



Erste Schritte mit Storage und Virtualisierung

NetApp virtualization solutions

NetApp
July 21, 2026

Inhalt

Erste Schritte mit Storage und Virtualisierung	1
Erfahren Sie mehr über die Integration von NetApp Trident mit Red Hat OpenShift	1
NetApp Trident	1
Trident Protect	1
NetApp ONTAP Hardwaremodelle und unterstützte Protokolle	2
Trident Treiber für ONTAP Storage	3
Installieren Sie Trident auf einem Red Hat OpenShift Cluster und erstellen Sie Speicherobjekte	4
Schritt 1: Trident installieren	5
Schritt 2: Bereiten Sie das Speicher-Backend und StorageClass Konfigurationsdateien für Ihre Umgebung vor	9
Schritt 3: Erstellen Sie eine VolumeSnapshotClass Konfigurationsdatei	29
Schritt 4: Wenden Sie die Konfigurationsdateien auf Ihren Cluster an	29
Schritt 5: Legen Sie die standardmäßigen Trident Storage- und Snapshot-Klassen fest	31
Voraussetzungen für die Bereitstellung von Red Hat OpenShift Virtualization mit ONTAP	32
Voraussetzungen	32
NetApp Trident CSI, Speicherprofile, Datenträgerzugriffsmodi für OpenShift Virtualisierung	33
Erstellen einer VM mit ONTAP Speicher und Red Hat OpenShift Virtualization	38
VM erstellen	39
Videodemonstration	42
Erstellen einer VM mit Red Hat OpenShift Virtualization unter Verwendung von ONTAP Storage durch alternative Methoden	42
Erstellen einer VM aus einer YAML-Datei mit Red Hat OpenShift Virtualization	42
Erstellen Sie eine VM aus einer Snapshot-Kopie mit Red Hat OpenShift Virtualization	44
Klonen einer VM in OpenShift Virtualisierung mit Trident	48
Migrieren Sie eine VM zwischen zwei Knoten in einem Red Hat OpenShift-Cluster	52
VM Live Migration	52
Speichermigration von VMs zwischen Speicherklassen in OpenShift Virtualisierung	54

Erste Schritte mit Storage und Virtualisierung

Erfahren Sie mehr über die Integration von NetApp Trident mit Red Hat OpenShift

Speicher ist in containerisierten Anwendungen von entscheidender Bedeutung, da er sicherstellt, dass Daten in dynamischen und oft kurzlebigen Umgebungen dauerhaft verfügbar und effizient verwaltet werden. Moderne Speicherlösungen für Container sind so konzipiert, dass sie nahtlos mit den Anwendungen skalieren und die Flexibilität bieten, Speicherressourcen dynamisch entsprechend den Anforderungen der Anwendungen bereitzustellen, wodurch sowohl Wachstum als auch wechselnde Arbeitslasten unterstützt werden. Da OpenShift Virtualization es den VMs ermöglicht, als Container ausgeführt zu werden, verwenden sie denselben Speicher wie andere Anwendungen in den OpenShift Clustern.

NetApp ONTAP ist eine robuste und hochskalierbare Datenmanagement-Software, die als Grundlage für die Speicherlösungen der Enterprise-Klasse von NetApp dient. Die Software bietet eine einheitliche Speicherarchitektur, die sich nahtlos in verschiedene Speicherumgebungen integrieren lässt, darunter On-Premises-, Cloud- und Hybrid-Setups. ONTAP unterstützt eine Vielzahl von Protokollen wie NFS, CIFS/SMB, iSCSI, FC und NVMe über TCP, was eine nahtlose Integration mit unterschiedlichen Anwendungen und Workloads ermöglicht. Die Konsolidierung von SAN, NAS und Objektspeicher auf einer einzigen Plattform wird ermöglicht, wodurch das Management vereinfacht und Kosten reduziert werden. NetApp ONTAP ist sowohl On-Premises als auch als Managed Service in der Cloud verfügbar.

NetApp Trident

Trident ist ein Open-Source-Speicherorchestrator, der von NetApp entwickelt wurde und dazu dient, die Nutzung von ONTAP Storage in containerisierten Umgebungen zu erleichtern. Er ermöglicht die dynamische Bereitstellung und Verwaltung von persistentem Speicher für containerisierte Anwendungen.

Hauptmerkmale von Trident

CSI-Compliance Gewährleistung der Kompatibilität mit den neuesten Kubernetes-Speicherfunktionen und -Standards. **Deklarative Konfiguration** Ermöglicht es Nutzern, Speicheranforderungen in YAML-Dateien zu definieren, die anschließend automatisch von Trident verwaltet werden. **Treiber** Trident unterstützt Treiber für NFS-, CIFS/SMB-, iSCSI-, FC- und NVMe/TCP-Protokolle, die von ONTAP unterstützt werden. **Hybrid- und Multi-Cloud-Unterstützung** Trident unterstützt Treiber für ONTAP Storage vor Ort und verwaltete Storage-Services bei den Hyperscalern, namentlich FSx für NetApp ONTAP in AWS, Azure NetApp Files in Azure und Google Cloud NetApp Volumes in Google Cloud. **Erweitertes Datenmanagement** Nutzung der Snapshot- und Klonfunktionen von ONTAP für effizienten Datenschutz und die schnelle Bereitstellung von Test- und Entwicklungsumgebungen.

Vollständige Details zu Trident finden sich unter NetApp Trident ["Dokumentation hier"](#)

Trident Protect

Trident Protect ist eine erweiterte Funktion von Trident, die verbesserten Datenschutz und Sicherheit für containerisierte Umgebungen bietet. Sie wurde entwickelt, um sicherzustellen, dass kritische Daten gegen potenzielle Bedrohungen und Datenverlust-Szenarien geschützt sind. Diese Funktion ist für alle Trident Nutzer kostenlos verfügbar und kann in jedem OpenShift Cluster mit installiertem Trident einfach aktiviert werden.

Hauptmerkmale von Trident Protect

Automatisierte Backups Trident Protect ermöglicht automatisierte, richtlinienbasierte Backups von Containeranwendungen und VMs, sodass diese regelmäßig gesichert werden, ohne dass eine manuelle Intervention erforderlich ist.

Snapshot-Management Es nutzt die Snapshot-Funktionen von ONTAP, um häufige, zeitpunktgenaue Kopien von Containeranwendungen und VMs zu erstellen und stellt so einen zuverlässigen Mechanismus für die Wiederherstellung bereit.

Datenverschlüsselung Stellt sicher, dass Daten sowohl im Ruhezustand als auch während der Übertragung verschlüsselt sind, wodurch strenge Sicherheitsanforderungen erfüllt und sensible Informationen geschützt werden.

Compliance und Auditing Bietet Tools und Funktionen, die Organisationen dabei unterstützen, Compliance-Anforderungen zu erfüllen und regelmäßige Audits ihrer Datenschutzpraktiken durchzuführen.

Disaster Recovery Integriert sich in die Replikationsfunktionen von ONTAP, um robuste Disaster-Recovery-Lösungen bereitzustellen und die Geschäftskontinuität auch im Falle katastrophaler Ereignisse zu gewährleisten.

Vollständige Informationen zu Trident Protect finden sich unter NetApp Trident Protect "[Dokumentation hier](#)".

NetApp ONTAP Hardwaremodelle und unterstützte Protokolle

NetApp ONTAP Software läuft auf einer Vielzahl von Hardwaremodellen, die jeweils auf unterschiedliche Leistungs- und Kapazitätsanforderungen ausgelegt sind. Trident erweitert seine umfassenden Funktionen, um verschiedene NetApp ONTAP Systeme zu unterstützen, darunter All Flash FAS (AFF), All SAN Array (ASA) und ASA R2 Systeme. Diese Unterstützung stellt sicher, dass Unternehmen das volle Potenzial ihrer Hochleistungs-Storage-Lösungen in containerisierten Umgebungen ausschöpfen können.

All Flash FAS (AFF) Systeme AFF Systeme bieten außergewöhnliche Leistung und geringe Latenz, was sie ideal für anspruchsvolle Anwendungen macht.

All SAN Array (ASA) Systeme ASA Systeme sind für dedizierte SAN-Umgebungen konzipiert und bieten robuste Leistung und Zuverlässigkeit.

ASA R2 Systems ASA R2-Systeme bieten erweiterte Funktionen und Möglichkeiten für SAN-Umgebungen.

AFX NetApp AFX ist eine speziell entwickelte, disaggregierte All-Flash-Storage-Architektur, die für KI-Workloads der Enterprise-Klasse konzipiert ist. Sie bietet leistungsstarkes NAS und ermöglicht die unabhängige Skalierung von Rechenleistung und Kapazität. NetApp AFX-Speichersysteme unterscheiden sich von anderen ONTAP-basierten Systemen (ASA, AFF und FAS) in der Implementierung ihrer Speicherschicht. AFX verwendet ein verteiltes Dateisystem, das hohe Leistung und Skalierbarkeit ermöglicht, während andere ONTAP-basierte Systeme eine traditionelle Speicherarchitektur nutzen.

1. Unterstützte Protokolle für AFF: NFS, CIFS/SMB, iSCSI, FC, NVMe/TCP
2. Unterstützte Protokolle für ASA: iSCSI, FC, NVMe/TCP
3. Für ASA R2 unterstützte Protokolle: iSCSI, FC, NVMe/TCP
4. Unterstützte Protokolle für AFX: Ab Trident 25.10 wird nur das NFS-Protokoll unterstützt; das SMB-Protokoll wird nicht unterstützt.

Trident Treiber für ONTAP Storage

Trident umfasst die folgenden CSI-Treiber, die jeweils ein Volume im Backend-Storage bereitstellen können

Für NAS-Protokolle auf unterstützten Hardwaremodellen

1. `ontap-nas`: Ein PVC ist auf 1 PV abgebildet, das ein dediziertes FlexVolume ist.
2. `ontap-nas-economy`: Jedes bereitgestellte PV ist ein Qtree mit einer konfigurierbaren Anzahl von Qtrees pro FlexVolume (Standardwert ist 200, konfigurierbar zwischen 50 und 300). Ein PVC entspricht einem PV, das ein Qtree auf einem gemeinsam genutzten FlexVolume ist.



Sie können alle Festplatten der VMs als Qtrees in einem einzigen Volume ablegen. Es ist jedoch zu beachten, dass Snapshots, Klone und Replikationsfunktionen von Trident nicht unterstützt werden. Wenn diese Funktionen vom Storage-Administrator verwaltet werden, erfolgt die Verwaltung nicht auf PV-Ebene.

