



存储和虚拟化入门

NetApp virtualization solutions

NetApp
July 21, 2026

This PDF was generated from <https://docs.netapp.com/zh-cn/netapp-solutions-virtualization/openshift/osv-storage-getting-started.html> on July 21, 2026. Always check docs.netapp.com for the latest.

目录

存储和虚拟化入门	1
了解NetApp Trident与 Red Hat OpenShift 的集成	1
NetApp Trident	1
Trident Protect	1
NetApp ONTAP 硬件型号和协议支持	2
适用于 ONTAP 存储的 Trident 驱动程序	2
在 Red Hat OpenShift 集群上安装 Trident 并创建存储对象	3
步骤 1: 安装 Trident	4
步骤 2: 为您的环境准备存储后端和 StorageClass 配置文件	8
步骤 3: 创建 VolumeSnapshotClass 配置文件	28
步骤 4: 将配置文件应用于集群	28
第 5 步: 设置默认 Trident 存储和快照类	30
使用ONTAP部署 Red Hat OpenShift Virtualization 的要求	31
前提条件	31
NetApp Trident CSI、存储配置文件、OpenShift 虚拟化的卷访问模式	32
使用带有 Red Hat OpenShift 虚拟化的 ONTAP 存储创建虚拟机	36
创建 VM	37
视频演示	40
通过替代方法使用 ONTAP 存储, 利用 Red Hat OpenShift Virtualization 创建虚拟机	40
使用红帽 OpenShift 虚拟化从 YAML 文件创建虚拟机	40
使用 Red Hat OpenShift Virtualization 从快照副本创建 VM	42
使用 Trident 在 OpenShift Virtualization 中克隆虚拟机	46
在 Red Hat OpenShift 集群中的两个节点之间迁移虚拟机	50
虚拟机实时迁移	50
OpenShift Virtualization 中存储类之间 VM 的存储迁移	51

存储和虚拟化入门

了解NetApp Trident与 Red Hat OpenShift 的集成

存储在容器化应用程序中至关重要，可确保数据在动态且通常是短暂的环境中持续可用和有效管理。面向容器的现代存储解决方案旨在与应用程序无缝扩展，提供根据应用程序需求动态分配存储资源的灵活性，支持增长和不同工作负载。由于 OpenShift 虚拟化使虚拟机能够作为容器运行，因此它们使用与 OpenShift 集群中其他应用程序相同的存储。

NetApp ONTAP 是一款强大且高度可扩展的数据管理软件，可作为 NetApp 企业级存储解决方案的支柱。它提供了一个统一的存储架构，可与各种存储环境（包括本地、云和混合设置）无缝集成。ONTAP 支持各种协议，如 NFS、CIFS/SMB、iSCSI、FC 和 NVMe over TCP，可与不同的应用程序和工作负载无缝集成。它可以将 SAN、NAS 和对象存储整合到单个平台中，从而简化管理并降低成本。NetApp ONTAP 可在本地部署，也可作为云中的托管服务使用。

NetApp Trident

Trident 是由 NetApp 开发的开源存储编排器，旨在促进在容器化环境中使用 ONTAP 存储。它可以为容器化应用程序实现永久存储的动态配置和管理。

Trident 的主要功能

CSI 合规性 确保与最新的 Kubernetes 存储功能和标准兼容。声明性配置 允许用户在 YAML 文件中定义存储要求，然后由 Trident 自动管理。驱动程序 Trident 支持 ONTAP 所支持的 NFS、CIFS/SMB、iSCSI、FC 和 NVMe/TCP 协议的驱动程序。混合云和多云支持 Trident 支持本地 ONTAP 存储以及超大规模云服务提供商中托管存储服务的驱动程序，包括 AWS 中的 FSx for NetApp ONTAP、Azure 中的 Azure NetApp Files 以及 Google Cloud 中的 Google Cloud NetApp Volumes。高级数据管理 利用 ONTAP 的快照和克隆功能，实现高效的数据保护以及测试和开发环境的快速配置。

有关 Trident 的完整详细信息，请参阅 NetApp Trident ["此处文档"](#)

Trident Protect

Trident Protect 是 Trident 的一项高级功能，为容器化环境提供增强的数据保护和安全性。它旨在确保关键数据免受潜在威胁和数据丢失情况的影响。它对所有 Trident 用户免费提供，并且可以在安装了 Trident 的任何 OpenShift 集群中轻松启用。

Trident Protect 的主要功能

自动备份 Trident Protect 支持容器应用程序和虚拟机的自动策略驱动备份，确保它们定期备份，而无需人工干预。

快照管理 它利用 ONTAP 的快照功能为容器应用程序和虚拟机创建频繁的时间点副本，提供可靠的恢复机制。

数据加密 确保数据在静态和传输过程中都得到加密，满足严格的安全要求并保护敏感信息。

合规性和审计 提供工具和功能，帮助组织满足合规性要求并对其数据保护实践进行定期审计。

灾难恢复 与 ONTAP 的复制功能集成，提供强大的灾难恢复解决方案，即使在发生灾难性事件时也能确保业务连续性。

有关 Trident Protect 的完整详细信息，请参阅 NetApp Trident Protect ["此处文档"](#)。

NetApp ONTAP 硬件型号和协议支持

NetApp ONTAP 软件运行在各种硬件型号上，每种型号都旨在满足不同的性能和容量要求。Trident 扩展了其强大的功能，以支持各种 NetApp ONTAP 系统，包括全闪存 FAS (AFF)、全 SAN 阵列 (ASA) 和 ASA R2 系统。这种支持可确保组织能够在容器化环境中充分利用其高性能存储解决方案的潜力。

全闪存 **FAS (AFF)** 系统 AFF 系统提供卓越的性能和低延迟，使其成为高需求应用的理想选择。

All SAN Array (ASA) Systems ASA 系统专为专用 SAN 环境而设计，可提供强大的性能和可靠性。

ASA R2 Systems ASA R2 系统为 SAN 环境提供了增强的特性和功能。

AFX NetApp AFX 是一种专为企业 AI 工作负载设计的专用、分解式全闪存存储架构。它提供高性能 NAS，允许独立扩展计算和容量。NetApp AFX 存储系统在存储层的实现方面不同于其他基于 ONTAP 的系统（ASA、AFF 和 FAS）。AFX 使用分布式文件系统，可实现高性能和可扩展性，而其他基于 ONTAP 的系统则使用传统的存储架构。

1. AFF 支持的协议：NFS、CIFS/SMB、iSCSI、FC、NVMe/TCP
2. ASA 支持的协议：iSCSI、FC、NVMe/TCP
3. ASA R2 支持的协议：iSCSI、FC、NVMe/TCP
4. AFX 支持的协议：从 Trident 25.10 开始，仅支持 NFS 协议；不支持 SMB 协议。

适用于 ONTAP 存储的 Trident 驱动程序

Trident 包括以下 CSI 驱动程序，每个驱动程序都能够后端存储中配置卷

对于支持的硬件型号上的 **NAS** 协议

1. ontap-nas：一个 pvc 映射到 1 个 pv，这是一个专用的 flexvolume
2. ontap-nas-economy：配置的每个 PV 都是一个 qtree，每个 FlexVolume 具有可配置数量的 qtree（默认值为 200，可配置范围为 50 到 300）。一个 pvc 映射到 1 个 pv，即共享 flexvolume 上的一个 qtree



您可以将虚拟机的所有磁盘作为 qtree 放在单个卷中。但是，请注意，Snapshots、Clones 和 Replication 功能不受 Trident 支持。当这些功能由存储管理员管理时，它们不具备 PV 粒度。

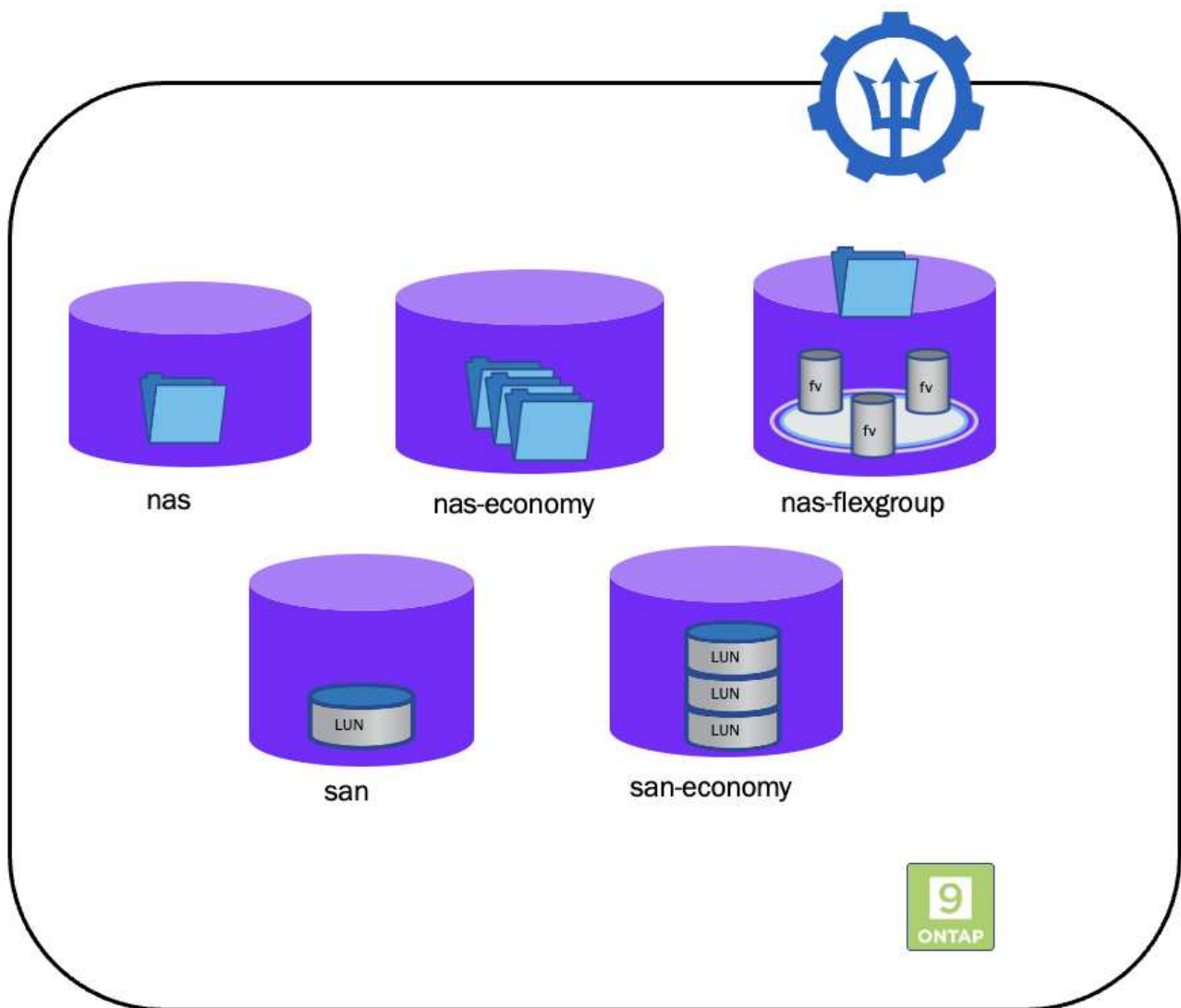
1. ontap-nas-flexgroup：配置的每个 PV 都是一个完整的 ONTAP FlexGroup，并使用分配给 SVM 的所有聚合。

对于支持的硬件型号上的 **SAN** 协议

1. ontap-san：一个 pvc 映射到 1 个 pv，即专用 flexvolume 上的 LUN
2. ontap-san-economy：一个 pvc 映射到 1 pv，这是共享 flexvolume 上的 LUN。您可以在共享 flexvolume 中有多个 LUN。（默认值为 100，每个 flexvol 卷可配置 50 到 200 个 LUN/PV）



您可以将虚拟机的所有磁盘作为 LUN 放置在单个卷中。但是，此驱动程序虽然支持 Snapshots 和 Clones，但不支持 Replication。当 Replication 由存储管理员管理时，它不是 PV 粒度的。



有关通过 Trident 驱动程序公开的功能以及必须由存储管理员应用的功能的完整列表，请参阅 Trident 文档中的["ONTAP 后端驱动程序"](#)部分。

在 Red Hat OpenShift 集群上安装 Trident 并创建存储对象

使用 Red Hat Certified Trident Operator 安装 Trident 并为 ONTAP 和 Amazon FSx for NetApp ONTAP 创建存储对象，以启用容器和虚拟机的动态卷配置。在需要时为块访问准备工作节点。

开始之前

- 在安装 OpenShift Virtualization 之前，请完成此页面上的步骤。OpenShift Virtualization 需要默认的 Trident 支持的 StorageClass 和 VolumeSnapshotClass 来为 VM 模板创建黄金映像。
- 如果您在配置 Trident 之前已经安装了 OpenShift Virtualization，请删除使用不同存储类创建的所有黄金映像。将 Trident 设置为默认值后，OpenShift Virtualization 会使用 Trident 存储重新创建黄金映像。

```
oc delete dv,VolumeSnapshot -n openshift-virtualization-os-images
--selector=cdi.kubevirt.io/dataImportCron
```

