



Introdução ao armazenamento e à virtualização

NetApp virtualization solutions

NetApp
July 14, 2026

Índice

Introdução ao armazenamento e à virtualização	1
Saiba mais sobre a integração do NetApp Trident com o Red Hat OpenShift	1
NetApp Trident	1
Trident Protect	1
NetApp ONTAP modelos de hardware e protocolos suportados	2
Drivers Trident para storage ONTAP	2
Instale o Trident em um cluster Red Hat OpenShift e crie objetos de storage	4
Passo 1: Instalar Trident	5
Etapa 2: Prepare o backend de armazenamento e os arquivos de configuração StorageClass para o seu ambiente	9
Passo 3: criar um arquivo de configuração VolumeSnapshotClass	29
Etapa 4: aplique os arquivos de configuração ao seu cluster	29
Etapa 5: defina as classes padrão de storage e snapshot do Trident	31
Requisitos para implantar o Red Hat OpenShift Virtualization com ONTAP	32
Pré-requisitos	32
NetApp Trident CSI, perfis de storage, modos de acesso a volumes para OpenShift Virtualization.	33
Crie uma máquina virtual usando o storage ONTAP com Red Hat OpenShift Virtualization	37
Criar VM	38
Demonstração em vídeo	41
Crie uma VM com Red Hat OpenShift Virtualization usando storage ONTAP por meio de métodos alternativos	41
Crie uma máquina virtual a partir de um arquivo YAML com Red Hat OpenShift Virtualization	41
Crie uma VM a partir de uma cópia de snapshot com o Red Hat OpenShift Virtualization	43
Clone uma VM na OpenShift Virtualization usando Trident	47
Migrar uma VM entre dois nós em um cluster Red Hat OpenShift	51
Migração ao vivo de VM	51
Migração de storage de VMs entre classes de storage em OpenShift Virtualization	53

Introdução ao armazenamento e à virtualização

Saiba mais sobre a integração do NetApp Trident com o Red Hat OpenShift

O armazenamento é crucial em aplicações containerizadas, garantindo que os dados estejam disponíveis de forma persistente e gerenciados com eficiência em ambientes dinâmicos e frequentemente efêmeros. As soluções modernas de armazenamento para contêineres são projetadas para escalar perfeitamente com as aplicações, oferecendo a flexibilidade de alocar recursos de storage de forma dinâmica com base nas necessidades da aplicação, suportando tanto o crescimento quanto cargas de trabalho variáveis. Como a OpenShift Virtualization permite que as VMs sejam executadas como contêineres, elas utilizam o mesmo armazenamento que outras aplicações nos clusters OpenShift.

NetApp ONTAP é um software de gerenciamento de dados robusto e altamente escalável que serve como base para as soluções de storage de nível empresarial da NetApp. Ele fornece uma arquitetura de storage unificado que se integra perfeitamente a diversos ambientes de storage, incluindo locais, em nuvem e nuvem híbrida. ONTAP suporta uma variedade de protocolos, como NFS, CIFS/SMB, iSCSI, FC e NVMe sobre TCP, permitindo integração perfeita com diferentes aplicativos e cargas de trabalho. Ele possibilita a consolidação de SAN, NAS e storage de objetos em uma plataforma única, simplificando o gerenciamento e reduzindo custos. NetApp ONTAP está disponível localmente e também como serviços gerenciados na nuvem.

NetApp Trident

Trident é um orquestrador de storage de código aberto desenvolvido pela NetApp, projetado para facilitar o uso do storage ONTAP em ambientes containerizados. Ele permite o provisionamento e gerenciamento dinâmico de storage persistente para aplicações containerizadas.

Principais características do Trident

Conformidade com CSI Garantindo compatibilidade com os recursos e padrões de storage mais recentes do Kubernetes. **Configuração declarativa** Permitindo que você defina os requisitos de storage em arquivos YAML, que são gerenciados automaticamente pelo Trident. **Drivers** Trident oferece suporte a drivers para os protocolos NFS, CIFS/SMB, iSCSI, FC e NVMe/TCP suportados pelo ONTAP. **Suporte a nuvem híbrida e multicloud** Trident oferece suporte a drivers para storage ONTAP local e serviço de storage gerenciado nos hiperescaladores, como FSx for NetApp ONTAP na AWS, Azure NetApp Files no Azure e Google Cloud NetApp Volumes no Google Cloud. **Gerenciamento avançado de dados** Aproveitando os recursos de Snapshot e clonagem do ONTAP para proteção eficiente de dados e provisionamento rápido de ambientes de teste e desenvolvimento.

Para obter detalhes completos sobre Trident, consulte NetApp Trident ["documentação aqui"](#)

Trident Protect

Trident Protect é um recurso avançado do Trident que oferece proteção e segurança de dados aprimoradas para ambientes containerizados. Ele foi projetado para garantir que seus dados críticos estejam protegidos contra possíveis ameaças e cenários de perda de dados. Está disponível gratuitamente para todos os usuários do Trident e pode ser facilmente ativado em qualquer cluster OpenShift com Trident instalado.

Principais características do Trident Protect

Backups automatizados Trident Protect permite backups automatizados e voltados a políticas de aplicativos em contêineres e VMs, garantindo que eles sejam copiados regularmente sem intervenção manual.

Gerenciamento de Snapshots Ele aproveita os recursos de snapshot do ONTAP para criar cópias frequentes de ponto no tempo de aplicativos em contêineres e VMs, fornecendo um mecanismo confiável de recuperação.

Criptografia de Dados Garante que os dados sejam criptografados tanto em repouso quanto em trânsito, atendendo a rigorosos requisitos de segurança e protegendo informações confidenciais.

Conformidade e Auditoria Oferece ferramentas e recursos que ajudam as organizações a atender aos requisitos de conformidade e realizar auditorias regulares de suas práticas de proteção de dados.

Recuperação de Desastres integra-se aos recursos de replicação do ONTAP para fornecer soluções robustas de recuperação de desastres, garantindo a continuidade dos negócios mesmo diante de eventos catastróficos.

Para obter detalhes completos sobre o Trident Protect, consulte NetApp Trident Protect ["documentação aqui"](#).

NetApp ONTAP modelos de hardware e protocolos suportados

NetApp ONTAP software é executado em diversos modelos de hardware, cada um projetado para atender a diferentes requisitos de desempenho e capacidade. Trident amplia seus recursos robustos para oferecer suporte a vários sistemas NetApp ONTAP, incluindo All Flash FAS (AFF), All SAN Array (ASA) e ASA R2. Esse suporte garante que as organizações possam aproveitar todo o potencial de suas soluções de storage de alto desempenho em ambientes containerizados.

Sistemas All Flash FAS (AFF) Os sistemas AFF oferecem desempenho excepcional e baixa latência, tornando-os ideais para aplicações de alta demanda.

All SAN Array (ASA) Systems Os sistemas ASA são projetados para ambientes SAN dedicados, oferecendo desempenho robusto e confiabilidade.

ASA R2 Systems Os sistemas ASA R2 oferecem recursos e capacidades aprimorados para ambientes SAN.

AFX NetApp AFX é uma arquitetura de storage all-flash desagregada, desenvolvida especificamente para cargas de trabalho de IA corporativas. Oferece NAS de alto desempenho, permitindo o escalonamento independente de computação e capacidade. Os sistemas de storage NetApp AFX diferem de outros sistemas baseados em ONTAP (ASA, AFF e FAS) na implementação de sua camada de storage. AFX utiliza um sistema de arquivos distribuído que permite alto desempenho e escalabilidade, enquanto outros sistemas baseados em ONTAP utilizam uma arquitetura de storage tradicional.

1. Protocolos suportados para AFF: NFS, CIFS/SMB, iSCSI, FC, NVMe/TCP
2. Protocolos suportados para ASA: iSCSI, FC, NVMe/TCP
3. Protocolos suportados para ASA R2: iSCSI, FC, NVMe/TCP
4. Protocolos suportados para AFX: a partir do Trident 25.10, apenas o protocolo NFS é suportado; o protocolo SMB não é suportado.

Drivers Trident para storage ONTAP

Trident inclui os seguintes drivers CSI, cada um capaz de provisionar um volume no storage de backend

Para protocolos NAS em modelos de hardware compatíveis

1. ontap-nas: Um PVC corresponde a um PV, que é um flexvolume dedicado.
2. ontap-nas-economy: Cada PV provisionado é uma qtree, com um número configurável de qtrees por FlexVolume (o padrão é 200, configurável entre 50 e 300). Um pvc corresponde a 1 pv, que é uma qtree em um flexvolume compartilhado.



Você pode colocar todos os discos das VMs como qtrees em um único volume. No entanto, esteja ciente de que os recursos de Snapshot, Clones e Replicação não são suportados pelo Trident. Quando esses recursos são gerenciados pelo administrador de storage, eles não são granulares em nível de PV.

