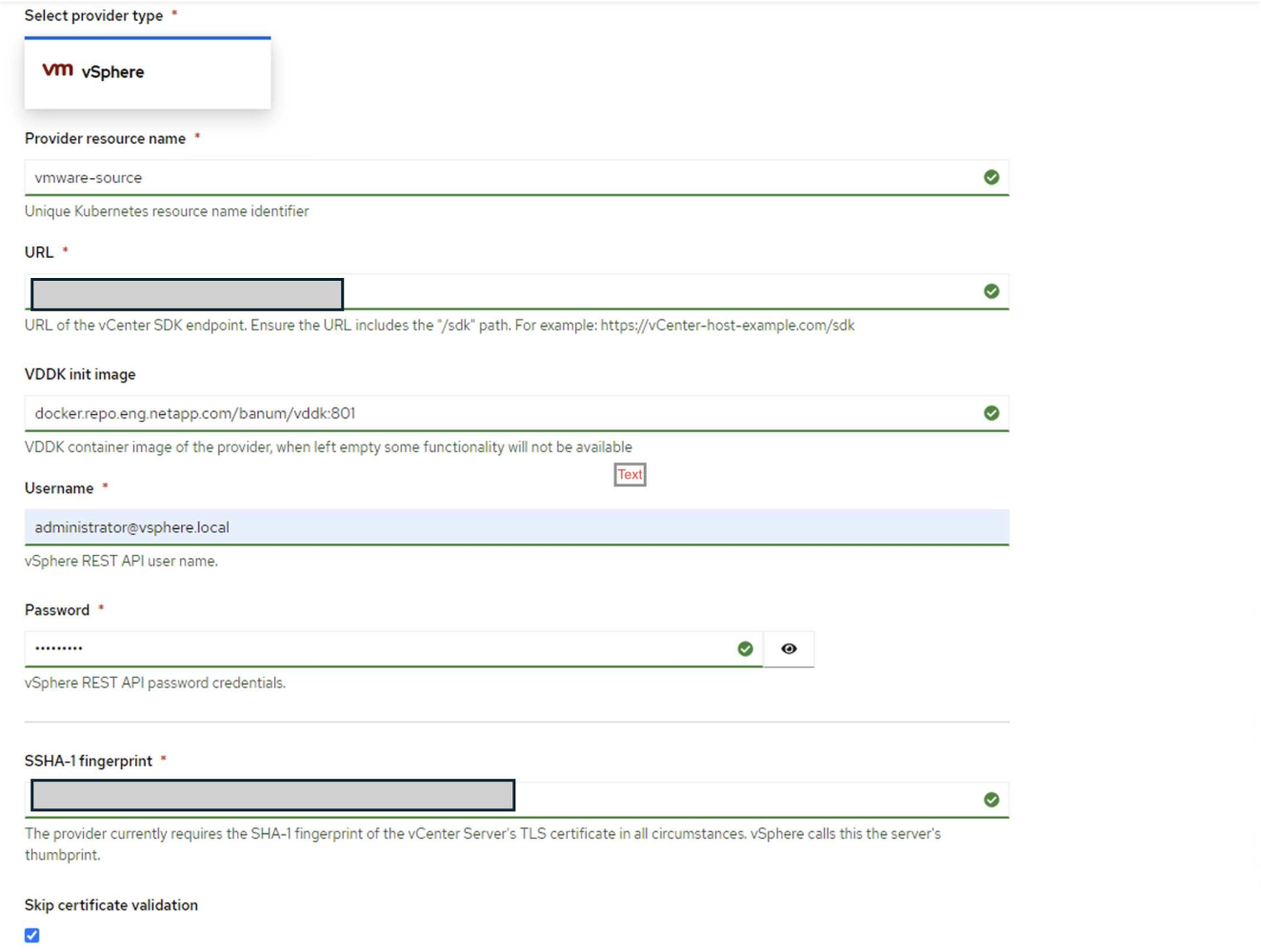
创建源提供程序

要将RHEL VM从VMware迁移到OpenShift虚拟化、您需要先为VMware创建源提供程序。请参阅说明 "此处" 以创建源提供程序。

要创建VMware源提供程序、您需要满足以下条件：

- vCenter URL
- vCenter凭据
- vCenter Server指纹
- 存储库中的VDDK映像

创建源提供程序的示例：



> 虚拟化迁移工具包(Migration Toolkit for Virtualization、Mv）使用VMware虚拟磁盘开发工具包(Virtual Disk Development Kit、VDDK) SDK来加快从VMware vSphere传输虚拟磁盘的速度。因此、强烈建议创建VDDK映像、尽管这是可选的。
> 要使用此功能、请下载VMware虚拟磁盘开发工具包(VDDK)、构建VDDK映像、然后将VDDK映像推送到映像注册表。

