



스토리지 및 가상화 시작하기

NetApp virtualization solutions

NetApp
July 21, 2026

목차

스토리지 및 가상화 시작하기	1
Red Hat OpenShift와 NetApp Trident 통합에 대해 알아보세요	1
NetApp Trident	1
Trident Protect	1
NetApp ONTAP 하드웨어 모델 및 지원 프로토콜	2
ONTAP 스토리지용 Trident 드라이버	2
Red Hat OpenShift 클러스터에 Trident를 설치하고 스토리지 객체를 생성하세요	4
1단계: Trident 설치	4
2단계: 사용 환경에 맞는 스토리지 백엔드 및 StorageClass 구성 파일을 준비합니다	9
3단계: VolumeSnapshotClass 구성 파일 생성	29
4단계: 구성 파일을 클러스터에 적용합니다	29
5단계: 기본 Trident 스토리지 및 스냅샷 클래스 설정	31
ONTAP 사용하여 Red Hat OpenShift Virtualization을 배포하기 위한 요구 사항	32
필수 조건	32
NetApp Trident CSI, 스토리지 프로파일, OpenShift 가상화를 위한 볼륨 액세스 모드	33
Red Hat OpenShift Virtualization을 사용하여 ONTAP 스토리지로 VM 생성	37
VM 생성	38
비디오 데모	41
Red Hat OpenShift Virtualization에서 대체 방법을 통해 ONTAP 스토리지를 사용하여 VM 생성	41
Red Hat OpenShift Virtualization을 사용하여 YAML 파일에서 VM 생성	41
Red Hat OpenShift Virtualization을 사용하여 스냅샷 복사본에서 VM 만들기	43
OpenShift Virtualization에서 Trident를 사용하여 VM 복제	47
Red Hat OpenShift 클러스터의 두 노드 간에 VM 마이그레이션	51
VM 라이브 마이그레이션	51
OpenShift Virtualization에서 스토리지 클래스 간 VM 스토리지 마이그레이션	53

스토리지 및 가상화 시작하기

Red Hat OpenShift와 NetApp Trident 통합에 대해 알아보세요

컨테이너화된 애플리케이션에서 스토리지는 매우 중요하며, 데이터의 지속적인 가용성을 보장하고 동적이고 종종 일시적인 환경에서도 효율적으로 관리할 수 있도록 해줍니다. 최신 컨테이너용 스토리지 솔루션은 애플리케이션과 함께 원활하게 확장되도록 설계되어 애플리케이션 요구 사항에 따라 스토리지 리소스를 동적으로 할당할 수 있는 유연성을 제공하며, 성장과 다양한 워크로드를 모두 지원합니다. OpenShift Virtualization을 사용하면 VM이 컨테이너로 실행되므로 OpenShift 클러스터의 다른 애플리케이션과 동일한 스토리지를 사용합니다.

NetApp ONTAP은 NetApp의 엔터프라이즈급 스토리지 솔루션의 핵심 역할을 하는 강력하고 확장성이 뛰어난 데이터 관리 소프트웨어입니다. 온프레미스, 클라우드, 하이브리드 환경을 포함한 다양한 스토리지 환경과 원활하게 통합되는 유니파이드 스토리지 아키텍처를 제공합니다. ONTAP은 NFS, CIFS/SMB, iSCSI, FC, NVMe over TCP 등 다양한 프로토콜을 지원하여 여러 애플리케이션 및 워크로드와의 원활한 통합을 가능하게 합니다. SAN, NAS, 객체 스토리지를 단일 플랫폼으로 통합하여 관리를 간소화하고 비용을 절감할 수 있습니다. NetApp ONTAP은 온프레미스 환경뿐 아니라 클라우드 관리형 서비스로도 제공됩니다.

NetApp Trident

Trident는 NetApp이 개발한 오픈 소스 스토리지 오케스트레이터로, 컨테이너 환경에서 ONTAP 스토리지를 쉽게 사용할 수 있도록 설계되었습니다. 이를 통해 컨테이너화된 애플리케이션을 위한 영구 스토리지의 동적 프로비저닝 및 관리가 가능합니다.

Trident의 주요 기능

CSI 규격 준수 최신 Kubernetes 스토리지 기능 및 표준과의 호환성을 보장합니다. 선언적 구성 사용자가 YAML 파일에 스토리지 요구 사항을 정의할 수 있으며, 이는 Trident에 의해 자동으로 관리됩니다. 드라이버 Trident는 ONTAP에서 지원하는 NFS, CIFS/SMB, iSCSI, FC 및 NVMe/TCP 프로토콜용 드라이버를 지원합니다. 하이브리드 및 멀티 클라우드 지원 Trident는 온프레미스 ONTAP 스토리지와 하이퍼스케일러의 관리형 스토리지 서비스(AWS의 FSx for NetApp ONTAP, Azure의 Azure NetApp Files, Google Cloud의 Google Cloud NetApp Volumes)용 드라이버를 지원합니다. 고급 데이터 관리 ONTAP의 스냅샷 및 클론 기능을 활용하여 효율적인 데이터 보호와 테스트 및 개발 환경의 신속한 프로비저닝을 지원합니다.

Trident에 대한 자세한 내용은 NetApp Trident ["여기에 문서가 있습니다."](#)를 참조하십시오.

Trident Protect

Trident Protect는 컨테이너화된 환경을 위한 향상된 데이터 보호 및 보안 기능을 제공하는 Trident의 고급 기능입니다. 잠재적인 위험 및 데이터 손실 시나리오로부터 중요 데이터를 안전하게 보호하도록 설계되었습니다. 모든 Trident 사용자에게 무료로 제공되며, Trident가 설치된 모든 OpenShift 클러스터에서 간편하게 활성화할 수 있습니다.