步骤 1：安装 Trident

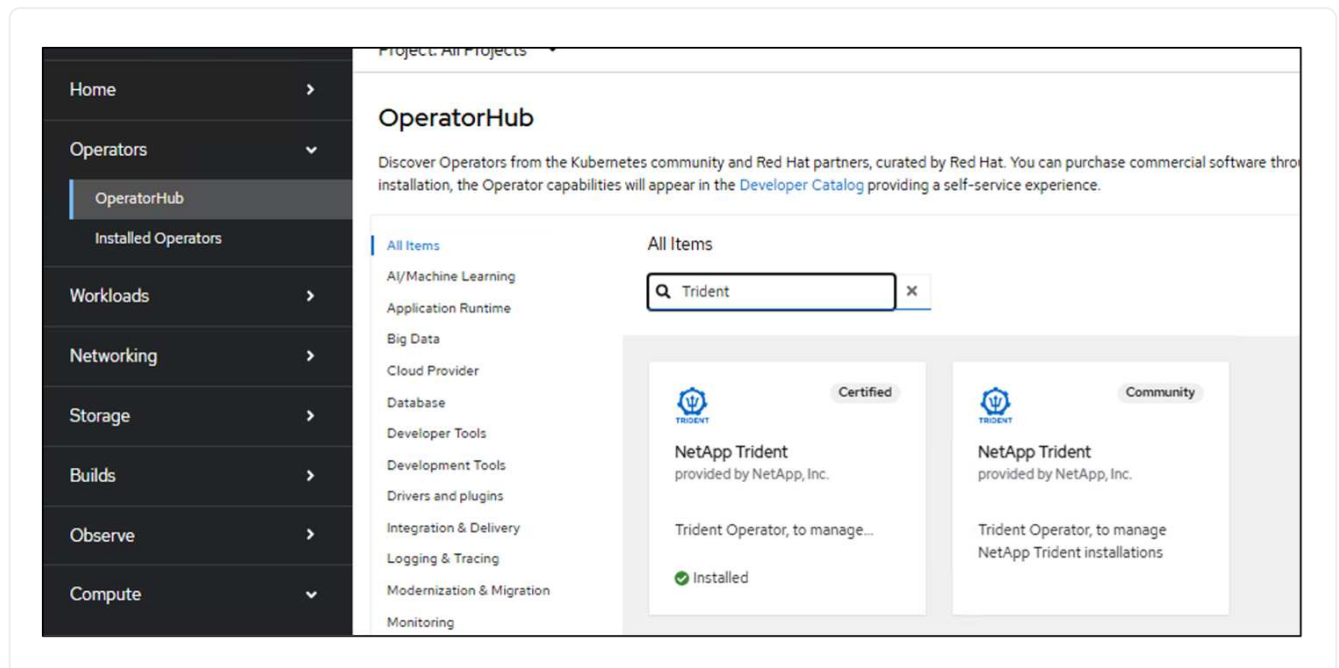
红帽认证 Trident Operator 由 NetApp 支持，适用于本地、公共云和 ROSA 等托管服务中的 OpenShift。从 Trident 25.02 开始，当您使用 Amazon FSx for NetApp ONTAP 并计划运行 OpenShift Virtualization VM 工作负载时，操作员还可以为 iSCSI 准备工作节点。

有关其他安装选项，请参见 "[Trident 文档](#)"。

步骤

1. 在 **OperatorHub** 中，选择 **Certified NetApp Trident**。

显示示例



2. 在 **Install** 页面上，保留最新版本并选择 **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 *

stable

Version *

25.2.1

25.2.1

25.2.0

Operator will be available in all namespaces.

☐

A specific namespace on the cluster

This mode is not supported by this Operator

Installed Namespace *

PR openshift-operators

Update approval *

☒ Automatic

☐ Manual

Install

Cancel

3. 操作员安装后，选择 查看操作员 并创建 Trident Orchestrator 的实例。

如果要为 iSCSI 准备工作节点，请切换到 YAML 视图并添加 `iscsi`` 到 ``nodePrep``。

显示示例

Create TridentOrchestrator

Create by completing the form. Default values may be provided by the Operator authors.

Configure via: ☐ Form view ☒ YAML view

```
1 kind: TridentOrchestrator
2 apiVersion: trident.netapp.io/v1
3 metadata:
4   name: trident
5 spec:
6   IPv6: false
7   debug: true
8   nodePrep:
9     - iscsi
10  imagePullSecrets: []
11  imageRegistry: ''
12  namespace: trident
13  silenceAutosupport: false
14
```

4. 确认所有 Trident Pod 都在集群中运行。

显示示例

```
[root@localhost ~]# oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-controller-84cb9bff89-lkx6k 6/6     Running   0           16h
trident-node-linux-d88b9             2/2     Running   0           16h
trident-node-linux-ld4b8             2/2     Running   0           16h
trident-node-linux-mj5r8             2/2     Running   0           16h
trident-node-linux-mkmmmp            2/2     Running   0           16h
trident-node-linux-qhgr7             2/2     Running   0           16h
trident-node-linux-vt9tp             2/2     Running   0           16h
[root@localhost ~]#
```

5. 如果已启用 iSCSI 节点准备，请登录到工作节点并验证 `iscsid` 和 `multipathd` 是否处于活动状态，以及 `multipath.conf` 是否包含条目。

显示示例

```
sh-5.1# systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-04-25 00:23:49 UTC; 3 days ago
 TriggeredBy: ● iscsid.socket
    Docs: man:iscsid(8)
          man:iscsiuio(8)
          man:iscsiadm(8)
   Main PID: 74787 (iscsid)
    Status: "Ready to process requests"
     Tasks: 1 (limit: 410912)
    Memory: 1.8M
       CPU: 6ms
    CGroup: /system.slice/iscsid.service
            └─74787 /usr/sbin/iscsid -f

Apr 25 00:23:49 ocp11-worker1 systemd[1]: Starting Open-iSCSI...
Apr 25 00:23:49 ocp11-worker1 systemd[1]: Started Open-iSCSI.
sh-5.1#
```

显示示例

```
sh-5.1# systemctl status multipathd
● multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-04-25 00:23:50 UTC; 3 days ago
 TriggeredBy: ● multipathd.socket
    Process: 74905 ExecStartPre=/sbin/modprobe -a scsi_dh_alua scsi_dh_emc scsi_dh_rdac dm-multipath (code=exited, status=0/SUCCESS)
    Process: 74906 ExecStartPre=/sbin/multipath -A (code=exited, status=0/SUCCESS)
   Main PID: 74907 (multipathd)
    Status: "up"
     Tasks: 7
    Memory: 18.3M
       CPU: 23.008s
    CGroup: /system.slice/multipathd.service
            └─74907 /sbin/multipathd -d -s

Apr 25 00:23:50 ocp11-worker1 systemd[1]: Starting Device-Mapper Multipath Device Controller...
Apr 25 00:23:50 ocp11-worker1 multipathd[74907]: -----start up-----
Apr 25 00:23:50 ocp11-worker1 multipathd[74907]: read /etc/multipath.conf
Apr 25 00:23:50 ocp11-worker1 multipathd[74907]: path checkers start up
Apr 25 00:23:50 ocp11-worker1 systemd[1]: Started Device-Mapper Multipath Device Controller.
sh-5.1#
```

```
sh-5.1# cat /etc/multipath.conf
defaults {
    find_multipaths no
}
blacklist {
    device {
        product .*
        vendor  .*
    }
}
blacklist_exceptions {
    device {
        product LUN
        vendor  NETAPP
    }
}
sh-5.1#
```

视频演示

以下视频展示了使用红帽认证 Trident Operator 安装 Trident 的演示。

[在 OpenShift 中使用经过认证的Trident Operator 安装Trident 25.02.1](#)

步骤 2：为您的环境准备存储后端和 **StorageClass** 配置文件

为您的环境创建 TridentBackendConfig 和 StorageClass 定义。您可以在环境中配置多个存储协议。为要使用的每个协议创建 YAML 文件，并将占位符值替换为特定的配置详细信息。



根据您的环境完成下面的本地或 ROSA 部分，然后继续执行步骤 3。

本地 OpenShift 集群

为要配置的每个协议创建 YAML 文件。您可以配置以下一个或多个协议：NAS 用于基于 NFS 的文件存储、iSCSI 用于 iSCSI 块存储、NVMe/TCP 用于高性能 NVMe over TCP 块存储，或 FC 用于光纤通道块存储。

为每个协议创建 TridentBackendConfig 和 StorageClass。后端配置包括存储在 Kubernetes Secret 中的凭据。

后端配置基本选项的基本准则

1. 后端配置文件引用您的 **ONTAP SVM** 和管理 **LIF**。+
2. 您可以使用用户名和密码向 ONTAP 进行身份验证。可以是集群管理员或 vservers 管理员的用户名和密码。+
3. 您还可以使用客户端证书对 ONTAP 进行身份验证。有关如何生成客户端证书、私钥和 CA 证书的详细信息，请参阅 ["Trident 文档"](#)。

NAS 经济驱动程序后端配置选项的基本准则

对于 **NAS economy driver** 配置，可以包括以下参数：

1. **qtreesPerFlexvol** 该值必须在50-300的范围内，默认值为200。+
2. **limitVolumePoolSize** 该值是在 ontap-nas-economy 后端使用 qtree 时请求的最大 FlexVol 大小。未设置默认值，默认情况下不强制执行此参数。如果设置此参数，当当前 FlexVol 达到指定大小时，Trident 操作员将创建一个新的 FlexVol。此参数可通过限制用于配置 PV 的 FlexVols 大小来帮助控制集群中的容量利用率。+
3. **denyNewVolumePools** 此参数限制 ontap-nas-economy 后端创建新 FlexVol 卷以包含其 qtree。仅使用预先存在的 FlexVol 来配置新的 PV。

SAN 驱动程序后端配置选项的基本准则

以下参数可包含在 **SAN driver** 配置中：

1. **limitVolumeSize** 该值是您在 SAN 后端时请求的最大卷大小。未设置默认值，默认情况下不强制执行此参数。如果设置此参数，则当请求的卷大小超过此值时，Trident 操作员将无法进行配置。您可以使用此参数来限制其为 LUN 管理的卷的最大大小。+
2. **lunsPerFlexvol** 每个 FlexVol 的最大 LUN 数，必须在 [50, 200] 范围内，默认值为 100。如果设置此参数，当当前 FlexVol 达到指定数量的 LUN 时，Trident 操作员会创建一个新的 FlexVol。此参数可通过限制每个用于调配 PV 的 FlexVol 的 LUN 数量，帮助您控制集群中的容量利用率。

SAN economy 驱动程序后端配置选项的基本准则

以下参数可包含在 **SAN economy driver** 配置中：

1. **limitVolumePoolSize** 该值是您在 SAN 经济后端时请求的最大 FlexVol 大小。未设置默认值，默认情况下不强制执行此参数。如果设置此参数，Trident 运算符将限制可以为在 flexvol 中创建 LUN 而创建的最大 flexvol 大小。此参数可以通过限制用于配置 PV 的 FlexVols 大小来帮助您控制集群中的容量利用率。+
2. **denyNewVolumePools** 此参数限制 SAN economy 后端创建新 FlexVol 卷以包含其 LUN。仅使用预先存在的 FlexVol 来配置新的 PV。



在为每个协议提供的示例 yaml 文件中，如果未提供 storagePrefix，则使用默认前缀 "trident"。当使用 ontap-nas-economy 且 storagePrefix 为 24 个或更多字符时，qtree 将不会嵌入存储前缀，但它将包含在卷名称中。

当您直接从系统管理器或 CLI 在 ONTAP 中创建卷时，如果未为所有可选参数提供值，则在卷配置期间会自动设置一些默认值，如下所示。

```
NSOL-NetApp-A70-T19U05:> volume create -vserver oc-test-automation -volume test_ontap_default -aggregate NSOL_NetApp_A70-T19U05a_NVME_SSD_1 -size 10GB
[Job 22836] Job succeeded: Successful
```

```
NSOL-NetApp-A70-T19U05:> volume show -vserver oc-test-automation -volume test_ontap_default -fields snapshot-policy,qos-adaptive-policy-group,snapshot-reserve-available,tiering-policy,unix-permissions,snapdir-access,space-guarantee
vserver      volume      unix-permissions space-guarantee snapdir-access snapshot-policy qos-adaptive-policy-group snapshot-reserve-available tiering-policy
-----
oc-test-automation test_ontap_default ---rwxr-xr-x      none          true          default      -              511.8MB      none
```

以下是使用 Trident 后端配置创建的卷示例。除名称模板外，它没有在默认值部分中显式设置任何选项。

```
oc describe tbc -n trident tbc-nas
### output truncated for brevity
Spec:
  Backend Name:  tbc-nas
  Credentials:
    Name:  tbc-nas-secret
  Defaults: # There are no defaults set except for the volume name
template.
  Name Template:      {{ .config.StoragePrefix }}_{{
.volume.Namespace }}_{{ .volume.RequestName }}
  Management LIF:     10.192.102.50
  Storage Driver Name: ontap-nas
  Storage Prefix:     testautomation
  Svm:                oc-test-automation
```

使用此后端配置创建卷时，未在后端配置文件中设置的选项将设置为默认值，如下所示。

```
NSOL-NetApp-A70-T19U05:> volume show -vserver oc-test-automation -volume testautomation_default_test_trident_defaults_91653 -fields snapshot-policy,qos-adaptive-policy-group,snapshot-reserve-available,tiering-policy,unix-permissions,snapdir-access,space-guarantee
vserver      volume      unix-permissions space-guarantee snapdir-access snapshot-policy qos-adaptive-policy-group snapshot-reserve-available tiering-policy
-----
oc-test-automation testautomation_default_test_trident_defaults_91653 ---rwxrwxrwx      none          false         none         -              0B              none
```

但是，当您在 Trident 后端配置的默认部分中设置值时，您可以创建对这些参数具有特定值的卷。在以下示例中，snapshotPolicy 在后端配置 yaml 的默认部分中设置为默认策略（如果未显式设置任何内容，则为 none），snapshotDir 设置为 true。通过这些设置，您可以看到后端卷获取了这些值。（snapdir-access 为 true，snapshot-policy 为默认值）

```
[root@00-50-56-b5-01-5b pvcs-br]# oc describe tbc -n trident tbc-nas
Name:          tbc-nas
Namespace:     trident
Labels:        <none>
Annotations:   <none>
API Version:   trident.netapp.io/v1
Kind:          TridentBackendConfig
Metadata:
  Creation Timestamp: 2026-03-30T20:06:59Z
  Finalizers:
    trident.netapp.io
  Generation:        3
  Resource Version:   81358056
  UID:                4dc24339-ff66-4345-acc1-37ec73d14140
Spec:
  Backend Name: tbc-nas
  Credentials:
    Name: tbc-nas-secret
  Defaults:
    Name Template:  {{ .config.StoragePrefix }}_{{ .volume.Namespace }}_{{ .volume.RequestName }}
    Snapshot Dir:   true
    Snapshot Policy: default
    Management LIF: 10.192.102.50
    Storage Driver Name: ontap-nas
    Storage Prefix:  testautomation
    Svm:             oc-test-automation
    Version:         1
Status:
  Backend Info:
    Backend Name:      tbc-nas
    Backend UUID:      210cc722-b394-489d-8e9e-49018d3e8710
    Deletion Policy:   delete
    Last Operation Status: Success
    Message:           Backend 'tbc-nas' updated
    Phase:             Bound
```

```
NSOL-NetApp-A70-T19U05:> volume show -vserver oc-test-automation -volume testautomation_default_test_trident_set_defaults_7567a -fields snapshot-policy,qos-adaptive-policy-group,snapshot-reserve-available,tiering-policy,unix-permissions,snapdir-access,space-guarantee
vserver      volume
le tiering-policy
-----
oc-test-automation testautomation_default_test_trident_set_defaults_7567a ---rwxrwxrwx  none      true      default  -      269.5MB
none
```

```
NSOL-NetApp-A70-T19U05:> volume show -vserver oc-test-automation -volume testautomation_default_test_trident_set_defaults_7567a -fields snapshot-policy,snapdir-access
vserver      volume
snapdir-access snapshot-policy
-----
oc-test-automation testautomation_default_test_trident_set_defaults_7567a true      default
```