按照提供的说明进行操作 "此处" 创建VDDK映像并将其推送到可从OpenShift集群访问的注册表。

创建目标提供程序

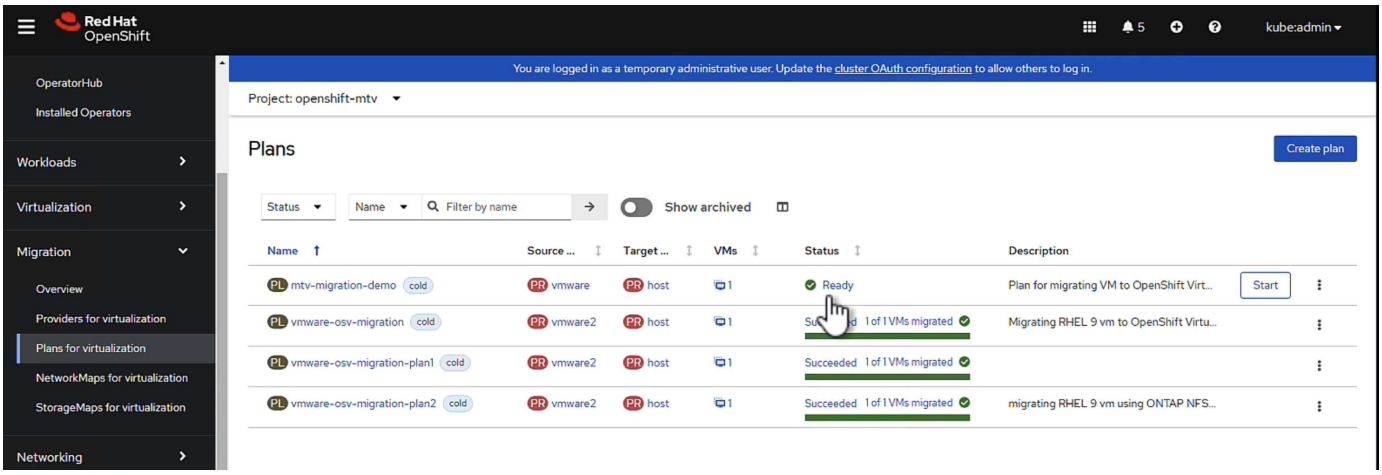
由于OpenShift虚拟化提供程序是源提供程序、因此会自动添加主机集群。

创建迁移计划

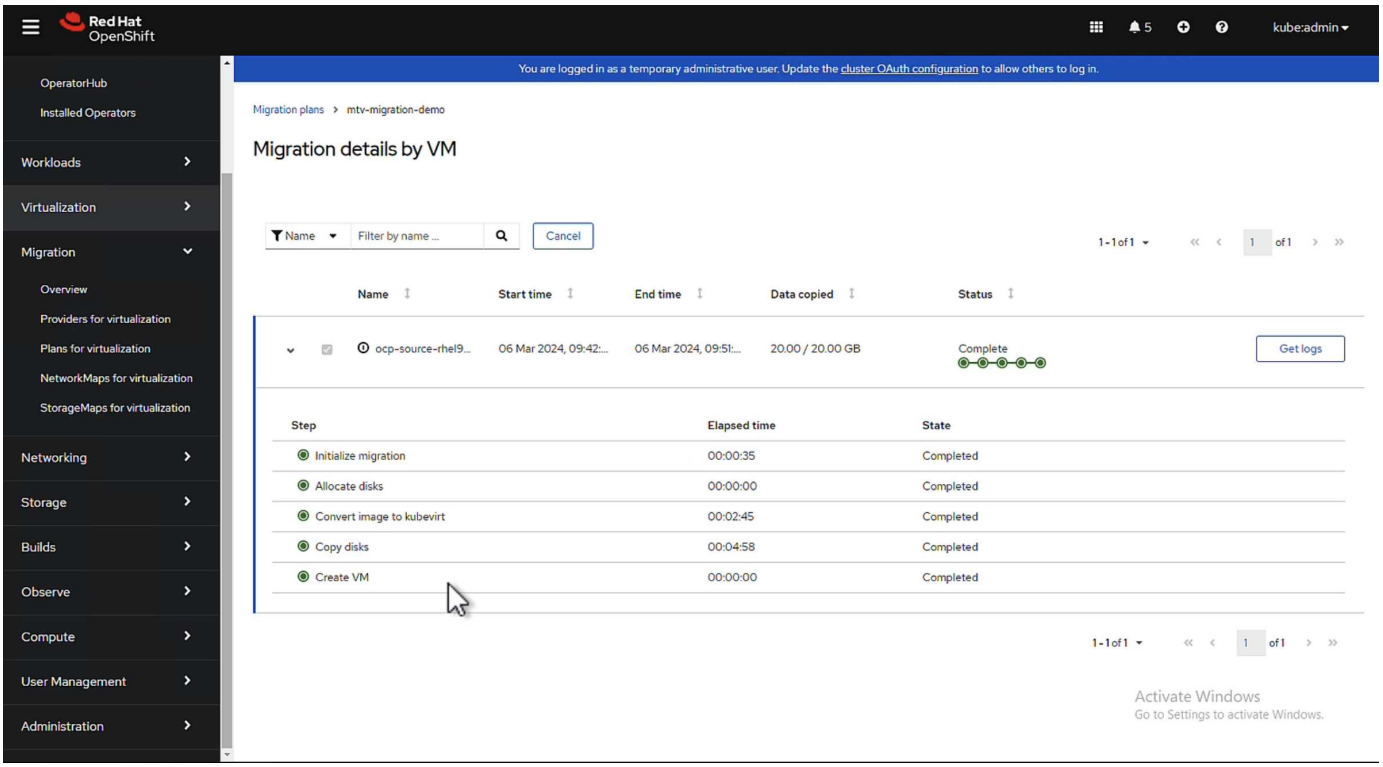
按照提供的说明进行操作 "此处" 以创建迁移计划。

创建计划时，如果尚未创建，则需要创建以下内容：

- 用于将源网络映射到目标网络的网络映射。
- 用于将源数据存储库映射到目标存储类的存储映射。为此、您可以选择ONTAP SAN存储类。
  创建迁移计划后，该计划的状态应显示*Ready*，现在您应该能够*Start*该计划。



单击*Start*将运行一系列步骤来完成虚拟机的迁移。



完成所有步骤后，您可以通过单击左侧导航菜单中“Virtualization”(虚拟化)下的*virtual Machines*来查看迁移的VM。
其中提供了访问虚拟机的说明 "此处"。

您可以登录到虚拟机并验证pos正在 使用的数据库的内容。此表中的数据库、表和条目应与在源VM上创建的相同。

借助 NetApp 在 Red Hat OpenShift 上为 Kubernetes 提供高级集群管理

适用于 Kubernetes 的高级集群管理：采用 NetApp 的 Red Hat OpenShift

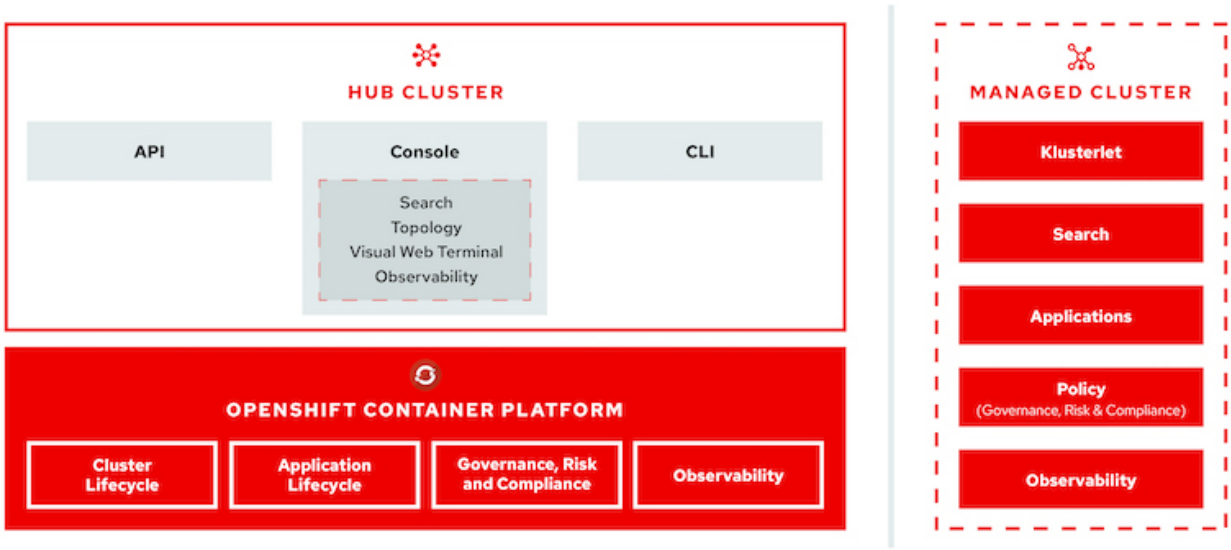
随着容器化应用程序从开发过渡到生产，许多组织需要使用多个 Red Hat OpenShift 集群来支持该应用程序的测试和部署。同时，企业通常会在 OpenShift 集群上托管多个应用程序或工作负载。因此，每个组织最终都会管理一组集群，因此 OpenShift 管理员必须面对在跨多个内部数据中心和公有云的一系列环境中管理和维护多个集群这一额外挑战。为了应对这些挑战， Red Hat 推出了适用于 Kubernetes 的高级集群管理。

使用 Red Hat Advanced Cluster Management for Kubernetes 可以执行以下任务：

1. 跨数据中心和公有云创建，导入和管理多个集群。
2. 从一个控制台在多个集群上部署和管理应用程序或工作负载。
3. 监控和分析不同集群资源的运行状况和状态
4. 监控并强制实施多个集群的安全合规性。

Red Hat Advanced Cluster Management for Kubernetes 作为 Red Hat OpenShift 集群的附加组件进行安装，并使用此集群作为其所有操作的中央控制器。此集群称为集线器集群，它会为用户提供一个管理平面以连接到高级集群管理。通过高级集群管理控制台导入或创建的所有其他 OpenShift 集群均由集线器集群管理，称为受管集

群。它会在受管集群上安装一个名为 Kluterlet 的代理，将其连接到中心集群，并处理与集群生命周期管理，应用程序生命周期管理，可观察性和安全合规性相关的不同活动请求。



有关详细信息，请参见文档 "此处"。

部署

部署适用于 **Kubernetes** 的高级集群管理

前提条件

1. 用于集线器集群的 Red Hat OpenShift 集群（版本 4.5 以上）
2. 适用于受管集群的 Red Hat OpenShift 集群（高于 4.5.3 版）
3. 对 Red Hat OpenShift 集群的集群管理员访问
4. 适用于 Kubernetes 的 Red Hat 高级集群管理订阅

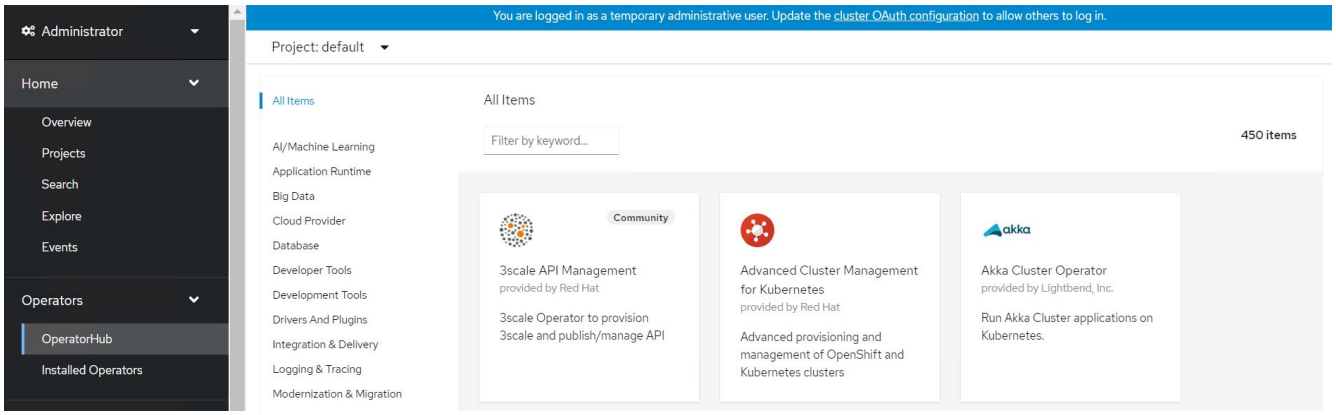
高级集群管理是 OpenShift 集群的一个附加功能，因此，根据在集线器和受管集群中使用的功能，硬件资源具有某些要求和限制。在对集群进行规模估算时，您需要考虑这些问题。请参见文档 "此处" 有关详细信息：

或者，如果集线器集群具有专用节点来托管基础架构组件，并且您希望仅在这些节点上安装高级集群管理资源，则需要相应地为这些节点添加容错和选择器。有关详细信息，请参见文档 "此处"。

部署适用于 **Kubernetes** 的高级集群管理

要在 OpenShift 集群上安装适用于 Kubernetes 的高级集群管理，请完成以下步骤：

1. 选择一个 OpenShift 集群作为中心集群，并使用 cluster-admin 权限登录到该集群。
2. 导航到 Operators > Operators Hub ，然后搜索适用于 Kubernetes 的高级集群管理。



3. 选择适用于 Kubernetes 的高级集群管理，然后单击安装。

4. 在 Install Operator 屏幕上，提供必要的详细信息（ NetApp 建议保留默认参数），然后单击 Install 。

OperatorHub > Operator Installation

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *

- ○ release-2.0
- ○ release-2.1
- ◉ release-2.2

Installation mode *

- ○ All namespaces on the cluster (default)
  This mode is not supported by this Operator
- ◉ A specific namespace on the cluster
  Operator will be available in a single Namespace only.