Trident Protect의 주요 기능

자동 백업 Trident Protect는 컨테이너 애플리케이션 및 VM의 자동화된 정책 기반 백업을 지원하여 수동 개입 없이 정기적으로 백업이 이루어지도록 합니다.

스냅샷 관리 ONTAP의 스냅샷 기능을 활용하여 컨테이너 애플리케이션 및 VM의 특정 시점 복사본을 자주

생성함으로써 안정적인 복구 메커니즘을 제공합니다.

데이터 암호화 데이터가 저장될 때와 전송될 때 모두 암호화되어 엄격한 보안 요구 사항을 충족하고 중요한 정보를 보호합니다.

규정 준수 및 감사 조직이 규정 준수 요건을 충족하고 데이터 보호 관행에 대한 정기적인 감사를 수행하는 데 도움이 되는 도구와 기능을 제공합니다.

재해 복구 ONTAP의 복제 기능과 통합되어 강력한 재해 복구 솔루션을 제공함으로써 재앙적인 사태 발생 시에도 비즈니스 연속성을 보장합니다.

Trident Protect에 대한 자세한 내용은 NetApp Trident Protect ["여기에 문서가 있습니다."](#)를 참조하십시오.

NetApp ONTAP 하드웨어 모델 및 지원 프로토콜

NetApp ONTAP 소프트웨어는 다양한 하드웨어 모델에서 실행되며, 각 모델은 서로 다른 성능 및 용량 요구 사항을 충족하도록 설계되었습니다. Trident는 강력한 기능을 확장하여 All Flash FAS(AFF), All SAN Array(ASA) 및 ASA R2 시스템을 포함한 다양한 NetApp ONTAP 시스템을 지원합니다. 이러한 지원을 통해 기업은 컨테이너화된 환경에서 고성능 스토리지 솔루션의 잠재력을 최대한 활용할 수 있습니다.

All Flash FAS(AFF) 시스템 AFF 시스템은 탁월한 성능과 낮은 지연 시간을 제공하여 고성능 애플리케이션에 이상적입니다.

All SAN Array (ASA) 시스템 ASA 시스템은 전용 SAN 환경을 위해 설계되었으며, 강력한 성능과 안정성을 제공합니다.

ASA R2 시스템 ASA R2 시스템은 SAN 환경을 위한 향상된 기능과 성능을 제공합니다.

AFX NetApp AFX는 기업용 AI 워크로드를 위해 특별히 설계된 분산형 올플래시 스토리지 아키텍처입니다. 고성능 NAS를 제공하여 컴퓨팅 및 용량을 독립적으로 확장할 수 있습니다. NetApp AFX 스토리지 시스템은 스토리지 계층 구현 방식에서 다른 ONTAP 기반 시스템(ASA, AFF, FAS)과 차별화됩니다. AFX는 고성능과 확장성을 제공하는 분산 파일 시스템을 사용하는 반면, 다른 ONTAP 기반 시스템은 기존 스토리지 아키텍처를 사용합니다.

1. AFF에 지원되는 프로토콜: NFS, CIFS/SMB, iSCSI, FC, NVMe/TCP
2. ASA에 지원되는 프로토콜: iSCSI, FC, NVMe/TCP
3. ASA R2에서 지원하는 프로토콜: iSCSI, FC, NVMe/TCP
4. AFX에서 지원하는 프로토콜: Trident 25.10부터는 NFS 프로토콜만 지원되며, SMB 프로토콜은 지원되지 않습니다.

ONTAP 스토리지용 Trident 드라이버

Trident에는 백엔드 스토리지에 볼륨을 프로비저닝할 수 있는 다음과 같은 CSI 드라이버가 포함되어 있습니다.

지원되는 하드웨어 모델의 **NAS** 프로토콜

1. ontap-nas: 하나의 pvc는 전용 flexvolume인 하나의 pv에 매핑됩니다.
2. ontap-nas-economy: 프로비저닝된 각 PV는 큐트리이며, FlexVolume당 구성 가능한 수의 큐트리를 가집니다 (기본값 200, 50~300 사이로 구성 가능). 하나의 PVC는 공유 FlexVolume 상의 큐트리인 하나의 PV에 매핑됩니다.



VM의 모든 디스크를 단일 볼륨에 qtree로 배치할 수 있습니다. 그러나 Snapshots, Clones 및 Replication 기능은 Trident에서 지원되지 않습니다. 이러한 기능이 스토리지 관리자에 의해 관리되는 경우, PV 단위로 세부 관리가 불가능합니다.