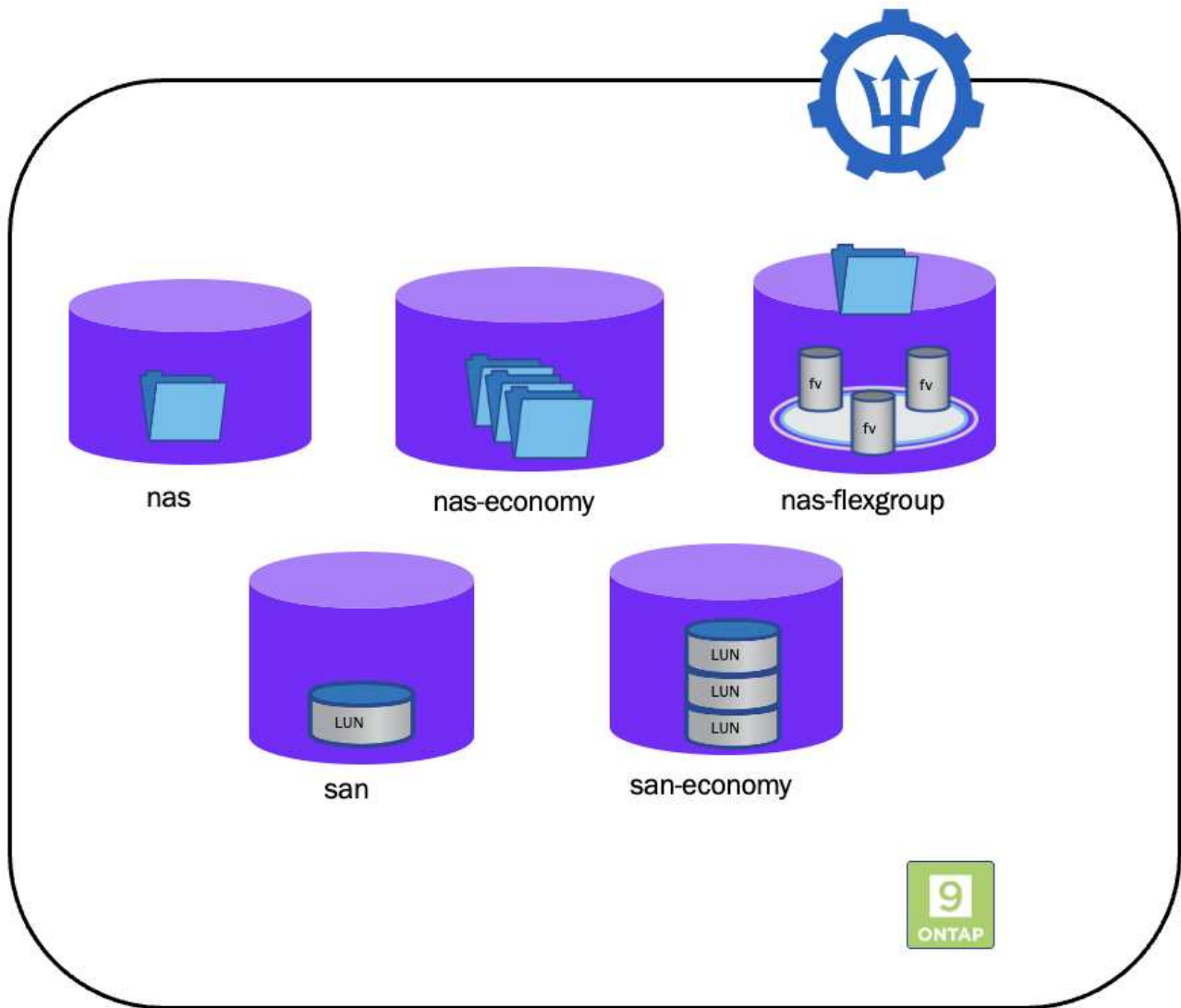
1. `ontap-nas-flexgroup`: Jeder bereitgestellte PV ist ein vollständiger ONTAP FlexGroup, und alle einem SVM zugewiesenen Aggregate werden verwendet.

Für SAN-Protokolle auf unterstützten Hardwaremodellen

1. `ontap-san`: Ein PVC ist einem PV zugeordnet, das eine LUN auf einem dedizierten FlexVolume ist.
2. `ontap-san-economy`: Ein PVC entspricht einem PV, das eine LUN auf einem gemeinsam genutzten Flexvolume ist. Mehrere LUNs können sich in dem gemeinsam genutzten Flexvolume befinden (Standard ist 100, konfigurierbar zwischen 50 und 200 LUNs/PVs pro FlexVol Volume).



Sie können alle Festplatten der VMs als LUNs in einem einzigen Volume ablegen. Dieser Treiber unterstützt zwar Snapshots und Klone, jedoch keine Replikation. Wenn die Replikation vom Storage-Administrator verwaltet wird, ist sie nicht PV-granular.



Eine vollständige Liste der über Trident Treiber bereitgestellten Funktionen sowie derjenigen, die vom Speicheradministrator angewendet werden müssen, ist im "[ONTAP-Backend-Treiber](#)" Abschnitt in der Trident-Dokumentation zu finden.

Installieren Sie Trident auf einem Red Hat OpenShift Cluster und erstellen Sie Speicherobjekte

Installieren Sie Trident mit dem Red Hat Certified Trident Operator und erstellen Sie Speicherobjekte für ONTAP und Amazon FSx für NetApp ONTAP, um die dynamische Volumenbereitstellung für Container und VMs zu aktivieren. Bereiten Sie Worker-Knoten bei Bedarf für den Blockzugriff vor.

Bevor Sie beginnen

- Führen Sie die Schritte auf dieser Seite durch, bevor Sie OpenShift Virtualization installieren. OpenShift Virtualization erfordert eine standardmäßige, von Trident unterstützte StorageClass und VolumeSnapshotClass, um Golden Images für VM-Vorlagen zu erstellen.
- Wenn Sie OpenShift Virtualization bereits vor der Konfiguration von Trident installiert haben, löschen Sie

alle Golden Images, die mit einer anderen Speicherklasse erstellt wurden. Nachdem Sie Trident als Standard festgelegt haben, erstellt OpenShift Virtualization die Golden Images mit Trident-Speicher neu.

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

Schritt 1: Trident installieren

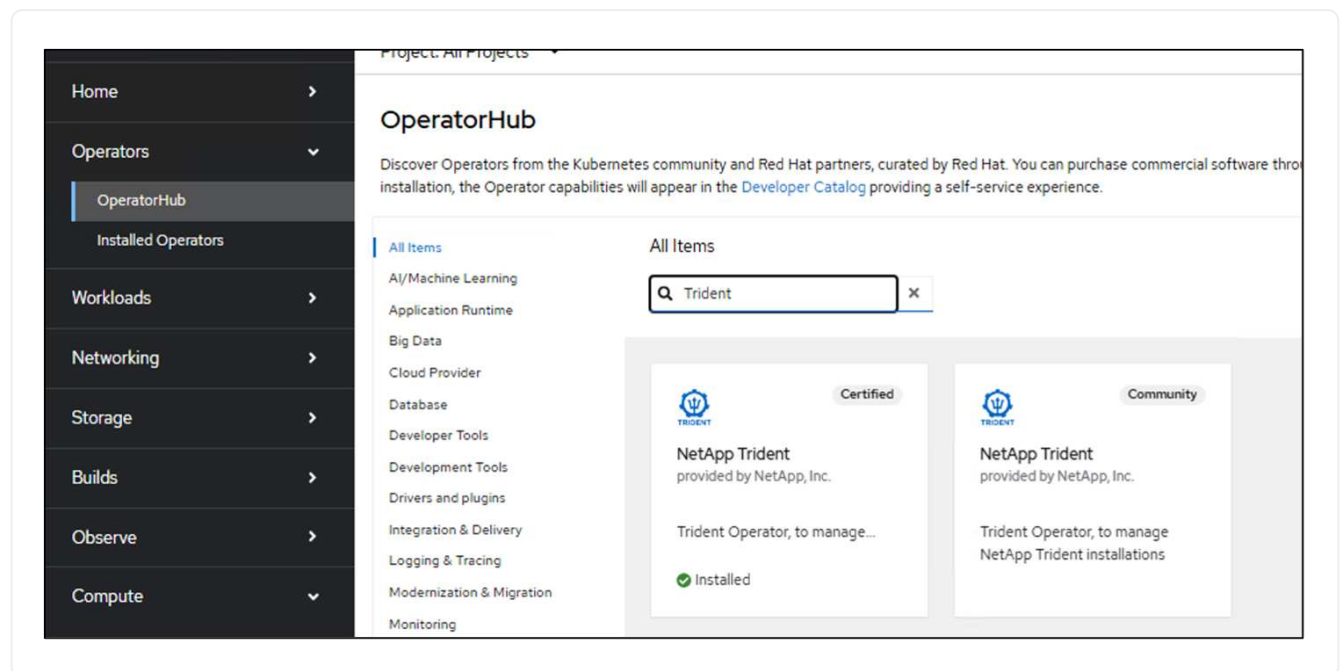
Der Red Hat Certified Trident Operator wird von NetApp für OpenShift On-Premises, in öffentlichen Clouds und in Managed Services wie ROSA unterstützt. Ab Trident 25.02 kann der Operator auch Worker-Knoten für iSCSI vorbereiten, wenn Sie Amazon FSx for NetApp ONTAP verwenden und planen, OpenShift Virtualization VM-Workloads auszuführen.

Weitere Installationsoptionen finden Sie unter ["die Trident Dokumentation"](#).

Schritte

1. In **OperatorHub** wählen Sie **Certified NetApp Trident**.

Beispiel anzeigen



2. Auf der Seite **Install** die neueste Version beibehalten und **Install** auswählen.

Beispiel anzeigen

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. Nach der Installation des Operators wählen Sie **View operator** und erstellen eine Instanz des Trident Orchestrator.

Wenn Sie Worker-Knoten für iSCSI vorbereiten möchten, wechseln Sie zur YAML-Ansicht und fügen Sie `iscsi` zu `nodePrep` hinzu.

Beispiel anzeigen

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. Vergewissern Sie sich, dass alle Trident Pods im Cluster ausgeführt werden.

Beispiel anzeigen

```
[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. Wenn Sie die iSCSI-Knotenvorbereitung aktiviert haben, melden Sie sich an den Worker-Knoten an und überprüfen Sie, ob `iscsid` und `multipathd` aktiv sind und dass `multipath.conf` Einträge enthält.

Beispiel anzeigen