Installed Namespace *

- ◉ Operator recommended Namespace: PR open-cluster-management

Namespace creation
Namespace **open-cluster-management** does not exist and will be created.

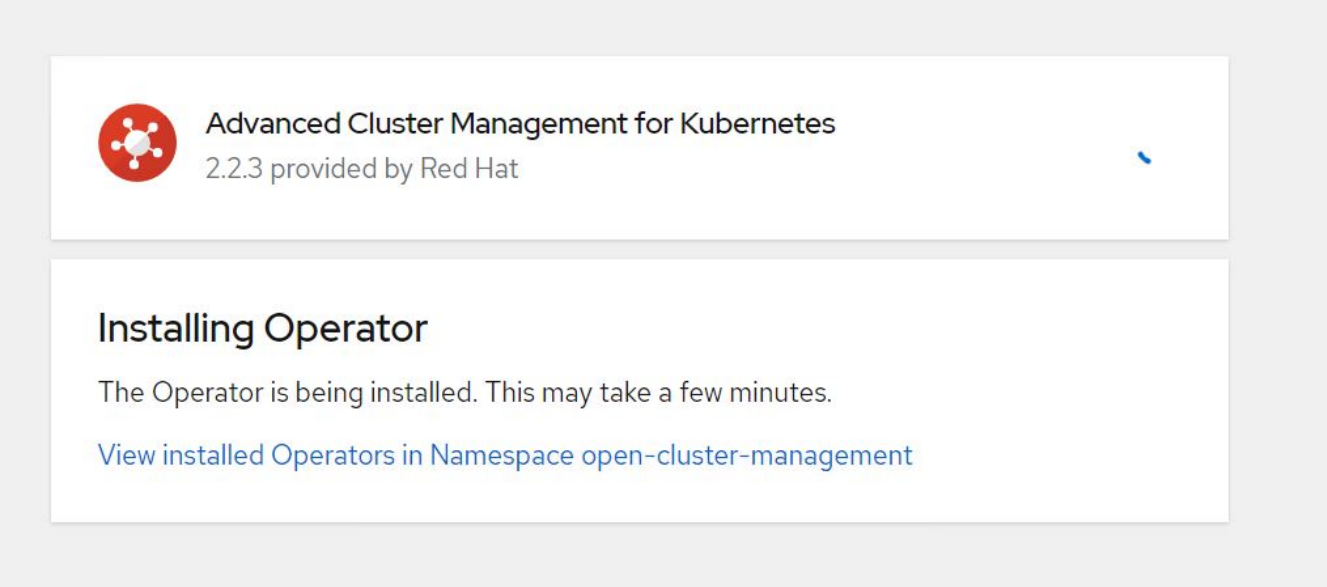
- ○ Select a Namespace

Approval strategy *

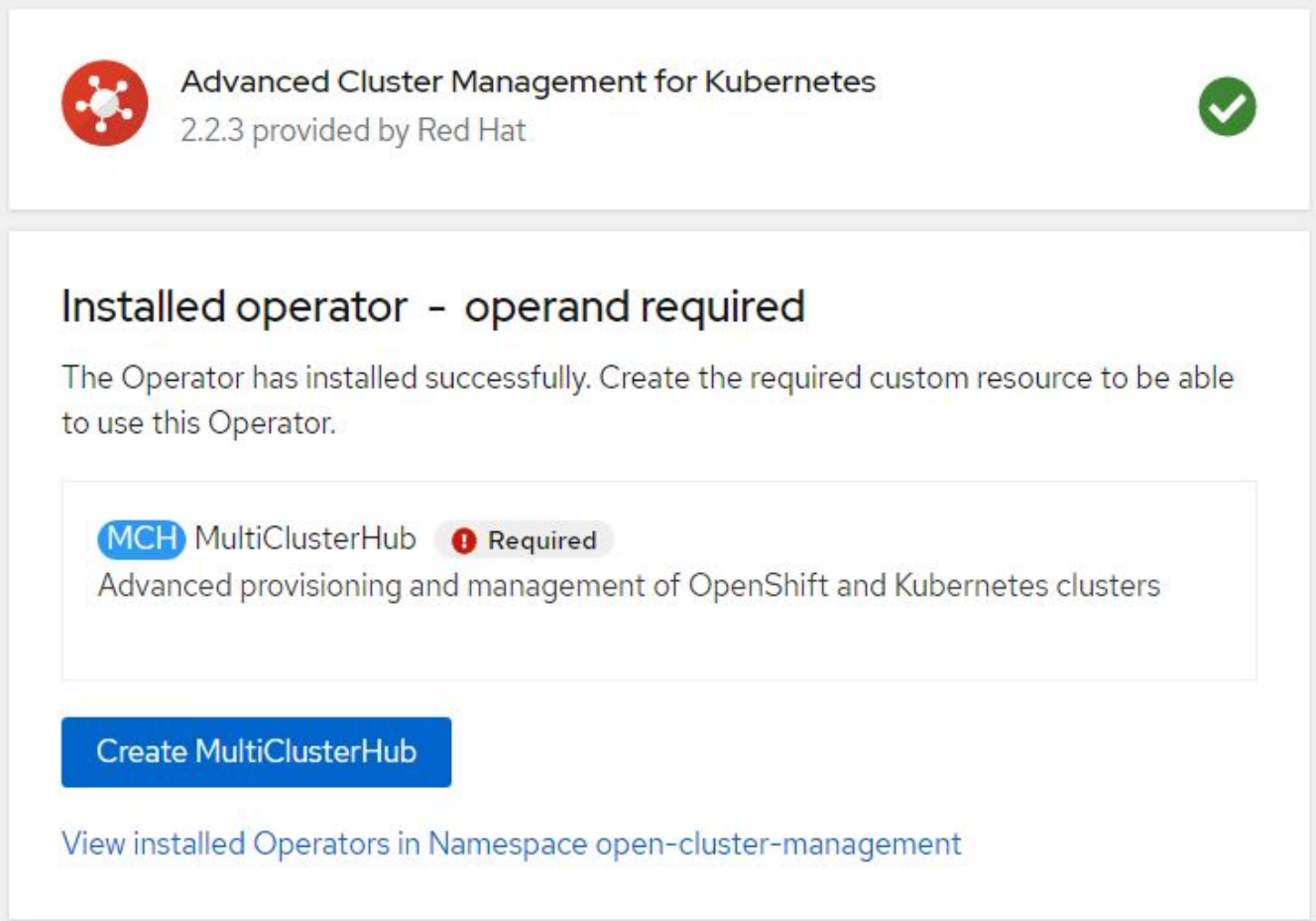
- ◉ Automatic
- ○ Manual

Install Cancel