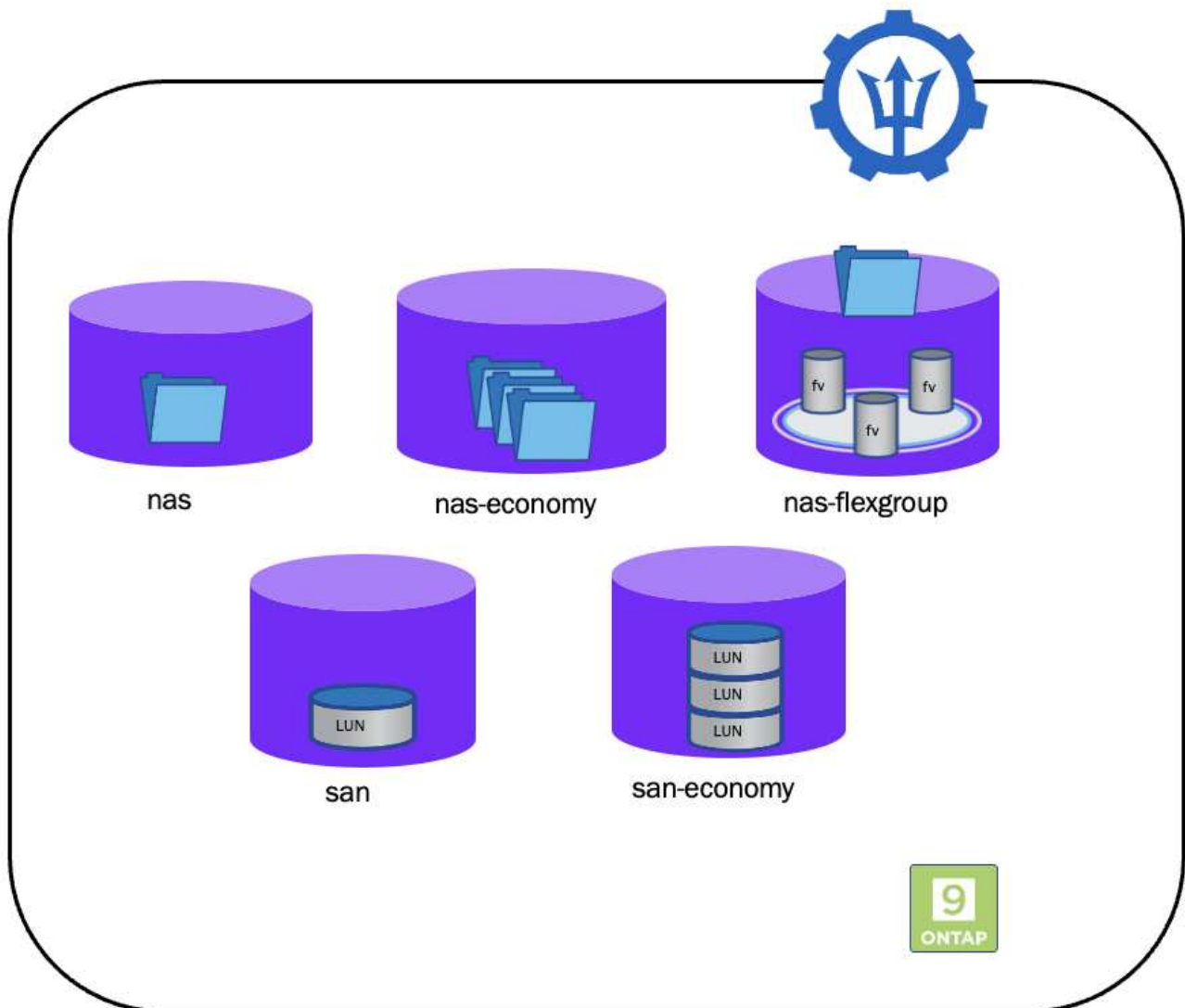
1. ontap-nas-flexgroup: 프로비저닝된 각 PV는 완전한 ONTAP FlexGroup이며, SVM에 할당된 모든 애그리게이트가 사용됩니다.

지원되는 하드웨어 모델의 **SAN** 프로토콜

1. ontap-san: 하나의 PVC는 전용 FlexVol의 LUN인 하나의 PV에 매핑됩니다.
2. ontap-san-economy: 하나의 PVC는 하나의 PV(공유 FlexVolume의 LUN)에 매핑됩니다. 공유 FlexVolume에는 여러 개의 LUN을 저장할 수 있습니다.(기본값은 100개이며, FlexVolume당 50개에서 200개 사이의 LUN/PV로 구성 가능합니다.)



모든 VM 디스크를 단일 볼륨에 LUN으로 배치할 수 있습니다. 하지만 이 드라이버는 스냅샷과 클론은 지원하지 않지만 복제는 지원하지 않습니다. 스토리지 관리자가 복제를 관리하는 경우 PV 단위로 세부적으로 관리되지 않습니다.



Trident 드라이버를 통해 제공되는 모든 기능과 스토리지 관리자가 적용해야 하는 기능에 대한 전체 목록은 Trident 설명서의 "[ONTAP 백엔드 드라이버](#)" 섹션을 참조하십시오.

Red Hat OpenShift 클러스터에 Trident를 설치하고 스토리지 객체를 생성하세요

Red Hat 인증 Trident Operator를 사용하여 Trident를 설치하고 ONTAP 및 Amazon FSx for NetApp ONTAP용 스토리지 객체를 생성하여 컨테이너 및 VM에 대한 동적 볼륨 프로비저닝을 활성화합니다. 필요한 경우 블록 액세스를 위해 작업자 노드를 준비합니다.

시작하기 전에

- OpenShift Virtualization을 설치하기 전에 이 페이지의 절차를 완료하십시오. OpenShift Virtualization은 VM 템플릿용 골든 이미지를 생성하기 위해 기본 Trident 기반 StorageClass 및 VolumeSnapshotClass가 필요합니다.
- Trident를 구성하기 전에 OpenShift Virtualization을 이미 설치한 경우 다른 스토리지 클래스로 생성된 골든 이미지를 모두 삭제하십시오. Trident를 기본값으로 설정하면 OpenShift Virtualization이 Trident 스토리지를 사용하여 골든 이미지를 다시 생성합니다.