对于所有其他附加参数，请查看 Trident 文档：

1. "Trident NAS后端配置选项" +
2. "Trident SAN 后端配置选项"

NAS

为 ONTAP NAS 创建 TridentBackendConfig 和 StorageClass，以启用基于 NFS 的持久存储配置。后端配置包括存储在 Kubernetes Secret 中的凭据，并引用您的 ONTAP SVM 和管理 LIF。

使用用户名和密码身份验证（另存为 tbc-nas.yaml）的后端密钥和后端配置文件示例：

```
# tbc-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-nas-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-nas
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: <ONTAP management LIF>
  backendName: tbc-nas
  svm: zoneb #<replace with your SVM name>
  storagePrefix: testzoneb #<replace with your prefix>
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace
  }}"
    credentials:
      name: tbc-nas-secret
```

使用客户端证书身份验证的后端密钥和后端配置文件示例（另存为 tbc-nas.yaml）：

```
# tbc-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-secret
  namespace: trident
type: Opaque
stringData:
  clientPrivateKey: <client private>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: <ONTAP management LIF>
  backendName: tbc-nas
  svm: zoneb #<replace with your SVM name>
  storagePrefix: testzoneb #<replace with your prefix>
  credentials:
    name: ontap-nas-secret
    clientCertificate: <client certificate>
```

StorageClass 定义（另存为 sc-nas.yaml）：

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

NAS 经济

创建 TridentBackendConfig 和 StorageClass，用于 ONTAP NAS economy 驱动程序，以启用使用 qtree 的基于 NFS 的持久存储配置。后端配置包括存储在 Kubernetes Secret 中的凭据，并引用您的 ONTAP

SVM 和管理 LIF。

使用用户名和密码身份验证（另存为 `tbc-nas-economy.yaml`）的后端密钥和后端配置文件示例：

```
# tbc-nas-economy.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-nas-economy-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-nas-economy
spec:
  version: 1
  storageDriverName: ontap-nas-economy
  managementLIF: <ONTAP management LIF>
  backendName: tbc-nas-eco
  svm: <replace with your SVM name>
  storagePrefix: <replace with your prefix>
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace
  }}"
    credentials:
      name: tbc-nas-economy-secret
```

StorageClass 定义（另存为 `sc-nas-economy.yaml`）：

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



有关可以包含在 ontap-nas-economy 驱动程序后端配置文件中的其他参数的详细信息，请参阅上述 NAS economy 驱动程序后端配置选项的基本指南部分。

iSCSI SAN

创建 TridentBackendConfig 和 StorageClass，用于支持 iSCSI 的 ONTAP SAN，以启用基于 iSCSI 的块存储配置。后端配置使用 ontap-san 驱动程序，并包含存储在 Kubernetes Secret 中的凭据。如果省略 sanType 参数，则默认使用 iSCSI 协议。

使用用户名密码身份验证的后端 secret 和后端配置文件（另存为 tbc-iscsi.yaml）：

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

使用客户端证书身份验证的后端 secret 和后端配置文件（另存为 tbc-iscsi.yaml）：

```
# tbc-iscsi.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-iscsi-secret
type: Opaque
stringData:
  clientPrivateKey: <client private key>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: ontap-iscsi
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <management LIF>
  backendName: ontap-iscsi
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-iscsi-secret
    clientCertificate: <client certificate>
```

StorageClass 定义 (另存为 sc-iscsi.yaml) :

```
# sc-iscsi.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-iscsi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

NVMe/TCP SAN

为 ONTAP SAN 创建 TridentBackendConfig 和 StorageClass, 使用 NVMe over TCP 以实现高性能块存储调配。后端配置使用针对 NVMe/TCP 传输优化的 ontap-san 驱动程序, 并包括存储在 Kubernetes Secret 中的凭据。

使用用户名密码身份验证的后端 secret 和后端配置文件（另存为 tbc-nvme.yaml）：

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

使用客户端证书身份验证的后端 secret 和后端配置文件（另存为 tbc-nvme.yaml）：

```

# tbc-nvme.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-ontap-nvme-secret
type: Opaque
stringData:
  clientPrivateKey: <client private key>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nvme
spec:
  version: 1
  storageDriverName: ontap-san
  sanType: nvme
  managementLIF: <ONTAP management LIF>
  backendName: backend-tbc-ontap-nvme
  svm: <SVM name>
  credentials:
    name: backend-tbc-ontap-nvme-secret
    clientCertificate: <client certificate>

```

StorageClass 定义（另存为 `sc-nvme.yaml`）：

```

# sc-nvme.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-nvme
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true

```



如果您计划将 ReadWriteMany (RWX) 卷与 NVMe 协议一起使用，则单个 RWX NVMe 卷不能由 Kubernetes 集群中超过 64 个节点挂载。有关此限制的详细信息，以及 Trident 26.02 中引入的超级子系统模型和 Trident 26.02 之前的每卷子系统模型的信息，请参见 Trident 文档：<https://docs.netapp.com/us-en/trident/trident-use/ontap-san-prep.html#nvme-tcp-considerations>。

FC SAN

为具有光纤通道的 ONTAP SAN 创建 TridentBackendConfig 和 StorageClass，以启用基于 FC 的块存储配置。后端配置使用指定了 FCP 协议的 ontap-san 驱动程序，并包括存储在 Kubernetes Secret 中的凭据。

使用用户名/密码身份验证方法或客户端证书身份验证方法通过 ONTAP 进行身份验证。有关如何生成客户端证书、私钥和 CA 证书的详细信息，请查看"[Trident 文档](#)"。下面提供了两种方法的示例。

显示示例

使用 ONTAP 用户名和密码身份验证（另存为 `tbc-fc.yaml`）的后端密钥和后端配置文件：

```
# tbc-fc.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-fc-secret
type: Opaque
stringData:
  username: <cluster admin username>
  password: <cluster admin password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-fc
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <ONTAP management LIF>
  backendName: tbc-fc
  svm: openshift-fc #<replace with your SVM name>
  sanType: fcp
  storagePrefix: demofc #<replace with your prefix>
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace
  }}"_{{ .volume.RequestName }}"
  credentials:
    name: tbc-fc-secret
```

使用客户端证书身份验证的后端 `secret` 和后端配置文件（另存为 `tbc-fc.yaml`）：

```
# tbc-fc.yaml
apiVersion: v1
kind: Secret
metadata:
  name: tbc-fc-secret
type: Opaque
stringData:
  clientPrivateKey: <client private key>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-fc
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <ONTAP management LIF>
  backendName: tbc-fc
  svm: openshift-fc #<replace with your SVM name>
  sanType: fcp
  storagePrefix: demofc #<replace with your prefix>
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace
  }}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-fc-secret
    clientCertificate: <client certificate>
```

```
# sc-fc.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-fc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

为使用 iSCSI 的 ONTAP SAN economy 创建 TridentBackendConfig 和 StorageClass，以启用基于 iSCSI 的块存储配置。后端配置使用带有 iSCSI 协议的 ontap-san-economy 驱动程序，并包括存储在 Kubernetes Secret 中的凭据。

使用客户端证书身份验证的后端 secret 和后端配置文件（另存为 tbc-san-eco.yaml）：

```
# tbc-san-eco.yaml
---
apiVersion: v1
kind: Secret
metadata:
  name: ontap-san-cert-auth
  namespace: trident
type: Opaque
stringData:
  clientPrivateKey: <client private key>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: tbc-san-eco-cert
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-san-economy
  managementLIF: "<ONTAP management LIF>"
  backendName: tbc_san_eco
  svm: <replace with your SVM name>
  storagePrefix: <replace with your prefix>
  credentials:
    name: ontap-san-cert-auth
    clientCertificate: <client certificate>
```

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

查看 Trident 文档，了解 yaml 文件的其他示例，以及有关 SAN 驱动程序和 NAS 驱动程序支持的访问模式和卷模式的信息。

1. ["使用 NAS 驱动程序的示例"](#)
2. ["使用 SAN 驱动程序的示例"](#)

使用 **Amazon FSx for NetApp ONTAP** 的 **ROSA** 集群

为要配置的每个协议创建 YAML 文件。可以配置以下协议之一或两者：用于基于 NFS 的文件存储的 NAS 或用于块存储的 iSCSI。

NAS

为 Amazon FSx for NetApp ONTAP 创建 TridentBackendConfig 和 StorageClass，并使用 ONTAP NAS，以便在 ROSA 集群上启用基于 NFS 的持久存储配置。后端配置使用 Amazon FSx for NetApp ONTAP 的 DNS 名称来管理和数据 LIF，并包含存储在 trident 命名空间的 Kubernetes Secret 中的凭据。

后端 secret 和后端配置文件（另存为 tbc-fsx-nas.yaml）：

```
# tbc-fsx-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-fsx-ontap-nas-secret
type: Opaque
stringData:
  username: <FSx for ONTAP, for example fsxadmin>
  password: <FSx for ONTAP password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-fsx-ontap-nas
spec:
  version: 1
  backendName: fsx-ontap
  storageDriverName: ontap-nas
  managementLIF: <Management DNS name>
  dataLIF: <NFS DNS name>
  svm: <SVM NAME>
  credentials:
    name: backend-fsx-ontap-nas-secret
```

StorageClass 定义（另存为 sc-fsx-nas.yaml）：

```
# sc-fsx-nas.yaml (storage class name is trident-csi)
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: trident-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  fsType: "ext4"
allowVolumeExpansion: true
reclaimPolicy: Retain
```

iSCSI

为 Amazon FSx for NetApp ONTAP 配置 ONTAP SAN，创建 TridentBackendConfig 和 StorageClass，以便在 ROSA 集群上启用基于 iSCSI 的块存储配置。后端配置使用 ontap-san 驱动，并包含存储在 Kubernetes Secret 中的凭据。确保工作节点已为 iSCSI 访问做好准备。

后端 secret 和后端配置文件（另存为 tbc-fsx-iscsi.yaml）：

```
# tbc-fsx-iscsi.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-tbc-fsx-iscsi-secret
type: Opaque
stringData:
  username: <FSx for ONTAP, for example fsxadmin>
  password: <FSx for ONTAP password>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: fsx-iscsi
spec:
  version: 1
  storageDriverName: ontap-san
  managementLIF: <Management DNS name>
  backendName: fsx-iscsi
  svm: <SVM name>
  credentials:
    name: backend-tbc-fsx-iscsi-secret
```

StorageClass 定义（另存为 sc-fsx-iscsi.yaml）：

```
# sc-fsx-iscsi.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: sc-fsx-iscsi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  media: "ssd"
  provisioningType: "thin"
  fsType: ext4
  snapshots: "true"
allowVolumeExpansion: true
```

步骤 3：创建 **VolumeSnapshotClass** 配置文件

为本地和 ROSA 部署创建 VolumeSnapshotClass 定义。此配置为持久卷启用基于快照的操作。

VolumeSnapshotClass 定义（另存为 `snapshot-class.yaml`）：

```
# snapshot-class.yaml
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: trident-snapshotclass
driver: csi.trident.netapp.io
deletionPolicy: Retain
```

步骤 4：将配置文件应用于集群

将前面步骤中创建的配置文件应用于 OpenShift 集群。

步骤

1. 为您配置的每个协议应用 TridentBackendConfig 和 StorageClass 文件。

对于本地集群：

```
oc create -f tbc-nas.yaml -n trident
oc create -f sc-nas.yaml

oc create -f tbc-nas-economy.yaml -n trident
oc create -f sc-nas-economy.yaml

oc create -f tbc-iscsi.yaml -n trident
oc create -f sc-iscsi.yaml

oc create -f tbc-nvme.yaml -n trident
oc create -f sc-nvme.yaml

oc create -f tbc-fc.yaml -n trident
oc create -f sc-fc.yaml
```

对于 ROSA 集群：

```
oc create -f tbc-fsx-nas.yaml -n trident
oc create -f sc-fsx-nas.yaml

oc create -f tbc-fsx-iscsi.yaml -n trident
oc create -f sc-fsx-iscsi.yaml
```

2. 应用 VolumeSnapshotClass 配置。

```
oc create -f snapshot-class.yaml
```

3. 验证资源是否已成功创建。

检查 TridentBackendConfig 对象：

```
oc get tbc -n trident
```

检查 StorageClass 对象：

```
oc get storageclass
```

检查 VolumeSnapshotClass：

```
oc get volumesnapshotclass
```

第 5 步：设置默认 Trident 存储和快照类

将 Trident StorageClass 和 VolumeSnapshotClass 设置为 OpenShift 集群中的默认值。这是 OpenShift Virtualization 为 VM 模板创建黄金映像源所必需的。

步骤

1. 设置默认 Trident StorageClass。

将 Trident 支持的 StorageClass 设置为集群默认值，以便在未指定存储类时 PersistentVolumeClaims 自动使用它。您需要配置两个注释：一个用于集群范围的默认值，一个特定于 OpenShift Virtualization。

a. 设置集群范围的默认 StorageClass 注释。

确保仅将一个 StorageClass 设置为默认值。如果另一个 StorageClass 已设置为默认值，则将其注释设置为 false。

在控制台中，编辑标注：

```
storageclass.kubernetes.io/is-default-class: "true"
```

从 CLI：

```
kubectl patch storageclass <storage class name> -p '{"metadata":  
{"annotations":{"storageclass.kubernetes.io/is-default-  
class":"true"}}}'
```

b. 设置 OpenShift Virtualization 特定的默认注释。

OpenShift Virtualization 使用优先于集群常规 `is-default-class` 注释的特定注释。如果另一个 StorageClass 已设置为默认值，则将其注释设置为 false。

在控制台中，编辑标注：

```
storageclass.kubevirt.io/is-default-virt-class: "true"
```

从 CLI：

```
kubectl patch storageclass <storage class name> -p '{"metadata":  
{"annotations":{"storageclass.kubevirt.io/is-default-virt-class":  
"true"}}}'
```