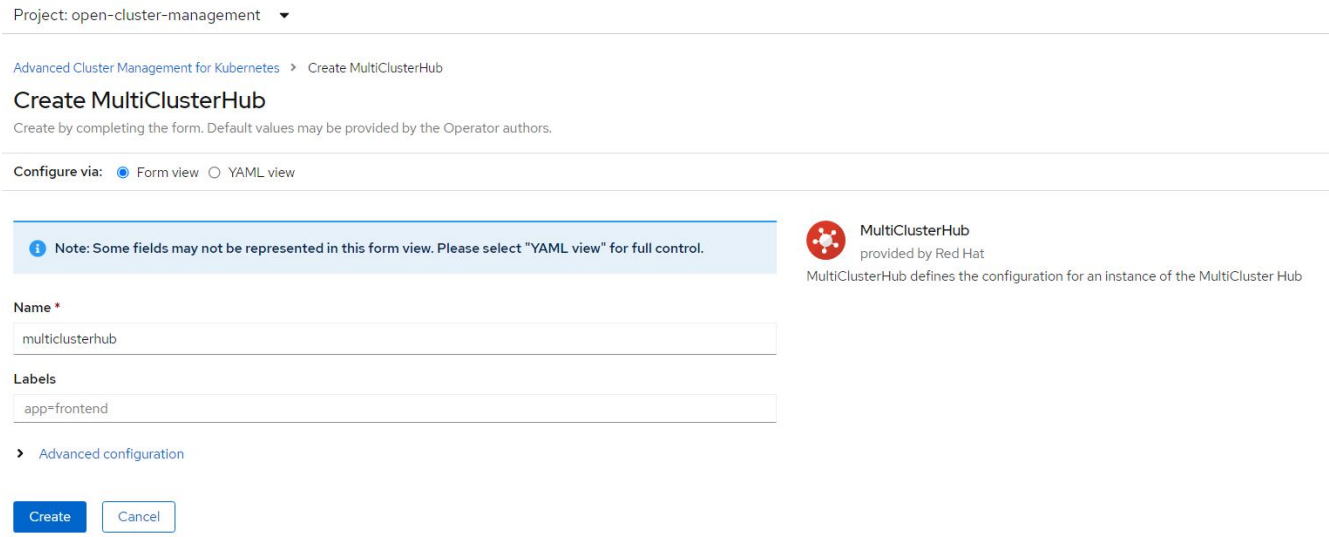
5. 等待操作员安装完成。



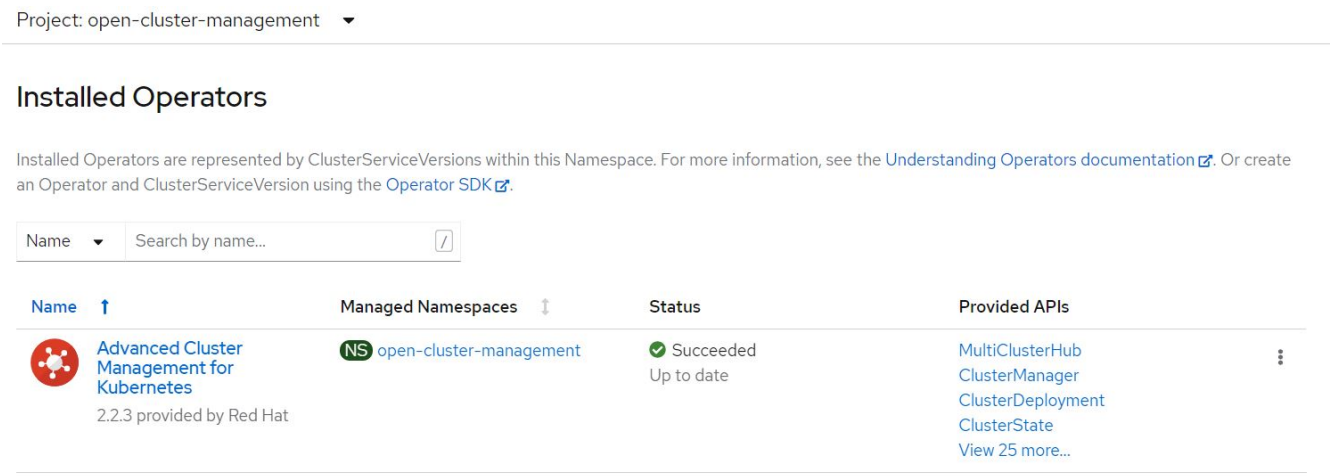
6. 安装操作员后，单击创建多集群中心。



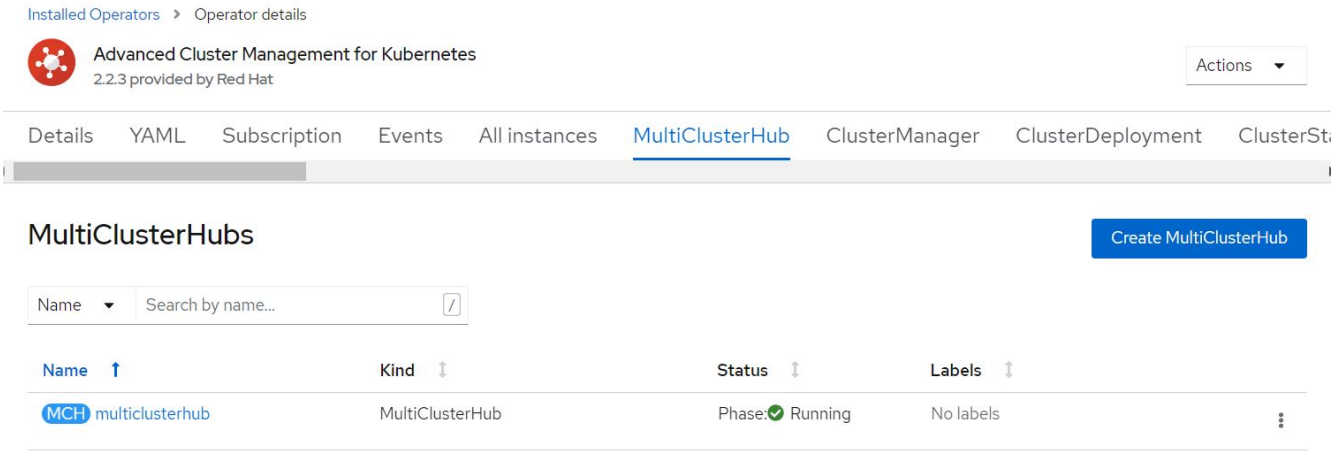
7. 在 "Create MultiClusterHub " 屏幕上，在提供详细信息后单击 "Create 。此操作将启动多集群集线器的安装。



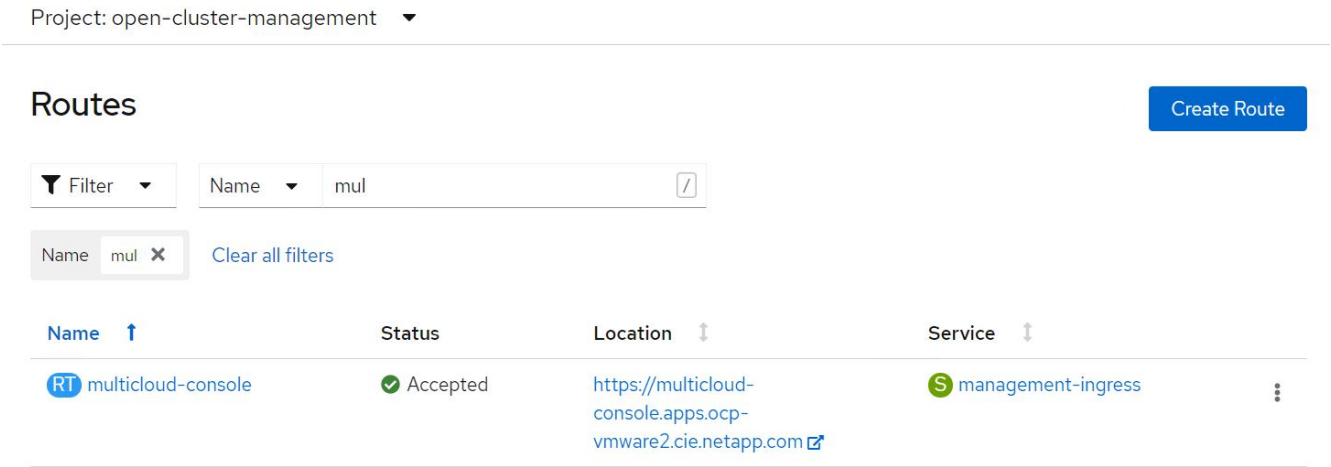
8. 在打开集群管理命名空间中的所有 Pod 均移至运行状态且操作员移至成功状态后，将安装适用于 Kubernetes 的高级集群管理。