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

1단계: Trident 설치

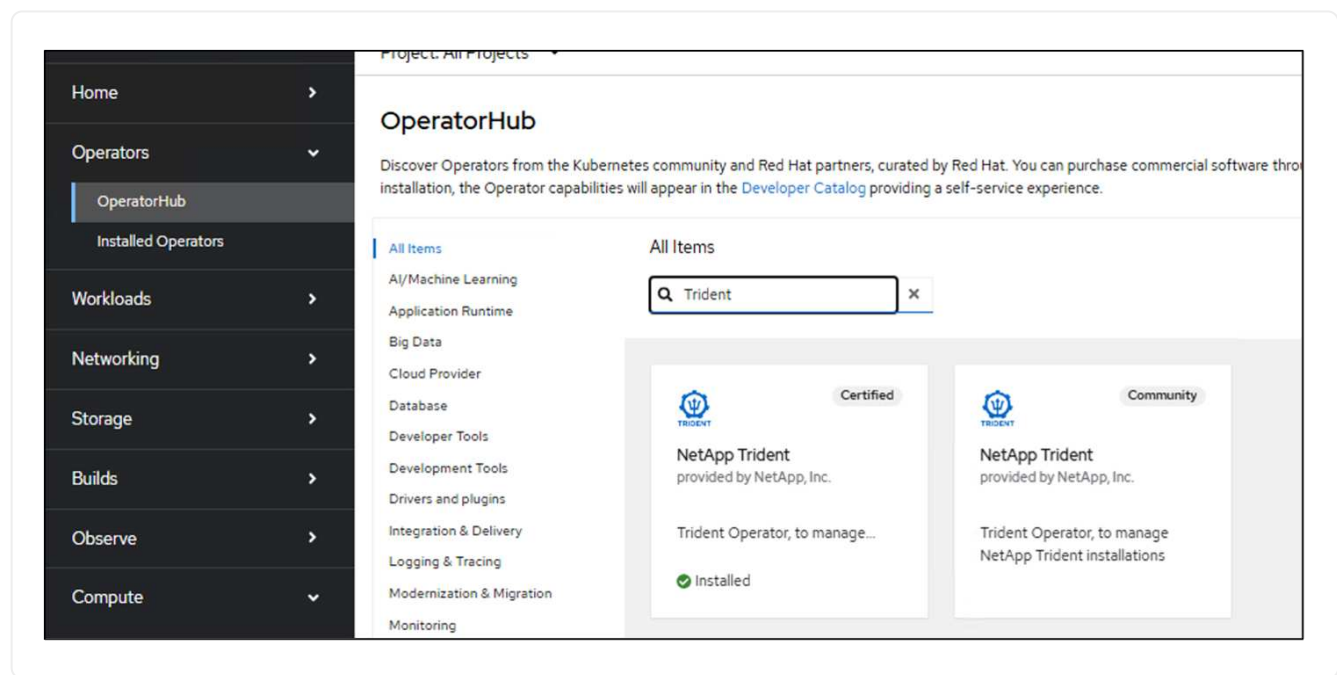
Red Hat 인증 Trident Operator는 온프레미스, 퍼블릭 클라우드 및 ROSA와 같은 관리형 서비스에서 NetApp이 OpenShift를 위해 지원합니다. Trident 25.02부터 Amazon FSx for NetApp ONTAP를 사용하고 OpenShift Virtualization VM 워크로드를 실행할 계획인 경우 Operator가 iSCSI용 작업자 노드를 준비할 수도 있습니다.

다른 설치 옵션은 "[Trident 문서](#)"를 참조하십시오.

단계

1. *OperatorHub*에서 *Certified NetApp Trident*을 선택하십시오.

예를 보여주세요



2. 설치 페이지에서 최신 버전을 유지하고 *설치*를 선택합니다.

예를 보여주세요

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. 오퍼레이터 설치가 완료되면 *View operator*를 선택하고 Trident Orchestrator 인스턴스를 생성하십시오.
iSCSI를 위해 워커 노드를 준비하려면 YAML 보기로 전환하고 `iscsi`을 `nodePrep`에 추가하세요.

예를 보여주세요

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. 클러스터에서 모든 Trident Pod가 실행 중인지 확인하십시오.

예를 보여주세요

```
[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. iSCSI 노드 준비를 활성화한 경우 워커 노드에 로그인하여 `iscsid` 및 ``multipathd``이(가) 활성 상태이고 ``multipath.conf``에 항목이 있는지 확인하십시오.

예를 보여주세요