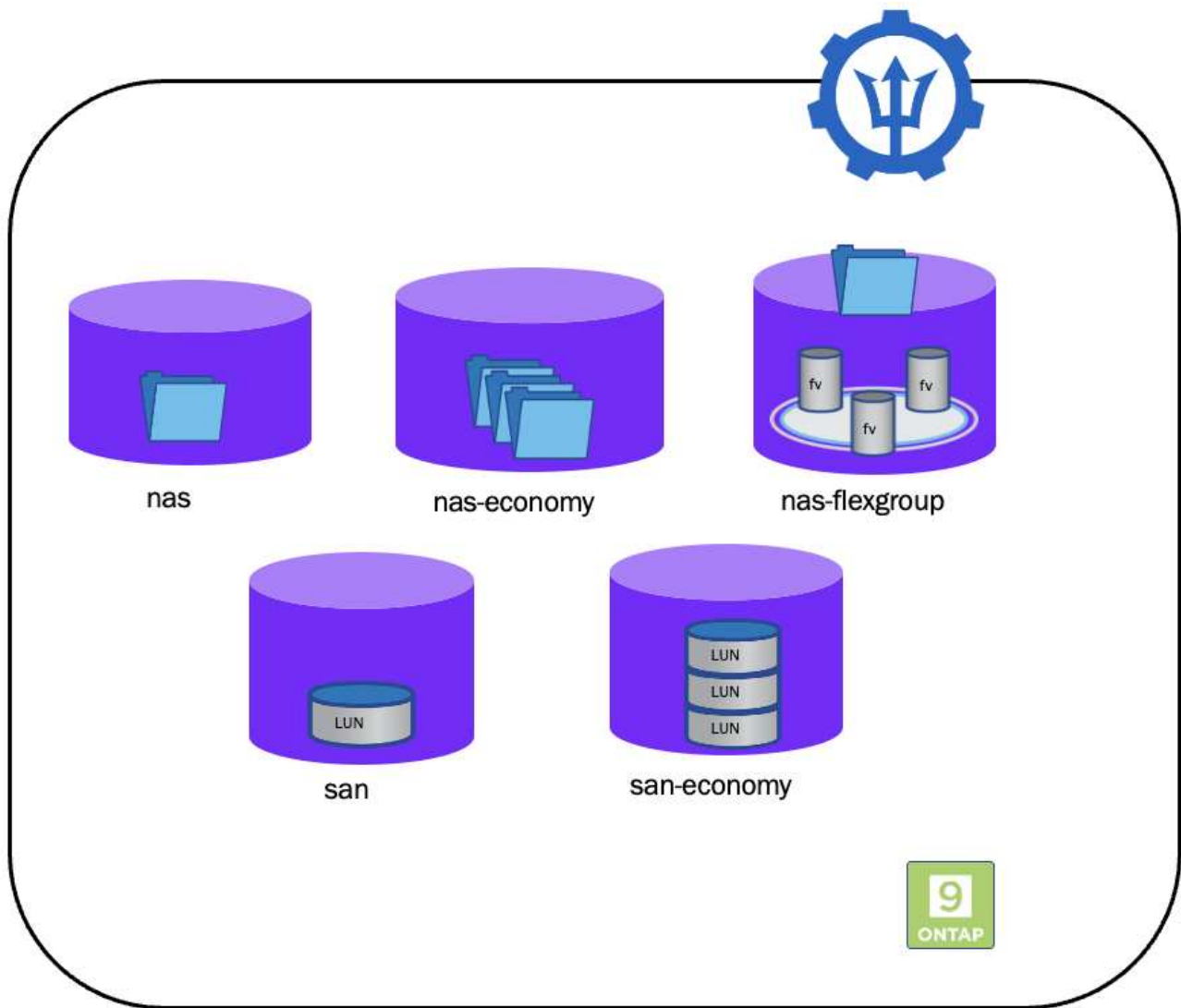
1. ontap-nas-flexgroup: Cada PV provisionado é um ONTAP FlexGroup, e todos os agregados atribuídos a um SVM são utilizados.

Para protocolos SAN em modelos de hardware compatíveis

1. ontap-san: Um PVC corresponde a um PV, que é um LUN em um flexvolume dedicado
2. ontap-san-economy: Um PVC corresponde a um PV, que é um LUN em um flexvolume compartilhado. Você pode ter vários LUNs no flexvolume compartilhado (o padrão é 100, configurável entre 50 e 200 LUNs/PVs por volume flexvol).



Você pode colocar todos os discos das VMs como LUNs em um único volume. No entanto, embora este driver suporte Snapshots e Clones, ele não suporta replicação. Quando a replicação é gerenciada pelo administrador de storage, ela não é granular por PV.



Para obter a lista completa de recursos expostos pelos drivers Trident e daqueles que devem ser aplicados pelo administrador de storage, consulte a seção "[Drivers de backend ONTAP](#)" na documentação do Trident.

Instale o Trident em um cluster Red Hat OpenShift e crie objetos de storage

Instale Trident com o Trident Operator certificado da Red Hat e crie objetos de storage para ONTAP e Amazon FSx for NetApp ONTAP para habilitar o provisionamento dinâmico de volumes para contêineres e VMs. Prepare os nós de trabalho para acesso a blocos quando necessário.

Antes de começar

- Conclua os procedimentos desta página antes de instalar OpenShift Virtualization. OpenShift Virtualization requer um StorageClass padrão com suporte do Trident e um VolumeSnapshotClass padrão para criar imagens douradas para modelos de máquinas virtuais.
- Se você já instalou OpenShift Virtualization antes de configurar Trident, exclua quaisquer imagens douradas criadas com uma classe de armazenamento diferente. Depois de definir Trident como padrão,

OpenShift Virtualization recria as imagens douradas usando o storage Trident.

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

Passo 1: Instalar Trident

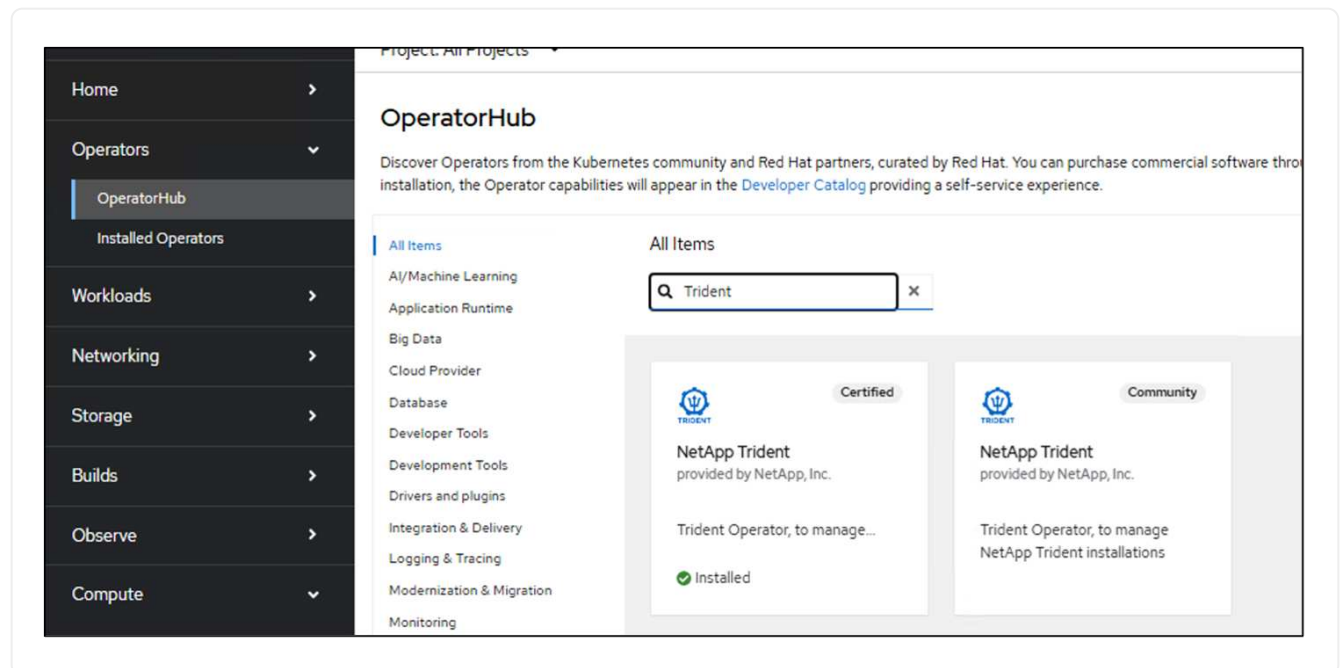
O operador Trident certificado pela Red Hat é compatível com NetApp para OpenShift em ambientes locais, nuvens públicas e serviços gerenciados como o ROSA. A partir do Trident 25.02, o operador também pode preparar nós de trabalho para iSCSI quando você usa Amazon FSx for NetApp ONTAP e planeja executar cargas de trabalho de VM de Virtualização do OpenShift.

Para outras opções de instalação, consulte ["a documentação do Trident"](#).

Passos

1. Em **OperatorHub**, selecione **Certified NetApp Trident**.

Mostrar exemplo



2. Na página **Install**, mantenha a versão mais recente e selecione **Install**.

Mostrar exemplo

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. Após a instalação do operador, selecione **View operator** e crie uma instância do Trident Orchestrator.

Se você deseja preparar nós de trabalho para iSCSI, mude para a visualização YAML e adicione `iscsi` a `nodePrep`.

Mostrar exemplo

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. Confirme que todos os pods do Trident estão em execução no cluster.

Mostrar exemplo

```
[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. Se você habilitou a preparação do nó iSCSI, faça login nos nós de trabalho e verifique se `iscsid` e `multipathd` estão ativos e se `multipath.conf` tem entradas.

Mostrar exemplo