```
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#
```

Beispiel anzeigen

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

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

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

Videodemonstration

Das folgende Video zeigt eine Demonstration der Installation von Trident mit dem Red Hat Certified Trident Operator.

[Installieren von Trident 25.02.1 mit dem zertifizierten Trident Operator in OpenShift](#)

Schritt 2: Bereiten Sie das Speicher-Backend und StorageClass Konfigurationsdateien für Ihre Umgebung vor

Erstellen Sie TridentBackendConfig und StorageClass Definitionen für Ihre Umgebung. Sie können mehrere Speicherprotokolle innerhalb Ihrer Umgebung konfigurieren. Erstellen Sie YAML-Dateien für jedes Protokoll, das Sie verwenden möchten, und ersetzen Sie die Platzhalterwerte durch Ihre spezifischen Konfigurationsdetails.



Füllen Sie je nach Ihrer Umgebung entweder den Abschnitt „On-Premises“ oder „ROSA“ unten aus und fahren Sie dann mit Schritt 3 fort.

Lokale OpenShift Cluster

Erstellen Sie YAML-Dateien für jedes Protokoll, das Sie konfigurieren möchten. Sie können eines oder mehrere der folgenden Protokolle konfigurieren: NAS für NFS-basierte Dateispeicherung, iSCSI für iSCSI-Blockspeicherung, NVMe/TCP für leistungsstarke NVMe über TCP-Blockspeicherung oder FC für Fibre Channel-Blockspeicherung.

Erstellen Sie eine TridentBackendConfig und eine StorageClass für jedes Protokoll. Die Backend-Konfiguration umfasst Anmeldeinformationen, die in einem Kubernetes Secret gespeichert sind.

Wesentliche Richtlinien für grundlegende Optionen der Backend-Konfiguration

1. Die Backend-Konfigurationsdateien verweisen auf Ihre **ONTAP SVM** und **Management LIF**. +
2. Sie können sich mit **Benutzername und Passwort** bei ONTAP authentifizieren. Es kann sich um den Benutzernamen und das Passwort des Cluster-Administrators oder des vServer-Administrators handeln. +
3. Sie können auch ein **Clientzertifikat** zur Authentifizierung mit ONTAP verwenden. Einzelheiten dazu, wie das Clientzertifikat, der private Schlüssel und das CA-Zertifikat generiert werden, finden sich unter "[Trident Dokumentation hier](#)".

Wesentliche Richtlinien für Backend-Konfigurationsoptionen für den NAS Economy-Treiber

Für die Konfiguration des **NAS economy driver** können die folgenden Parameter einbezogen werden:

1. **qtreesPerFlexvol** Der Wert muss im Bereich von 50 bis 300 liegen, der Standardwert ist 200. +
2. **limitVolumePoolSize** Der Wert ist die maximale FlexVol-Größe, die bei der Verwendung von Qtrees im ontap-nas-economy Backend angefordert wird. Es ist kein Standardwert festgelegt, und dieser Parameter wird standardmäßig nicht erzwungen. Wenn dieser Parameter gesetzt ist, erstellt der Trident Operator einen neuen FlexVol, sobald der aktuelle FlexVol die angegebene Größe erreicht. Dieser Parameter kann dabei helfen, die Kapazitätsauslastung im Cluster zu steuern, indem die Größe der für die Bereitstellung von PVs verwendeten FlexVols beschränkt wird. +
3. **denyNewVolumePools** Dieser Parameter verhindert, dass ontap-nas-economy Backends neue FlexVol Volumes zur Aufnahme ihrer Qtrees erstellen. Für die Bereitstellung neuer PVs werden ausschließlich bereits vorhandene FlexVols verwendet.

Wesentliche Richtlinien für Backend-Konfigurationsoptionen für den SAN Treiber

Die folgenden Parameter können für die **SAN Treiber**-Konfiguration einbezogen werden:

1. **limitVolumeSize** Der Wert ist die maximale Volume-Größe, die bei Verwendung eines SAN-Backends angefordert wird. Es ist kein Standardwert festgelegt, und dieser Parameter wird standardmäßig nicht erzwungen. Wenn dieser Parameter festgelegt ist, schlägt die Bereitstellung durch den Trident Operator fehl, wenn die angeforderte Volume-Größe diesen Wert überschreitet. Mit diesem Parameter lässt sich die maximale Größe der von ihm für LUNs verwalteten Volumes beschränken. +
2. **lunsPerFlexvol** Maximale Anzahl an LUNs pro FlexVol, muss im Bereich [50, 200] liegen, der Standardwert ist 100. Wenn Sie diesen Parameter festlegen, erstellt der Trident Operator ein neues FlexVol, wenn das aktuelle FlexVol die angegebene Anzahl an LUNs erreicht. Dieser Parameter kann Ihnen dabei helfen, die Kapazitätsauslastung in Ihrem Cluster zu steuern, indem die Anzahl der LUNs pro FlexVol, die für die Bereitstellung von PVs verwendet werden, begrenzt wird.

Wesentliche Richtlinien für Backend-Konfigurationsoptionen für den SAN Economy-Treiber

Die folgenden Parameter können für die Konfiguration des **SAN economy driver** einbezogen werden:

1. **limitVolumePoolSize** Der Wert ist die maximale FlexVol-Größe, die bei Verwendung des SAN economy Backends angefordert wird. Es ist kein Standardwert festgelegt, und dieser Parameter wird standardmäßig nicht erzwungen. Wenn dieser Parameter festgelegt ist, begrenzt der Trident Operator die maximale FlexVol-Größe, die für die Erstellung von LUNs im FlexVol erstellt werden kann. Dieser Parameter kann dazu beitragen, die Kapazitätsauslastung im Cluster zu steuern, indem die Größe der für die Bereitstellung von PVs verwendeten FlexVols eingeschränkt wird. +
2. **denyNewVolumePools** Dieser Parameter verhindert, dass SAN Economy Backends neue FlexVol Volumes zur Aufnahme ihrer LUNs erstellen. Für die Bereitstellung neuer PVs werden ausschließlich bereits vorhandene FlexVols verwendet.

Wesentliche Richtlinien für Backend-Konfigurationsoptionen zur Bereitstellung von Volumes



In den für jedes Protokoll bereitgestellten YAML-Beispieldateien wird, falls storagePrefix nicht angegeben ist, das Standardpräfix „trident“ verwendet. Bei Verwendung von ontap-nas-economy und einem storagePrefix von 24 oder mehr Zeichen wird das Speicherpräfix nicht in die Qtrees eingebettet, ist aber im Volume-Namen enthalten.

Wenn in ONTAP direkt über den System Manager oder über die CLI ein Volume erstellt wird, ohne Werte für alle optionalen Parameter anzugeben, werden während der Bereitstellung eines Volumes automatisch einige Standardwerte festgelegt, wie unten dargestellt.

```
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
```

Hier ist ein Beispiel für ein Volume, das mit der Trident Backend-Konfiguration erstellt wurde. Im Abschnitt „Standardeinstellungen“ sind außer der Namensvorlage keine Optionen explizit festgelegt.

```
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
```

Wenn das Volume mit dieser Backend-Konfiguration erstellt wird, werden die Optionen, die nicht in der Backend-Konfigurationsdatei festgelegt sind, auf die unten dargestellten Standardwerte gesetzt.

```
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
```

Wenn Sie jedoch im Standardabschnitt der Trident-Backend-Konfiguration Werte festlegen, können Volumes erstellt werden, die spezifische Werte für diese Parameter aufweisen. Im folgenden Beispiel ist snapshotPolicy auf die Standardrichtlinie gesetzt (sie ist none, wenn nichts explizit festgelegt wird), und snapshotDir ist im Standardabschnitt der Backend-Konfigurations-YAML auf true gesetzt. Mit diesen Einstellungen erhält das Backend-Volume diese Werte. (snapdir-access ist true und snapshot-policy ist 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
```

Alle weiteren Parameter sind in der Trident Dokumentation aufgeführt:

1. ["Trident NAS Backend Konfigurationsoptionen"](#) +
2. ["Trident SAN Backend-Konfigurationsoptionen"](#)

NAS

Erstellen Sie eine TridentBackendConfig und StorageClass für ONTAP NAS, um die Bereitstellung von persistentem Speicher auf NFS-Basis zu ermöglichen. Die Backend-Konfiguration enthält Anmeldeinformationen, die in einem Kubernetes Secret gespeichert sind, und verweist auf Ihre ONTAP SVM und Management-LIF.

Beispiel für ein Backend-Geheimnis und eine Backend Konfigurationsdatei mit Authentifizierung über Benutzernamen und Passwort (speichern als `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
```

Beispiel für ein Backend-Geheimnis und eine Backend Konfigurationsdatei mit Clientzertifikat-Authentifizierung (speichern unter `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 definition (speichern als `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

Erstellung einer TridentBackendConfig und StorageClass für den ONTAP NAS economy Treiber, um die NFS-basierte Bereitstellung von persistentem Speicher mithilfe von Qtrees zu ermöglichen. Die Backend-

Konfiguration enthält Anmeldeinformationen, die in einem Kubernetes Secret gespeichert sind, und verweist auf Ihre ONTAP SVM und Management-LIF.

Beispiel für ein Backend-Geheimnis und eine Backend Konfigurationsdatei mit Authentifizierung über Benutzername und Passwort (speichern als `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
  }}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-nas-economy-secret
```

StorageClass-Definition (speichern als `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
```



Einzelheiten zu zusätzlichen Parametern, die in die Backend Konfigurationsdatei für den ontap-nas-economy Treiber aufgenommen werden können, sind im obigen Abschnitt „Wichtige Richtlinien für Backend Konfigurationsoptionen für den NAS economy Treiber“ zu finden.

iSCSI SAN

Eine TridentBackendConfig und StorageClass für ONTAP SAN mit iSCSI ermöglichen die Bereitstellung von iSCSI-basiertem Blockspeicher. Die Backend-Konfiguration verwendet den ontap-san Treiber und enthält Anmeldeinformationen, die in einem Kubernetes Secret gespeichert sind. Der sanType Parameter verwendet standardmäßig das iSCSI-Protokoll, wenn er weggelassen wird.

Backend-Geheimnis und Backend Konfigurationsdatei mit Benutzername-Passwort-Authentifizierung (speichern als `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

```

Backend-Geheimnis und Backend Konfigurationsdatei mit Clientzertifikats-Authentifizierung (speichern als 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 (speichern als `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

Erstellen Sie eine `TridentBackendConfig` und `StorageClass` für ONTAP SAN mit NVMe über TCP, um die Bereitstellung von Blockspeicher mit hoher Leistung zu ermöglichen. Die Backend-Konfiguration verwendet den für NVMe/TCP-Transport optimierten `ontap-san`-Treiber und enthält Anmeldeinformationen, die in einem Kubernetes Secret gespeichert sind.

Backend-Geheimnis und Backend Konfigurationsdatei mit Benutzername-Passwort-Authentifizierung (speichern als `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
```

Backend-Geheimnis und Backend Konfigurationsdatei mit Clientzertifikats-Authentifizierung (speichern als `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 (speichern als `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
```



Wenn Sie planen, ReadWriteMany (RWX)-Volumes mit dem NVMe-Protokoll zu verwenden, kann ein einzelnes RWX-NVMe-Volume von nicht mehr als 64 Knoten in einem Kubernetes-Cluster eingebunden werden. Weitere Informationen zu dieser Einschränkung sowie Informationen zum in Trident 26.02 eingeführten Super-Subsystem-Modell und zum Per-Volume-Subsystem-Modell vor Trident 26.02 enthält die Trident Dokumentation: <https://docs.netapp.com/us-en/trident/trident-use/ontap-san-prep.html#nvme-tcp-considerations>.

FC SAN

Erstellen Sie eine TridentBackendConfig und StorageClass für ONTAP SAN mit Fibre Channel, um die FC-basierte Blockspeicherbereitstellung zu ermöglichen. Die Backend-Konfiguration verwendet den ontap-san-Treiber mit dem angegebenen FCP-Protokoll und enthält Anmeldeinformationen, die in einem Kubernetes Secret gespeichert sind.

Das Authentifizierungsverfahren mit Benutzername/Passwort oder das Authentifizierungsverfahren mit Clientzertifikat kann zur Authentifizierung bei ONTAP verwendet werden. Einzelheiten zur Generierung des Clientzertifikats, des privaten Schlüssels und des CA-Zertifikats finden sich unter "[Trident Dokumentation hier](#)". Beispiele für beide Verfahren sind unten aufgeführt.

Beispiel anzeigen

Backend-Geheimnis und Backend Konfigurationsdatei mit ONTAP Benutzernamen- und Passwortauthentifizierung (speichern unter `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
  }}"
    credentials:
      name: tbc-fc-secret
```

Backend-Geheimnis und Backend Konfigurationsdatei mit Clientzertifikats-Authentifizierung (speichern als `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
```

Eine TridentBackendConfig und StorageClass für ONTAP SAN Economy mit iSCSI ermöglichen die Bereitstellung von iSCSI-basiertem Blockspeicher. Die Backend-Konfiguration verwendet den `ontap-san-economy` Treiber mit dem iSCSI-Protokoll und enthält Anmeldeinformationen, die in einem Kubernetes Secret gespeichert sind.

Backend-Geheimnis und Backend Konfigurationsdatei mit Clientzertifikats-Authentifizierung (speichern als `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"
```

In der Trident-Dokumentation finden sich weitere Beispiele für YAML-Dateien sowie Informationen zu den von SAN-Treibern und NAS-Treibern unterstützten Zugriffsmodi und Volumenmodi.

1. ["Beispiele mit NAS-Treibern"](#)
2. ["Beispiele mit SAN Treibern"](#)

ROSA clusters mit Amazon FSx for NetApp ONTAP

Erstellen Sie YAML-Dateien für jedes Protokoll, das Sie konfigurieren möchten. Sie können eines oder beide der folgenden Protokolle konfigurieren: NAS für NFS-basierte Dateispeicherung oder iSCSI für Blockspeicherung.

NAS

Erstellen Sie eine TridentBackendConfig und StorageClass für Amazon FSx for NetApp ONTAP mit ONTAP NAS, um die Bereitstellung von NFS-basiertem persistentem Speicher auf ROSA-Clustern zu ermöglichen. Die Backend-Konfiguration verwendet Amazon FSx for NetApp ONTAP DNS-Namen für Management- und Daten-LIFs und enthält Anmeldeinformationen, die in einem Kubernetes Secret im trident Namespace gespeichert sind.

Backend-Geheimnis und Backend-Konfigurationsdatei (speichern als `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 (speichern als `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

Erstellen Sie eine TridentBackendConfig und StorageClass für Amazon FSx for NetApp ONTAP mit ONTAP SAN, um die iSCSI-basierte Blockspeicher-Bereitstellung auf ROSA-Clustern zu ermöglichen. Die Backend-Konfiguration verwendet den ontap-san-Treiber und enthält Anmeldeinformationen, die in einem Kubernetes Secret gespeichert sind. Stellen Sie sicher, dass die Worker-Knoten für iSCSI-Zugriff vorbereitet sind.

Backend-Geheimnis und Backend-Konfigurationsdatei (speichern als `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-Definition (speichern als `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
```

Schritt 3: Erstellen Sie eine VolumeSnapshotClass Konfigurationsdatei

Erstellen Sie eine VolumeSnapshotClass-Definition für lokale und ROSA-Bereitstellungen. Diese Konfiguration ermöglicht Snapshot-basierte Operationen für persistente Volumes.