```
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#
```

예를 보여주세요

```
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#
```

예를 보여주세요

```
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#
```

비디오 데모

다음 비디오는 Red Hat 인증 Trident Operator를 사용하여 Trident를 설치하는 방법을 보여줍니다.

[OpenShift에서 인증된 Trident Operator를 사용하여 Trident 25.02.1 설치](#)

2단계: 사용 환경에 맞는 스토리지 백엔드 및 **StorageClass** 구성 파일을 준비합니다

TridentBackendConfig 및 StorageClass 정의를 사용자 환경에 맞게 생성하십시오. 환경 내에서 여러 스토리지 프로토콜을 구성할 수 있습니다. 사용하려는 각 프로토콜에 대한 YAML 파일을 생성하고 자리 표시자 값을 특정 구성 세부 정보로 바꾸십시오.



사용 환경에 따라 아래의 온프레미스 또는 ROSA 섹션 중 하나를 완료한 후 3단계로 진행하십시오.

온프레미스 OpenShift 클러스터

구성하려는 각 프로토콜에 대한 YAML 파일을 생성하십시오. 다음 프로토콜 중 하나 이상을 구성할 수 있습니다. NFS 기반 파일 스토리지를 위한 NAS, iSCSI 블록 스토리지를 위한 iSCSI, TCP를 통한 고성능 NVMe 블록 스토리지를 위한 NVMe/TCP 또는 Fibre Channel 블록 스토리지를 위한 FC.

각 프로토콜에 대해 TridentBackendConfig 및 StorageClass를 생성합니다. 백엔드 구성에는 Kubernetes Secret에 저장된 자격 증명이 포함됩니다.

백엔드 구성 기본 옵션에 대한 필수 지침

1. 백엔드 구성 파일은 **ONTAP SVM** 및 관리 **LIF**를 참조합니다. +
2. 사용자 이름과 비밀번호를 사용하여 ONTAP에 인증할 수 있습니다. 클러스터 관리자 또는 vserver 관리자 사용자 이름과 비밀번호를 사용하면 됩니다. +
3. ONTAP에서 인증을 위해 클라이언트 인증서를 사용할 수도 있습니다. 클라이언트 인증서, 개인 키 및 CA 인증서 생성 방법에 대한 자세한 내용은 "[Trident 관련 문서는 여기에서 확인할 수 있습니다.](#)"을 참조하십시오.

NAS 이코노미 드라이버의 백엔드 구성 옵션에 대한 필수 지침

NAS 이코노미 드라이버 구성에 포함할 수 있는 매개변수는 다음과 같습니다.

1. **qtreesPerFlexvol** 값은 50~300 범위 내에 있어야 하며, 기본값은 200입니다. +
2. **limitVolumePoolSize** 이 값은 ontap-nas-economy 백엔드에서 qtree를 사용할 때 요청하는 최대 FlexVol 크기입니다. 기본값은 설정되어 있지 않으며, 이 매개변수는 기본적으로 적용되지 않습니다. 이 매개변수를 설정하면 현재 FlexVol이 지정된 크기에 도달할 때 Trident 운영자가 새 FlexVol을 생성합니다. 이 매개변수는 PV 프로비저닝에 사용되는 FlexVols의 크기를 제한하여 클러스터의 용량 활용률을 제어하는 데 도움이 될 수 있습니다. +
3. **denyNewVolumePools** 이 매개변수는 ontap-nas-economy 백엔드가 qtree를 저장할 새 FlexVol 볼륨을 생성하는 것을 제한합니다. 새 PV를 프로비저닝할 때는 기존 FlexVol만 사용됩니다.

SAN 드라이버 백엔드 구성 옵션에 대한 필수 지침

SAN 드라이버 구성에 다음 매개변수를 포함할 수 있습니다.

1. **limitVolumeSize** 이 값은 SAN 백엔드를 사용할 때 요청하는 최대 볼륨 크기입니다. 기본값은 설정되어 있지 않으며, 이 매개변수는 기본적으로 적용되지 않습니다. 이 매개변수를 설정하면 요청된 볼륨 크기가 이 값보다 클 경우 Trident 오퍼레이터의 프로비저닝이 실패합니다. 이 매개변수를 사용하여 관리되는 LUN 볼륨의 최대 크기를 제한할 수 있습니다. +
2. **lunsPerFlexvol** FlexVol당 최대 LUN 개수입니다. [50, 200] 범위 내에 있어야 하며 기본값은 100입니다. 이 매개변수를 설정하면 현재 FlexVol의 LUN 개수가 지정된 개수에 도달할 때 Trident 운영자가 새 FlexVol을 생성합니다. 이 매개변수를 사용하면 PV 프로비저닝에 사용되는 FlexVol당 LUN 개수를 제한하여 클러스터의 용량 활용률을 제어할 수 있습니다.

SAN 이코노미 드라이버 구성에 포함할 수 있는 매개변수는 다음과 같습니다.

1. **limitVolumePoolSize** 이 값은 SAN 이코노미 백엔드를 사용할 때 요청하는 최대 FlexVol 크기입니다. 기본값은 설정되어 있지 않으며, 이 매개변수는 기본적으로 적용되지 않습니다. 이 매개변수를 설정하면 Trident 운영자가 FlexVol에 LUN을 생성하기 위해 생성할 수 있는 최대 FlexVol 크기를 제한합니다. 이 매개변수는 PV 프로비저닝에 사용되는 FlexVols 크기를 제한하여 클러스터의 용량 활용률을 제어하는 데 도움이 될 수 있습니다. +
2. **denyNewVolumePools** 이 매개변수는 SAN 이코노미 백엔드가 LUN을 담을 새로운 FlexVol 볼륨을 생성하는 것을 제한합니다. 기존 FlexVol만 새로운 PV 프로비저닝에 사용됩니다.



각 프로토콜에 제공된 샘플 YAML 파일에서 storagePrefix가 제공되지 않으면 기본 접두사 "trident"가 사용됩니다. ontap-nas-economy를 사용하고 storagePrefix가 24자 이상인 경우, qtree에는 스토리지 접두사가 포함되지 않지만 볼륨 이름에는 포함됩니다.

ONTAP에서 시스템 관리자 또는 CLI를 통해 직접 볼륨을 생성할 때 모든 선택적 매개변수에 값을 제공하지 않으면 아래와 같이 볼륨 프로비저닝 중에 일부 기본값이 자동으로 설정됩니다.

```
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,
x-permissions,snapsdir-access,space-guarantee
vserver      volume      unix-permissions space-guarantee snapsdir-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
```

다음은 Trident 백엔드 구성을 사용하여 생성된 볼륨의 예입니다. 이름 템플릿을 제외하고는 기본값 섹션에서 명시적으로 설정된 옵션이 없습니다.

```
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
```

이 백엔드 구성을 사용하여 볼륨을 생성하면 백엔드 구성 파일에 설정되지 않은 옵션이 아래와 같이 기본값으로 설정됩니다.