```
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#
```

Mostrar exemplo

```
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#
```

Mostrar exemplo

```
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# █
```

Demonstração em vídeo

O vídeo a seguir mostra uma demonstração da instalação do Trident usando o Trident Operator certificado pela Red Hat.

[Instalando o Trident 25.02.1 usando o Trident Operator certificado no OpenShift](#)

Etapa 2: Prepare o backend de armazenamento e os arquivos de configuração StorageClass para o seu ambiente

Crie definições de TridentBackendConfig e StorageClass para o seu ambiente. Você pode configurar vários protocolos de storage dentro do seu ambiente. Crie arquivos YAML para cada protocolo que deseja usar e substitua os valores de espaço reservado pelos seus detalhes específicos de configuração.



Preencha a seção on-premises ou ROSA abaixo, dependendo do seu ambiente, e prossiga para a Etapa 3.

Clusters locais OpenShift

Crie arquivos YAML para cada protocolo que você deseja configurar. Você pode configurar um ou mais dos seguintes protocolos: NAS para storage de arquivos baseado em NFS, iSCSI para storage em bloco iSCSI, NVMe/TCP para storage em bloco NVMe sobre TCP de alto desempenho ou FC para storage em bloco Fibre Channel.

Crie um `TridentBackendConfig` e um `StorageClass` para cada protocolo. A configuração do backend inclui credenciais armazenadas em um segredo do Kubernetes.

Diretrizes essenciais para opções básicas de configuração de backend

1. Os arquivos de configuração do backend fazem referência ao seu **ONTAP SVM** e **management LIF**.
+
2. Você pode usar **nome de usuário e senha** para autenticar no ONTAP. Pode ser o nome de usuário e a senha do administrador do cluster ou do administrador do vserver. +
3. Você também pode usar um **certificado de cliente** para autenticação com ONTAP. Para obter detalhes sobre como gerar o certificado de cliente, a chave privada e o certificado da CA, consulte ["Documentação do Trident aqui"](#).

Diretrizes essenciais para opções de configuração de backend para o driver NAS economy

Os seguintes parâmetros podem ser incluídos para a configuração do **driver de economia NAS**:

1. **qtreesPerFlexvol** O valor deve estar no intervalo de 50 a 300, o valor padrão é 200. +
2. **limitVolumePoolSize** O valor é o tamanho máximo de FlexVol que você solicita ao usar qtrees no backend ontap-nas-economy. Nenhum valor padrão é definido e este parâmetro não é aplicado por padrão. Se você definir este parâmetro, o operador Trident cria um novo FlexVol quando o FlexVol atual atingir o tamanho especificado. Este parâmetro pode ajudar você a controlar a utilização da capacidade no seu cluster ao restringir o tamanho dos FlexVols usados para o provisionamento de PVs. +
3. **denyNewVolumePools** Este parâmetro restringe os backends ontap-nas-economy de criarem novos volumes FlexVol para conter suas qtrees. Somente FlexVols preexistentes são usados para o provisionamento de novos PVs.

Diretrizes essenciais para opções de configuração de backend para driver SAN

Os seguintes parâmetros podem ser incluídos para a configuração do **driver SAN**:

1. **limitVolumeSize** O valor representa o tamanho máximo do volume que você solicita ao usar o backend SAN. Não há valor padrão definido e este parâmetro não é aplicado por padrão. Se você definir este parâmetro, o operador Trident falhará no provisionamento se o tamanho do volume solicitado for superior a este valor. Você pode usar este parâmetro para restringir o tamanho máximo dos volumes que ele gerencia para LUNs. +
2. **lunsPerFlexvol** Número máximo de LUNs por FlexVol, deve estar no intervalo [50, 200], o padrão é 100. Se você definir este parâmetro, o operador Trident criará um novo FlexVol quando o FlexVol atual atingir o número especificado de LUNs. Este parâmetro pode ajudar você a controlar a utilização da capacidade em seu cluster, restringindo o número de LUNs por FlexVol usados para o provisionamento de PVs.

Diretrizes essenciais para opções de configuração de backend para o driver de economia SAN

Os seguintes parâmetros podem ser incluídos para a configuração do **SAN economy driver**:

1. **limitVolumePoolSize** O valor representa o tamanho máximo de FlexVol que você solicita ao usar o backend SAN economy. Nenhum valor padrão é definido e este parâmetro não é aplicado por padrão. Se você definir este parâmetro, o operador Trident limita o tamanho máximo do flexvol que pode ser criado para a criação de LUNs no flexvol. Este parâmetro pode ajudar você a controlar a utilização da capacidade em seu cluster, restringindo o tamanho dos FlexVols usados para o provisionamento de PVs. +
2. **denyNewVolumePools** Este parâmetro restringe os backends econômicos do SAN de criarem novos volumes FlexVol para conter seus LUNs. Somente FlexVols preexistentes são usados para o provisionamento de novos PVs.

Diretrizes essenciais para opções de configuração de backend para provisionamento de volumes



Nos arquivos yaml de exemplo fornecidos para cada protocolo, se storagePrefix não for especificado, o prefixo padrão "trident" será usado. Ao usar ontap-nas-economy e um storagePrefix com 24 ou mais caracteres, as qtrees não terão o storage prefix incorporado, embora ele esteja presente no nome do volume.

Ao criar um volume no ONTAP diretamente do system manager ou da CLI, sem fornecer valores para todos os parâmetros opcionais, alguns valores padrão são definidos automaticamente durante o provisionamento de um volume, conforme mostrado abaixo.

```
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
```

Aqui está um exemplo de um volume criado usando a configuração de backend do Trident. Ele não possui nenhuma opção definida explicitamente na seção de padrões, exceto o modelo de nome.

```
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
```

Quando o volume é criado usando esta configuração de backend, as opções que não estão definidas no arquivo de configuração de backend são definidas com os valores padrão, conforme mostrado abaixo.

```
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
```

No entanto, ao definir valores na seção padrão da configuração do backend Trident, você pode criar volumes que podem ter valores específicos para esses parâmetros. No exemplo a seguir, snapshotPolicy está definido como a política padrão (é none se você não definir nada explicitamente) e snapshotDir está definido como true na seção padrão do arquivo de configuração yaml do backend. Com essas configurações, você pode ver que o volume do backend recebe esses valores. (snapdir-access é true e snapshot-policy é default)


```
[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      unix-permissions space-guarantee snapdir-access snapshot-policy qos-adaptive-policy-group snapshot-reserve-availab
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
```

Para todos os outros parâmetros adicionais, consulte a documentação do Trident:

1. ["Opções de configuração do backend Trident NAS"](#) +
2. ["Opções de configuração do backend SAN Trident"](#)

NAS

Crie um `TridentBackendConfig` e `StorageClass` para ONTAP NAS para habilitar o provisionamento de storage persistente baseado em NFS. A configuração de backend inclui credenciais armazenadas em um segredo do Kubernetes e faz referência à sua ONTAP SVM e à LIF de gerenciamento.

Exemplo de segredo do backend e arquivo de configuração do backend usando autenticação por nome de usuário e senha (salve como `tbc-nas.yaml`):

```
# 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
```

Exemplo de segredo do backend e arquivo de configuração do backend usando autenticação por certificado de cliente (salve como `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 definição (salvar como `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 economy

Crie um `TridentBackendConfig` e um `StorageClass` para o driver de economia ONTAP NAS para habilitar o provisionamento de storage persistente baseado em NFS usando `qtrees`. A configuração de backend

inclui credenciais armazenadas em um Secret do Kubernetes e faz referência à sua ONTAP SVM e à LIF de gerenciamento.

Exemplo de segredo do backend e arquivo de configuração do backend usando autenticação por nome de usuário e senha (salve como `tbc-nas-economy.yaml`):

```
# 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 definição (salvar como `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
```



Consulte a seção Diretrizes essenciais para opções de configuração de backend para o driver NAS Economy acima para obter detalhes sobre parâmetros adicionais que podem ser incluídos no arquivo de configuração de backend para o driver ontap-nas-economy.

iSCSI SAN

Crie um TridentBackendConfig e um StorageClass para ONTAP SAN com iSCSI para habilitar o provisionamento de storage em bloco baseado em iSCSI. A configuração do backend usa o driver ontap-san e inclui credenciais armazenadas em um segredo do Kubernetes. O parâmetro sanType usa o protocolo iSCSI por padrão se omitido.

Segredo do backend e arquivo de configuração do backend usando autenticação por nome de usuário e senha (salve como 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

```

Segredo do backend e arquivo de configuração do backend usando autenticação por certificado de cliente (salve como `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 definition (salvar como 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

Crie um TridentBackendConfig e StorageClass para ONTAP SAN com NVMe sobre TCP para habilitar o provisionamento de storage de blocos de alto desempenho. A configuração do backend usa o driver ontap-san otimizado para transporte NVMe/TCP e inclui credenciais armazenadas em um segredo do Kubernetes.

Segredo do backend e arquivo de configuração do backend usando autenticação por nome de usuário e senha (salve como `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
```

Segredo do backend e arquivo de configuração do backend usando autenticação por certificado de cliente (salve como `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 definition (salvar como `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
```



Se você planeja usar volumes ReadWriteMany (RWX) com o protocolo NVMe, um único volume RWX NVMe não pode ser montado por mais de 64 nós em um cluster Kubernetes. Para obter mais informações sobre essa limitação e informações sobre o modelo de Super-subsystem introduzido no Trident 26.02 e o modelo de Per-volume subsystem anterior ao Trident 26.02, consulte a documentação do Trident: <https://docs.netapp.com/us-en/trident/trident-use/ontap-san-prep.html#nvme-tcp-considerations>.

FC SAN

Crie um `TridentBackendConfig` e `StorageClass` para ONTAP SAN com Fibre Channel para habilitar o provisionamento de storage em bloco baseado em FC. A configuração do backend usa o driver `ontap-san` com o protocolo FCP especificado e inclui credenciais armazenadas em um Kubernetes Secret.

Use o método de autenticação por nome de usuário/senha ou o método de autenticação por certificado de cliente para autenticar com ONTAP. Para obter detalhes sobre como gerar o certificado de cliente, a chave privada e o certificado da CA, consulte "[Documentação do Trident aqui](#)". Exemplos para ambos os métodos são fornecidos abaixo.

Mostrar exemplo

Segredo do backend e arquivo de configuração do backend usando autenticação com nome de usuário e senha do ONTAP (salve como `tbc-fc.yaml`):

```
# 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
```

Segredo do backend e arquivo de configuração do backend usando autenticação por certificado de cliente (salve como `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
```

Crie um `TridentBackendConfig` e um `StorageClass` para ONTAP SAN economy com iSCSI para habilitar o provisionamento de storage em bloco baseado em iSCSI. A configuração de backend utiliza o driver `ontap-san-economy` com o protocolo iSCSI e inclui credenciais armazenadas em um `Secret` do Kubernetes.

Segredo do backend e arquivo de configuração do backend usando autenticação por certificado de cliente (salve como `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"
```

Consulte a documentação do Trident para obter exemplos adicionais de arquivos yaml e informações sobre os modos de acesso e modos de volume suportados pelos drivers SAN e NAS.

1. ["Exemplos usando drivers NAS"](#)
2. ["Exemplos usando drivers SAN"](#)

Clusters ROSA com Amazon FSx for NetApp ONTAP

Crie arquivos YAML para cada protocolo que você deseja configurar. Você pode configurar um ou ambos os seguintes protocolos: NAS para storage de arquivos ou iSCSI para block storage.

NAS

Crie um TridentBackendConfig e StorageClass para Amazon FSx for NetApp ONTAP com ONTAP NAS para habilitar o provisionamento de storage persistente baseado em NFS em clusters ROSA. A configuração de backend usa nomes DNS do Amazon FSx for NetApp ONTAP para LIFs de gerenciamento e dados, e inclui credenciais armazenadas em um Secret do Kubernetes no namespace trident.

Segredo do backend e arquivo de configuração (salvar como `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 definition (salvar como `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

Crie um TridentBackendConfig e StorageClass para Amazon FSx for NetApp ONTAP com ONTAP SAN para habilitar o provisionamento de storage baseado em blocos via iSCSI em clusters ROSA. A configuração de backend utiliza o driver ontap-san e inclui credenciais armazenadas em um Kubernetes Secret. Certifique-se de que os nós de trabalho estejam preparados para acesso iSCSI.

Segredo do backend e arquivo de configuração (salvar como `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 definição (salvar como `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
```

Passo 3: criar um arquivo de configuração VolumeSnapshotClass

Crie uma definição de VolumeSnapshotClass para implantações locais e ROSA. Essa configuração permite operações baseadas em snapshots para volumes persistentes.

VolumeSnapshotClass definição (salvar como `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
```

Etapa 4: aplique os arquivos de configuração ao seu cluster

Aplique os arquivos de configuração que você criou nas etapas anteriores ao seu OpenShift cluster.