9. 完成集线器安装需要一些时间，完成后，多集群集线器将变为运行状态。



10. 它会在开放式集群管理命名空间中创建路由。连接到路由中的 URL 以访问高级集群管理控制台。

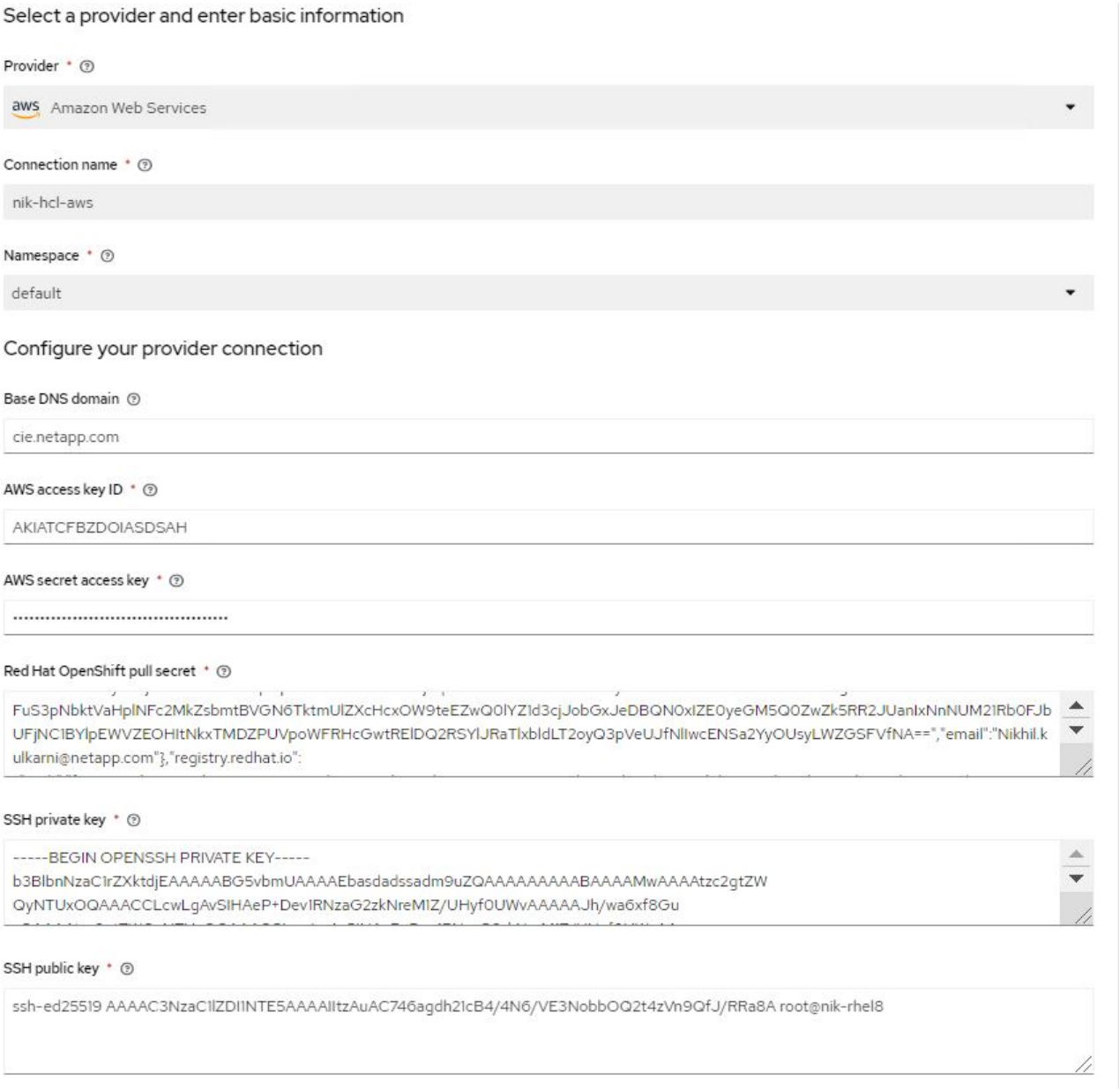


功能

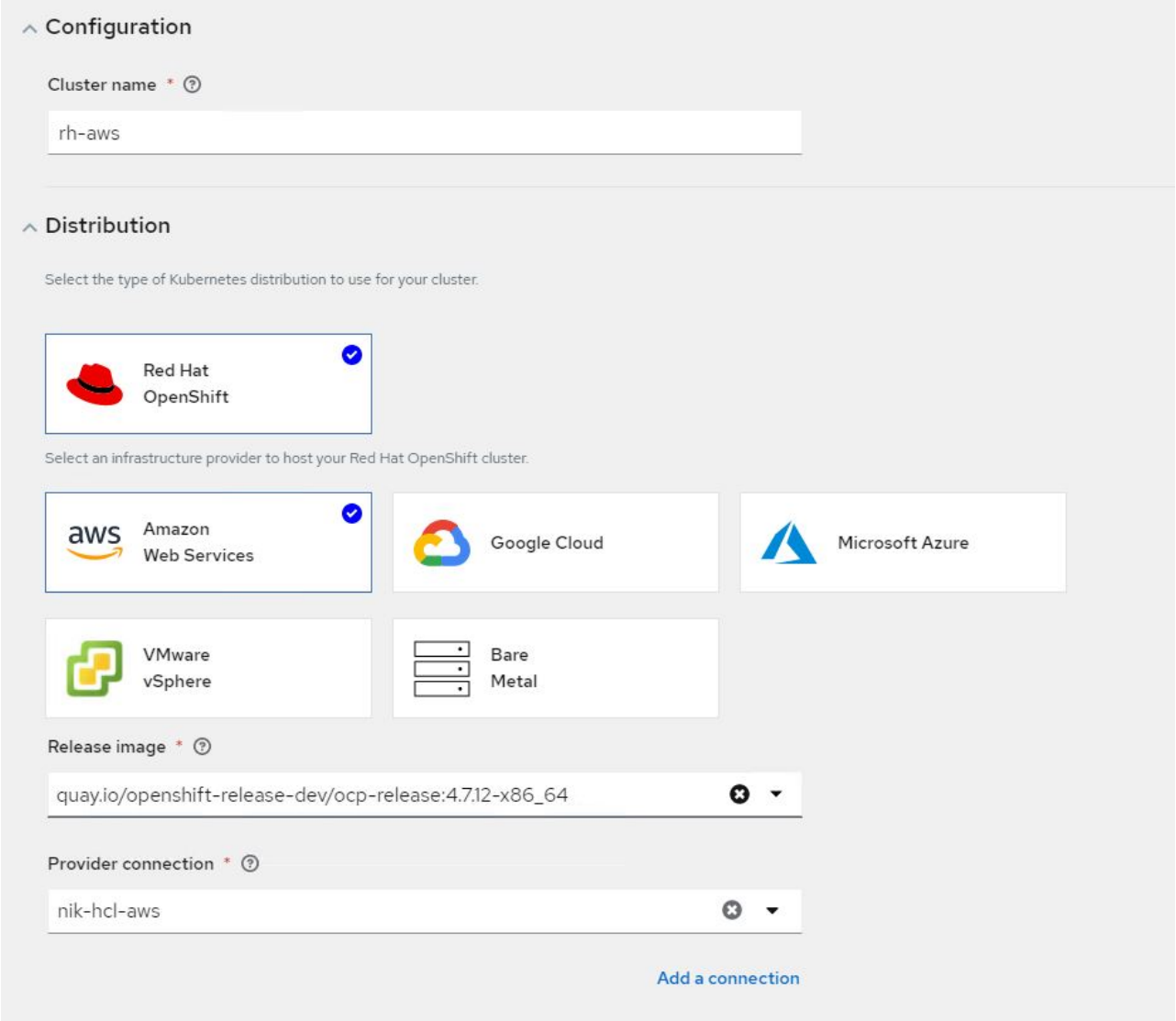
集群生命周期管理

要管理不同的 OpenShift 集群，您可以创建这些集群或将其导入到高级集群管理中。

1. 首先导航到 " 自动化基础架构 ">" 集群 " 。
2. 要创建新的 OpenShift 集群，请完成以下步骤：
   a. 创建提供程序连接：导航到 " 提供程序连接 " 并单击 " 添加连接 " ，提供与选定提供程序类型对应的所有详细信息，然后单击 " 添加 " 。