2. 设置默认 Trident VolumeSnapshotClass。

将 Trident 支持的 VolumeSnapshotClass 设置为集群默认值，为持久卷启用基于快照的操作。这可确保在未指定快照类时 VolumeSnapshots 自动使用 Trident CSI 驱动程序，并允许 OpenShift Virtualization 创建黄金

映像的快照。

确保仅将一个 VolumeSnapshotClass 设置为默认值。如果另一个 VolumeSnapshotClass 已设置为默认值，则将其注释设置为 false。

在控制台中，编辑标注：

```
snapshot.storage.kubernetes.io/is-default-class: "true"
```

从 CLI：

```
oc patch volumesnapshotclass <snapshot class name> --type=merge -p  
'{"metadata":{"annotations":{"snapshot.storage.kubernetes.io/is-default-  
class":"true"}}}'
```

使用ONTAP部署 Red Hat OpenShift Virtualization 的要求

查看使用 ONTAP 存储系统部署 OpenShift 虚拟化的先决条件和其他详细信息。

前提条件

- 安装在裸机基础架构上并带有 RHCOS 工作节点的 Red Hat OpenShift 集群（高于 4.6 版本）
- 部署机器健康检查以维护虚拟机的高可用性
- NetApp ONTAP集群，其中 SVM 配置了正确的协议。
- 在 OpenShift 集群上安装Trident
- 创建Trident后端配置
- 在 OpenShift 集群上配置 StorageClass，并使用Trident作为配置器
- 在 OpenShift 集群上配置的以 Trident 作为配置程序的 Trident VolumeSnapshotClass
- Trident StorageClass 配置为 OpenShift 集群或 OpenShift Virtualization 的默认值。

有关上述Trident先决条件，请参阅["Trident安装部分"](#)了解详情。

- 集群管理员访问 Red Hat OpenShift 集群
- NetApp ONTAP集群的管理员访问权限
- 安装了 tridentctl 和 oc 工具并添加到 \$PATH 的管理员工作站

由于 OpenShift 虚拟化由安装在 OpenShift 集群上的操作员管理，因此会给内存、CPU 和存储带来额外的开销，在规划集群的硬件要求时必须考虑这些开销。有关更多详细信息，请参见文档 ["此处"](#)。

也可以通过配置节点放置规则，指定 OpenShift 集群节点的子集来托管 OpenShift 虚拟化运算符、控制器和虚拟机。要为 OpenShift 虚拟化配置节点放置规则，请参阅文档 ["此处"](#)。

对于支持 OpenShift Virtualization 的存储，NetApp建议使用专用的 StorageClass，该 StorageClass 从特定

的Trident后端请求存储，而该后端又由专用的 SVM 支持。这对于为 OpenShift 集群上基于 VM 的工作负载提供的数据保持了一定程度的多租户级别。

NetApp Trident CSI、存储配置文件、OpenShift 虚拟化的卷访问模式

在 OpenShift 虚拟化中，引导源是用于从模板配置虚拟机 (VM) 的磁盘映像。它们通常存储为 openshift-virtualization-os-images 命名空间中的 DataVolumes 或持久卷声明 (PVC)。为了使 OpenShift 虚拟化正常运行，在使用操作员安装 OpenShift 虚拟化之前，必须执行两项存储配置任务：

1. 必须为集群配置默认 StorageClass。否则，OpenShift 虚拟化无法自动导入启动源映像。
2. 如果您的存储提供商未被 CDI 识别，则必须配置存储配置文件。存储配置文件根据关联的 StorageClass 提供推荐的存储设置。StorageProfile 的关键功能之一是为 StorageClass 定义克隆策略。这决定了卷的克隆方式（例如，使用快照、CSI 克隆或主机辅助复制），从而根据底层存储供应商的功能优化性能和资源利用率。+ Trident CSI 驱动程序可被 CDI 识别，并且系统会自动为集群中创建的任何 Trident StorageClass 创建默认存储配置文件。此默认存储配置文件的克隆策略设置为快照，这是 Trident CSI 驱动程序推荐的克隆策略。在安装 OpenShift Virtualization 之前，您必须为 Trident 创建 VolumeSnapshotClass 和 StorageClass，并将 StorageClass 设置为默认值。这可以确保引导源镜像在 openshift-virtualization-os-images 命名空间中作为 VolumeSnapshots 可用，从而能够高效且无误地配置虚拟机。

对于某些受支持的操作系统（如 RHEL、Fedora 或 CentOS），OpenShift 虚拟化会自动提供和更新引导源。这些文件在安装 OpenShift 虚拟化时下载，并自动管理和更新到最新操作系统版本。如果未自动提供操作系统，管理员必须手动准备并附加自定义映像。当您使用 NetApp Trident 作为 CSI 时，您将获得以下额外好处：

1. 启动源映像将在 openshift-virtualization-os-images 命名空间中作为快照提供。配置 VM 时，会从快照中克隆启动映像，从而使 VM 启动速度更快。
2. 系统会为每个创建的 Trident StorageClass 自动创建一个存储配置文件，并自动将克隆策略配置为快照。访问模式设置为 ReadWriteMany。无需手动配置。

在集群中安装 OpenShift Virtualization 之前，创建 Trident StorageClass 和 Trident VolumeSnapshotClass 并将其设置为默认值至关重要。这将确保引导源映像作为 VolumeSnapshots 可用，从而能够高效且无误地配置 VM。

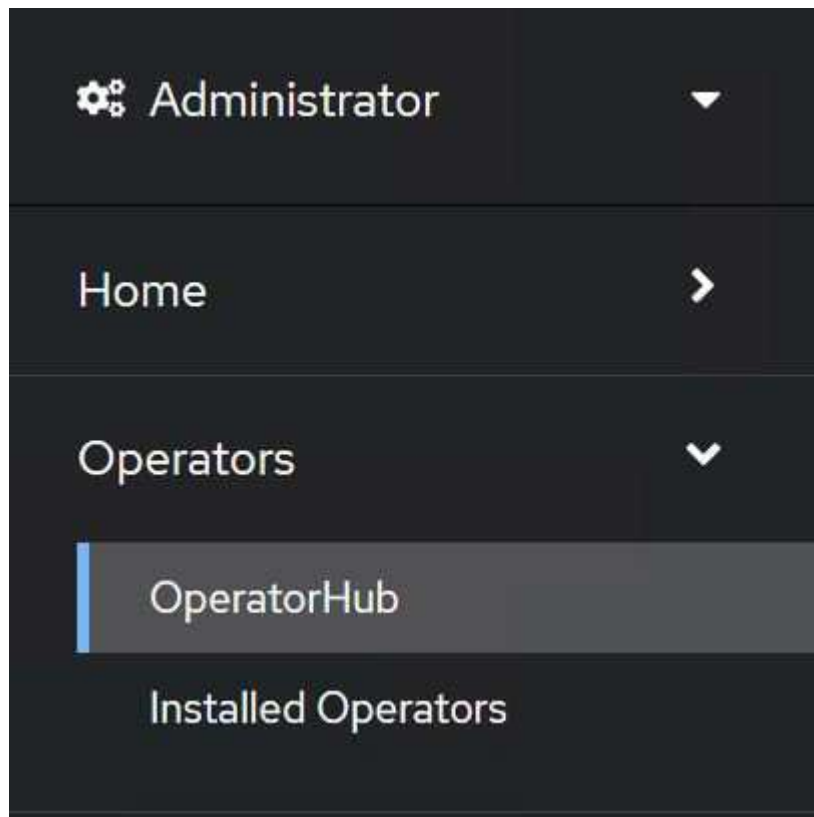
使用 Trident CSI 的另一个优势：在 OpenShift Virtualization 中，虚拟机的引导磁盘需要具有 RWX 访问模式才能进行虚拟机的实时迁移。Trident 支持对使用 NAS 协议以及使用 SAN 协议（当 volumeMode 设置为 Block 而不是 FileSystem 时）配置 national 的卷使用 **ReadWriteMany** 访问模式。您可以对引导磁盘使用任一协议。存储配置文件对象将根据 StorageClass 设置自动进行配置，并且在配置虚拟机时将相应地设置引导磁盘访问模式。

将 Trident StorageClass 设置为集群的默认值后，您可以为 Trident 创建具有不同参数的附加 StorageClasses，OpenShift Virtualization 将使用为其创建的适当存储配置文件自动识别它们。然后，您可以在 VM 定义中指定用于 VM 启动磁盘和数据磁盘的 StorageClass。

使用 OpenShift OperatorHub 中提供的 OpenShift Virtualization 运算符在 OpenShift 集群上安装 OpenShift Virtualization 是一个简单的过程。有关如何安装 OpenShift Virtualization 的详细步骤，请参见文档 ["此处"](#)。

在 Red Hat OpenShift 裸机集群上安装 OpenShift Virtualization。此过程包括使用集群管理员访问权限登录、导航到 OperatorHub 以及安装 OpenShift Virtualization 操作员。

1. 使用集群管理员访问权限登录 Red Hat OpenShift 裸机集群。
2. 从“视角”下拉菜单中选择“管理员”。
3. 导航到 Operators > OperatorHub 并搜索 OpenShift Virtualization。



4. 选择 OpenShift 虚拟化磁贴并单击安装。



OpenShift Virtualization

2.6.2 provided by Red Hat

[Install](#)

Latest version
2.6.2

Capability level

- ☒ Basic Install
- ☒ Seamless Upgrades
- ☒ Full Lifecycle
- ☐ Deep Insights
- ☐ Auto Pilot

Provider type
Red Hat

Provider
Red Hat

Requirements

Your cluster must be installed on bare metal infrastructure with Red Hat Enterprise Linux CoreOS workers.

Details

OpenShift Virtualization extends Red Hat OpenShift Container Platform, allowing you to host and manage virtualized workloads on the same platform as container-based workloads. From the OpenShift Container Platform web console, you can import a VMware virtual machine from vSphere, create new or clone existing VMs, perform live migrations between nodes, and more. You can use OpenShift Virtualization to manage both Linux and Windows VMs.

The technology behind OpenShift Virtualization is developed in the [KubeVirt](#) open source community. The KubeVirt project extends [Kubernetes](#) by adding additional virtualization resource types through [Custom Resource Definitions](#) (CRDs). Administrators can use Custom Resource Definitions to manage [VirtualMachine](#) resources alongside all other resources that Kubernetes provides.

5. 在“安装操作员”屏幕上，保留所有默认参数并单击“安装”。

Update channel *

- ☐ 2.1
- ☐ 2.2
- ☐ 2.3
- ☐ 2.4
- ☒ stable

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** openshift-cnv

i Namespace creation

Namespace **openshift-cnv** does not exist and will be created.

- ☐ Select a Namespace

Approval strategy *

- ☒ Automatic
- ☐ Manual

Install

Cancel



OpenShift Virtualization
provided by Red Hat

Provided APIs

HC OpenShift
Virtualization
Deployment

i Required

Represents the deployment of
OpenShift Virtualization

6. 等待操作员安装完成。



OpenShift Virtualization
2.6.2 provided by Red Hat



Installing Operator

The Operator is being installed. This may take a few minutes.

[View installed Operators in Namespace openshift-cnv](#)

7. 操作员安装完成后，单击创建超融合。



Installed operator - operand required

The Operator has installed successfully. Create the required custom resource to be able to use this Operator.



HyperConverged



Required

Creates and maintains an OpenShift Virtualization Deployment

Create HyperConverged

[View installed Operators in Namespace openshift-cnv](#)

8. 在“创建超融合”屏幕上，单击“创建”，接受所有默认参数。此步骤开始安装 OpenShift Virtualization。

Name *

Labels

Infra >

infra HyperConvergedConfig influences the pod configuration (currently only placement) for all the infra components needed on the virtualization enabled cluster but not necessarily directly on each node running VMs/VMLs.

Workloads >

workloads HyperConvergedConfig influences the pod configuration (currently only placement) of components which need to be running on a node where virtualization workloads should be able to run. Changes to Workloads HyperConvergedConfig can be applied only without existing workload.

Bare Metal Platform

☒ true

BareMetalPlatform indicates whether the infrastructure is baremetal.

Feature Gates >

featureGates is a map of feature gate flags. Setting a flag to "true" will enable the feature. Setting "false" or removing the feature gate, disables the feature.

Local Storage Class Name

LocalStorageClassName the name of the local storage class.

9. 当 openshift-cnv 命名空间中的所有 pod 都转为 Running 状态，并且 OpenShift Virtualization Operator 处于 Succeeded 状态后，该 Operator 即可使用。现在可以在 OpenShift 集群上创建虚拟机。

Project: openshift-cnv ▾

Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#) or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

Name ▾

Search by name...