Passos

1. Aplique os arquivos TridentBackendConfig e StorageClass para cada protocolo que você configurou.

Para clusters locais:

```
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
```

Para clusters ROSA:

```
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. Aplique a configuração VolumeSnapshotClass.

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

3. Verifique se os recursos foram criados com sucesso.

Verifique os objetos TridentBackendConfig:

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

Verifique os objetos StorageClass:

```
oc get storageclass
```

Verifique VolumeSnapshotClass:

```
oc get volumesnapshotclass
```


Etapa 5: defina as classes padrão de storage e snapshot do Trident

Defina o Trident StorageClass e VolumeSnapshotClass como padrões no cluster OpenShift. Isso é necessário para OpenShift Virtualization criar fontes de imagem golden para os templates de VM.

Passos

1. Defina o StorageClass padrão do Trident.

Defina um StorageClass com suporte Trident como padrão do cluster para que PersistentVolumeClaims o utilizem automaticamente quando nenhuma classe de armazenamento for especificada. Você precisa configurar duas anotações: uma para o padrão de todo o cluster e uma específica para OpenShift Virtualization.

- a. Defina a anotação padrão StorageClass para todo o cluster.

Certifique-se de que apenas um StorageClass esteja definido como padrão. Se outro StorageClass já estiver definido como padrão, defina sua anotação como falsa.

No console, edite a anotação:

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

A partir da linha de comando:

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

- b. Defina a anotação padrão específica para OpenShift Virtualization.

OpenShift Virtualization usa uma anotação específica que tem precedência sobre a anotação geral do cluster `is-default-class`. Se outra StorageClass já estiver definida como padrão, defina sua anotação como falsa.

No console, edite a anotação:

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

A partir da linha de comando:

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

2. Defina o Trident VolumeSnapshotClass padrão.

Defina um VolumeSnapshotClass com suporte do Trident como padrão do cluster para habilitar operações

baseadas em snapshots para volumes persistentes. Isso garante que VolumeSnapshots usem automaticamente o driver CSI do Trident quando nenhuma classe de snapshot for especificada e permite que a OpenShift Virtualization crie snapshots de imagens douradas.

Certifique-se de que apenas um VolumeSnapshotClass esteja definido como padrão. Se outro VolumeSnapshotClass já estiver definido como padrão, defina sua anotação como falsa.

No console, edite a anotação:

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

A partir da linha de comando:

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

Requisitos para implantar o Red Hat OpenShift Virtualization com ONTAP

Analise os pré-requisitos e detalhes adicionais para implementar a virtualização OpenShift com sistemas de armazenamento ONTAP.

Pré-requisitos

- Um cluster Red Hat OpenShift (posterior à versão 4.6) instalado em infraestrutura bare-metal com nós de trabalho RHCOS
- Implantar verificações de integridade da máquina para manter HA para VMs
- Um cluster NetApp ONTAP , com SVM configurado com o protocolo correto.
- Trident instalado no cluster OpenShift
- Uma configuração de backend Trident criada
- Um StorageClass configurado no cluster OpenShift com Trident como provisionador
- Um Trident VolumeSnapshotClass configurado no OpenShift cluster com Trident como provisionador
- Trident StorageClass configurado como padrão para o cluster OpenShift ou OpenShift Virtualization.

Para os pré-requisitos do Trident acima, consulte ["Seção de instalação do Trident"](#) para mais detalhes.

- Acesso de administrador de cluster ao cluster Red Hat OpenShift
- Acesso de administrador ao cluster NetApp ONTAP
- Uma estação de trabalho de administração com as ferramentas tridentctl e oc instaladas e adicionadas ao \$PATH

Como a OpenShift Virtualization é gerenciada por um operador instalado no cluster OpenShift, ela impõe uma sobrecarga adicional em memória, CPU e storage, que deve ser considerada ao planejar os requisitos de

hardware do cluster. Consulte a documentação ["aqui"](#) para mais detalhes.

Opcionalmente, você também pode especificar um subconjunto dos nós do cluster OpenShift para hospedar os operadores, controladores e VMs do OpenShift Virtualization, configurando regras de posicionamento de nós. Para configurar regras de posicionamento de nós para OpenShift Virtualization, consulte a documentação ["aqui"](#).

Para o armazenamento de suporte ao OpenShift Virtualization, a NetApp recomenda ter um StorageClass dedicado que solicita armazenamento de um backend Trident específico, que por sua vez é apoiado por um SVM dedicado. Isso mantém um nível de multilocação em relação aos dados fornecidos para cargas de trabalho baseadas em VM no cluster OpenShift.

NetApp Trident CSI, perfis de storage, modos de acesso a volumes para OpenShift Virtualization

Na OpenShift Virtualization, as fontes de inicialização são imagens de disco usadas para provisionar máquinas virtuais (VMs) a partir de templates. Elas são normalmente armazenadas como DataVolumes ou Persistent Volume Claims (PVCs) no namespace `openshift-virtualization-os-images`. Para que a OpenShift Virtualization funcione corretamente, há duas tarefas de configuração de storage que são obrigatórias antes de instalar a OpenShift Virtualization usando o operador:

1. Você deve configurar um StorageClass padrão para o cluster. Caso contrário, a OpenShift Virtualization não poderá importar automaticamente as imagens de origem de inicialização.
2. Você deve configurar perfis de armazenamento se o seu provedor de armazenamento não for reconhecido pelo CDI. Um perfil de armazenamento fornece configurações de armazenamento recomendadas com base no StorageClass associado. Uma das funções críticas do StorageProfile é definir estratégias de clonagem para um StorageClass. Isso determina como os volumes são clonados (por exemplo, usando snapshots, clones CSI ou cópias assistidas pelo host), otimizando o desempenho e o uso de recursos com base nas capacidades do fornecedor de storage subjacente. O driver Trident CSI é reconhecido pelo CDI, e um perfil de armazenamento padrão é criado automaticamente para qualquer StorageClass do Trident criado no cluster. A estratégia de clonagem para esse perfil de armazenamento padrão é definida como snapshot, que é a estratégia de clonagem recomendada para o driver Trident CSI. Você deve criar o VolumeSnapshotClass e o StorageClass para Trident e definir o StorageClass como padrão antes de instalar OpenShift Virtualization. Isso garante que as imagens de origem de inicialização estejam disponíveis como VolumeSnapshots no namespace `openshift-virtualization-os-images`, para que seja possível provisionar VMs de forma eficiente e sem erros.

Para determinados sistemas operacionais suportados (como RHEL, Fedora ou CentOS), OpenShift Virtualization fornece e atualiza automaticamente as fontes de inicialização. Essas são baixadas quando OpenShift Virtualization é instalado e são gerenciadas e atualizadas automaticamente para a versão mais recente do sistema operacional. Se um sistema operacional não for fornecido automaticamente, um administrador deve preparar e anexar manualmente uma imagem personalizada. Ao usar NetApp Trident como o CSI, você obtém os seguintes benefícios adicionais:

1. As imagens de inicialização estarão disponíveis como snapshots no namespace `openshift-virtualization-os-images`. Quando as VMs são provisionadas, a imagem de inicialização é clonada do snapshot, tornando a inicialização da VM muito mais rápida.
2. Um perfil de armazenamento é criado automaticamente para cada Trident StorageClass criado e é configurado automaticamente com a estratégia de clonagem definida como snapshot. O modo de acesso está definido como `ReadWriteMany`. Não é necessária nenhuma configuração manual.

É muito importante que você crie um Trident StorageClass e um Trident VolumeSnapshotClass e os defina como padrão antes de instalar o OpenShift Virtualization no seu cluster. Isso garantirá que as imagens de origem de inicialização estejam disponíveis como VolumeSnapshots no namespace `openshift-virtualization-os-`

imagens, permitindo o provisionamento eficiente de VMs sem erros.

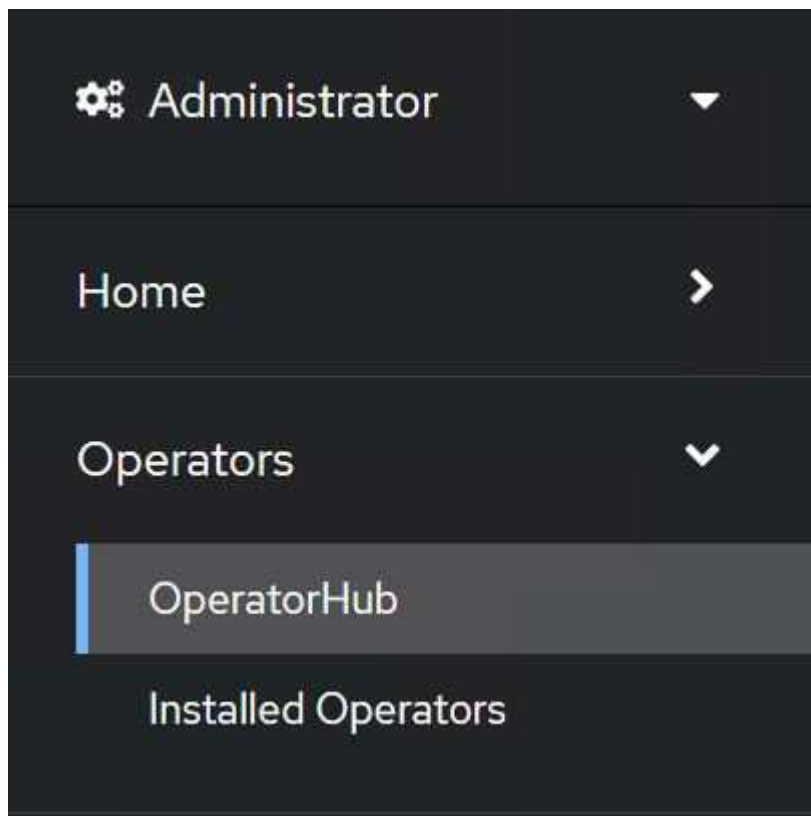
Outra vantagem de usar o Trident CSI: . No OpenShift Virtualization, os discos de boot das VMs precisam ter modos de acesso RWX para fazer a migração ao vivo (Live migration) de VMs. O Trident suporta o modo de acesso **ReadWriteMany** para volumes provisionados usando protocolos NAS, bem como usando protocolos SAN (quando o volumeMode está definido como Block e não FileSystem). Você pode usar qualquer um dos protocolos para o disco de boot. O objeto de perfil de storage será configurado automaticamente com base nas configurações de StorageClass e os modos de acesso do disco de boot serão definidos adequadamente ao provisionar as VMs.

Depois de definir o Trident StorageClass como padrão para o cluster, você pode criar StorageClasses adicionais para o Trident com parâmetros diferentes, e elas serão reconhecidas automaticamente pelo OpenShift Virtualization, com os perfis de storage apropriados criados para elas. Em seguida, você pode especificar o StorageClass a ser usado para o disco de boot e os discos de dados da máquina virtual na definição da VM.

Instalar a OpenShift Virtualization no cluster OpenShift é um processo simples usando o operador OpenShift Virtualization disponível no OpenShift OperatorHub. Para obter etapas detalhadas sobre como instalar a OpenShift Virtualization, consulte a documentação ["aqui"](#).

Instale o OpenShift Virtualization em um cluster bare-metal do Red Hat OpenShift. Este procedimento inclui fazer login com acesso de administrador de cluster, navegar até o OperatorHub e instalar o operador OpenShift Virtualization.

1. Efetue login no cluster bare-metal do Red Hat OpenShift com acesso de administrador de cluster.
2. Selecione Administrador no menu suspenso Perspectiva.
3. Navegue até Operadores > OperatorHub e procure por OpenShift Virtualization.