   b. 要创建新集群，请导航到集群，然后单击添加集群 > 创建集群。提供集群和相应提供程序的详细信息，然后单击创建。



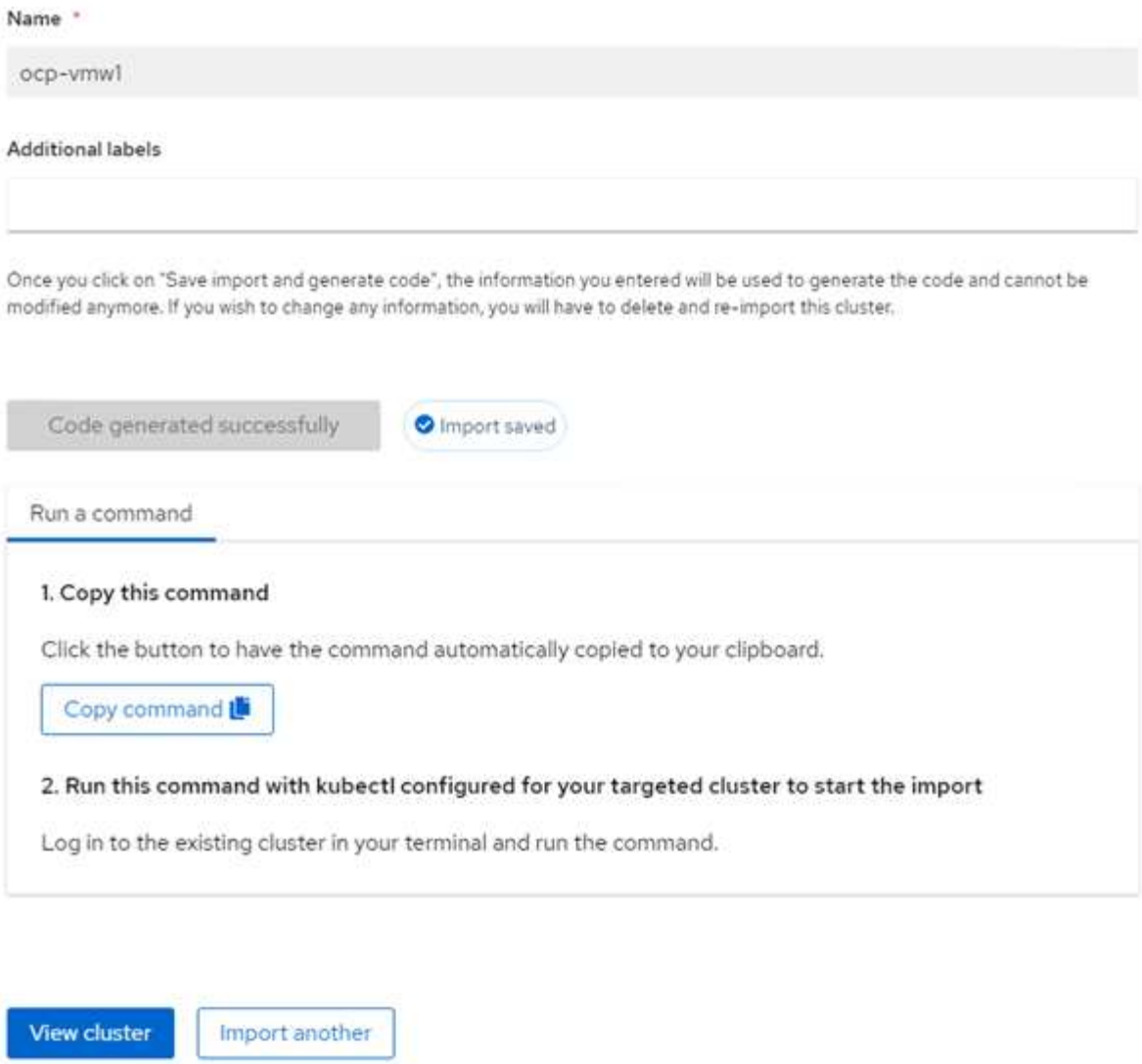
c. 创建集群后，该集群将显示在集群列表中，状态为 Ready 。

3. 要导入现有集群，请完成以下步骤：

   a. 导航到集群，然后单击添加集群 > 导入现有集群。

   b. 输入集群的名称，然后单击保存导入并生成代码。此时将显示一个用于添加现有集群的命令。

   c. 单击 Copy Command ，然后对要添加到集线器集群的集群运行命令。此操作将在集群上启动所需代理的安装，完成此过程后，集群将显示在集群列表中，并显示状态为 Ready 。



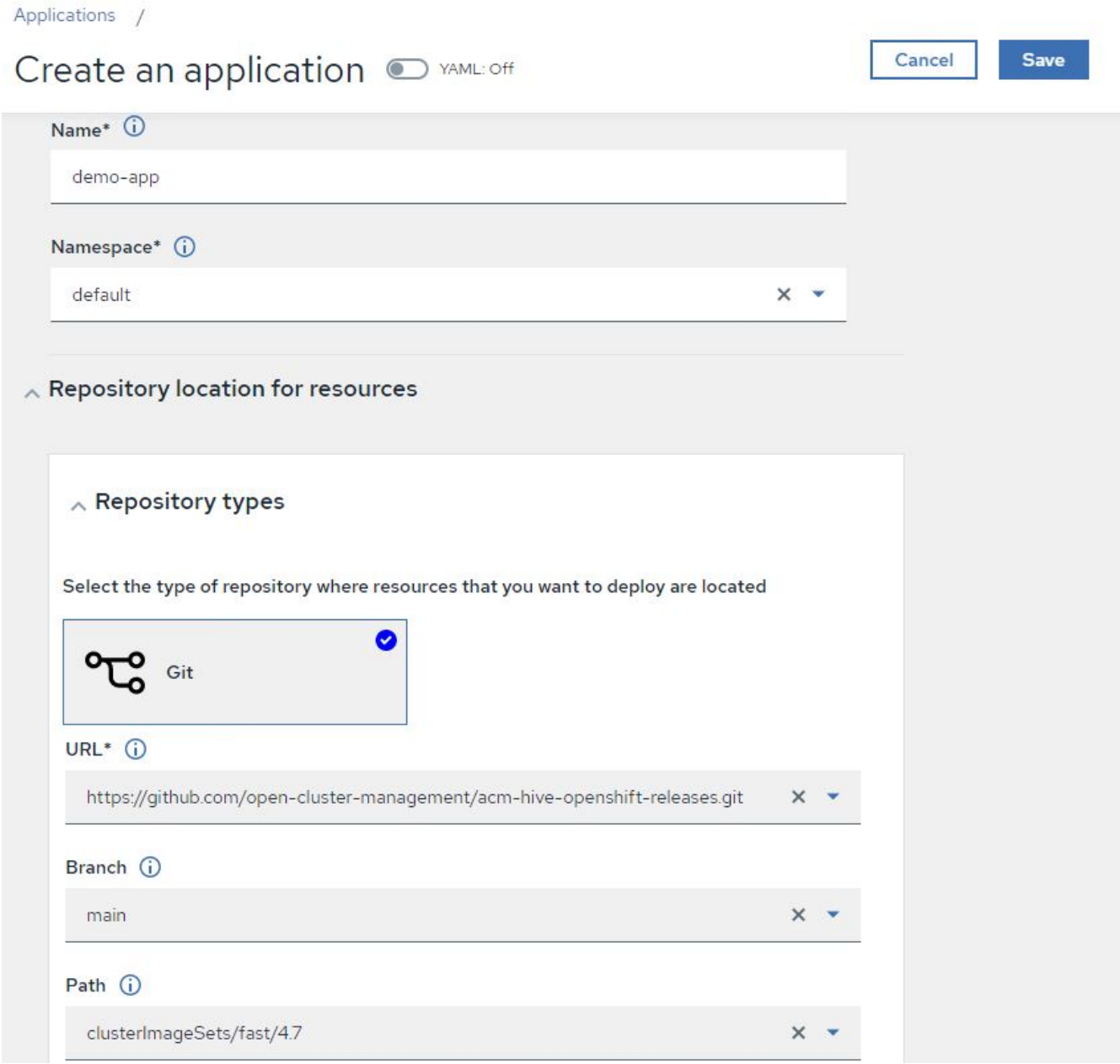
4. 创建并导入多个集群后，您可以从一个控制台监控和管理这些集群。

功能：借助 **NetApp** 在 **Red Hat OpenShift** 上为 **Kubernetes** 提供高级集群管理

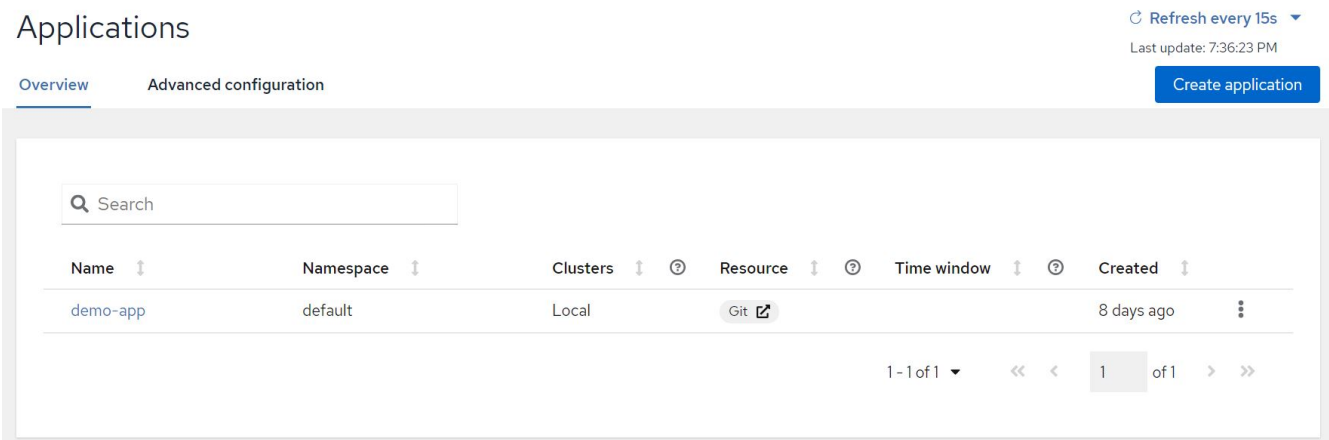
应用程序生命周期管理

要创建应用程序并在一组集群中对其进行管理，

1. 从边栏导航到管理应用程序，然后单击创建应用程序。提供要创建的应用程序的详细信息，然后单击保存。



2. 安装应用程序组件后，此应用程序将显示在列表中。



3. 现在，可以从控制台监控和管理此应用程序。

监管和风险

通过此功能，您可以为不同的集群定义合规性策略，并确保集群遵循此策略。您可以对策略进行配置，以通知或修复任何规则偏差或违规行为。

1. 从边栏导航到监管和风险。
2. 要创建合规性策略，请单击创建策略，输入策略标准的详细信息，然后选择应遵循此策略的集群。如果要自动修复此策略的违规，请选中 " 如果支持，则强制 " 复选框，然后单击 " 创建 " 。