```
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,snapsdir-access,space-guarantee
vserver      volume      unix-permissions space-guarantee snapsdir-access snapshot-policy qos-adaptive-policy-group snapshot-reserve-available t
tiering-policy
oc-test-automation testautomation_default_test_trident_defaults_91653 ---rwxrwxrwx      none          false      none      -          0B          n
one
```

하지만 Trident 백엔드 구성의 기본 섹션에서 값을 설정하면 해당 매개변수에 특정 값을 지정할 수 있는 볼륨을 생성할 수 있습니다. 다음 예에서는 snapshotPolicy를 기본 정책(명시적으로 설정하지 않으면 none)으로, snapshotDir을 백엔드 구성 YAML의 기본 섹션에서 true로 설정했습니다. 이러한 설정을 통해 백엔드 볼륨이 해당 값을 갖는 것을 확인할 수 있습니다. (snapsdir-access는 true이고 snapshot-policy는 기본 정책입니다.)

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

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

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

그 외 추가 매개변수에 대해서는 Trident 설명서를 참조하십시오.

1. "Trident NAS 백엔드 구성 옵션" +
2. "Trident SAN 백엔드 구성 옵션"

NAS

TridentBackendConfig 및 StorageClass를 생성하여 ONTAP NAS에 대한 NFS 기반 영구 스토리지 프로비저닝을 활성화합니다. 백엔드 구성에는 Kubernetes Secret에 저장된 자격 증명이 포함되며 ONTAP SVM 및 관리 LIF를 참조합니다.

사용자 이름과 비밀번호 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일 예시 (다음으로 저장 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 }}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-nas-secret
```

클라이언트 인증서 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일 예시 (다른 이름으로 저장 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 정의(다른 이름으로 저장 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 이코노미

TridentBackendConfig 및 StorageClass를 생성하여 ONTAP NAS 이코노미 드라이버에서 qtree를 사용한 NFS 기반 영구 스토리지 프로비저닝을 활성화합니다. 백엔드 구성에는 Kubernetes Secret에 저장된 자격 증명이

포함되며 ONTAP SVM 및 관리 LIF를 참조합니다.

사용자 이름과 비밀번호 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일 예시 (다음으로 저장 tbc-nas-economy.yaml):

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

StorageClass 정의(다른 이름으로 저장 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
```



ontap-nas-economy 드라이버의 백엔드 구성 파일에 포함할 수 있는 추가 매개변수에 대한 자세한 내용은 위의 "NAS 이코노미 드라이버 백엔드 구성 옵션에 대한 필수 지침" 섹션을 참조하십시오.

iSCSI SAN

iSCSI를 사용하는 ONTAP SAN에 대한 TridentBackendConfig 및 StorageClass를 생성하여 iSCSI 기반 블록 스토리지 프로비저닝을 활성화합니다. 백엔드 구성은 ontap-san 드라이버를 사용하며 Kubernetes Secret에 저장된 자격 증명을 포함합니다. sanType 매개변수를 생략하면 기본적으로 iSCSI 프로토콜이 사용됩니다.

사용자 이름 비밀번호 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일(다른 이름으로 저장 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

```

클라이언트 인증서 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일(다른 이름으로 저장 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 정의(다른 이름으로 저장 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

NVMe over TCP를 사용하는 ONTAP SAN용 TridentBackendConfig 및 StorageClass를 생성하여 고성능 블록 스토리지 프로비저닝을 활성화합니다. 백엔드 구성은 NVMe/TCP 전송에 최적화된 ontap-san 드라이버를 사용하며 Kubernetes Secret에 저장된 자격 증명을 포함합니다.

사용자 이름 비밀번호 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일(다른 이름으로 저장 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
```

클라이언트 인증서 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일(다른 이름으로 저장 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 정의(다른 이름으로 저장 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

```



ReadWriteMany(RWX) 볼륨을 NVMe 프로토콜과 함께 사용하려는 경우, Kubernetes 클러스터에서 단일 RWX NVMe 볼륨은 64개 이상의 노드에서 마운트할 수 없습니다. 이 제한 사항 및 Trident 26.02에서 도입된 슈퍼 서브시스템 모델과 Trident 26.02 이전의 볼륨별 서브시스템 모델에 대한 자세한 내용은 Trident 설명서를 참조하십시오.

<https://docs.netapp.com/us-en/trident/trident-use/ontap-san-prep.html#nvme-tcp-considerations>

FC SAN

TridentBackendConfig 및 StorageClass를 생성하여 Fibre Channel이 포함된 ONTAP SAN에 대해 FC 기반 블록 스토리지 프로비저닝을 활성화합니다. 백엔드 구성은 FCP 프로토콜이 지정된 ontap-san 드라이버를 사용하며 Kubernetes Secret에 저장된 자격 증명을 포함합니다.

사용자 이름/암호 인증 방법 또는 클라이언트 인증서 인증 방법을 사용하여 ONTAP으로 인증합니다. 클라이언트 인증서, 개인 키 및 CA 인증서 생성 방법에 대한 자세한 내용은 "[Trident 관련 문서는 여기에서 확인할 수 있습니다.](#)"을 참조하십시오. 두 가지 방법에 대한 예시는 아래에 제공됩니다.

예를 보여주세요

ONTAP 사용자 이름 및 비밀번호 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일(`tbc-fc.yaml`로 저장):

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

클라이언트 인증서 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일(다른 이름으로 저장 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
```

TridentBackendConfig 및 StorageClass를 생성하여 iSCSI를 사용하는 ONTAP SAN economy에 대해 iSCSI 기반 블록 스토리지 프로비저닝을 활성화합니다. 백엔드 구성은 iSCSI 프로토콜을 사용하는 ontap-san-economy 드라이버를 사용하며 Kubernetes Secret에 저장된 자격 증명을 포함합니다.

클라이언트 인증서 인증을 사용하는 백엔드 비밀 키 및 백엔드 구성 파일(다른 이름으로 저장 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"
```

Trident 설명서를 참조하시면 YAML 파일의 추가 샘플과 SAN 드라이버 및 NAS 드라이버에서 지원하는 액세스 모드 및 볼륨 모드에 대한 정보를 확인할 수 있습니다.