4. Selecione o bloco OpenShift Virtualization e clique em Instalar.



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. Na tela Instalar Operador, deixe todos os parâmetros padrão e clique em Instalar.

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



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



OpenShift
Virtualization
Deployment



Required

Represents the deployment of
OpenShift Virtualization

6. Aguarde a conclusão da instalação do operador.



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. Após a instalação do operador, clique em Criar HyperConverged.



OpenShift Virtualization
2.6.2 provided by Red Hat



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. Na tela Criar HyperConverged, clique em Criar, aceitando todos os parâmetros padrões. Esta etapa inicia a instalação do 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. Depois que todos os pods forem movidos para o estado Em execução no namespace openshift-cnv e o operador OpenShift Virtualization estiver no estado Bem-sucedido, o operador estará pronto para uso. Agora é possível criar VMs no cluster OpenShift.

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
 OpenShift Virtualization 2.6.2 provided by Red Hat	 openshift-cnv	 Succeeded Up to date
		Last updated  May 18, 8:02 pm
		Provided APIs OpenShift Virtualization Deployment HostPathProvisioner deployment

Crie uma máquina virtual usando o storage ONTAP com Red Hat OpenShift Virtualization

Crie uma VM com OpenShift Virtualization. Este procedimento inclui selecionar um template de sistema operacional, configurar StorageClasses e personalizar os parâmetros da VM para atender a requisitos específicos. Como pré-requisito, você já

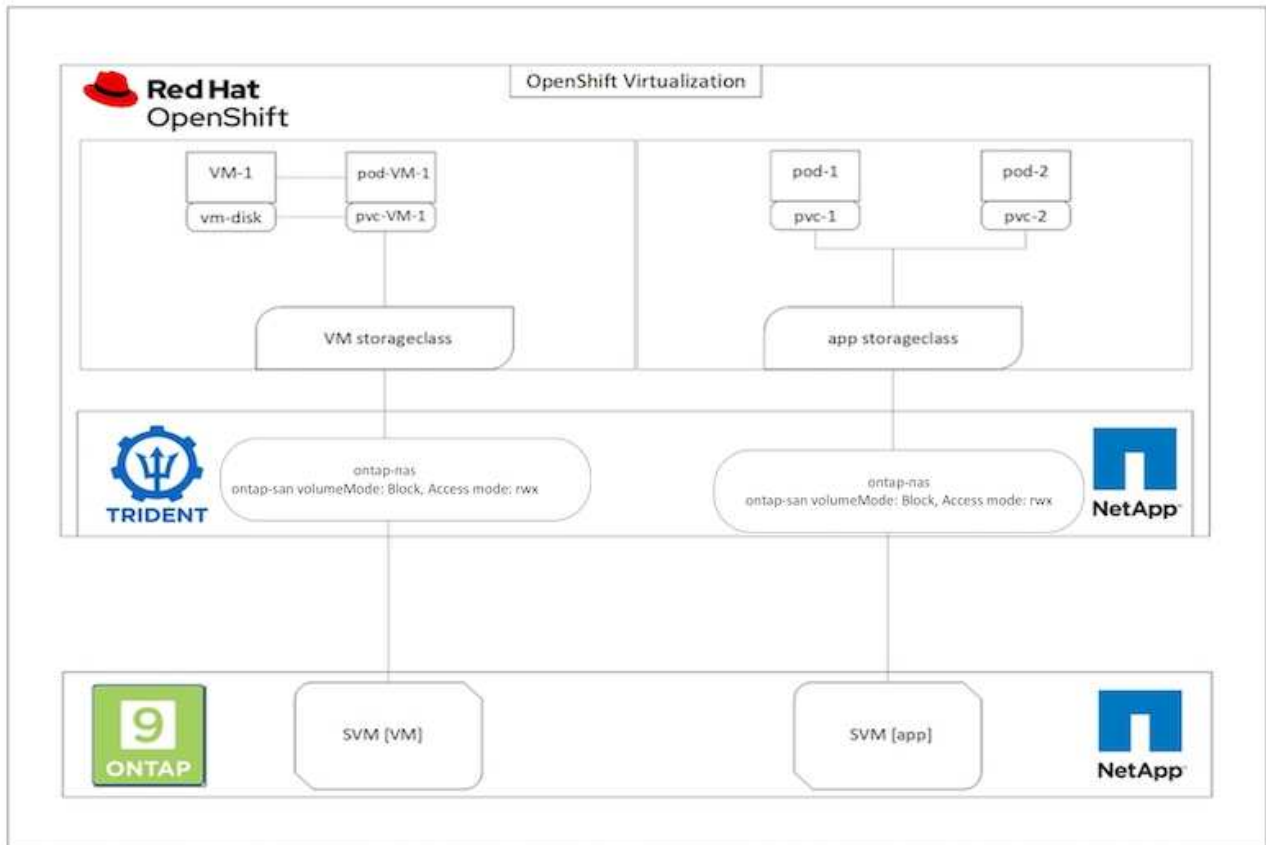
deve ter criado o backend Trident, o StorageClass e os objetos da classe de Snapshot de volume. Você pode consultar a "[Seção de instalação do Trident](#)" para obter detalhes.

Criar VM

VMs são implantações com estado que exigem volumes para hospedar o sistema operacional e os dados. Com o CNV, como as VMs são executadas como pods, elas são apoiadas por PVs hospedados no NetApp ONTAP por meio do Trident. Esses volumes são anexados como discos e armazenam todo o sistema de arquivos, incluindo a origem de inicialização da VM.



Trident StorageClass deve ser definido como padrão StorageClass no cluster para garantir que os PVCs criados para os discos das máquinas virtuais sejam provisionados usando o backend Trident. Você pode ter vários StorageClasses no cluster, mas o StorageClass padrão deve ser usado para o provisionamento do PVC dos discos de inicialização das máquinas virtuais. Discos adicionais podem usar outros StorageClasses, se necessário.



Para criar rapidamente uma máquina virtual no cluster OpenShift, conclua as seguintes etapas:

1. Navegue até Virtualização > Máquinas virtuais e clique em Criar.
2. Selecione Do modelo.
3. Selecione o sistema operacional desejado para o qual a fonte de inicialização está disponível.
4. Marque a caixa de seleção Iniciar a máquina virtual após a criação.
5. Clique em Criação rápida de máquina virtual.

A máquina virtual é criada, iniciada e entra no estado **Running**. Ela cria automaticamente um PVC e um PV correspondente para o disco de inicialização usando o StorageClass padrão. Para poder migrar a VM em tempo real no futuro, você deve garantir que o StorageClass usado para os discos suporte volumes RWX. Este é um requisito para a migração em tempo real. `ontap-nas` e `ontap-san` (`volumeMode` block para os protocolos iSCSI e NVMe/TCP) podem suportar modos de acesso RWX para os volumes criados usando os respectivos StorageClasses.

Para configurar o `ontap-san` StorageClass no cluster, consulte o ["Seção para configurar StorageClass"](#).



Clicar em **Quick create VirtualMachine** usará o StorageClass padrão para criar o PVC e o PV para o disco raiz inicializável da VM. Você pode selecionar um StorageClass diferente para o disco, acessando **Customize VirtualMachine** > **Customize VirtualMachine parameters** > **Disks** e, em seguida, editando o disco para usar o StorageClass necessário.

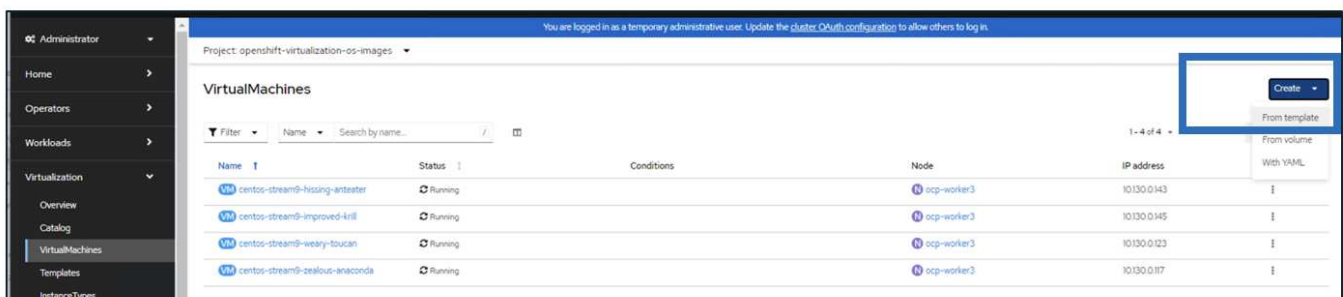
Normalmente, o modo de acesso em bloco é preferível em comparação aos sistemas de arquivos durante o provisionamento dos discos da VM.

Para personalizar a criação da máquina virtual depois de selecionar o modelo de sistema operacional, clique em **Personalizar máquina virtual** em vez de **Criação rápida**.

1. Se o sistema operacional selecionado tiver a origem de inicialização configurada, você pode clicar em **Personalizar parâmetros da VirtualMachine**.
2. Se o sistema operacional selecionado não tiver nenhuma fonte de inicialização configurada, você deverá configurá-lo. Você pode ver detalhes sobre os procedimentos mostrados no ["documentação"](#).
3. Após configurar o disco de inicialização, você pode clicar em **Personalizar parâmetros da VirtualMachine**.
4. Você pode personalizar a VM nas guias desta página. Por exemplo, clique na aba **Discos** e depois clique em **Adicionar disco** para adicionar outro disco à VM.
5. Clique em **Criar Máquina Virtual** para criar a máquina virtual; isso iniciará um pod correspondente em segundo plano.



Quando uma fonte de inicialização é configurada para um modelo ou um sistema operacional a partir de uma URL ou de um registro, ela cria um PVC no `openshift-virtualization-os-images` projeto e baixa a imagem do convidado KVM para o PVC. Você deve garantir que os PVCs de modelo tenham espaço provisionado suficiente para acomodar a imagem do convidado KVM para o sistema operacional correspondente. Esses PVCs são então clonados e anexados como disco raiz às máquinas virtuais quando são criados usando os respectivos modelos em qualquer projeto.



The screenshot shows the OpenShift Virtualization console interface. On the left is a sidebar with navigation links: Administrator, Home, Operators, Workloads, and Virtualization (selected). Under Virtualization, there are links for Overview, Catalog, VirtualMachines (active), Templates, and Instance Types. The main panel displays a table of VirtualMachines for the project 'openshift-virtualization-os-images'. The table has columns for Name, Status, Conditions, Node, and IP address. There are four VMs listed, all with a status of 'Running'. A 'Create' button is visible in the top right corner of the table area.

Name	Status	Conditions	Node	IP address
centos-stream9-hissing-artester	Running		ocp-worker3	10.130.0.143
centos-stream9-improved-kill	Running		ocp-worker3	10.130.0.145
centos-stream9-weary-toucan	Running		ocp-worker3	10.130.0.123
centos-stream9-zealous-anaconda	Running		ocp-worker3	10.130.0.117

Create new VirtualMachine

Select an option to create a VirtualMachine from.

Template catalog **InstanceTypes**

Template project
All projects ▾

Default templates

All items 13 items

Default templates

User templates

☐ Boot source available

Operating system

☐ CentOS

☐ Fedora

☐ Other

☐ RHEL

☐ Windows

Workload

☐ Desktop

☐ High performance

☐ Server

CentOS Stream 8 VM
centos-stream8-server-small

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

CentOS Stream 9 VM
centos-stream9-server-small

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

CentOS 7 VM
centos7-server-small

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

Fedora VM
fedora-server-small

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

Red Hat Enterprise Linux 7 VM
rhel7-server-small

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

Red Hat Enterprise Linux 8 VM
rhel8-server-small

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

Red Hat Enterprise Linux 9 VM
rhel9-server-small

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

Microsoft Windows 10 VM
windows10-desktop-medium

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

Microsoft Windows 11 VM
windows11-desktop-medium

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

Microsoft Windows Server 2012 R2 VM
windows2k12r2-server-medium

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

VirtualMachines > VirtualMachine details

VM centos-stream9-zealous-anaconda Running VAML Actions

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

Disks

[Add disk](#)

☐ Mount Windows drivers disk

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

Project: openshift-virtualization-os-images

Catalog > Customize template parameters > Customize VirtualMachine

Customize and create VirtualMachine

Template: CentOS Stream 9 VM

Overview | **YAML** | Scheduling | Environment | Network interfaces | Disks | Scripts | Metadata

Name

centos-stream9-pleased-hamster

Namespace

openshift-virtualization-os-images

Description

Not available

Operating system

CentOS Stream 9 VM

CPU | Memory

1 CPU | 2 GiB Memory

Machine type

pc-q35-rhel9.2.0

Boot mode

BIOS

Start in pause mode

☐

Workload profile

Server

Network interfaces (1)

Name	Network	Type
default	Pod networking	Masquerade

Disks (2)

Name	Drive	Size
rootdisk	Disk	30 GiB
cloudinitdisk	Disk	-

Hardware devices

GPU devices

Not available

Host devices

Not available

Headless mode

☐

Hostname

centos-stream9-pleased-hamster

☒ Start this VirtualMachine after creation

Create VirtualMachine Cancel

Demonstração em vídeo

O vídeo a seguir mostra uma demonstração de criação de uma VM no OpenShift Virtualization usando armazenamento iSCSI.

[Crie uma VM no OpenShift Virtualization usando armazenamento em bloco](#)

Crie uma VM com Red Hat OpenShift Virtualization usando storage ONTAP por meio de métodos alternativos

Além do método de criação de uma VM usando a interface de usuário do OpenShift Virtualization, existem outros métodos para criar VMs no OpenShift Virtualization. Esta seção fornece uma visão geral de alguns desses métodos.

Crie uma máquina virtual a partir de um arquivo YAML com Red Hat OpenShift Virtualization

Uma máquina virtual (VM) também pode ser criada aplicando um arquivo YAML com o `oc apply` comando. Esse método é útil para usuários que preferem usar a CLI ou que desejam automatizar a criação de VMs usando scripts. Para criar uma VM usando esse método, você pode criar um arquivo YAML que define a VM e seus recursos associados, como PVCs para os discos da VM. Em seguida, você pode aplicar o arquivo YAML usando o `oc apply -f <yaml-file>` comando para criar a VM no OpenShift cluster.

Mostrar exemplo