/

Name ↑	Managed Namespaces ⌵	Status	Last updated	Provided APIs
 OpenShift Virtualization 2.6.2 provided by Red Hat	NS openshift-cnv	✔ Succeeded Up to date	🕒 May 18, 8:02 pm	OpenShift Virtualization Deployment HostPathProvisioner deployment ⋮

使用带有 Red Hat OpenShift 虚拟化的 ONTAP 存储创建虚拟机

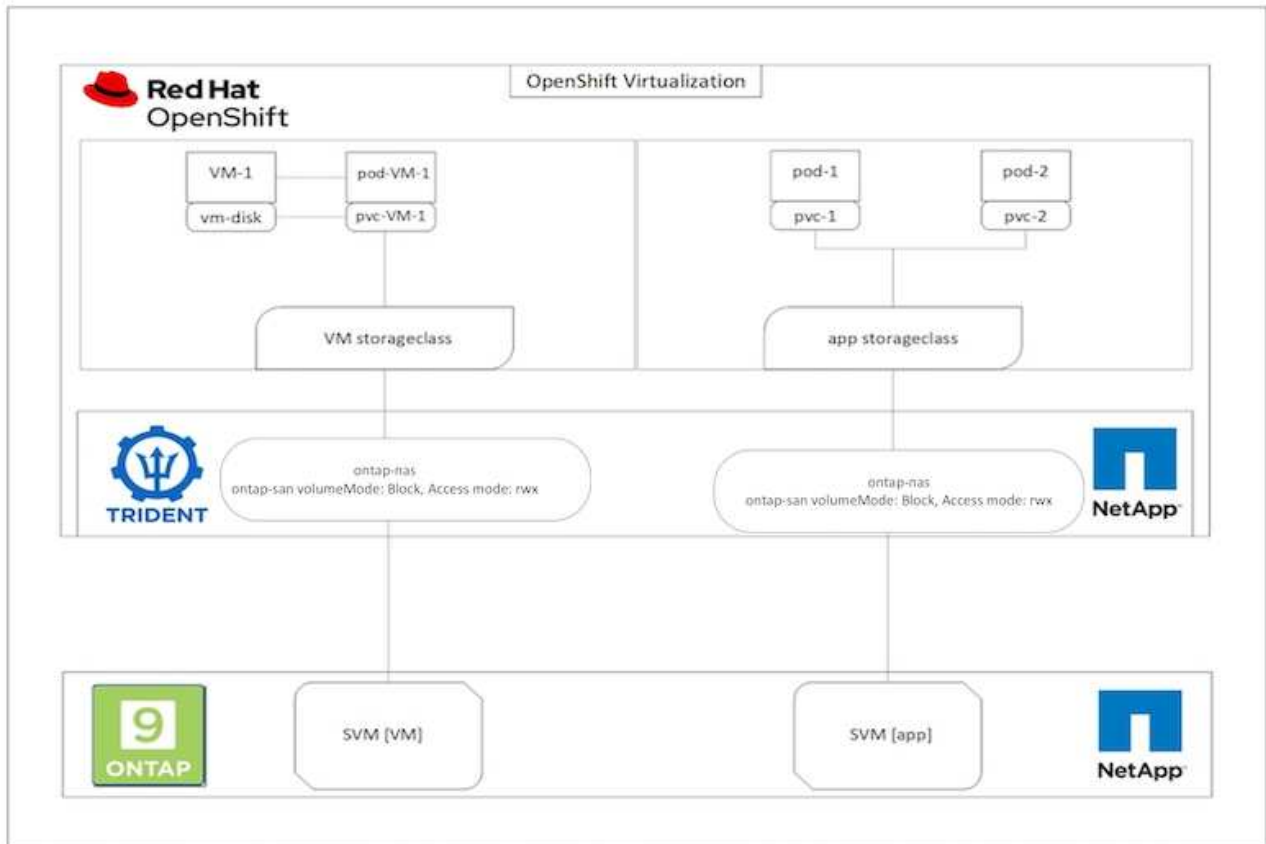
使用 OpenShift Virtualization 创建虚拟机。此过程包括选择操作系统模板、配置 StorageClasses 以及自定义 VM 参数以满足特定要求。作为先决条件，您应已创建 Trident 后端、StorageClass 和卷快照类对象。有关详细信息，请参阅 ["Trident安装部分"](#)。

创建 VM

虚拟机是有状态的部署，需要卷来托管操作系统和数据。使用 CNV，由于虚拟机作为 pod 运行，因此虚拟机由通过 Trident 托管在 NetApp ONTAP 上的 PV 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。



必须将 Trident StorageClass 设置为集群上的默认 StorageClass，以确保为 VM 磁盘创建的 PVC 使用 Trident 后端进行配置。集群上可以有多个 StorageClasses，但默认 StorageClass 必须用于为 VM 启动磁盘配置 PVC。如果需要，其他磁盘可以使用其他 StorageClasses。



要在 OpenShift 集群上快速创建虚拟机，请完成以下步骤：

1. 导航到“虚拟化”>“虚拟机”，然后单击“创建”。
2. 选择来自模板。
3. 选择所需的可用启动源的操作系统。
4. 选中复选框“创建后启动虚拟机”。
5. 单击快速创建虚拟机。

虚拟机已创建并启动，并进入 **Running** 状态。它使用默认 StorageClass 自动为启动磁盘创建 PVC 和相应的 PV。为了能够在将来对虚拟机执行实时迁移，您必须确保用于磁盘的 StorageClass 能够支持 RWX 卷。这是实时迁移的要求。ontap-nas 和 ontap-san (iSCSI 和 NVMe/TCP 协议的 volumeMode 块) 可以支持使用各自 StorageClasses 创建的卷的 RWX 访问模式。

要在集群上配置 ontap-san StorageClass，请参阅 ["用于配置 StorageClass 的部分"](#)。



单击 Quick create VirtualMachine 将使用默认的 StorageClass 来为 VM 的可引导根磁盘创建 PVC 和 PV。您可以通过选择 Customize VirtualMachine > Customize VirtualMachine parameters > Disks，然后编辑磁盘以使用所需的 StorageClass，从而为该磁盘选择其他 StorageClass。

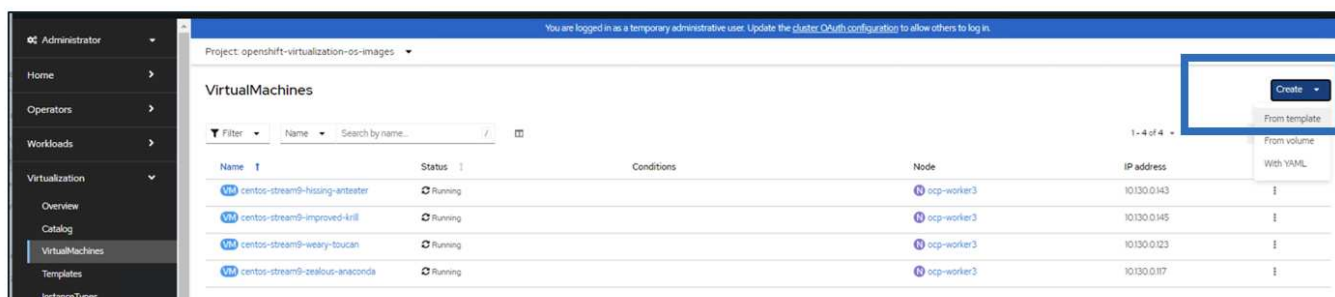
在配置虚拟机磁盘时，通常与文件系统相比，块访问模式更受欢迎。

要在选择操作系统模板后自定义虚拟机创建，请单击“自定义虚拟机”而不是“快速创建”。

1. 如果选择的操作系统已经配置了启动源，可以点击*自定义虚拟机参数*。
2. 如果所选操作系统没有配置启动源，则必须配置它。您可以查看["文档"](#)。
3. 配置完启动盘后，可以点击*自定义虚拟机参数*。
4. 您可以从此页面上的选项卡自定义虚拟机。例如，单击“磁盘”选项卡，然后单击“添加磁盘”向虚拟机添加另一个磁盘。
5. 单击“创建虚拟机”以创建虚拟机；这会在后台启动相应的 pod。



当从 URL 或注册表为模板或操作系统配置启动源时，它会在 `openshift-virtualization-os-images` 项目并将 KVM 客户映像下载到 PVC。您必须确保模板 PVC 有足够的配置空间来容纳相应操作系统的 KVM 客户机映像。然后，当使用任何项目中的相应模板创建虚拟机时，这些 PVC 将被克隆并作为根磁盘附加到虚拟机。



Create new VirtualMachine

Select an option to create a VirtualMachine from.

Template project

All projects

Default templates

User templates

☐ Boot source available

Operating system

☐ CentOS

☐ Fedora

☐ Other

☐ RHEL

☐ Windows

Workload

☐ Desktop

☐ High performance

☐ Server

Default templates

Q Filter by keyword...

13 items

Source available

 **CentOS Stream 8 VM**
centos-stream8-server-small

Project openshift
Boot source PVC (auto import)
Workload Server
CPU 1
Memory 2 GiB

Source available

 **CentOS Stream 9 VM**
centos-stream9-server-small

Project openshift
Boot source PVC (auto import)
Workload Server
CPU 1
Memory 2 GiB

Source available

 **CentOS 7 VM**
centos7-server-small

Project openshift
Boot source PVC (auto import)
Workload Server
CPU 1
Memory 2 GiB

Source available

 **Fedora VM**
fedora-server-small

Project openshift
Boot source PVC (auto import)
Workload Server
CPU 1
Memory 2 GiB

Source available

 **Red Hat Enterprise Linux 7 VM**
rhel7-server-small

Project openshift
Boot source PVC (auto import)
Workload Server
CPU 1
Memory 2 GiB

Source available

 **Red Hat Enterprise Linux 8 VM**
rhel8-server-small

Project openshift
Boot source PVC (auto import)
Workload Server
CPU 1
Memory 2 GiB

Source available

 **Red Hat Enterprise Linux 9 VM**
rhel9-server-small

Project openshift
Boot source PVC (auto import)
Workload Server
CPU 1
Memory 2 GiB

Source available

 **Microsoft Windows 10 VM**
windows10-desktop-medium

Project openshift
Boot source PVC
Workload Desktop
CPU 1
Memory 4 GiB

Source available

 **Microsoft Windows 11 VM**
windows11-desktop-medium

Project openshift
Boot source PVC
Workload Desktop
CPU 2
Memory 4 GiB

Source available

 **Microsoft Windows Server 2012 R2 VM**
windows2k12r2-server-medium

Project openshift
Boot source PVC
Workload Server
CPU 1
Memory 4 GiB

VirtualMachine details

VM centos-stream9-zealous-anaconda Running VAML Actions

Overview Details Metrics VAML **Configuration** Events Console Snapshots Diagnostics

Disk **Add disk**

Network interfaces

Scheduling

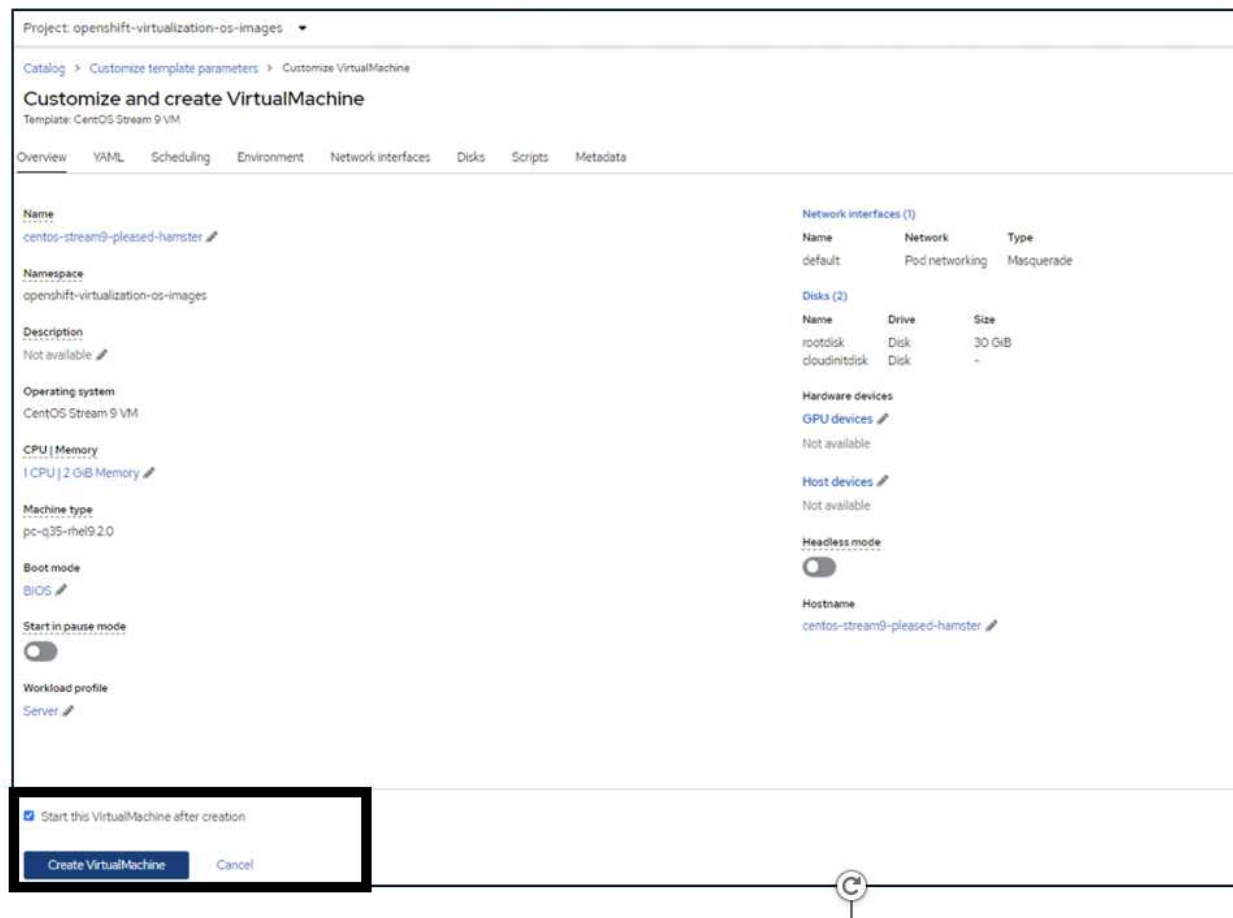
Environment

Scripts

Name	Source	Size	Drive	Interface	Storage class
cloudinitdisk	Other	-	Disk	virtio	-
data-disk1 Persistent Hotplug	 centos-stream9-zealous-anaconda-data-disk1	30.00 GiB	Disk	SCSI	ontap-san-block
rootdisk bootable	 centos-stream9-zealous-anaconda	30.00 GiB	Disk	virtio	ontap-san-block

File systems

Name	File system type	Mount point	Total bytes	Used bytes
vdal	xfs	/	29.94 GiB	1.30 GiB



视频演示

以下视频演示了如何使用 iSCSI 存储在 OpenShift Virtualization 中创建 VM。

使用块存储在 [OpenShift 虚拟化中创建虚拟机](#)

通过替代方法使用 ONTAP 存储，利用 Red Hat OpenShift Virtualization 创建虚拟机

除了使用 OpenShift 虚拟化 UI 创建虚拟机的方法外，还有其它方法可以在 OpenShift 虚拟化上创建虚拟机。本节概述了其中一些方法。

使用红帽 OpenShift 虚拟化从 YAML 文件创建虚拟机

也可以通过使用 `oc apply` 命令应用 YAML 文件来创建虚拟机。对于喜欢使用 CLI 或希望使用脚本自动创建虚拟机的用户，此方法非常有用。要使用此方法创建虚拟机，可以创建定义虚拟机及其关联资源（如虚拟机磁盘的 PVC）的 YAML 文件。然后，您可以使用 `oc apply -f <yaml-file>` 命令应用 YAML 文件，在 OpenShift 集群上创建虚拟机。

```
apiVersion: kubevirt.io/v1
kind: VirtualMachine
metadata:
  name: test-vms-vm
  namespace: test-5vms
  labels:
    app: test-vms-vm
    vm.kubevirt.io/template: centos-stream9-server-small
spec:
  dataVolumeTemplates:
    - apiVersion: cdi.kubevirt.io/v1beta1
      kind: DataVolume
      metadata:
        name: test-vms-vm-rootdisk
      spec:
        sourceRef:
          kind: DataSource
          name: centos-stream9
          namespace: openshift-virtualization-os-images
        storage:
          resources:
            requests:
              storage: 30Gi
  runStrategy: Always
  template:
    metadata:
      labels:
        kubevirt.io/domain: test-vms-vm
    spec:
      domain:
        cpu:
          cores: 1
          sockets: 1
          threads: 1
        devices:
          disks:
            - bootOrder: 1
              disk:
                bus: virtio
                name: rootdisk
            - disk:
                bus: virtio
                name: cloudinitdisk
```

```

    interfaces:
      - masquerade: {}
        model: virtio
        name: default
    resources:
      requests:
        memory: 2Gi
    networks:
      - name: default
        pod: {}
    terminationGracePeriodSeconds: 180
    volumes:
      - dataVolume:
          name: test-vms-vm-rootdisk
          name: rootdisk
      - cloudInitNoCloud:
          userData: |-
            #cloud-config
            user: admin
            password: <replace with your password>
            chpasswd: { expire: False }
          name: cloudinitdisk