Governance and risk / Policies /

Create policy YAML: Off

Name *

policy-complianceoperator

Namespace *

default

Specifications *

1 ComplianceOperator

Cluster selector

1 local-cluster: "true"

Standards

1 NIST-CSF

Categories

1 PR.IP Information Protection Processes and Procedures

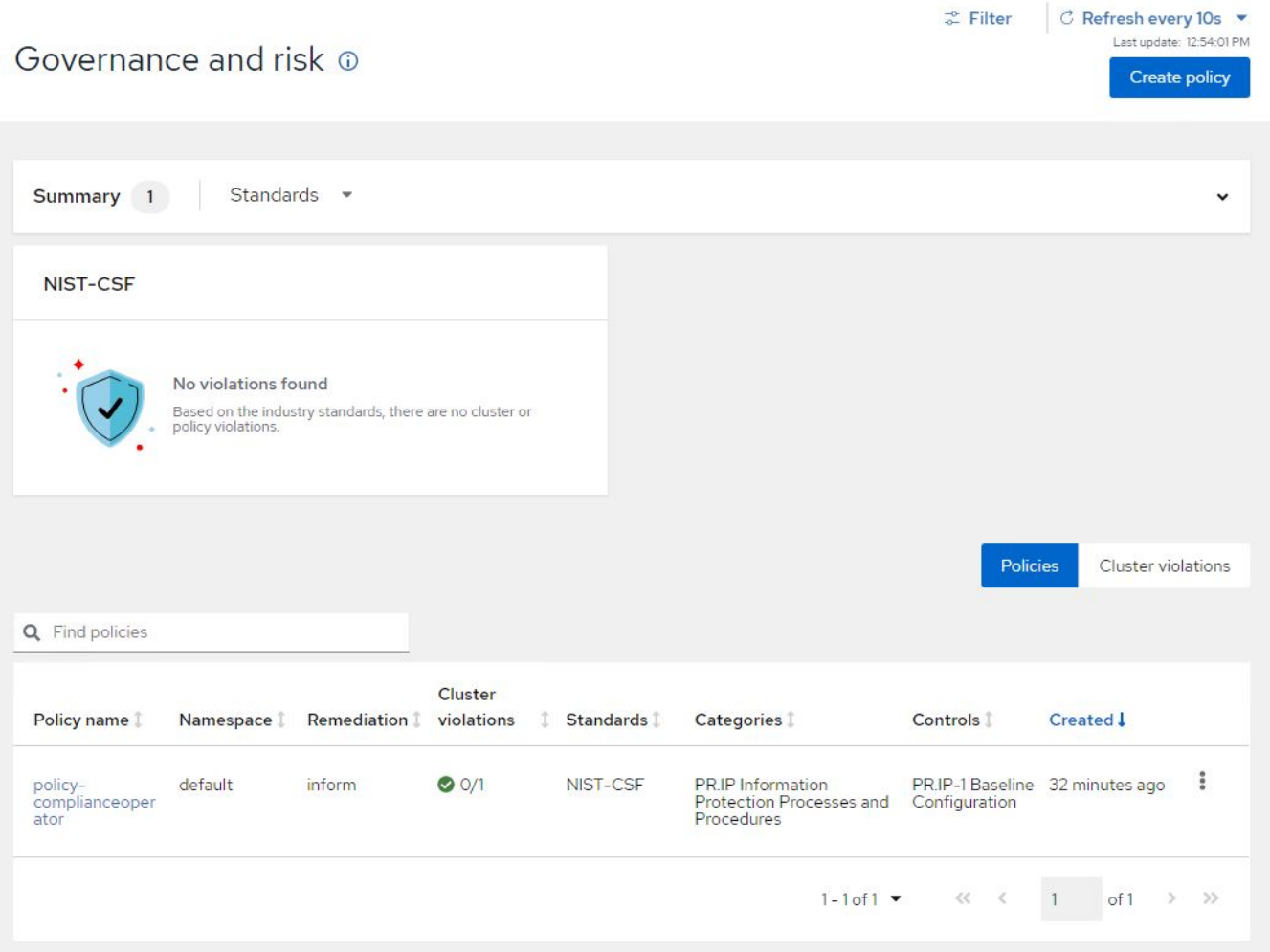
Controls

1 PR.IP-1 Baseline Configuration

Enforce if supported

Disable policy

3. 配置完所有必需的策略后，可以通过高级集群管理监控和修复任何策略或集群违规。

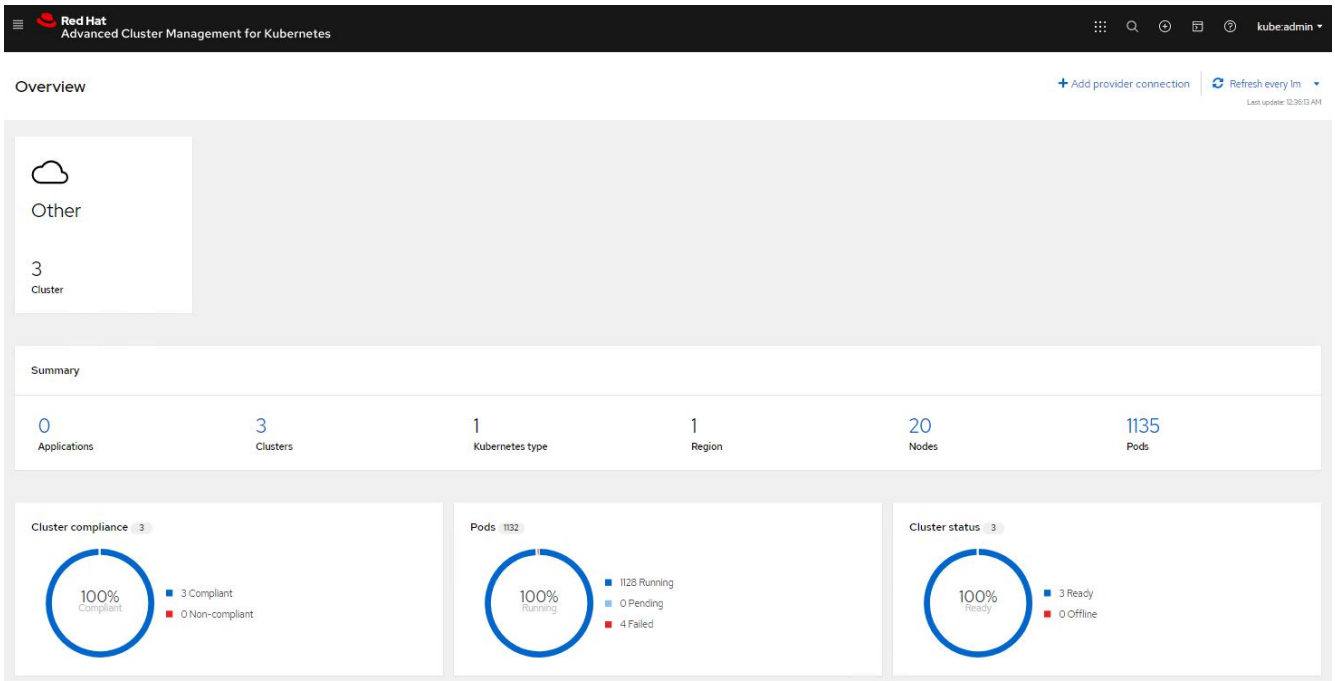


功能：借助 **NetApp** 在 **Red Hat OpenShift** 上为 **Kubernetes** 提供高级集群管理

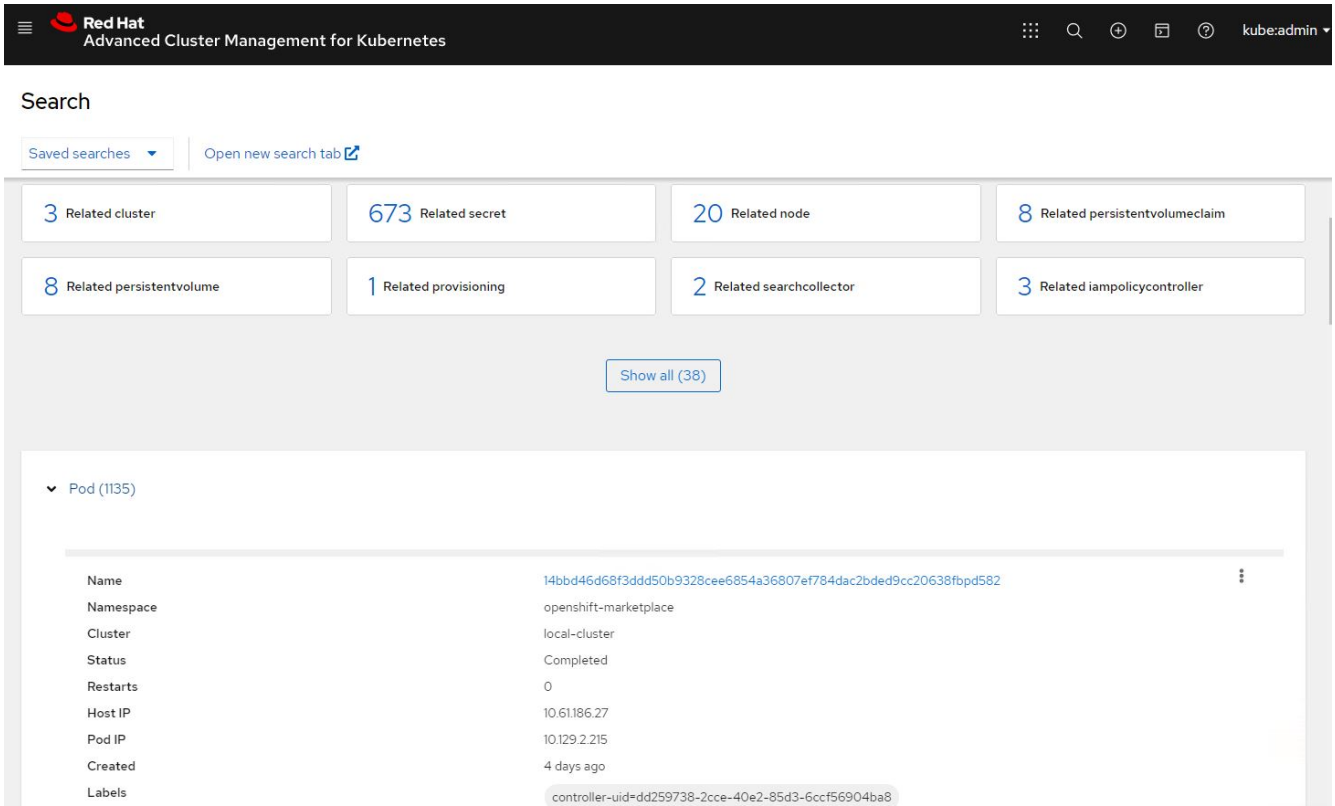
可观察性

适用于 Kubernetes 的高级集群管理提供了一种监控所有集群中的节点， Pod 以及应用程序和工作负载的方法。

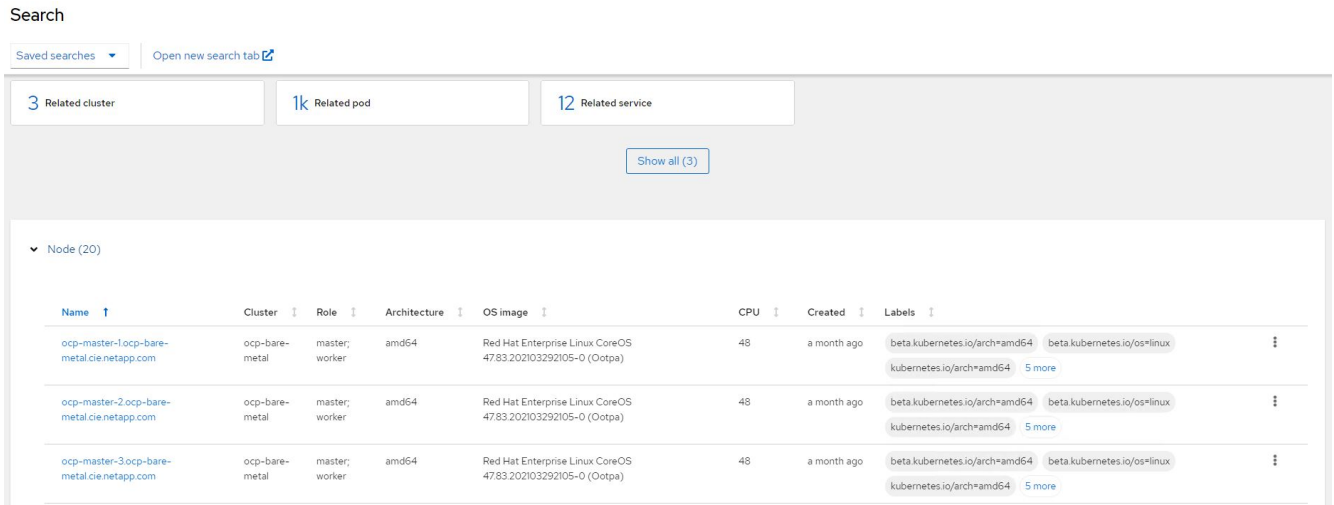
1. 导航到 " 观察环境 ">" 概述 " 。



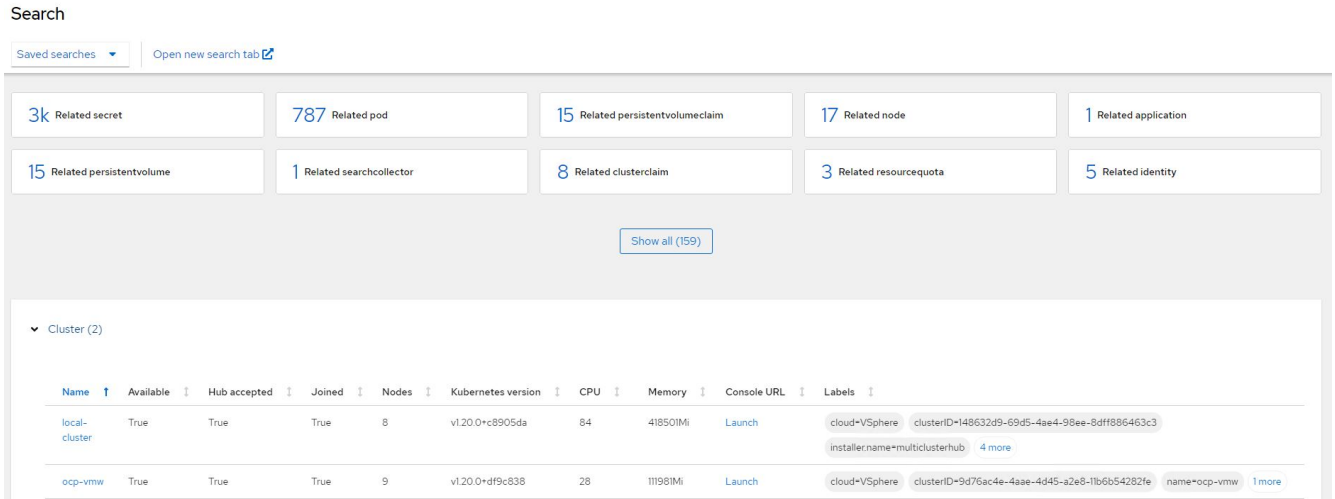
2. 所有集群中的所有 Pod 和工作负载都会根据各种筛选器进行监控和排序。单击 Pod 以查看相应数据。



3. 集群中的所有节点都会根据各种数据点进行监控和分析。单击节点可更深入地了解相应的详细信息。



4. 系统会根据不同的集群资源和参数监控和组织所有集群。单击集群可查看集群详细信息。



功能：借助 **NetApp** 在 **Red Hat OpenShift** 上为 **Kubernetes** 提供高级集群管理

在多个集群上创建资源

通过适用于 Kubernetes 的高级集群管理功能，用户可以从控制台同时在一个或多个受管集群上创建资源。例如，如果您的 OpenShift 集群位于不同站点，并由不同的 NetApp ONTAP 集群提供支持，并且希望在两个站点上配置 PVC ，则可以单击顶部栏上的（ + ）符号。然后，选择要创建 PVC 的集群，粘贴资源 YAML ，然后单击创建。

Create resource

Cancel | Create

Clusters | Select the clusters where the resource(s) will be deployed.

2 × local-cluster, ocp-vmw

Resource configuration | Enter the configuration manifest for the resource(s).

YAML

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: demo-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ocp-trident
```