```
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

```

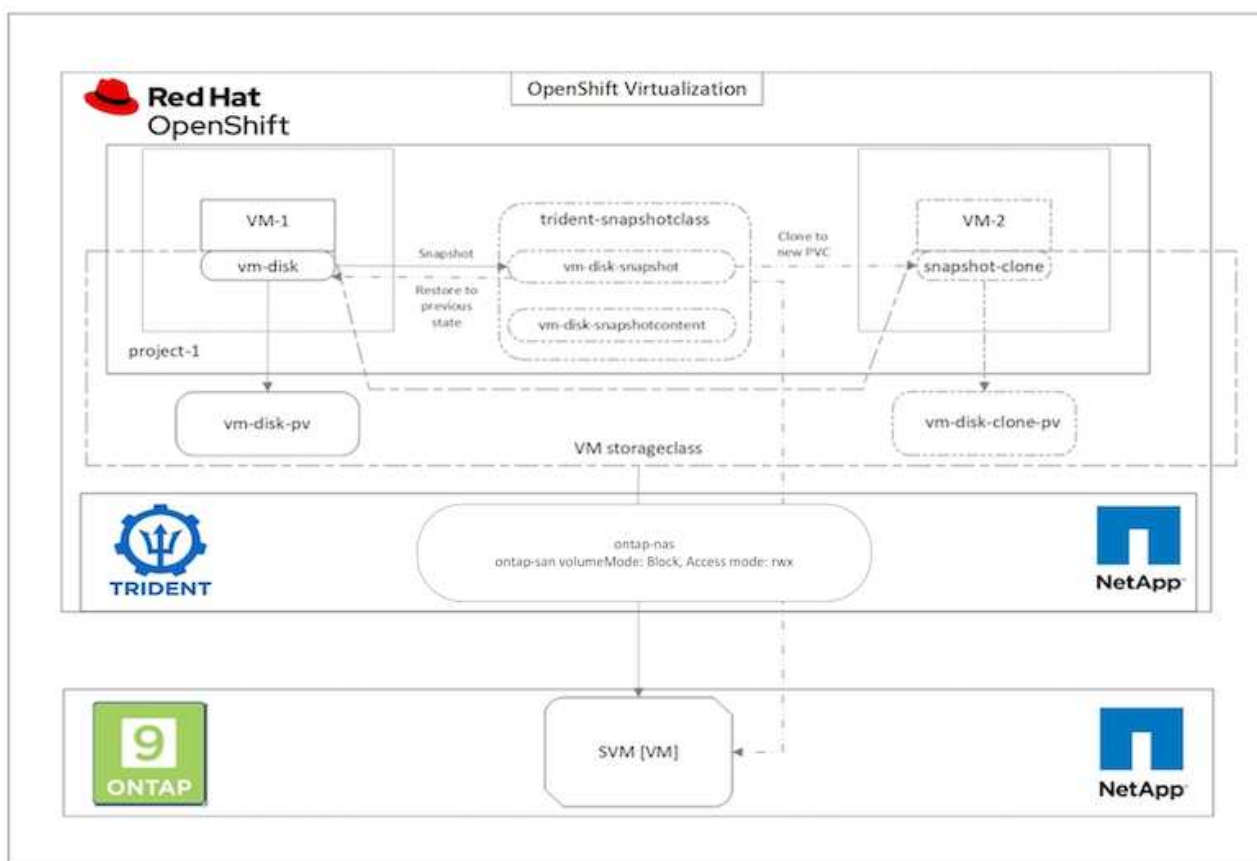
Crie uma VM a partir de uma cópia de snapshot com o Red Hat OpenShift Virtualization

Você pode criar uma máquina virtual a partir de um snapshot com OpenShift Virtualization. Este procedimento inclui criar uma `VolumeSnapshotClass`, tirar um snapshot da persistent volume claim (PVC) da máquina virtual, restaurar o snapshot para uma nova PVC e implantar uma nova máquina virtual que utiliza a PVC restaurada como disco raiz.

Com o Trident e o Red Hat OpenShift, os usuários podem tirar um snapshot de um volume persistente nas Classes de Armazenamento provisionadas por ele. Com esse recurso, os usuários podem fazer uma cópia pontual de um volume e usá-la para criar um novo volume ou restaurar o mesmo volume a um estado anterior. Isso permite ou oferece suporte a uma variedade de casos de uso, desde reversão até clones e restauração de dados.

Para operações de Snapshot no OpenShift, os recursos `VolumeSnapshotClass`, `VolumeSnapshot` e `VolumeSnapshotContent` devem ser definidos.

- Um `VolumeSnapshotContent` é o instantâneo real tirado de um volume no cluster. É um recurso de cluster análogo ao `PersistentVolume` para armazenamento.
- Um `VolumeSnapshot` é uma solicitação para criar um snapshot de um volume. É análogo a um `PersistentVolumeClaim`.
- `VolumeSnapshotClass` permite que o administrador especifique atributos diferentes para um `VolumeSnapshot`. Ele permite que você tenha atributos diferentes para diferentes instantâneos tirados do mesmo volume.



Mostrar exemplo

Pré-requisito

Você deve ter uma VM existente com um PVC anexado a ela, provisionado em uma Storage Class do Trident, e a VM deve estar em execução antes que você possa tirar um snapshot do PVC. Você deve ter uma VolumeSnapshotClass criada que possa ser usada para criar um VolumeSnapshot. Isso deve ter sido criado como parte dos pré-requisitos para instalar o OpenShift Virtualization. Você pode consultar a ["Passo 3 para criar uma configuração VolumeSnapshotClass"](#) para obter detalhes.

Para criar Snapshot de uma máquina virtual, siga os passos abaixo:

1. Identifique o PVC que está anexado à VM de origem e crie um instantâneo desse PVC. Navegar para `Storage > VolumeSnapshots` e clique em Criar VolumeSnapshots.
2. Selecione o PVC para o qual deseja criar o Snapshot, insira o nome do Snapshot ou aceite o padrão e selecione a VolumeSnapshotClass apropriada. Em seguida, clique em Criar.
3. Isso cria um instantâneo do PVC naquele momento.

Create VolumeSnapshot

[Edit YAML](#)

PersistentVolumeClaim *

PVC rhel8-short-frog-rootdisk-28dvb ▼

Name *

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

Snapshot Class *

VSC trident-snapshot-class ▼

Create

Cancel

Criar uma nova máquina virtual a partir do Snapshot

1. Primeiro, restaure o Snapshot em um novo PVC. Navegue até Armazenamento > VolumeSnapshots, clique nas reticências ao lado do Snapshot que você deseja restaurar e clique em Restaurar como novo PVC.
2. Insira os detalhes do novo PVC e clique em Restaurar. Isso cria um novo 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. Em seguida, crie uma nova VM a partir deste PVC. Navegue até Virtualização > Máquinas virtuais e clique em Criar > Com YAML.
2. Na seção spec > template > spec > volumes, especifique o novo PVC criado a partir do Snapshot em vez do disco do contêiner. Forneça todos os outros detalhes da nova VM de acordo com suas necessidades.


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

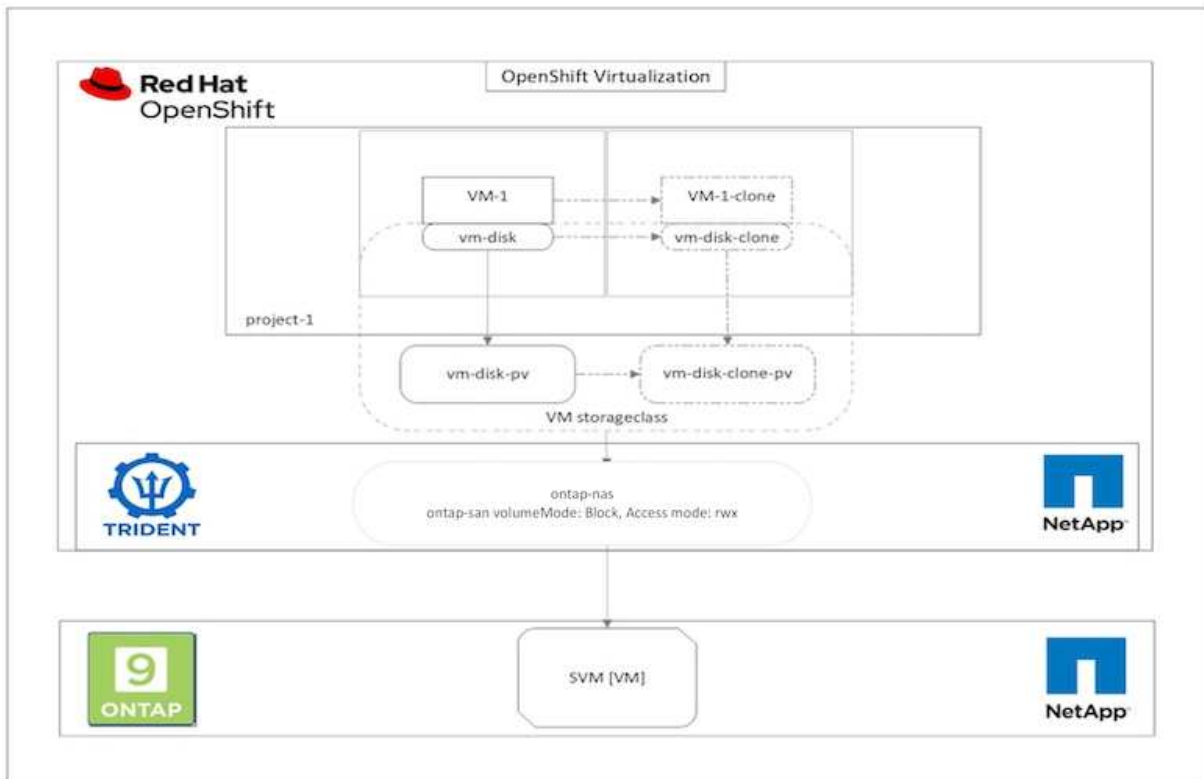
1. Clique em Criar para criar a nova VM.
2. Após a VM ser criada com sucesso, acesse e verifique se a nova VM tem o mesmo estado que o da VM cujo PVC foi usado para criar o snapshot no momento em que o snapshot foi criado.

Clone uma VM na OpenShift Virtualization usando Trident

Este procedimento inclui o uso da clonagem de volumes CSI do Trident, permitindo que você crie uma nova máquina virtual desligando a máquina virtual de origem ou mantendo-a em execução.

Clonagem de VM

A clonagem de uma VM existente no OpenShift é feita com o suporte do recurso de clonagem Volume CSI do Trident. A clonagem de volume CSI permite a criação de um novo PVC usando um PVC existente como fonte de dados, duplicando seu PV. Depois que o novo PVC é criado, ele funciona como uma entidade separada e sem qualquer vínculo ou dependência do PVC de origem.



Há certas restrições a serem consideradas na clonagem de volume CSI:

1. O PVC de origem e o PVC de destino devem estar no mesmo projeto.
2. A clonagem é suportada na mesma classe de armazenamento.
3. A clonagem só pode ser executada quando os volumes de origem e destino usam a mesma configuração VolumeMode; por exemplo, um volume de bloco só pode ser clonado para outro volume de bloco.

VMs em um cluster OpenShift podem ser clonadas de duas maneiras:

1. Ao desligar a VM de origem
2. Mantendo a VM de origem ativa