```

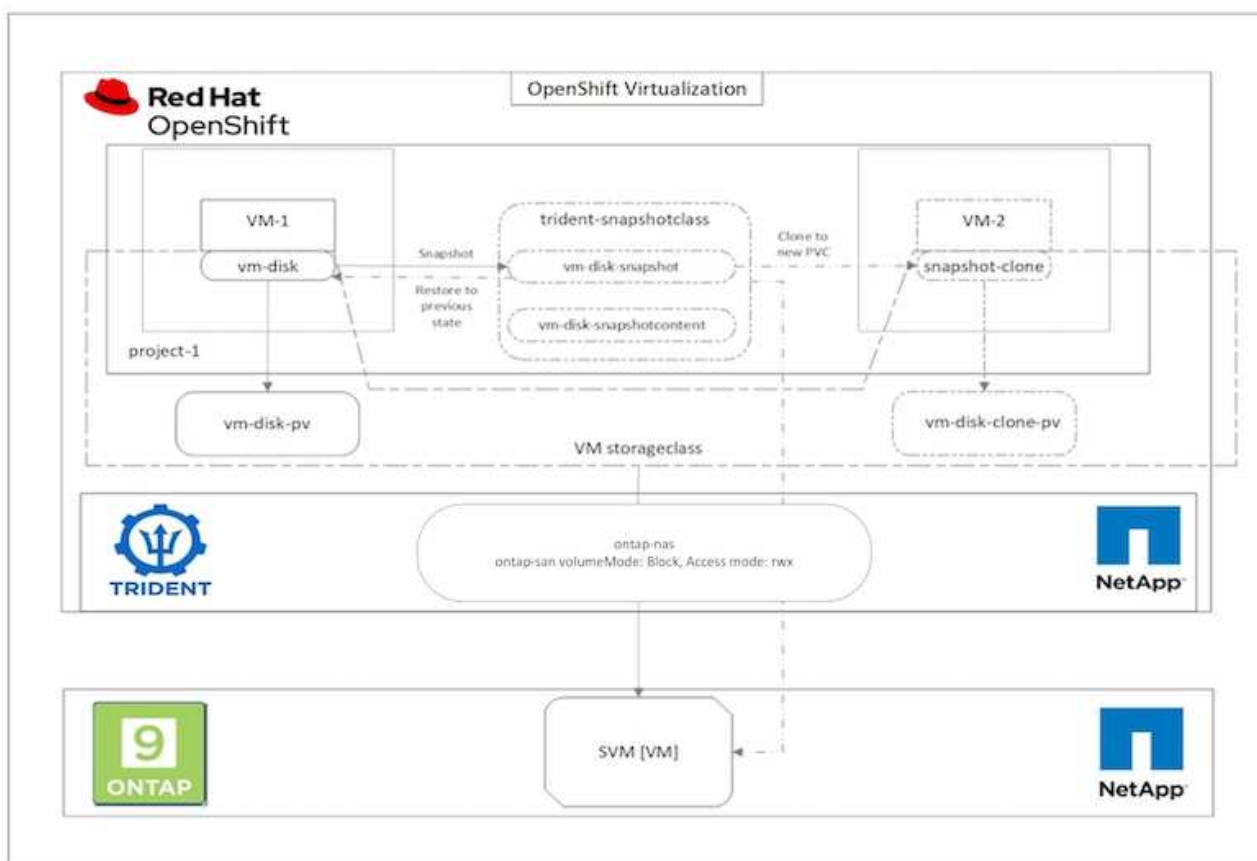
使用 Red Hat OpenShift Virtualization 从快照副本创建 VM

您可以使用 OpenShift Virtualization 从快照创建虚拟机。此过程包括创建 VolumeSnapshotClass、获取 VM 的持久卷声明 (PVC) 的快照、将快照还原到新的 PVC，以及部署使用还原的 PVC 作为根磁盘的新 VM。

借助 Trident 和 Red Hat OpenShift，用户可以对其所配置的存储类上的持久卷进行快照。利用此功能，用户可以获取卷的某个时间点的副本并使用它来创建新卷或将同一卷恢复到以前的状态。这启用或支持各种用例，从回滚到克隆到数据恢复。

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

- VolumeSnapshotContent 是从集群中的卷获取的实际快照。它是类似于用于存储的 PersistentVolume 的集群范围的资源。
- VolumeSnapshot 是创建卷快照的请求。它类似于 PersistentVolumeClaim。
- VolumeSnapshotClass 允许管理员为 VolumeSnapshot 指定不同的属性。它允许您对从同一卷获取的不同快照设置不同的属性。



先决条件

您必须在 Trident 存储类上配置已附加 PVC 的现有虚拟机，并且该虚拟机必须处于运行状态，然后才能拍摄 PVC 的快照。您必须创建一个 VolumeSnapshotClass，以便用于创建 VolumeSnapshot。这将作为安装 OpenShift 虚拟化的先决条件的一部分创建。有关详细信息，请参阅 ["第 3 步创建 VolumeSnapshotClass 配置"](#)。

要创建虚拟机的快照，请完成以下步骤：

1. 识别连接到源虚拟机的 PVC，然后创建该 PVC 的快照。导航至 `Storage > VolumeSnapshots` 并单击创建卷快照。
2. 选择要为其创建 Snapshot 的 PVC，输入 Snapshot 的名称或接受默认值，并选择适当的 VolumeSnapshotClass。然后单击“创建”。
3. 这将创建该时间点的 PVC 快照。

Create VolumeSnapshot

[Edit YAML](#)

PersistentVolumeClaim *

PVC rhel8-short-frog-rootdisk-28dvv

Name *

rhel8-short-frog-rootdisk-28dvv-snapshot

Snapshot Class *

VSC trident-snapshot-class

Create

Cancel

从快照创建新虚拟机

1. 首先，将快照恢复到新的 PVC 中。导航到存储 > VolumeSnapshots，单击要恢复的快照旁边的省略号，然后单击恢复为新 PVC。
2. 输入新 PVC 的详细信息，然后单击“恢复”。这将创建一个新的 PVC。

Restore as new PVC

When restore action for snapshot **rhel8-short-frog-rootdisk-28dvb-snapshot** is finished a new crash-consistent PVC copy will be created.

Name *

rhel8-short-frog-rootdisk-28dvb-snapshot-restore

Storage Class *



basic

Access Mode *

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

Size *

20

GiB

VolumeSnapshot details

Created at

 May 21, 12:46 am

Namespace

 default

Status

 Ready

API version

snapshot.storage.k8s.io/v1

Size

20 GiB

1. 接下来，从此 PVC 创建一个新的 VM。导航到“虚拟化”>“虚拟机”，然后单击“创建”>“使用 YAML”。
2. 在 `spec > template > spec > volumes` 部分中，指定从快照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
```

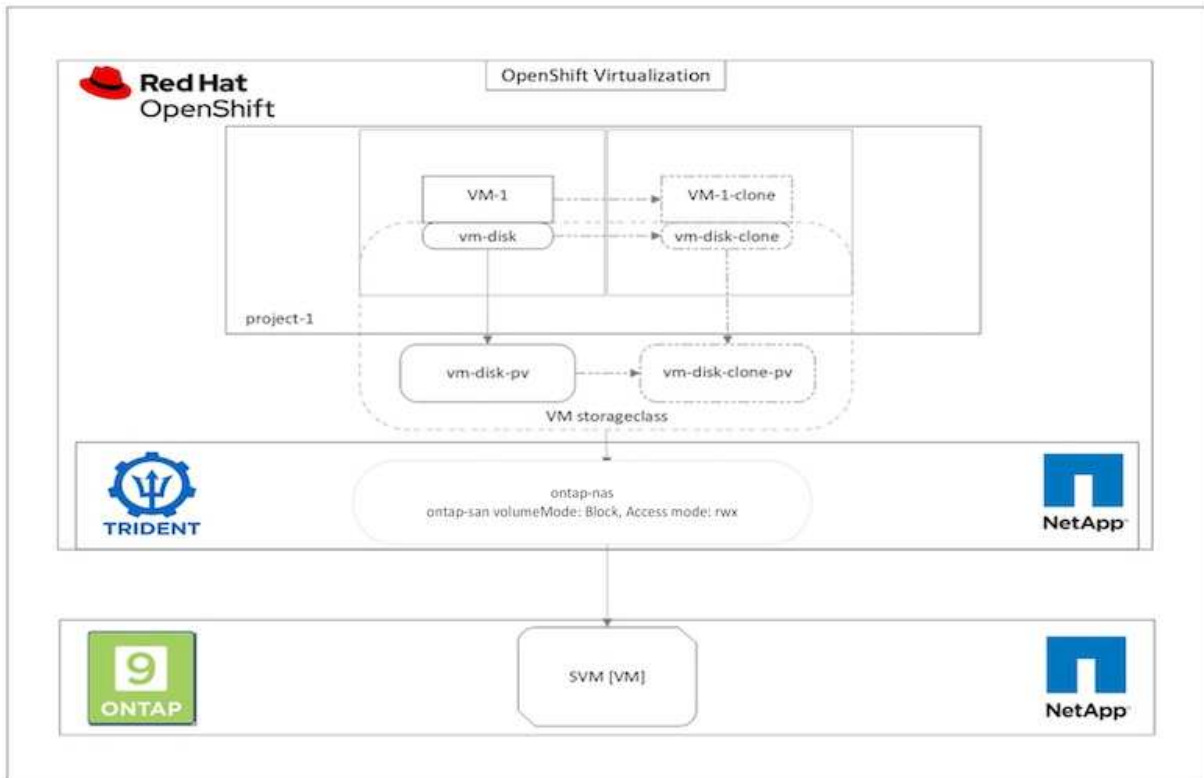
1. 单击“创建”以创建新的虚拟机。
2. 虚拟机创建成功后，访问并验证新虚拟机与创建快照时使用 PVC 创建的虚拟机的状态是否相同。

使用 **Trident** 在 **OpenShift Virtualization** 中克隆虚拟机

此过程包括利用 Trident CSI 卷克隆，允许您通过关闭源虚拟机或保持其运行来创建新的虚拟机。

虚拟机克隆

在 Trident 的 Volume CSI 克隆功能的支持下，可以克隆 OpenShift 中现有的 VM。CSI 卷克隆允许通过复制 PV 来使用现有 PVC 作为数据源创建新的 PVC。新的 PVC 创建新之后，它将作为一个独立的实体运行，并且与源 PVC 没有任何链接或依赖关系。



对于 CSI 卷克隆，需要考虑以下某些限制：

1. 源 PVC 和目标 PVC 必须在同一个项目中。
2. 同一存储类内支持克隆。
3. 仅当源卷和目标卷使用相同的 VolumeMode 设置时才可以执行克隆；例如，一个块卷只能克隆到另一个块卷。

OpenShift 集群中的虚拟机可以通过两种方式克隆：

1. 通过关闭源虚拟机
2. 通过保持源虚拟机处于活动状态

通过关闭源虚拟机

通过关闭 VM 来克隆现有 VM 是 OpenShift 的一项原生功能，是在 Trident 的支持下实现的。完成以下步骤来克隆虚拟机。

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。

- 单击克隆虚拟机并提供新虚拟机的详细信息。

Clone Virtual Machine

Name *	rhel8-short-frog-clone
Description	
Namespace *	default
	☒ Start virtual machine on clone
Configuration	Operating System Red Hat Enterprise Linux 8.0 or higher Flavor Small: 1 CPU 2 GiB Memory Workload Profile server NICs default - virtio Disks cloudinitdisk - cloud-init disk rootdisk - 20Gi - basic

 **The VM rhel8-short-frog is still running. It will be powered off while cloning.**

- 单击克隆虚拟机；这将关闭源虚拟机并启动克隆虚拟机的创建。
- 此步骤完成后，您可以访问并验证克隆的VM的内容。

通过保持源虚拟机处于活动状态

还可以通过克隆源 VM 的现有 PVC，然后使用克隆的 PVC 创建新 VM 来克隆现有 VM。此方法不需要您关闭源虚拟机。完成以下步骤即可在不关闭虚拟机的情况下克隆虚拟机。

1. 导航到存储 > 持久卷声明，然后单击连接到源虚拟机的 PVC 旁边的省略号。
2. 单击克隆 PVC 并提供新 PVC 的详细信息。

Clone

Name *

rhel8-short-frog-rootdisk-28dvb-clone

Access Mode *

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

Size *

20 GiB ▼

PVC details

Namespace	Requested capacity	Access mode
 default	20 GiB	Shared Access (RWX)
Storage Class	Used capacity	Volume mode
 basic	2.2 GiB	Filesystem

Cancel

Clone

3. 然后单击克隆。这会为新的 VM 创建 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 spec > template > spec > volumes 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
```