VolumeSnapshotClass-Definition (speichern als `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
```

Schritt 4: Wenden Sie die Konfigurationsdateien auf Ihren Cluster an

Wenden Sie die in den vorherigen Schritten erstellten Konfigurationsdateien auf Ihren OpenShift-Cluster an.

Schritte

1. Wenden Sie die TridentBackendConfig und StorageClass Dateien für jedes von Ihnen konfigurierte Protokoll an.

Für lokale Cluster:

```
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
```

Für ROSA-Cluster:

```
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. Wenden Sie die VolumeSnapshotClass-Konfiguration an.

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

3. Überprüfen Sie, ob die Ressourcen erfolgreich erstellt wurden.

Prüfen Sie TridentBackendConfig-Objekte:

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

Prüfen Sie StorageClass-Objekte:

```
oc get storageclass
```

Überprüfen Sie VolumeSnapshotClass:

```
oc get volumesnapshotclass
```

Schritt 5: Legen Sie die standardmäßigen Trident Storage- und Snapshot-Klassen fest

Legen Sie die Trident StorageClass und VolumeSnapshotClass als Standardwerte im OpenShift-Cluster fest. Dies ist erforderlich, damit OpenShift Virtualization Golden-Image-Quellen für VM-Vorlagen erstellen kann.

Schritte

1. Legen Sie die Standard-Trident-StorageClass fest.

Legen Sie eine von Trident unterstützte StorageClass als Standard für den Cluster fest, damit PersistentVolumeClaims diese automatisch verwenden, wenn keine Speicherklasse angegeben ist. Sie müssen zwei Annotationen konfigurieren: eine für den clusterweiten Standard und eine speziell für die OpenShift Virtualisierung.

a. Legen Sie die clusterweite Standard-StorageClass-Annotation fest.

Stellen Sie sicher, dass nur eine StorageClass als Standard festgelegt ist. Wenn eine andere StorageClass bereits als Standard festgelegt ist, setzen Sie deren Annotation auf false.

Bearbeiten Sie die Annotation in der Konsole:

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

Über die CLI:

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

b. Legen Sie die OpenShift-Virtualisierung-spezifische Standardannotation fest.

OpenShift Virtualisierung verwendet eine spezifische Annotation, die Vorrang vor der allgemeinen `is-default-class` Annotation des Clusters hat. Wenn eine andere StorageClass bereits als Standard festgelegt ist, setzen Sie deren Annotation auf false.

Bearbeiten Sie die Annotation in der Konsole:

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

Über die CLI:

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

2. Legen Sie die Standard-Trident-VolumeSnapshotClass fest.

Legen Sie eine von Trident unterstützte VolumeSnapshotClass als Standard für den Cluster fest, um Snapshot-basierte Operationen für persistente Volumes zu ermöglichen. Dadurch wird sichergestellt, dass VolumeSnapshots automatisch den Trident CSI-Treiber verwenden, wenn keine Snapshot-Klasse angegeben ist, und OpenShift Virtualization Snapshots von Golden Images erstellen kann.

Stellen Sie sicher, dass nur eine VolumeSnapshotClass als Standard festgelegt ist. Wenn eine andere VolumeSnapshotClass bereits als Standard festgelegt ist, setzen Sie deren Annotation auf false.

Bearbeiten Sie die Annotation in der Konsole:

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

Über die CLI:

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

Voraussetzungen für die Bereitstellung von Red Hat OpenShift Virtualization mit ONTAP

Die Voraussetzungen und weiteren Details zur Bereitstellung der OpenShift Virtualisierung mit ONTAP Speichersystemen werden aufgeführt.

Voraussetzungen

- Ein Red Hat OpenShift-Cluster (neuer als Version 4.6), installiert auf einer Bare-Metal-Infrastruktur mit RHCOS-Worker-Knoten
- Führen Sie Machine Health Checks durch, um HA für VMs aufrechtzuerhalten
- Ein NetApp ONTAP -Cluster mit SVM, das mit dem richtigen Protokoll konfiguriert ist.
- Trident auf dem OpenShift-Cluster installiert
- Eine erstellte Trident Backend-Konfiguration
- Eine auf dem OpenShift-Cluster konfigurierte StorageClass mit Trident als Provisioner
- Eine Trident VolumeSnapshotClass, die auf dem OpenShift-Cluster mit Trident als Provisionierer konfiguriert ist
- Trident StorageClass ist als Standard für den OpenShift Cluster oder OpenShift Virtualisierung konfiguriert.

Die oben genannten Trident -Voraussetzungen finden Sie unter ["Trident -Installationsabschnitt"](#) für Details.

- Cluster-Admin-Zugriff auf den Red Hat OpenShift-Cluster
- Administratorzugriff auf NetApp ONTAP -Cluster
- Eine Admin-Workstation mit installierten und zu \$PATH hinzugefügten Tridentctl- und OC-Tools

Da die OpenShift Virtualisierung von einem auf dem OpenShift Cluster installierten Operator verwaltet wird,

entsteht zusätzlicher Aufwand für Arbeitsspeicher, CPU und Speicher, der bei der Planung der Hardwareanforderungen für den Cluster berücksichtigt werden muss. Siehe die Dokumentation ["hier,"](#) für weitere Details.

Optional kann auch eine Teilmenge der OpenShift-Cluster-Knoten für das Hosting der OpenShift Virtualization-Operatoren, Controller und VMs durch die Konfiguration von Knotenplatzierungsregeln festgelegt werden. Zur Konfiguration von Knotenplatzierungsregeln für OpenShift Virtualization wird auf die Dokumentation verwiesen ["hier,"](#).

Für den Speicher, der die OpenShift-Virtualisierung unterstützt, empfiehlt NetApp eine dedizierte StorageClass, die Speicher von einem bestimmten Trident Backend anfordert, das wiederum von einer dedizierten SVM unterstützt wird. Dadurch wird ein gewisses Maß an Mandantenfähigkeit hinsichtlich der Daten aufrechterhalten, die für VM-basierte Workloads auf dem OpenShift-Cluster bereitgestellt werden.

NetApp Trident CSI, Speicherprofile, Datenträgerzugriffsmodi für OpenShift Virtualisierung

In OpenShift Virtualization sind Bootquellen Festplattenabbilder, die zur Bereitstellung virtueller Maschinen (VMs) anhand von Vorlagen verwendet werden. Sie werden typischerweise als DataVolumes oder Persistent Volume Claims (PVCs) im Namespace `openshift-virtualization-os-images` gespeichert. Damit OpenShift Virtualization ordnungsgemäß funktioniert, sind vor der Installation von OpenShift Virtualization mit dem Operator zwei Speicherkonfigurationsaufgaben obligatorisch:

1. Sie müssen eine Standard-StorageClass für den Cluster konfigurieren. Andernfalls kann OpenShift Virtualization Boot-Quellabbilder nicht automatisch importieren.
2. Sie müssen Speicherprofile konfigurieren, wenn Ihr Speicheranbieter von CDI nicht erkannt wird. Ein Speicherprofil stellt empfohlene Speichereinstellungen basierend auf der zugehörigen StorageClass bereit. Eine der wichtigsten Funktionen von StorageProfile ist die Definition von Klonstrategien für eine StorageClass. Dadurch wird festgelegt, wie Volumes geklont werden (z. B. mithilfe von Snapshots, CSI-Klonen oder hostgestützten Kopien), wobei Leistung und Ressourcennutzung entsprechend den Fähigkeiten des zugrunde liegenden Speicheranbieters optimiert werden. Der Trident CSI driver wird von CDI erkannt, und ein Standard-Speicherprofil wird automatisch für jede im Cluster erstellte Trident StorageClass angelegt. Die Klonstrategie für dieses Standard-Speicherprofil ist auf Snapshot festgelegt, was die empfohlene Klonstrategie für den Trident CSI driver ist. Die VolumeSnapshotClass und StorageClass für Trident müssen erstellt und die StorageClass als Standard festgelegt werden, bevor OpenShift Virtualization installiert wird. Dadurch wird sichergestellt, dass die Boot-Quellabbilder als VolumeSnapshots im Namespace `openshift-virtualization-os-images` verfügbar sind, sodass VMs effizient und fehlerfrei bereitgestellt werden können.

Für bestimmte unterstützte Betriebssysteme (wie RHEL, Fedora oder CentOS) stellt OpenShift Virtualization automatisch Bootquellen bereit und aktualisiert diese. Diese werden heruntergeladen, wenn OpenShift Virtualization installiert wird, und sie werden automatisch verwaltet und auf die neueste Betriebssystemversion aktualisiert. Wenn ein Betriebssystem nicht automatisch bereitgestellt wird, muss ein Administrator ein benutzerdefiniertes Image manuell vorbereiten und anhängen. Bei Verwendung von NetApp Trident als CSI ergeben sich folgende zusätzliche Vorteile:

1. Die Boot-Quellabbilder werden als Snapshots im Namespace `openshift-virtualization-os-images` verfügbar sein. Wenn VMs bereitgestellt werden, wird das Boot-Abbild aus dem Snapshot geklont, was das Starten der VM deutlich beschleunigt.
2. Ein Speicherprofil wird für jede erstellte Trident StorageClass automatisch angelegt und ist automatisch mit der Klonstrategie auf Snapshot konfiguriert. Der Zugriffsmodus ist auf `ReadWriteMany` gesetzt. Eine manuelle Konfiguration ist nicht erforderlich.

Es ist äußerst wichtig, dass Sie eine Trident StorageClass und eine Trident VolumeSnapshotClass erstellen

und diese als Standard festlegen, bevor Sie die OpenShift Virtualization in Ihrem Cluster installieren. Dadurch wird sichergestellt, dass die Boot-Quellimages als VolumeSnapshots im Namespace openshift-virtualization-os-images verfügbar sind, sodass VMs effizient und fehlerfrei bereitgestellt werden können.