Desligando a máquina virtual de origem

Clonar uma VM existente desligando-a é um recurso nativo do OpenShift implementado com suporte do Trident. Conclua as etapas a seguir para clonar uma VM.

1. Navegue até Cargas de trabalho > Virtualização > Máquinas virtuais e clique nas reticências ao lado da máquina virtual que você deseja clonar.
2. Clique em Clonar máquina virtual e forneça os detalhes da nova VM.

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.

Cancel

Clone Virtual Machine

3. Clique em Clonar máquina virtual; isso desliga a VM de origem e inicia a criação da VM clone.
4. Após a conclusão desta etapa, você poderá acessar e verificar o conteúdo da VM clonada.

Mantendo a VM de origem ativa

Uma VM existente também pode ser clonada clonando o PVC existente da VM de origem e, em seguida, criando uma nova VM usando o PVC clonado. Este método não exige que você desligue a VM de origem. Conclua as etapas a seguir para clonar uma VM sem desligá-la.

1. Navegue até Armazenamento > PersistentVolumeClaims e clique nas reticências ao lado do PVC que está anexado à VM de origem.
2. Clique em Clonar PVC e forneça os detalhes do novo PVC.

Clone

Name *

rhel8-short-frog-rootdisk-28dvv-clone

Access Mode *

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

Size *

20

GiB

PVC details

Namespace

 default

Requested capacity

20 GiB

Access mode

Shared Access (RWX)

Storage Class

 basic

Used capacity

2.2 GiB

Volume mode

Filesystem

Cancel

Clone

3. Em seguida, clique em Clonar. Isso cria um PVC para a nova VM.
4. Navegue até Cargas de trabalho > Virtualização > Máquinas virtuais e clique em Criar > Com YAML.
5. Na seção spec > template > spec > volumes, anexe o PVC clonado em vez do disco do contêiner. Forneça todos os outros detalhes da nova VM de acordo com suas necessidades.

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

1. Clique em Criar para criar a nova VM.
2. Depois que a VM for criada com sucesso, acesse e verifique se a nova VM é um clone da VM de origem.

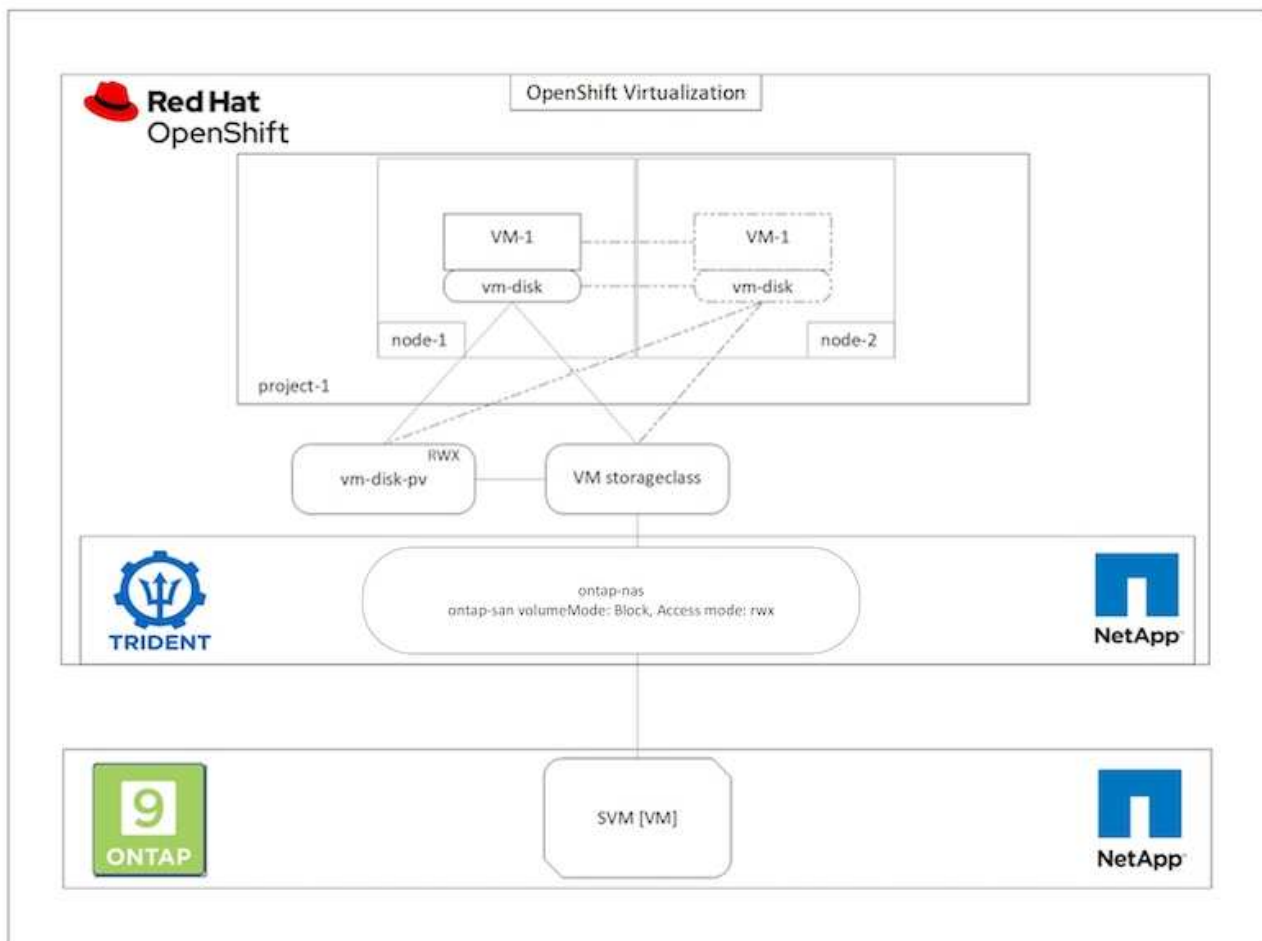
Migrar uma VM entre dois nós em um cluster Red Hat OpenShift

Migre uma VM no OpenShift Virtualization entre dois nós no cluster sem tempo de inatividade. Este procedimento inclui confirmar se os discos usam classes de armazenamento compatíveis com RWX, iniciar a migração e monitorar o progresso.

Migração ao vivo de VM

A migração ao vivo é um processo de migração de uma instância de VM de um nó para outro em um cluster OpenShift sem tempo de inatividade. Para que a migração ao vivo funcione em um cluster OpenShift, as VMs devem ser vinculadas a PVCs com modo de acesso ReadWriteMany compartilhado. Os backends Trident configurados usando drivers `ontap-nas` suportam o modo de acesso RWX para protocolos de sistema de arquivos `nfs` e `smb`. Consulte a documentação ["aqui"](#) . Os backends Trident configurados usando drivers `ontap-san` oferecem suporte ao modo de acesso RWX para `volumeMode` de bloco para protocolos `iSCSI` e `NVMe/TCP`. Consulte a documentação ["aqui"](#) .

Portanto, para que a migração ao vivo seja bem-sucedida, as VMs devem ser provisionadas com discos (discos de inicialização e discos `hot plug` adicionais) com PVCs usando classes de armazenamento `ontap-nas` ou `ontap-san` (`volumeMode`: `Block`). Quando os PVCs são criados, o Trident cria volumes ONTAP em um SVM habilitado para NFS ou iSCSI.

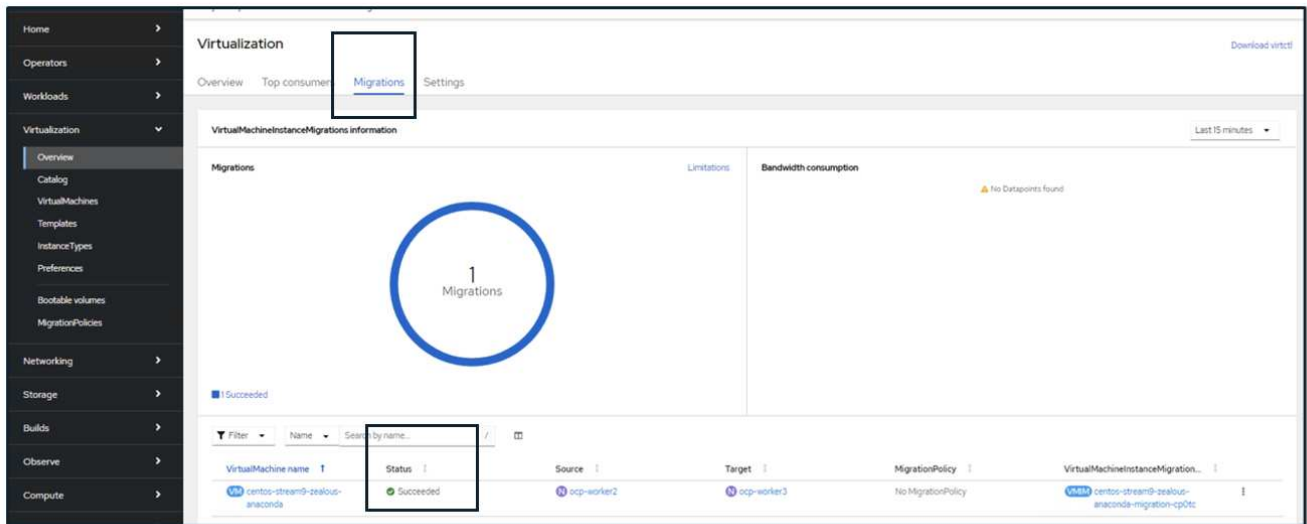
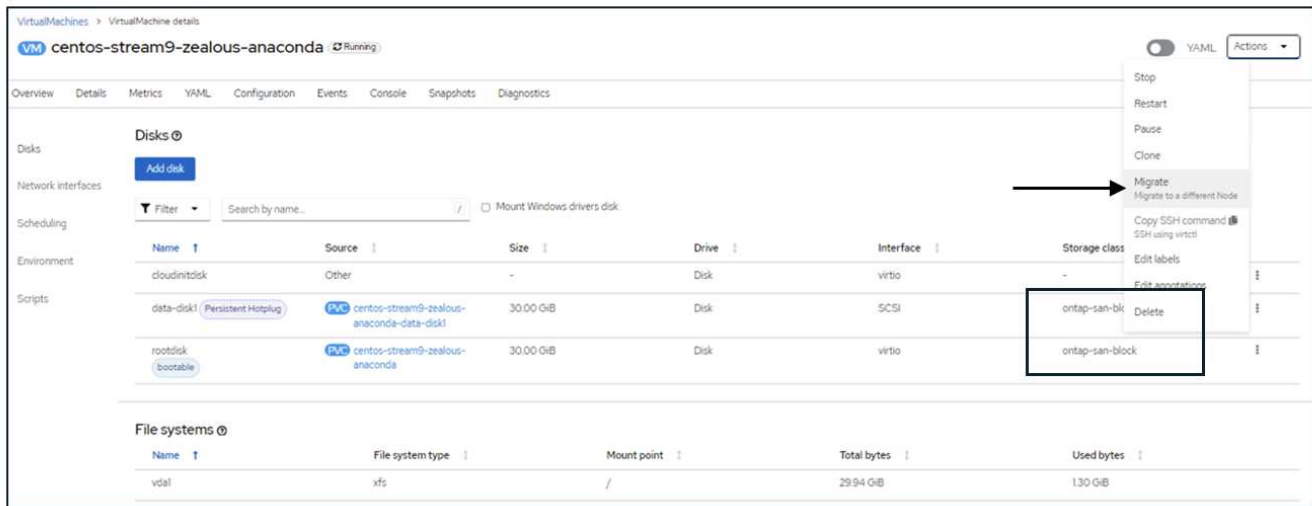


Para executar uma migração ao vivo de uma VM que foi criada anteriormente e está em estado de execução, execute as seguintes etapas:

1. Selecione a VM que você deseja migrar ao vivo.
2. Clique na aba **Configuração**.
3. Certifique-se de que todos os discos da VM sejam criados usando classes de armazenamento que possam suportar o modo de acesso RWX.
4. Clique em **Ações** no canto direito e depois selecione **Migrar**.
5. Para ver o progresso da migração, vá para Virtualização > Visão geral no menu do lado esquerdo e clique na aba **Migrações**. A migração da VM passará de **Pendente** para **Agendado** e depois para **Sucesso**



Uma instância de VM em um cluster OpenShift migra automaticamente para outro nó quando o nó original é colocado no modo de manutenção se evictionStrategy estiver definido como LiveMigrate.



Migração de storage de VMs entre classes de storage em OpenShift Virtualization

Na OpenShift Virtualization, você pode migrar os discos de uma máquina virtual (VM) para uma classe de armazenamento diferente diretamente pelo console da web, sem precisar excluir a VM. Esse processo permite mover todo o storage da VM ou volumes específicos.

Os passos a seguir descrevem como migrar a storage class de uma máquina virtual:

1. Certifique-se de que a classe de armazenamento de destino esteja configurada corretamente e disponível em seu OpenShift cluster. Isso inclui ter o backend de armazenamento necessário (por exemplo, NetApp ONTAP) configurado e acessível.

Mostrar exemplo