1. 单击“创建”以创建新的虚拟机。
2. VM创建成功后，访问并验证新VM是否为源VM的克隆。

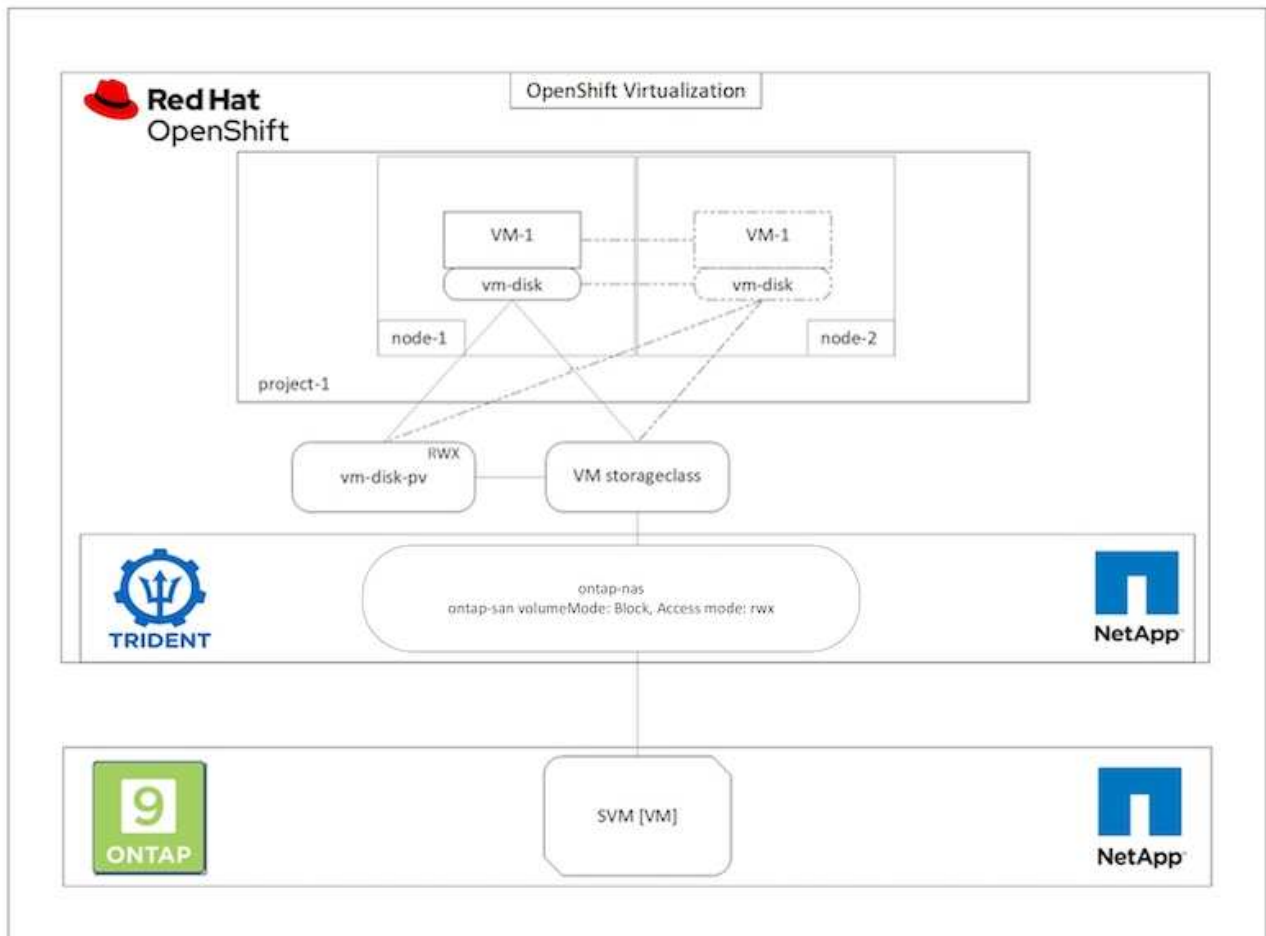
在 Red Hat OpenShift 集群中的两个节点之间迁移虚拟机

在集群中的两个节点之间迁移 OpenShift Virtualization 中的 VM，无需停机。此过程包括确认磁盘使用与 RWX 兼容的存储类、启动迁移以及监控进度。

虚拟机实时迁移

实时迁移是将虚拟机实例从 OpenShift 集群中的一个节点迁移到另一个节点的过程，无需停机。为了在 OpenShift 集群中进行实时迁移，虚拟机必须绑定到具有共享 ReadWriteMany 访问模式的 PVC。使用 ontap-nas 驱动程序配置的Trident后端支持文件系统协议 nfs 和 smb 的 RWX 访问模式。请参阅文档[此处](#)。使用 ontap-san 驱动程序配置的Trident后端支持 iSCSI 和 NVMe/TCP 协议的块卷模式的 RWX 访问模式。请参阅文档[此处](#)。

因此，为了成功实现实时迁移，必须使用 ontap-nas 或 ontap-san (volumeMode: Block) 存储类为虚拟机配置具有 PVC 的磁盘（启动磁盘和附加热插拔磁盘）。创建 PVC 时，Trident会在启用 NFS 或 iSCSI 的 SVM 中创建ONTAP卷。

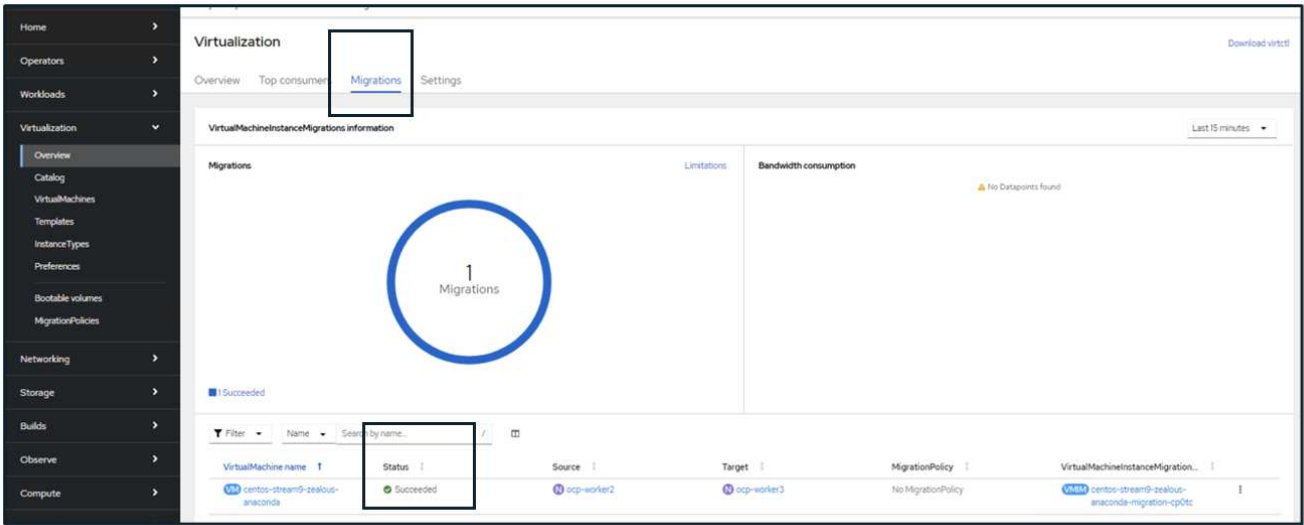
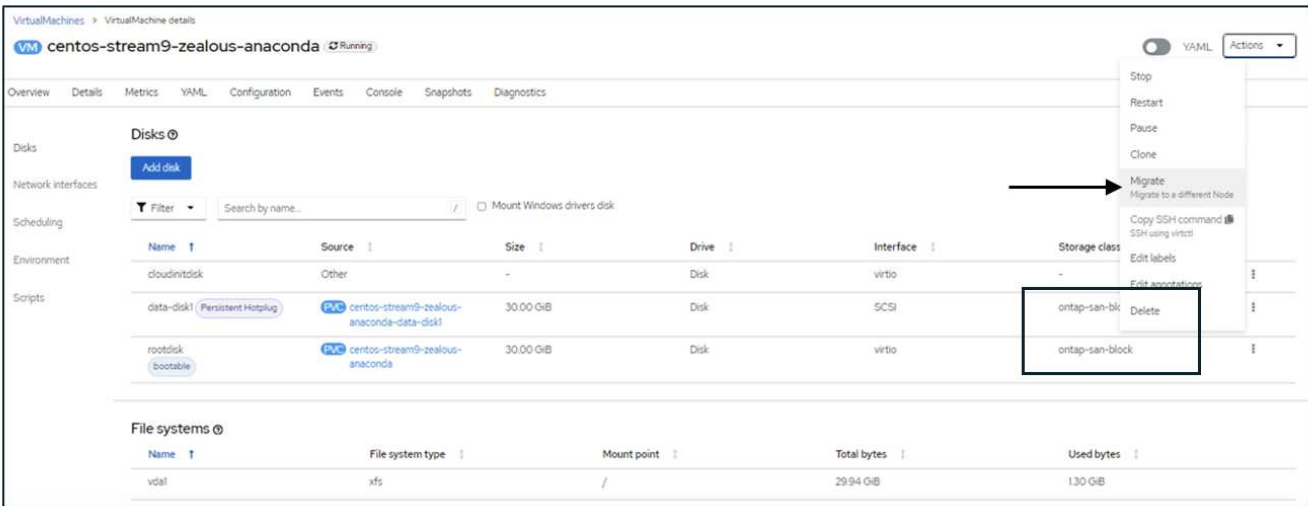


要对先前已创建且处于运行状态的 VM 执行实时迁移，请执行以下步骤：

1. 选择您想要实时迁移的虚拟机。
2. 单击“配置”选项卡。
3. 确保虚拟机的所有磁盘都是使用可以支持 RWX 访问模式的存储类创建的。
4. 点击右上角的*操作*，然后选择*迁移*。
5. 要查看迁移的进度，请转到左侧菜单上的“虚拟化”>“概述”，然后单击“迁移”选项卡。VM 的迁移将从 待定 过渡到 计划 再到 成功



如果将 evictionStrategy 设置为 LiveMigrate，则当原始节点处于维护模式时，OpenShift 集群中的 VM 实例会自动迁移到另一个节点。



OpenShift Virtualization 中存储类之间 VM 的存储迁移

在 OpenShift 虚拟化中，您可以通过 Web 控制台将虚拟机（VM）的磁盘迁移到其他

存储类，而无需删除 VM。此过程允许您移动整个虚拟机存储或特定卷。

以下步骤概述了如何迁移虚拟机的存储类：

- 1. 请确保目标存储类已正确配置并在 OpenShift 集群中可用。这包括设置和访问必要的存储后端（例如，NetApp ONTAP）。

显示示例

```
[root@00-50-56-b5-2b-9b ~]# oc get tbe -n trident
```

NAME	BACKEND	BACKEND UUID
tbe-k2vqg	tbc-san	3e9dc8ae-3b29-44c8-99a9-affdeaa6b4cc
tbe-mtp5n	tbc-nas-eco	a59e8053-bb00-4410-9b12-1dc4d283928b
tbe-nwqvq	tbc-nas-50	8b49f02f-3bc3-4720-b60f-bf90c3600008
tbe-wtp9h	tbc_san_eco	61745526-04cd-4acb-8f1b-daa9aa79f05a
tbe-zfp46	tbc-nas	259a1c2a-0a84-488d-ab62-ccea45499e09


```
[root@00-50-56-b5-2b-9b ~]# oc get sc
```

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
sc-nas (default)	csi.trident.netapp.io	Delete	Immediate	true	35d
sc-nas-50	csi.trident.netapp.io	Delete	Immediate	true	22d
sc-nas-economy	csi.trident.netapp.io	Delete	Immediate	true	20d
sc-san	csi.trident.netapp.io	Delete	Immediate	false	21d

- 2. 在 OpenShift 容器平台 Web 控制台中，导航到虚拟化视角，然后选择要迁移的虚拟机。

显示示例

VirtualMachines > VirtualMachine details

VM rhel9-sc-migration Running

OverviewMetricsYAMLConfigurationEventsConsoleSnapshotsDiagnostics

Details

Name

rhel9-sc-migration

Status

Running

Created

May 14, 2026, 1:31 PM (17 minutes ago)

Operating system

Red Hat Enterprise Linux 9.7 (Plow)

CPU | Memory

1 CPU | 2 GiB Memory

Time zone

EDT

Template

rhel9-server-small

Hostname

rhel9-sc-migration

VNC console

Open web console

Alerts (0)

General

Namespace

test-sc-migration

Node

00-50-56-b5-b6-fa

VirtualMachineInstance

rhel9-sc-migration

Pod

virt-launcher-rhel9-sc-migration-5jdw

Owner

No owner

Snapshots (0)

Take snapshot

VirtualMachines > VirtualMachine details

VM rhel9-sc-migration Running

OverviewMetricsYAMLConfigurationEventsConsoleSnapshotsDiagnostics

Configuration

Disks

Add disk

Filter

Search by name...