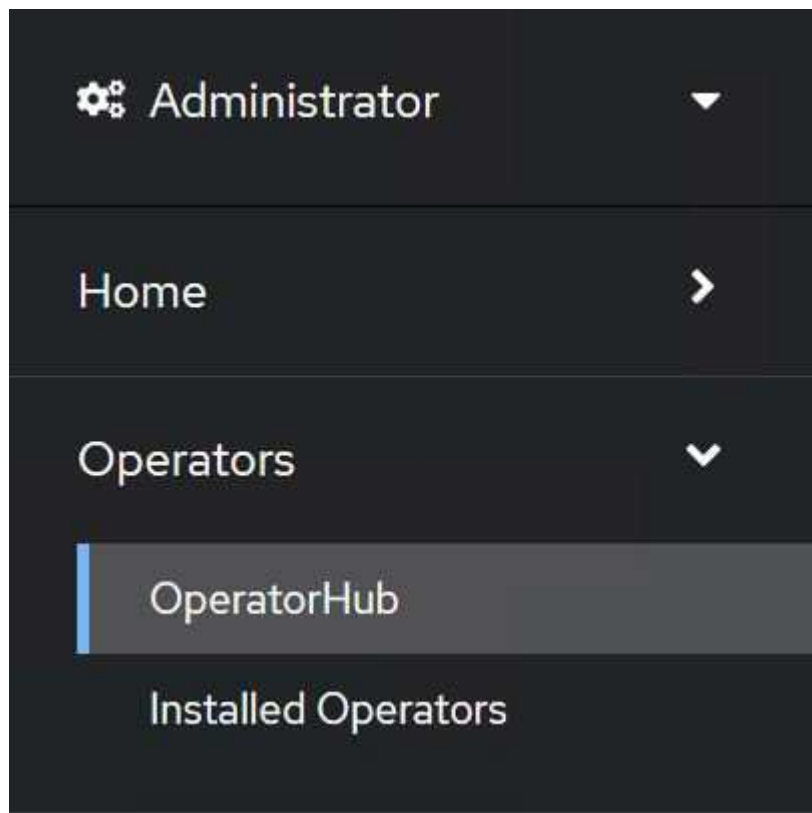
Ein weiterer Vorteil der Verwendung von Trident CSI: . In OpenShift Virtualization müssen die Boot-Disks der VMs über RWX-Zugriffsmodi verfügen, um eine Live-Migration der VMs durchzuführen. Trident unterstützt den Zugriffsmodus **ReadWriteMany** für Volumes, die mit NAS-Protokollen bereitgestellt werden, sowie mit SAN-Protokollen (wenn volumeMode auf Block und nicht auf FileSystem gesetzt ist). Für die Boot-Disk kann eines der beiden Protokolle verwendet werden. Das Storage-Profile-Objekt wird automatisch auf Basis der StorageClass-Einstellungen konfiguriert, und die Zugriffsmodi der Boot-Disk werden bei der Bereitstellung der VMs entsprechend festgelegt.

Nachdem Sie die Trident StorageClass als Standard für den Cluster festgelegt haben, können Sie zusätzliche StorageClasses für Trident mit unterschiedlichen Parametern erstellen, die von OpenShift Virtualization automatisch erkannt werden, wobei entsprechende Speicherprofile für sie erstellt werden. Anschließend kann die zu verwendende StorageClass für die Bootdisk und die Datendisks der VM in der VM-Definition angegeben werden.

Die Installation der OpenShift Virtualisierung auf dem OpenShift Cluster ist ein unkomplizierter Vorgang mithilfe des OpenShift Virtualization Operators, der im OpenShift OperatorHub verfügbar ist. Detaillierte Schritte zur Installation der OpenShift Virtualisierung sind in der Dokumentation zu finden ["hier,"](#).

Installieren Sie OpenShift Virtualization auf einem Red Hat OpenShift Bare-Metal-Cluster. Dieses Verfahren umfasst die Anmeldung mit Cluster-Administratorzugriff, die Navigation zum OperatorHub und die Installation des OpenShift Virtualization-Operators.

1. Melden Sie sich mit Cluster-Admin-Zugriff beim Red Hat OpenShift Bare-Metal-Cluster an.
2. Wählen Sie „Administrator“ aus der Dropdown-Liste „Perspektive“ aus.
3. Navigieren Sie zu Operatoren > OperatorHub und suchen Sie nach OpenShift Virtualization.



4. Wählen Sie die Kachel „OpenShift-Virtualisierung“ aus und klicken Sie auf „Installieren“.



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. Behalten Sie auf dem Bildschirm „Operator installieren“ alle Standardparameter bei und klicken Sie auf „Installieren“.

Update channel *

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

Installation mode *

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

Installed Namespace *

- ☒ Operator recommended Namespace: **PR** openshift-cnv

i Namespace creation

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

- ☐ Select a Namespace

Approval strategy *

- ☒ Automatic
- ☐ Manual

Install

Cancel

 OpenShift Virtualization
provided by Red Hat

Provided APIs

HC OpenShift
Virtualization
Deployment **Required**

Represents the deployment of
OpenShift Virtualization

6. Warten Sie, bis die Operatorinstallation abgeschlossen ist.



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. Klicken Sie nach der Installation des Operators auf „Hyperkonvergente erstellen“.



Installed operator - operand required

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

HC HyperConverged **Required**

Creates and maintains an OpenShift Virtualization Deployment

Create HyperConverged

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

8. Klicken Sie auf dem Bildschirm „Hyperkonvergente erstellen“ auf „Erstellen“ und akzeptieren Sie alle Standardparameter. Mit diesem Schritt wird die Installation von OpenShift Virtualization gestartet.

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/VMIs.

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. Nachdem alle Pods im OpenShift-CNV-Namespace in den Status „Ausgeführt“ gewechselt sind und sich der OpenShift-Virtualisierungsoperator im Status „Erfolgreich“ befindet, ist der Operator einsatzbereit. VMs können jetzt auf dem OpenShift-Cluster erstellt werden.

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

Erstellen einer VM mit ONTAP Speicher und Red Hat OpenShift Virtualization

Eine VM mit OpenShift Virtualization erstellen. Dieses Verfahren umfasst die Auswahl einer Betriebssystemvorlage, die Konfiguration von StorageClasses und die Anpassung der VM-Parameter an spezifische Anforderungen. Als Voraussetzung sollten das Trident

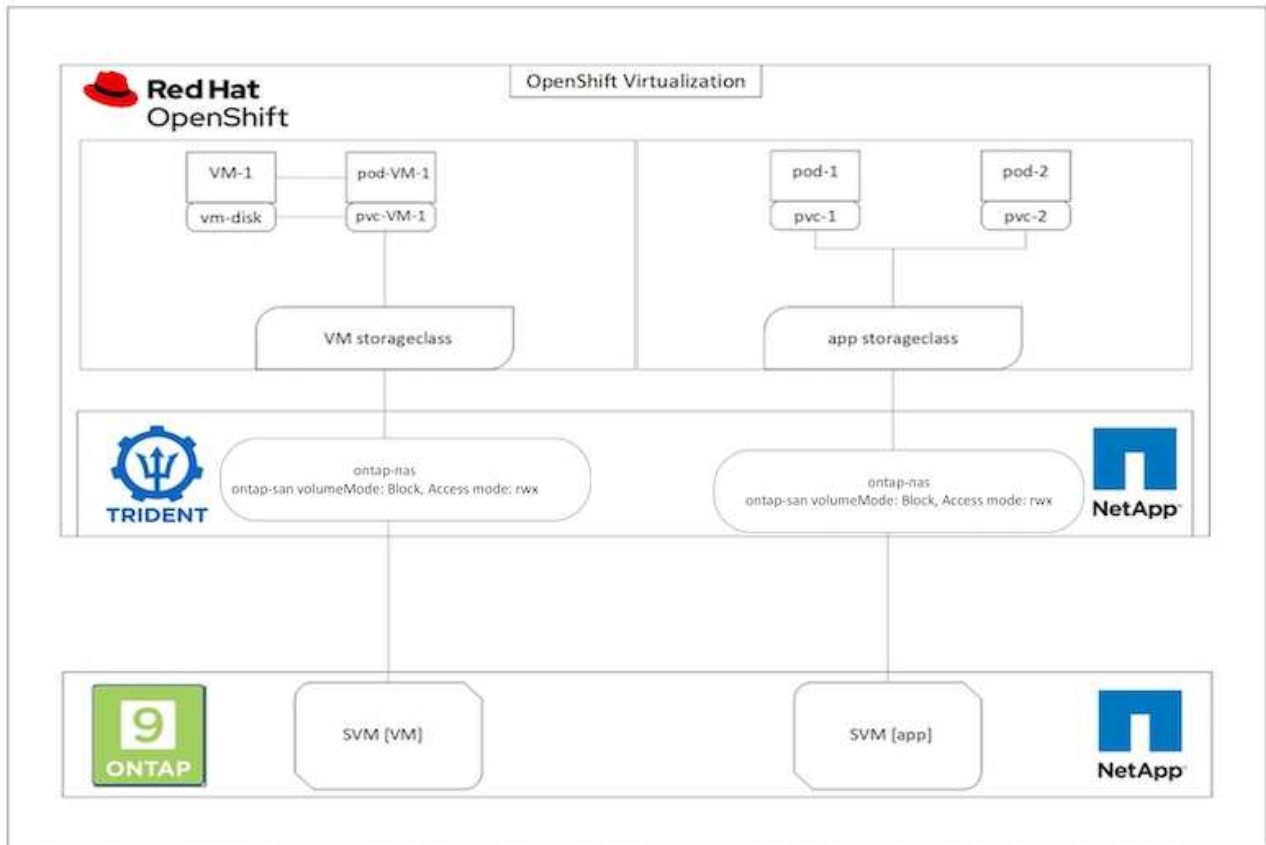
Backend, die StorageClass und die Volume Snapshot-Klassenobjekte bereits erstellt worden sein. Details sind in der "[Trident -Installationsabschnitt](#)" zu finden.

VM erstellen

VMs sind zustandsbehaftete Bereitstellungen, die Volumes zum Hosten des Betriebssystems und der Daten benötigen. Da die VMs bei CNV als Pods ausgeführt werden, werden sie durch PVs unterstützt, die über Trident auf NetApp ONTAP gehostet werden. Diese Volumes werden als Datenträger angeschlossen und speichern das gesamte Dateisystem einschließlich der Bootquelle der VM.



Trident StorageClass muss als Standard-StorageClass im Cluster festgelegt sein, damit die für die VM-Festplatten erstellten PVCs mit dem Trident-Backend bereitgestellt werden. Im Cluster können mehrere StorageClasses vorhanden sein, aber die Standard-StorageClass muss für die Bereitstellung des PVC für VM-Boot-Festplatten verwendet werden. Zusätzliche Festplatten können bei Bedarf andere StorageClasses verwenden.



Führen Sie die folgenden Schritte aus, um schnell eine virtuelle Maschine im OpenShift-Cluster zu erstellen:

1. Navigieren Sie zu Virtualisierung > Virtuelle Maschinen und klicken Sie auf Erstellen.
2. Wählen Sie „Aus Vorlage“ aus.
3. Wählen Sie das gewünschte Betriebssystem aus, für das die Bootquelle verfügbar ist.
4. Aktivieren Sie das Kontrollkästchen „Virtuelle Maschine nach der Erstellung starten“.
5. Klicken Sie auf „Virtuelle Maschine schnell erstellen“.

Die virtuelle Maschine wird erstellt, gestartet und erreicht den Status **Wird ausgeführt**. Es wird automatisch ein PVC und ein entsprechendes PV für die Boot-Disk mithilfe der Standard-StorageClass erstellt. Um die VM zukünftig live migrieren zu können, muss sichergestellt sein, dass die für die Festplatten verwendete StorageClass RWX-Volumes unterstützt. Dies ist eine Voraussetzung für die Live-Migration. `ontap-nas` und `ontap-san` (volumeMode block für iSCSI- und NVMe/TCP-Protokolle) können RWX-Zugriffsmodi für die mit den jeweiligen StorageClasses erstellten Volumes unterstützen.

Zur Konfiguration von `ontap-san` StorageClass auf dem Cluster siehe ["Abschnitt zur Konfiguration von StorageClass"](#).



Durch Klicken auf Quick create VirtualMachine wird die Standard-StorageClass verwendet, um die PVC und PV für die bootfähige Root-Disk der VM zu erstellen. Eine andere StorageClass für die Festplatte kann ausgewählt werden, indem Customize VirtualMachine > Customize VirtualMachine parameters > Disks ausgewählt und anschließend die Festplatte bearbeitet wird, um die gewünschte StorageClass zu verwenden.