```
[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. No console da web do OpenShift Container Platform, navegue até a perspectiva Virtualização e selecione a VM que você deseja migrar.

Mostrar exemplo

The first screenshot shows the 'Overview' tab for a VM named 'rhel9-sc-migration'. It displays details such as Name, Status (Running), Created time, Operating system (Red Hat Enterprise Linux 9.7), CPU/Memory, Time zone, Template, and Hostname. A VNC console window is also visible.

The second screenshot shows the 'Configuration' tab, specifically the 'Disks' section. It lists the disks attached to the VM:

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

Below the disk list, a terminal window shows the output of the command 'oc get vm,pvc -n test-sc-migration', displaying details for the VM and its associated PVCs.

3. Clique em Actions→Migrate→Migrate Storage. Selecione os volumes a serem migrados e a classe de storage de destino, depois clique em Migrate. O processo de migração será iniciado e você poderá monitorar o progresso na guia Events da VM. Assim que a migração for concluída, os discos da VM estarão usando a nova storage class.

Mostrar exemplo

VirtualMachines > VirtualMachine details

VM **rhel9-sc-migration** Running

Overview

Metrics

YAML

Configuration

Events

Console

Snapshots

Diagnostics

Details

Name	rhel9-sc-migration
Status	Running
Created	May 14, 2026, 1:31 PM (19 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

Compute

Namespace

Node

VirtualMachineInstance

Pod

Owner

Migrate VirtualMachine to a different Node

Migrate VirtualMachine storage to a different StorageClass

The VirtualMachine is running

VM

rhel9-sc-migration

virt-launcher-rhel9-sc-migration-5jkdw

No owner

Snapshots (0)

Take snapshot

Migrate VirtualMachine storage

Migrate VirtualMachine storage to a different StorageClass.

1 Migration details

2 Destination StorageClass

3 Review

Migration details

Select the storage to migrate for **rhel9-sc-migration**

☒ The entire VirtualMachine

☐ Selected volumes

Back

Next

Cancel

Migrate VirtualMachine storage

Migrate VirtualMachine storage to a different StorageClass.

1 Migration details

2 Destination StorageClass

3 Review

Destination StorageClass

Select the destination storage for the VirtualMachine storage migration.

sc

sc-nas-economy

Back

Next

Cancel

×

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. Após a conclusão da migração, você pode verificar se os discos da VM agora estão usando a nova classe de armazenamento consultando os detalhes da VM no console da web ou usando o OpenShift CLI.

57

Mostrar exemplo

```
[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 ~]#
```

```
[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-4hd7    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-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-ccea45499e09
                                         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-864d-abf72e710a2f    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 -n test-sc-migration pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
Name:                                pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
Labels:                               <none>
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
Volume Mode:                           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_OGBGMAIOLG/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_8ccf			
▼	oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546dbdb3b	oc-automation-svm	trident_qtree_pool_oc_automation_OGBGMIAOLG	/trident_qtree_pool_oc_automation_OGBGMIAOLG/oc_...

Informações sobre direitos autorais

Copyright © 2026 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTES DOCUMENTOS. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALIENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTES DOCUMENTOS, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.