☐ Mount Windows drivers disk

	Name	Source	Size	Drive	Interface	Storage class
Scheduling	cloudinitdisk	Other	-	Disk	virtio	-
SSH	disk-rhel9-nas	PVC dv-rhel9-migration-disk-emerald-wren-75-d88usc	53 GiB	Disk	virtio	sc-nas
Initial run	rootdisk	PVC rhel9-migration	31.8 GiB	Disk	virtio	sc-nas
Metadata	bootable					

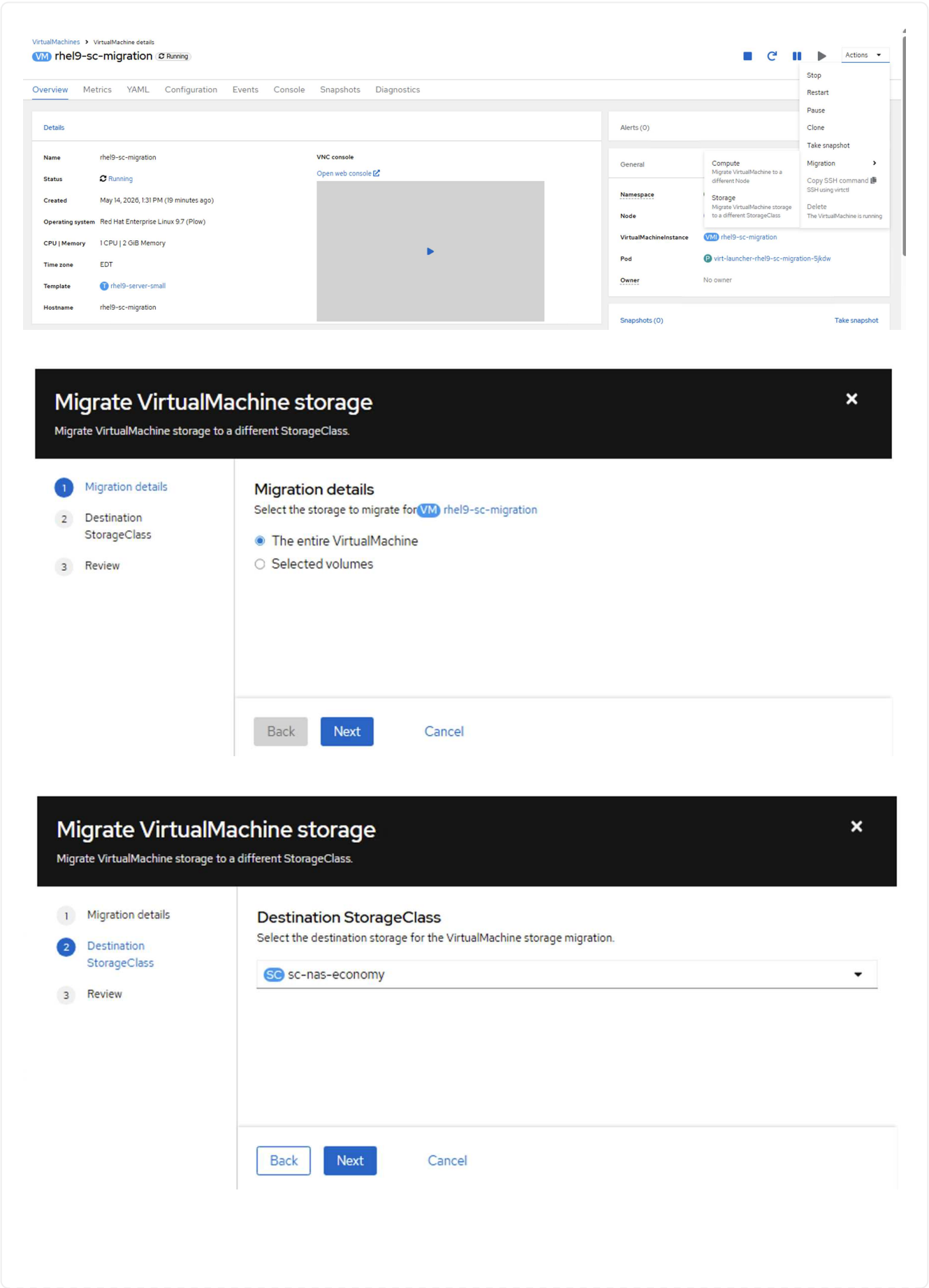
[root@00-50-56-b5-2b-9b ~]# oc get vm,pvc -n test-sc-migration

NAME	AGE	STATUS	READY
virtualmachine.kubevirt.io/rhel9-sc-migration	4m19s	Running	True

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	VOLUMEATTRIBUTESCLASS	AGE
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc	Bound	pvc-bb54342f-c01f-453e-864d-e8f72e710a2f	53Gi	RWX	sc-nas	<unset>	4m19s
persistentvolumeclaim/rhel9-migration	Bound	pvc-46a34903-942f-4ddc-8e6d-97a008f05d09	34144990004	RWX	sc-nas	<unset>	4m19s

[root@00-50-56-b5-2b-9b ~]#

3. 单击操作→迁移→迁移存储。选择要迁移的卷和目标存储类，然后单击迁移。迁移过程将开始，您可以在 VM 的"事件"选项卡中监控进度。迁移完成后，VM 的磁盘将使用新的存储类。



Migrate VirtualMachine storage

Migrate VirtualMachine storage to a different StorageClass.

1 Migration details

2 Destination StorageClass

3 Review

Review

Verify the details and click **Migrate VirtualMachine storage** to start the migration

Migration type

VirtualMachine storage

Destination StorageClass

sc-nas-economy

Migrating 1 out of 3

disk-rhel9-nas

Back

Migrate VirtualMachine storage

Cancel

Migrate VirtualMachine storage

Migrate VirtualMachine storage to a different StorageClass.

In progress

Started on

May 14, 2026, 1:54 PM

Status

Scheduling

Stop

Cancel

View report

4. 迁移完成后，您可以通过在 Web 控制台中检查虚拟机的详细信息或使用 OpenShift CLI 来验证虚拟机的磁盘现在是否正在使用新的存储类。

显示示例

```
[root@00-50-56-b5-2b-9b ~]# oc get vm,pvc -n test-sc-migration
NAME                                     AGE      STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration  24m     Migrating True

NAME                                     STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc  Bound   pvc-bb54342f-c01f-453e-864d-e8f72e710a2f  53Gi      Rwx          sc-nas        <unset>                24m
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb  Bound   pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b  60322815673  Rwx          sc-nas-economy <unset>                2m17s
persistentvolumeclaim/rhel9-migration  Bound   pvc-46a34903-942f-4ddc-8e6d-974008f05d09  34144990004  Rwx          sc-nas        <unset>                24m
```

```
[root@00-50-56-b5-2b-9b ~]# oc get all -n test-sc-migration
Warning: apps.openshift.io/v1 DeploymentConfig is deprecated in v1.24*, unavailable in v1.10000+
Warning: kubevirt.io/v1 VirtualMachineInstancePresets is now deprecated and will be removed in v2.

NAME                                     READY  STATUS    RESTARTS  AGE
pod/virt-launcher-rhel9-sc-migration-44dc7  1/1    Running   0          2m50s
pod/virt-launcher-rhel9-sc-migration-5jkdw  0/1    Completed 0          25m

NAME      TYPE      CLUSTER-IP  EXTERNAL-IP  PORT(S)    AGE
service/headless  ClusterIP  None        <none>       5434/TCP   25m

NAME                                     PHASE    PROGRESS  RESTARTS  AGE
datavolume.cdi.kubevirt.io/dv-rhel9-migration-disk-emerald-wren-75-d88usc  Succeeded 100.0%    0          25m
datavolume.cdi.kubevirt.io/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb  Succeeded 100.0%    0          3m7s
datavolume.cdi.kubevirt.io/rhel9-migration  Succeeded 100.0%    0          25m

NAME                                     PHASE    VMI
virtualmachineinstancemigration.kubevirt.io/kubevirt-workload-update-lbfj9  Succeeded  rhel9-sc-migration

NAME                                     AGE    PHASE    IP          NODENAME  READY
virtualmachineinstance.kubevirt.io/rhel9-sc-migration  25m    Running  10.131.0.22  00-50-56-b5-84-74  True

NAME      AGE    STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration  25m    Running   True
[root@00-50-56-b5-2b-9b ~]#
[root@00-50-56-b5-2b-9b ~]# oc get vm,pvc -n test-sc-migration
NAME                                     AGE      STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration  26m     Running   True

NAME                                     STATUS  VOLUME                                     CAPACITY  ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc  Bound   pvc-bb54342f-c01f-453e-864d-e8f72e710a2f  53Gi      Rwx          sc-nas        <unset>                26m
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb  Bound   pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b  60322815673  Rwx          sc-nas-economy <unset>                3m28s
persistentvolumeclaim/rhel9-migration  Bound   pvc-46a34903-942f-4ddc-8e6d-974008f05d09  34144990004  Rwx          sc-nas        <unset>                26m
[root@00-50-56-b5-2b-9b ~]#
```

```
[root@00-50-56-b5-2b-9b ~]# oc describe pv pvc-46a34903-942f-4ddc-8e6d-974008f05d09 -n test-sc-migration
Name:      pvc-46a34903-942f-4ddc-8e6d-974008f05d09
Labels:    <none>
Annotations:  pv.kubernetes.io/provisioned-by: csi.trident.netapp.io
            volume.kubernetes.io/provisioner-deletion-secret-name:
            volume.kubernetes.io/provisioner-deletion-secret-namespace:
Finalizers: [external-provisioner.volume.kubernetes.io/finalizer kubernetes.io/pv-protection external-attacher/csi-trident-netapp-io]
StorageClass:  sc-nas
Status:        Bound
Claim:         test-sc-migration/rhel9-migration
Reclaim Policy: Delete
Access Modes:  Rwx
VolumeMode:    Filesystem
Capacity:      34144990004
Node Affinity: <none>
Message:
Source:
  Type:          CSI (a Container Storage Interface (CSI) volume source)
  Driver:        csi.trident.netapp.io
  FSType:
  VolumeHandle:  pvc-46a34903-942f-4ddc-8e6d-974008f05d09
  ReadOnly:      false
  VolumeAttributes:  backendUUID=259a1c2a-0a84-488d-ab62-ccca45499e09
                    internalName=oc_automation_pvc_46a34903_942f_4ddc_8e6d_974008f05d09
                    name=pvc-46a34903-942f-4ddc-8e6d-974008f05d09
                    protocol=file
                    storage.kubernetes.io/csiProvisionerIdentity=1778505459781-8296-csi.trident.netapp.io
Events:         <none>
[root@00-50-56-b5-2b-9b ~]#
```

```

NSOL-NetApp-A70-T19U01:> volume show -vserver oc-automation-svm -volume oc_automation_pvc_46a34903_942f_4ddc_8e6d_974008f05d09

Vserver Name: oc-automation-svm
Volume Name: oc_automation_pvc_46a34903_942f_4ddc_8e6d_974008f05d09
Aggregate Name: NSOL_NetApp_A70_T19U01a_NVME_SSD_1
List of Aggregates for FlexGroup Constituents: NSOL_NetApp_A70_T19U01a_
NVME_SSD_1
Encryption Type: none
List of Nodes Hosting the Volume: NSOL-NetApp-A70-T19U01a
Volume Size: 31.80GB
Volume Data Set ID: 1716
Volume Master Data Set ID: 2152408880
Volume State: online
Volume Style: flex
Extended Volume Style: flexvol
FlexCache Endpoint Type: none
Is Cluster-Mode Volume: true
Is Constituent Volume: false
Number of Constituent Volumes: -
Export Policy: default
User ID: 0
Group ID: 0
Security Style: unix
UNIX Permissions: ---rwxrwxrwx
Junction Path: /oc_automation_pvc_46a34903_942f_4ddc_8e6d_974008f05d09
Junction Path Source: RW_volume
Junction Active: true
Junction Parent Volume: oc_automation-svm_root
Comment:
Available Size: 30.03GB
Filesystem Size: 31.80GB
Total User-Visible Size: 31.80GB
Used Size: 1.76GB
Used Percentage: 5%
Volume Nearly Full Threshold Percent: 95%
Volume Full Threshold Percent: 98%
Maximum Autosize: 38.16GB
Minimum Autosize: 31.80GB

```

```

[root@00-50-56-b5-2b-9b ~]# oc get all -n test-sc-migration
Warning: apps.openshift.io/v1 DeploymentConfig is deprecated in v4.14+, unavailable in v4.10000+
Warning: kubevirt.io/v1 VirtualMachineInstancePresets is now deprecated and will be removed in v2.
NAME                                READY    STATUS    RESTARTS   AGE
pod/virt-launcher-rhel9-sc-migration-44dc7    1/1      Running    0           2m50s
pod/virt-launcher-rhel9-sc-migration-5jkdw    0/1      Completed  0           25m

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/headless                    ClusterIP     None          <none>         5434/TCP    25m

NAME                                PHASE    PROGRESS    RESTARTS   AGE
datavolume.cdi.kubevirt.io/dv-rhel9-migration-disk-emerald-wren-75-d88usc    Succeeded  100.0%      0           25m
datavolume.cdi.kubevirt.io/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb    Succeeded  100.0%      0           3m7s
datavolume.cdi.kubevirt.io/rhel9-migration                                Succeeded  100.0%      0           25m

NAME                                PHASE    VMI
virtualmachineinstancemigration.kubevirt.io/kubevirt-workload-update-lbfj9    Succeeded  rhel9-sc-migration

NAME                                AGE    PHASE    IP            NODENAME    READY
virtualmachineinstance.kubevirt.io/rhel9-sc-migration    25m    Running  10.131.0.22   00-50-56-b5-84-74    True

NAME                                AGE    STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration    25m    Running    True
[root@00-50-56-b5-2b-9b ~]#
[root@00-50-56-b5-2b-9b ~]#
[root@00-50-56-b5-2b-9b ~]# oc get vm,pvc -n test-sc-migration
NAME                                AGE    STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration    26m    Running    True

NAME                                STATUS    VOLUME    CAPACITY    ACCESS MODES    STORAGECLASS    VOLUMEATTRIBUTESCLASS    AGE
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc    Bound    pvc-bb5d342f-c01f-453e-86ad-abf72e710a2f    536i    Rwx    sc-nas    <unset>    26m
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb    Bound    pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b    60322815673    Rwx    sc-nas-economy    <unset>    3m28s
persistentvolumeclaim/rhel9-migration                                Bound    pvc-46a34903-942f-4ddc-8e6d-974008f05d09    34144990004    Rwx    sc-nas    <unset>    26m
[root@00-50-56-b5-2b-9b ~]#

```

```

[root@00-50-56-b5-2b-9b ~]# oc describe pv -n test-sc-migration pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
Name:
Labels:
Annotations:
pv.kubernetes.io/provisioned-by: csi.trident.netapp.io
volume.kubernetes.io/provisioner-deletion-secret-name:
volume.kubernetes.io/provisioner-deletion-namespace:
Finalizers: [external-provisioner.volume.kubernetes.io/finalizer kubernetes.io/pv-protection external-attacher/csi-trident-netapp-io]
StorageClass: sc-nas-economy
Status: Bound
Claim: test-sc-migration/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb
Reclaim Policy: Delete
Access Modes: Rwx
VolumeMode: Filesystem
Capacity: 60322815673
Node Affinity: <none>
Message:
Source:
Type: CSI (a Container Storage Interface (CSI) volume source)
Driver: csi.trident.netapp.io
FSType:
VolumeHandle: pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
ReadOnly: false
VolumeAttributes:
  backendUUID: a59e8053-bb00-4410-9b12-1dc4d283928b
  internalID: /svm/oc-automation-svm/flexvol/trident_qtree_pool_oc_automation_OGBGMIOLG/qtree/oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546...
  internalName: oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546dbdb3b
  name: pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
  protocol: file
  storage.kubernetes.io/csiProvisionerIdentity: 1778505459781-8296-csi.trident.netapp.io
Events:
<none>

```

Qtrees

+ Add

🔍 ⬇️ 🗃️ ☰

☐	Name	Storage VM	Volume name	Mount path
	oc_automation_pvc_75a885f0_824e_407c_ab29_8cct			
▼	oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546dbdb3b	oc-automation-svm	trident_qtree_pool_oc_automation_OGBGMIAOLG	/trident_qtree_pool_oc_automation_OGBGMIAOLG/oc_...

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。