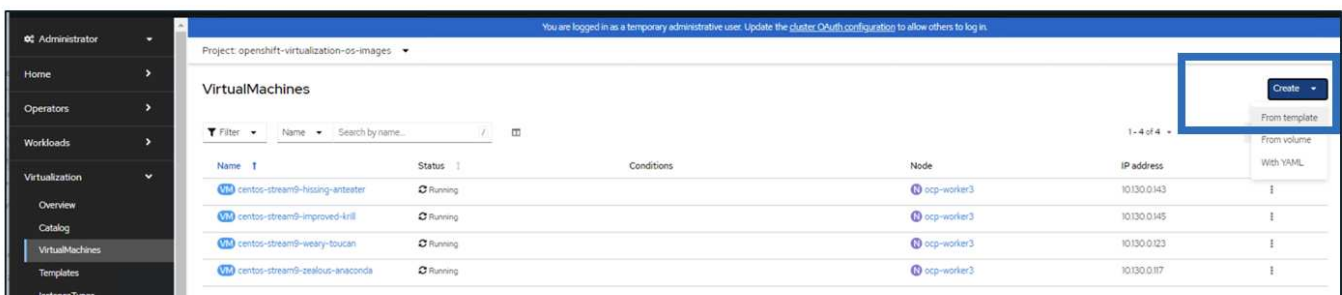
Normalerweise wird beim Bereitstellen der VM-Festplatten der Blockzugriffsmodus gegenüber Dateisystemen bevorzugt.

Um die Erstellung der virtuellen Maschine anzupassen, nachdem Sie die Betriebssystemvorlage ausgewählt haben, klicken Sie auf „Virtuelle Maschine anpassen“ statt auf „Schnell erstellen“.

1. Wenn für das ausgewählte Betriebssystem eine Startquelle konfiguriert ist, können Sie auf **Parameter der virtuellen Maschine anpassen** klicken.
2. Wenn für das ausgewählte Betriebssystem keine Startquelle konfiguriert ist, müssen Sie diese konfigurieren. Details zu den dargestellten Verfahren finden Sie in der ["Dokumentation"](#).
3. Nachdem Sie die Startdiskette konfiguriert haben, können Sie auf **Parameter der virtuellen Maschine anpassen** klicken.
4. Sie können die VM über die Registerkarten auf dieser Seite anpassen. Klicken Sie beispielsweise auf die Registerkarte **Datenträger** und dann auf **Datenträger hinzufügen**, um der VM einen weiteren Datenträger hinzuzufügen.
5. Klicken Sie auf „Virtuelle Maschine erstellen“, um die virtuelle Maschine zu erstellen. Dadurch wird im Hintergrund ein entsprechender Pod gestartet.



Wenn eine Boot-Quelle für eine Vorlage oder ein Betriebssystem von einer URL oder einer Registrierung konfiguriert wird, erstellt sie einen PVC in der `openshift-virtualization-os-images` Projekt und lädt das KVM-Gastimage auf das PVC herunter. Sie müssen sicherstellen, dass die PVC-Vorlagen über genügend Speicherplatz verfügen, um das KVM-Gastimage für das entsprechende Betriebssystem aufzunehmen. Diese PVCs werden dann geklont und als Rootdisk an virtuelle Maschinen angehängt, wenn sie mithilfe der entsprechenden Vorlagen in einem beliebigen Projekt erstellt werden.



Create new VirtualMachine

Select an option to create a VirtualMachine from.

Template project

All projects

All items

Default templates

Filter by keyword...

13 items

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 (auto import)

Workload Desktop

CPU 1

Memory 4 GiB

Microsoft Windows 11 VM

windows11-desktop-medium

Project openshift

Boot source PVC (auto import)

Workload Desktop

CPU 2

Memory 4 GiB

Microsoft Windows Server 2012 R2 VM

windows2k12r2-server-medium

Project openshift

Boot source PVC (auto import)

Workload Server

CPU 1

Memory 4 GiB

VirtualMachine details

VM centos-stream9-zealous-anaconda Running

VAML Actions

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

Disks

Add disk

Filter Search by name... Mount Windows drivers disk

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

Videodemonstration

Das folgende Video zeigt eine Demonstration der Erstellung einer VM in OpenShift Virtualization mithilfe von iSCSI-Speicher.

[Erstellen Sie eine VM in OpenShift Virtualization mithilfe von Block Storage](#)

Erstellen einer VM mit Red Hat OpenShift Virtualization unter Verwendung von ONTAP Storage durch alternative Methoden

Neben der Methode zur Erstellung einer VM über die OpenShift Virtualization UI gibt es weitere Methoden, um VMs auf OpenShift Virtualization zu erstellen. Dieser Abschnitt bietet einen Überblick über einige dieser Methoden.

Erstellen einer VM aus einer YAML-Datei mit Red Hat OpenShift Virtualization

Eine VM kann auch durch Anwenden einer YAML-Datei mit dem `oc apply` Befehl erstellt werden. Diese Methode ist für Benutzer nützlich, die die CLI bevorzugen oder die VM-Erstellung mithilfe von Skripten automatisieren möchten. Zur Erstellung einer VM mit dieser Methode kann eine YAML-Datei erstellt werden, die die VM und die zugehörigen Ressourcen wie PVCs für die VM-Festplatten definiert. Anschließend kann die YAML-Datei mit dem `oc apply -f <yaml-file>` Befehl angewendet werden, um die VM im OpenShift Cluster zu erstellen.

Beispiel anzeigen

```
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

```

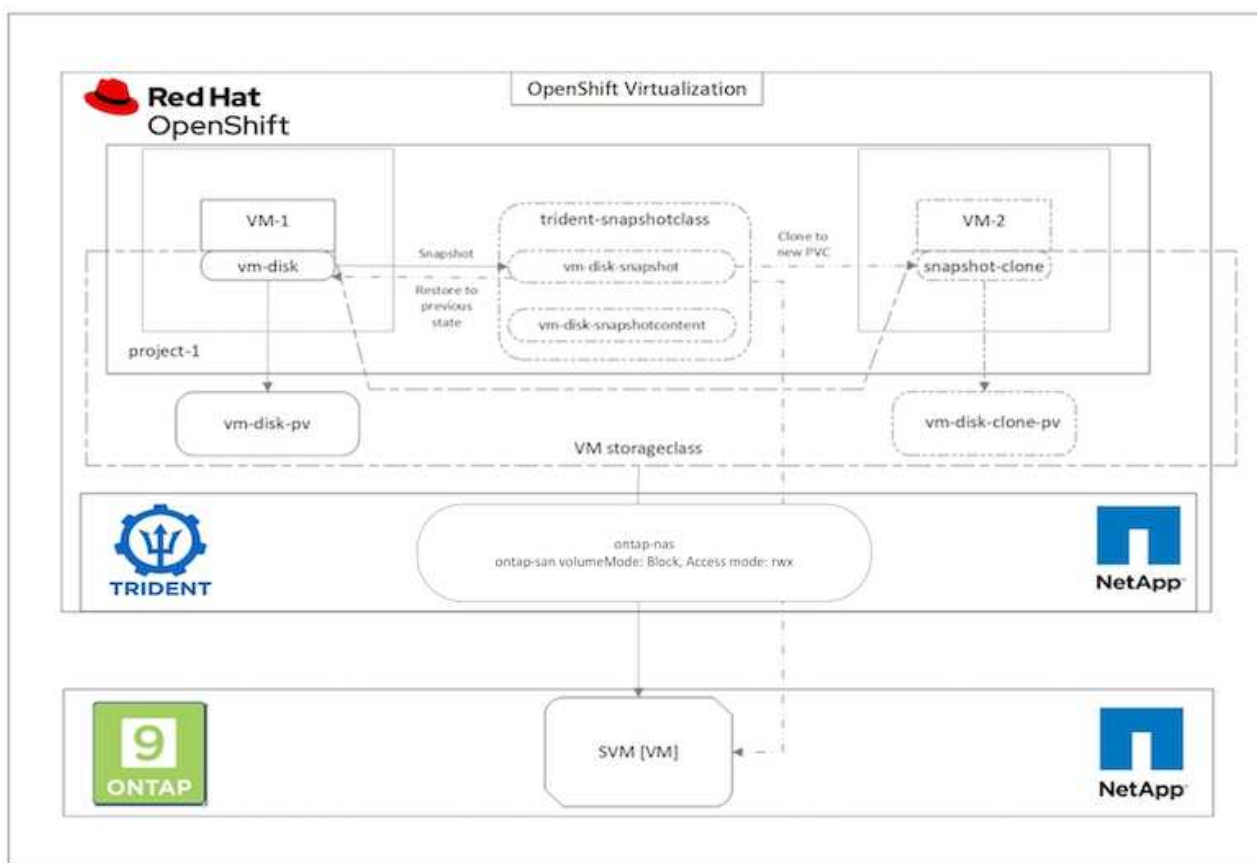
Erstellen Sie eine VM aus einer Snapshot-Kopie mit Red Hat OpenShift Virtualization

Sie können mit OpenShift Virtualization eine VM aus einem Snapshot erstellen. Dieses Verfahren umfasst das Erstellen einer VolumeSnapshotClass, das Erstellen eines Snapshots des Persistent Volume Claim (PVC) der VM, das Wiederherstellen des Snapshots in einem neuen PVC und das Bereitstellen einer neuen VM, die den wiederhergestellten PVC als Root-Disk verwendet.

Mit Trident und Red Hat OpenShift können Benutzer einen Snapshot eines persistenten Volumes auf den von ihm bereitgestellten Speicherklassen erstellen. Mit dieser Funktion können Benutzer eine zeitpunktbezogene Kopie eines Volumes erstellen und damit ein neues Volume erstellen oder dasselbe Volume in einen früheren Zustand zurückversetzen. Dies ermöglicht oder unterstützt eine Vielzahl von Anwendungsfällen, vom Rollback über Klone bis hin zur Datenwiederherstellung.

Für Snapshot-Operationen in OpenShift müssen die Ressourcen VolumeSnapshotClass, VolumeSnapshot und VolumeSnapshotContent definiert werden.

- Ein VolumeSnapshotContent ist der tatsächliche Snapshot, der von einem Volume im Cluster erstellt wurde. Es handelt sich um eine clusterweite Ressource analog zu PersistentVolume für die Speicherung.
- Ein VolumeSnapshot ist eine Anforderung zum Erstellen des Snapshots eines Volumes. Es ist analog zu einem PersistentVolumeClaim.
- Mit VolumeSnapshotClass kann der Administrator verschiedene Attribute für einen VolumeSnapshot angeben. Sie können damit unterschiedliche Attribute für unterschiedliche Snapshots festlegen, die vom selben Datenträger erstellt wurden.



Beispiel anzeigen

Voraussetzung

Es muss eine vorhandene VM mit einem daran angehängten PVC vorhanden sein, das auf einer Trident Storage Class bereitgestellt wurde, und die VM muss sich im ausgeführten Zustand befinden, bevor ein Snapshot des PVC erstellt werden kann. Es muss eine VolumeSnapshotClass erstellt worden sein, die anschließend zur Erstellung eines VolumeSnapshot verwendet werden kann. Dies wäre als Teil der Voraussetzungen für die Installation von OpenShift Virtualization erstellt worden. Einzelheiten können der ["Schritt 3 zum Erstellen einer VolumeSnapshotClass Konfiguration"](#) entnommen werden.

Um einen Snapshot einer VM zu erstellen, sind die folgenden Schritte erforderlich:

1. Identifizieren Sie den PVC, der an die Quell-VM angeschlossen ist, und erstellen Sie dann einen Snapshot dieses PVC. Navigieren Sie zu `Storage > VolumeSnapshots` und klicken Sie auf „VolumeSnapshots erstellen“.
2. Wählen Sie das PVC aus, für das Sie den Snapshot erstellen möchten, geben Sie den Namen des Snapshots ein oder akzeptieren Sie den Standardnamen und wählen Sie die entsprechende VolumeSnapshotClass aus. Klicken Sie dann auf Erstellen.
3. Dadurch wird der Snapshot des PVC zu diesem Zeitpunkt erstellt.

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

Eine neue VM aus dem Snapshot erstellen

1. Stellen Sie zunächst den Snapshot in einem neuen PVC wieder her. Navigieren Sie zu `Speicher > VolumeSnapshots`, klicken Sie auf die Auslassungspunkte neben dem Snapshot, den Sie wiederherstellen möchten, und klicken Sie auf Als neues PVC wiederherstellen.
2. Geben Sie die Details des neuen PVC ein und klicken Sie auf Wiederherstellen. Dadurch entsteht ein neues 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. Erstellen Sie als Nächstes eine neue VM aus diesem PVC. Navigieren Sie zu Virtualisierung > Virtuelle Maschinen und klicken Sie auf Erstellen > Mit YAML.
2. Geben Sie im Abschnitt „spec > template > spec > volumes“ den neuen PVC an, der aus Snapshot und nicht aus der Containerfestplatte erstellt wurde. Geben Sie alle weiteren Details zur neuen VM entsprechend Ihren Anforderungen an.

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

1. Klicken Sie auf „Erstellen“, um die neue VM zu erstellen.
2. Nachdem die VM erfolgreich erstellt wurde, greifen Sie darauf zu und überprüfen Sie, ob die neue VM denselben Status hat wie die VM, deren PVC zum Zeitpunkt der Erstellung des Snapshots zum Erstellen des Snapshots verwendet wurde.

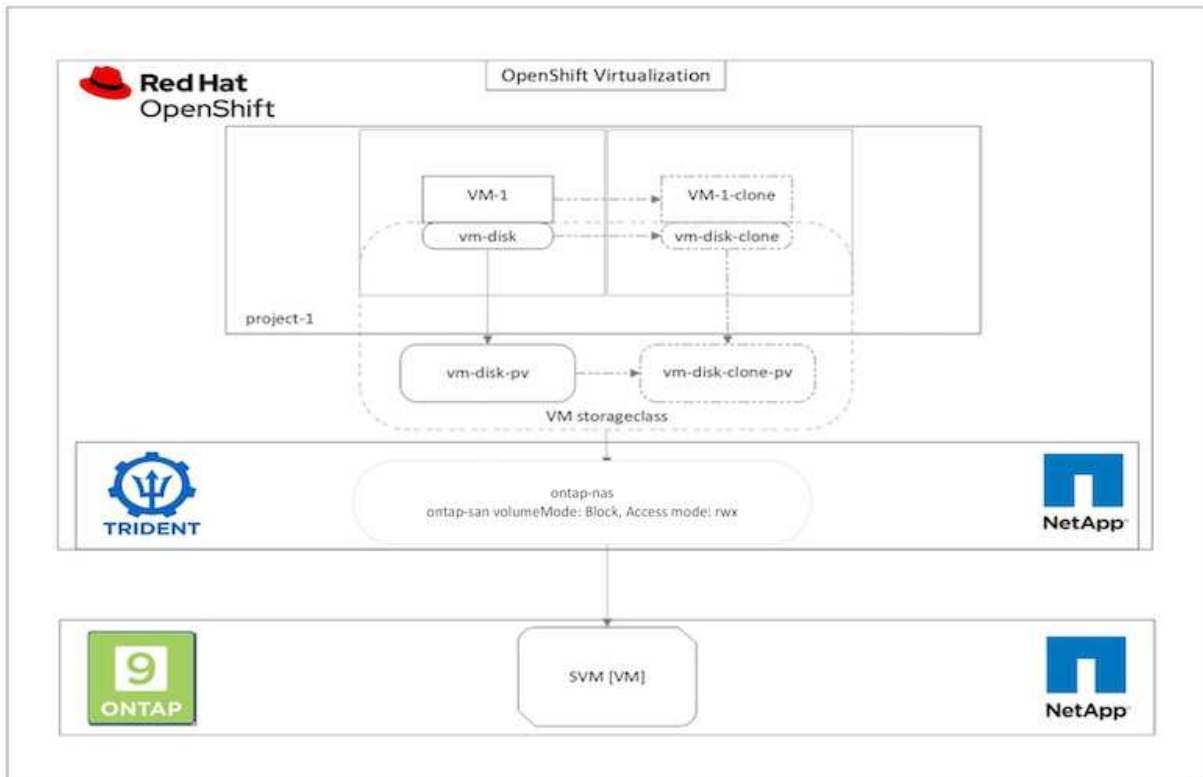
Klonen einer VM in OpenShift Virtualisierung mit Trident

Dieses Verfahren beinhaltet die Nutzung der Trident CSI Volume-Klonfunktion, wodurch eine neue VM erstellt werden kann, indem die Quell-VM heruntergefahren oder weiter betrieben wird.

Beispiel anzeigen

VM-Klonen

Das Klonen einer vorhandenen VM in OpenShift wird mit Unterstützung der Volume-CSI-Klonfunktion von Trident erreicht. Durch das Klonen von CSI-Volumes kann ein neues PVC erstellt werden, indem ein vorhandenes PVC als Datenquelle verwendet wird, indem dessen PV dupliziert wird. Nachdem der neue PVC erstellt wurde, funktioniert er als separate Einheit und ohne jegliche Verbindung zum Quell-PVC oder Abhängigkeit davon.



Beim Klonen von CSI-Volumes sind bestimmte Einschränkungen zu beachten:

1. Quell-PVC und Ziel-PVC müssen sich im selben Projekt befinden.
2. Das Klonen wird innerhalb derselben Speicherkategorie unterstützt.
3. Das Klonen kann nur durchgeführt werden, wenn Quell- und Zielvolumes dieselbe VolumeMode-Einstellung verwenden. Beispielsweise kann ein Blockvolume nur in ein anderes Blockvolume geklont werden.

VMs in einem OpenShift-Cluster können auf zwei Arten geklont werden:

1. Durch Herunterfahren der Quell-VM
2. Indem die Quell-VM aktiv bleibt

Durch das Herunterfahren der Quell-VM

Das Klonen einer vorhandenen VM durch Herunterfahren der VM ist eine native OpenShift-Funktion, die mit Unterstützung von Trident implementiert wird. Führen Sie die folgenden Schritte aus, um eine VM zu

klonen.

1. Navigieren Sie zu Workloads > Virtualisierung > Virtuelle Maschinen und klicken Sie auf die Auslassungspunkte neben der virtuellen Maschine, die Sie klonen möchten.
2. Klicken Sie auf „Virtuelle Maschine klonen“ und geben Sie die Details für die neue VM ein.

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. Klicken Sie auf „Virtuelle Maschine klonen“. Dadurch wird die Quell-VM heruntergefahren und die Erstellung der Klon-VM eingeleitet.

4. Nachdem dieser Schritt abgeschlossen ist, können Sie auf den Inhalt der geklonten VM zugreifen und ihn überprüfen.

Indem die Quell-VM aktiv bleibt

Eine vorhandene VM kann auch geklont werden, indem der vorhandene PVC der Quell-VM geklont und dann mit dem geklonten PVC eine neue VM erstellt wird. Bei dieser Methode müssen Sie die Quell-VM nicht herunterfahren. Führen Sie die folgenden Schritte aus, um eine VM zu klonen, ohne sie herunterzufahren.

1. Navigieren Sie zu Speicher > PersistentVolumeClaims und klicken Sie auf die Auslassungspunkte neben dem PVC, das an die Quell-VM angehängt ist.
2. Klicken Sie auf „PVC klonen“ und geben Sie die Details für das neue PVC an.

Clone

Name *

Access Mode *

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

Size *

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. Klicken Sie dann auf „Klonen“. Dadurch wird ein PVC für die neue VM erstellt.
4. Navigieren Sie zu Workloads > Virtualisierung > Virtuelle Maschinen und klicken Sie auf Erstellen > Mit YAML.
5. Fügen Sie im Abschnitt „spec > template > spec > volumes“ das geklonte PVC anstelle der

Containerfestplatte an. Geben Sie alle weiteren Details zur neuen VM entsprechend Ihren Anforderungen an.

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

1. Klicken Sie auf „Erstellen“, um die neue VM zu erstellen.
2. Nachdem die VM erfolgreich erstellt wurde, greifen Sie darauf zu und überprüfen Sie, ob die neue VM ein Klon der Quell-VM ist.

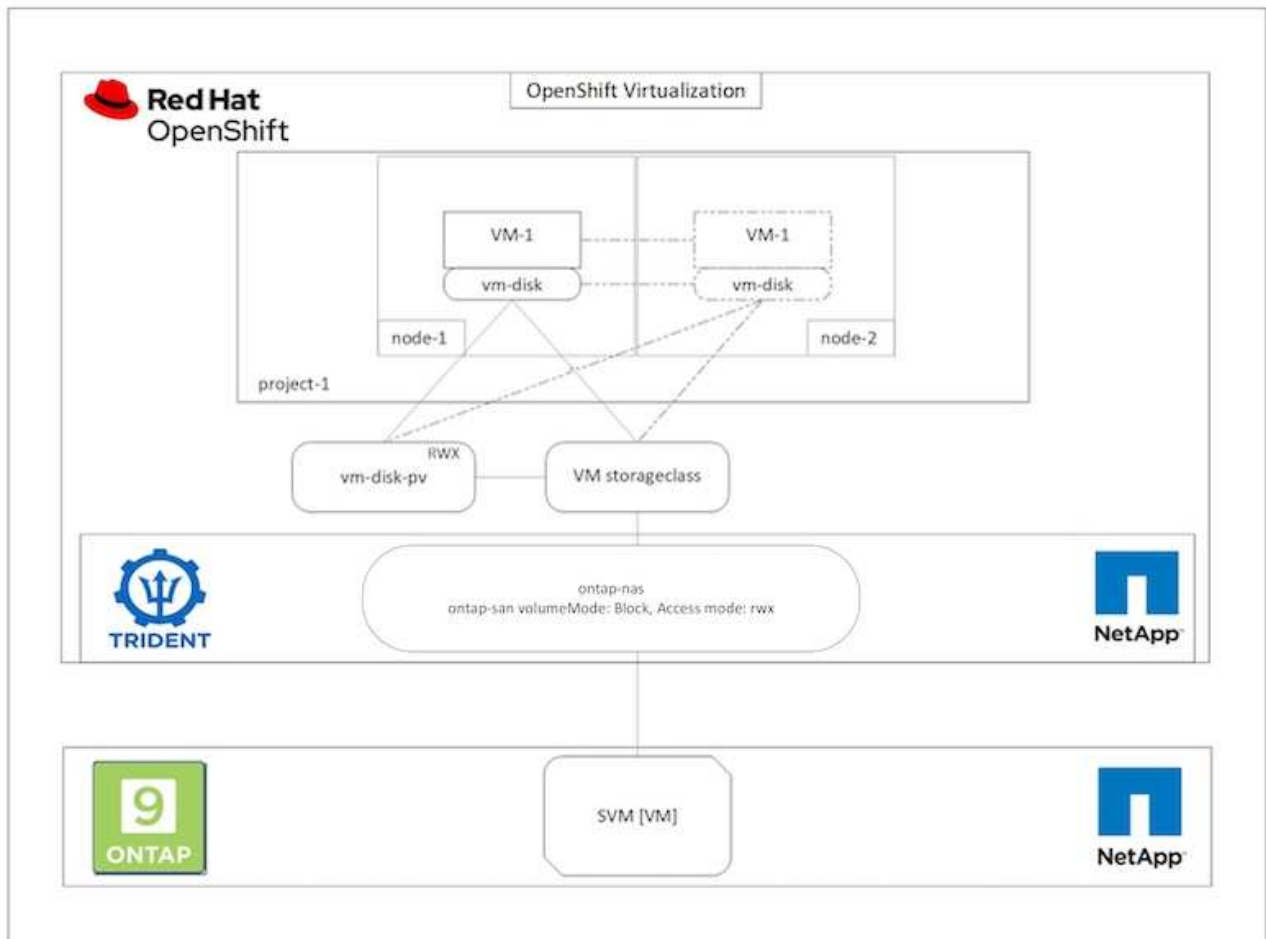
Migrieren Sie eine VM zwischen zwei Knoten in einem Red Hat OpenShift-Cluster

Migrieren Sie eine VM in OpenShift Virtualization ohne Ausfallzeiten zwischen zwei Knoten im Cluster. Dieses Verfahren umfasst die Bestätigung, dass die Datenträger RWX-kompatible Speicherklassen verwenden, das Initiieren der Migration und das Überwachen des Fortschritts.

VM Live Migration

Bei der Livemigration handelt es sich um einen Prozess, bei dem eine VM-Instanz ohne Ausfallzeiten von einem Knoten zu einem anderen in einem OpenShift-Cluster migriert wird. Damit die Livemigration in einem OpenShift-Cluster funktioniert, müssen VMs an PVCs mit dem gemeinsamen ReadWriteMany-Zugriffsmodus gebunden sein. Mit `ontap-nas`-Treibern konfigurierte Trident -Backends unterstützen den RWX-Zugriffsmodus für die Dateisystemprotokolle NFS und SMB. Weitere Informationen finden Sie in der Dokumentation ["hier,"](#) . Mit `ontap-san`-Treibern konfigurierte Trident -Backends unterstützen den RWX-Zugriffsmodus für den Block-VolumeMode für iSCSI- und NVMe/TCP-Protokolle. Weitere Informationen finden Sie in der Dokumentation ["hier,"](#) .

Damit die Livemigration erfolgreich ist, müssen die VMs daher mit Festplatten (Boot-Festplatten und zusätzlichen Hot-Plug-Festplatten) mit PVCs unter Verwendung der Speicherklassen `ontap-nas` oder `ontap-san` (VolumeMode: Block) ausgestattet werden. Wenn die PVCs erstellt werden, erstellt Trident ONTAP Volumes in einer SVM, die NFS- oder iSCSI-fähig ist.

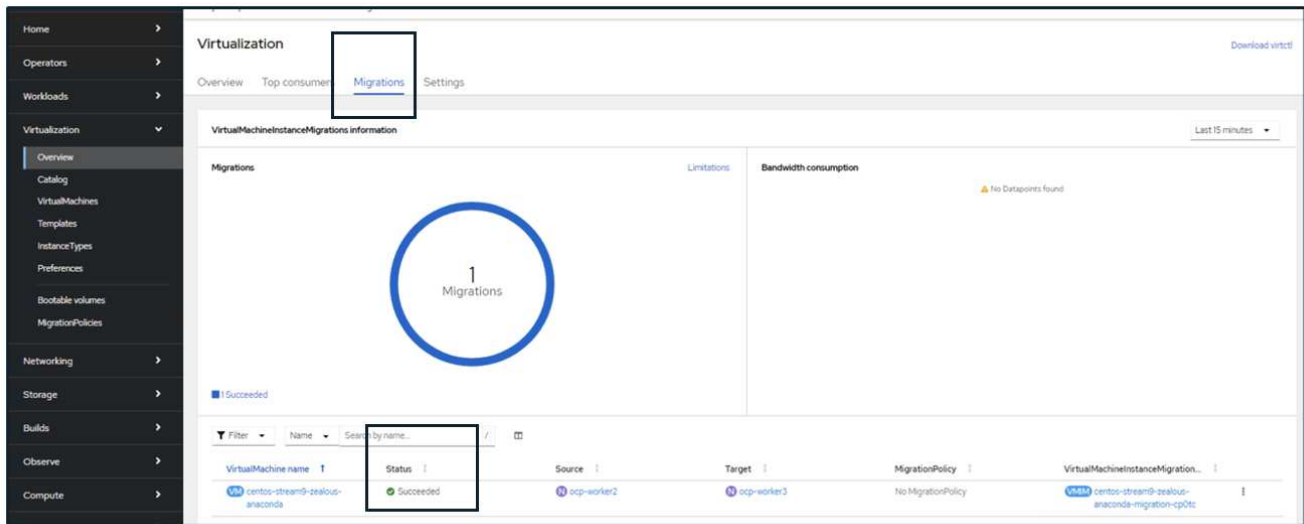
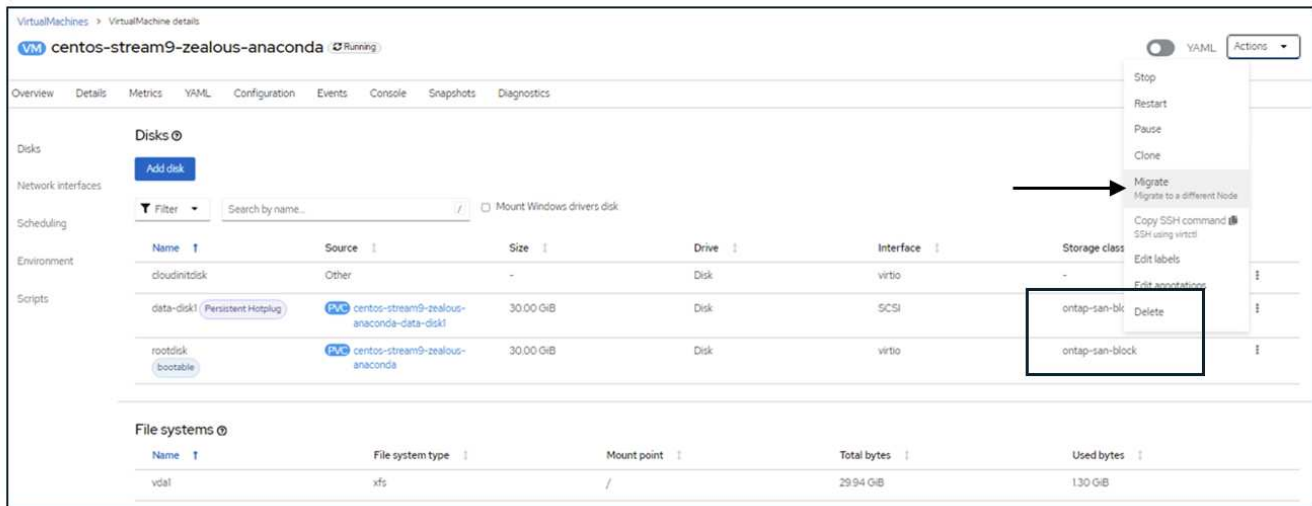


Um eine Livemigration einer VM durchzuführen, die zuvor erstellt wurde und sich im Status „Ausgeführt“ befindet, führen Sie die folgenden Schritte aus:

1. Wählen Sie die VM aus, die Sie live migrieren möchten.
2. Klicken Sie auf die Registerkarte **Konfiguration**.
3. Stellen Sie sicher, dass alle Datenträger der VM mit Speicherklassen erstellt werden, die den RWX-Zugriffsmodus unterstützen.
4. Klicken Sie in der rechten Ecke auf **Aktionen** und wählen Sie dann **Migrieren** aus.
5. Um den Fortschritt der Migration anzuzeigen, gehen Sie im Menü auf der linken Seite zu Virtualisierung > Übersicht und klicken Sie dann auf die Registerkarte **Migrationen**. Die Migration der VM wechselt von **Ausstehend** zu **Geplant** zu **Erfolgreich**



Eine VM-Instanz in einem OpenShift-Cluster wird automatisch auf einen anderen Knoten migriert, wenn der ursprüngliche Knoten in den Wartungsmodus versetzt wird und die `evictionStrategy` auf `LiveMigrate` gesetzt ist.



Speichermigration von VMs zwischen Speicherklassen in OpenShift Virtualisierung

In OpenShift Virtualization kann die Festplatte einer virtuellen Maschine (VM) direkt über die Webkonsole in eine andere Speicherklasse migriert werden, ohne die VM zu löschen. Mit diesem Vorgang lässt sich entweder der gesamte VM-Speicher oder bestimmte Volumes verschieben.

Die folgenden Schritte zeigen, wie die Speicherklasse einer VM migriert werden kann:

1. Stellen Sie sicher, dass die Zielspeicherklasse korrekt konfiguriert und in Ihrem OpenShift-Cluster verfügbar ist. Dies schließt ein, dass das erforderliche Speicher-Backend (z. B. NetApp ONTAP) eingerichtet und zugänglich ist.

Beispiel anzeigen

```
[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. In der Webkonsole der OpenShift Container Platform zur Virtualisierungsperspektive navigieren und die VM auswählen, die migriert werden soll.

Beispiel anzeigen

VirtualMachines > VirtualMachine details

VM rhel9-sc-migration Running

OverviewMetricsYAMLConfigurationEventsConsoleSnapshotsDiagnostics

Details

Namerhel9-sc-migration

StatusRunning

CreatedMay 14, 2020, 1:31 PM (17 minutes ago)

Operating systemRed Hat Enterprise Linux 9.7 (Plow)

CPU | Memory1 CPU | 2 GiB Memory

Time zoneEDT

Templatelabel rhel9-server-small

Hostnamerhel9-sc-migration

VNC console

Open web console

Alerts (0)

General

NamespaceNS test-sc-migration

Node00-50-56-b5-bd-fa

VirtualMachineInstanceVM rhel9-sc-migration

Podvirt-launcher-rhel9-sc-migration-5jkdw

OwnerNo owner

Snapshots (0)Take snapshot

VirtualMachines > VirtualMachine details

VM rhel9-sc-migration Running

OverviewMetricsYAMLConfigurationEventsConsoleSnapshotsDiagnostics

Details

Storage

Add disk

Network

Filter

Search by name...

☐ Mount Windows drivers disk

Scheduling

SSH

Initial run

Metadata

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
bootable					

```
[root@00-50-56-b5-2b-9b ~]# oc get vm,pvc -n test-sc-migration
NAME                                AGE    STATUS    READY
virtualmachine.kubevirt.io/rhel9-sc-migration 4m19s Running True

NAME                                STATUS    VOLUME                                CAPACITY    ACCESS MODES    STORAGECLASS    VOLUMEATTRIBUTESCLASS    AGE
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc Bound     pvc-bb54342f-c01f-453e-864d-e8f72e710a2f 53Gi        RWX            sc-nas          <unset>                  4m19s
persistentvolumeclaim/rhel9-migration Bound     pvc-46a34983-942f-4ddc-8e6d-974008f05d09 34144990004 RWX            sc-nas          <unset>                  4m19s
[root@00-50-56-b5-2b-9b ~]#
```

- Klicken Sie auf Actions→Migrate→Migrate Storage. Die zu migrierenden Volumes und die Zielspeicherklasse auswählen, dann auf Migrate klicken. Der Migrationsprozess startet und der Fortschritt kann auf der Registerkarte Events der VM überwacht werden. Nach Abschluss der Migration verwenden die Festplatten der VM die neue Speicherklasse.

Beispiel anzeigen

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 VM 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. Nach Abschluss der Migration kann überprüft werden, ob die Festplatten der VM nun die neue Speicherklasse verwenden, indem die Details der VM in der Webkonsole oder mit der OpenShift CLI geprüft werden.

Beispiel anzeigen

```
[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-5jdkw    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-bb5d1342f-c01f-453e-86ad-abf72e710a2f    536i    Rwx    sc-nas    <unset>    26m
persistentvolumeclaim/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb    Bound    pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b    60322815673    Rwx    sc-nas-economy    <unset>    3m28s
persistentvolumeclaim/rhel9-migration                                Bound    pvc-46a34903-942f-4ddc-8e6d-974008f05d09    34144990004    Rwx    sc-nas    <unset>    26m
[root@00-50-56-b5-2b-9b ~]#

```

```

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

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.