1. ["NAS 드라이버를 사용한 샘플"](#)
2. ["SAN 드라이버를 사용하는 샘플"](#)

Amazon FSx for NetApp ONTAP를 사용하는 ROSA 클러스터

구성하려는 각 프로토콜에 대한 YAML 파일을 생성하십시오. NFS 기반 파일 스토리지를 위한 NAS 또는 블록 스토리지를 위한 iSCSI 중 하나 또는 둘 다를 구성할 수 있습니다.

NAS

ROSA 클러스터에서 NFS 기반 영구 스토리지 프로비저닝을 활성화하기 위해 Amazon FSx for NetApp ONTAP의 ONTAP NAS용 TridentBackendConfig 및 StorageClass를 생성합니다. 백엔드 구성은 관리 및 데이터 LIF를 위해 Amazon FSx for NetApp ONTAP DNS 이름을 사용하며, trident 네임스페이스의 Kubernetes Secret에 저장된 자격 증명을 포함합니다.

백엔드 시크릿 및 백엔드 구성 파일(다른 이름으로 저장 `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 정의(다른 이름으로 저장 `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

ROSA 클러스터에서 iSCSI 기반 블록 스토리지 프로비저닝을 활성화하기 위해 Amazon FSx for NetApp ONTAP의 ONTAP SAN용 TridentBackendConfig 및 StorageClass를 생성합니다. 백엔드 구성은 ontap-san 드라이버를 사용하며, Kubernetes Secret에 저장된 자격 증명을 포함합니다. 작업자 노드가 iSCSI 액세스를 위해 준비되어 있는지 확인합니다.

백엔드 시크릿 및 백엔드 구성 파일(다른 이름으로 저장 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 정의(다른 이름으로 저장 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
```

3단계: VolumeSnapshotClass 구성 파일 생성

온프레미스 및 ROSA 배포 환경 모두에 대해 VolumeSnapshotClass 정의를 생성합니다. 이 구성을 통해 영구 볼륨에 대한 스냅샷 기반 작업을 활성화할 수 있습니다.

VolumeSnapshotClass 정의(다른 이름으로 저장 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
```

4단계: 구성 파일을 클러스터에 적용합니다

이전 단계에서 생성한 구성 파일을 OpenShift 클러스터에 적용합니다.

단계

1. 구성한 각 프로토콜에 대해 TridentBackendConfig 및 StorageClass 파일을 적용하십시오.

온프레미스 클러스터의 경우:

```
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
```

ROSA 클러스터의 경우:

```
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. VolumeSnapshotClass 구성을 적용합니다.

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

3. 리소스가 성공적으로 생성되었는지 확인하십시오.

TridentBackendConfig 객체를 확인하세요:

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

StorageClass 객체를 확인하세요:

```
oc get storageclass
```

VolumeSnapshotClass 확인:

```
oc get volumesnapshotclass
```

5단계: 기본 Trident 스토리지 및 스냅샷 클래스 설정

OpenShift 클러스터에서 Trident StorageClass와 VolumeSnapshotClass를 기본값으로 설정합니다. 이는 OpenShift Virtualization이 VM 템플릿을 위한 골든 이미지 소스를 생성하는 데 필요합니다.

단계

1. 기본 Trident StorageClass를 설정합니다.

Trident 기반 StorageClass를 클러스터 기본값으로 설정하여 스토리지 클래스가 지정되지 않은 경우 PersistentVolumeClaims가 자동으로 사용하도록 합니다. 클러스터 전체 기본값용 어노테이션 하나와 OpenShift Virtualization 전용 어노테이션 하나, 총 두 개의 어노테이션을 구성해야 합니다.

a. 클러스터 전체의 기본 StorageClass 어노테이션을 설정합니다.

하나의 StorageClass만 기본값으로 설정되어 있는지 확인하십시오. 다른 StorageClass가 이미 기본값으로 설정되어 있는 경우 해당 주석을 false로 설정하십시오.

콘솔에서 주석을 편집합니다.

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

CLI에서:

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

b. OpenShift Virtualization 관련 기본 주석을 설정합니다.

OpenShift Virtualization은 클러스터의 일반 is-default-class 어노테이션보다 우선순위가 높은 특정 어노테이션을 사용합니다. 다른 StorageClass가 이미 기본값으로 설정되어 있는 경우 해당 어노테이션을 false로 설정하십시오.

콘솔에서 주석을 편집합니다.

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

CLI에서:

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

2. 기본 Trident VolumeSnapshotClass를 설정합니다.

Trident 기반 VolumeSnapshotClass를 클러스터 기본값으로 설정하면 영구 볼륨에 대한 스냅샷 기반 작업이 활성화됩니다. 이렇게 하면 스냅샷 클래스가 지정되지 않은 경우 VolumeSnapshots가 자동으로 Trident CSI 드라이버를 사용하게 되며 OpenShift Virtualization에서 골든 이미지의 스냅샷을 생성할 수 있습니다.

VolumeSnapshotClass는 기본값으로 하나만 설정해야 합니다. 이미 다른 VolumeSnapshotClass가 기본값으로 설정되어 있는 경우, 해당 클래스의 어노테이션을 false로 설정하십시오.

콘솔에서 주석을 편집합니다.

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

CLI에서:

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

ONTAP 사용하여 Red Hat OpenShift Virtualization을 배포하기 위한 요구 사항

OpenShift 가상화를 ONTAP 스토리지 시스템과 함께 배포하기 위한 필수 조건 및 추가 세부 정보를 검토하십시오.

필수 조건

- RHCOS 작업자 노드가 있는 베어 메탈 인프라에 설치된 Red Hat OpenShift 클러스터(버전 4.6 이상)
- VM의 HA를 유지하기 위해 머신 상태 점검 배포
- SVM이 올바른 프로토콜로 구성된 NetApp ONTAP 클러스터입니다.
- OpenShift 클러스터에 Trident 설치
- Trident 백엔드 구성이 생성되었습니다.
- Trident 프로비저너로 사용하여 OpenShift 클러스터에 구성된 StorageClass
- Trident를 프로비저너로 하여 OpenShift 클러스터에 구성된 Trident VolumeSnapshotClass
- Trident StorageClass가 OpenShift 클러스터 또는 OpenShift Virtualization의 기본값으로 구성되어 있습니다.

위의 Trident 전제 조건은 다음을 참조하세요. ["Trident 설치 섹션"](#) 자세한 내용은.

- Red Hat OpenShift 클러스터에 대한 클러스터 관리자 액세스
- NetApp ONTAP 클러스터에 대한 관리자 액세스
- \$PATH에 tridentctl 및 oc 도구가 설치되고 추가된 관리자 워크스테이션

OpenShift 가상화는 OpenShift 클러스터에 설치된 오퍼레이터에 의해 관리되므로 메모리, CPU 및 스토리지에 추가적인 오버헤드가 발생합니다. 따라서 클러스터의 하드웨어 요구 사항을 계획할 때 이러한 오버헤드를 고려해야

합니다. 자세한 내용은 설명서 "[여기](#)"를 참조하십시오.

선택적으로 노드 배치 규칙을 구성하여 OpenShift 가상화 운영자, 컨트롤러 및 VM을 호스팅할 OpenShift 클러스터 노드의 하위 집합을 지정할 수도 있습니다. OpenShift 가상화에 대한 노드 배치 규칙을 구성하려면 설명서를 참조하십시오 "[여기](#)".

OpenShift Virtualization을 지원하는 스토리지의 경우, NetApp 특정 Trident 백엔드에서 스토리지를 요청하는 전용 StorageClass를 사용하는 것을 권장하며, 이는 전용 SVM에 의해 지원됩니다. 이를 통해 OpenShift 클러스터의 VM 기반 워크로드에 제공되는 데이터와 관련하여 다중 테넌시 수준이 유지됩니다.

NetApp Trident CSI, 스토리지 프로파일, OpenShift 가상화를 위한 볼륨 액세스 모드

OpenShift Virtualization에서 부팅 소스는 템플릿을 사용하여 가상 머신(VM)을 프로비저닝하는 데 사용되는 디스크 이미지입니다. 이러한 부팅 소스는 일반적으로 DataVolumes 또는 openshift-virtualization-os-images 네임스페이스 내의 Persistent Volume Claim(PVC)으로 저장됩니다. OpenShift Virtualization이 올바르게 작동하려면 운영자를 사용하여 OpenShift Virtualization을 설치하기 전에 두 가지 스토리지 구성 작업을 반드시 수행해야 합니다.

1. 클러스터에 대한 기본 StorageClass를 구성해야 합니다. 그렇지 않으면 OpenShift Virtualization에서 부팅 소스 이미지를 자동으로 가져올 수 없습니다.
2. 스토리지 공급업체가 CDI에서 인식되지 않는 경우 스토리지 프로필을 구성해야 합니다. 스토리지 프로파일은 관련 StorageClass를 기반으로 권장 스토리지 설정을 제공합니다. StorageProfile의 중요한 기능 중 하나는 StorageClass에 대한 복제 전략을 정의하는 것입니다. 이를 통해 볼륨 복제 방식(예: 스냅샷, CSI 복제 또는 호스트 지원 복사본 사용)이 결정되며, 기본 스토리지 공급업체의 기능을 기반으로 성능과 리소스 사용량을 최적화합니다. + Trident CSI 드라이버는 CDI에서 인식되며, 클러스터에 생성된 모든 Trident StorageClass에 대해 기본 스토리지 프로파일로 생성됩니다. 이 기본 스토리지 프로파일의 복제 전략은 스냅샷으로 설정되며, 이는 Trident CSI 드라이버에 권장되는 복제 전략입니다. OpenShift Virtualization을 설치하기 전에 Trident용 VolumeSnapshotClass 및 StorageClass를 생성하고 StorageClass를 기본값으로 설정해야 합니다. 이를 통해 부팅 소스 이미지를 openshift-virtualization-os-images 네임스페이스에서 VolumeSnapshots로 사용할 수 있으므로, 오류 없이 효율적으로 VM을 프로비저닝할 수 있습니다.

특정 지원 운영 체제(예: RHEL, Fedora 또는 CentOS)의 경우 OpenShift Virtualization은 부팅 소스를 자동으로 제공하고 업데이트합니다. 이러한 부팅 소스는 OpenShift Virtualization 설치 시 다운로드되며, 최신 OS 버전으로 자동 관리 및 업데이트됩니다. OS가 자동으로 제공되지 않는 경우 관리자는 사용자 지정 이미지를 수동으로 준비하여 연결해야 합니다. NetApp Trident를 CSI로 사용하면 다음과 같은 추가 이점을 얻을 수 있습니다.

1. 부팅 소스 이미지는 openshift-virtualization-os-images 네임스페이스에 스냅샷 형태로 제공됩니다. 가상 머신이 프로비저닝될 때 부팅 이미지가 스냅샷에서 복제되므로 가상 머신 부팅 속도가 훨씬 빨라집니다.
2. Trident StorageClass가 생성될 때마다 스토리지 프로파일이 자동으로 생성되고, 복제 전략이 스냅샷으로 자동 구성됩니다. 액세스 모드는 ReadWriteMany로 설정됩니다. 수동 구성이 필요하지 않습니다.

클러스터에 OpenShift Virtualization을 설치하기 전에 Trident StorageClass 및 Trident VolumeSnapshotClass를 생성하고 이를 기본값으로 설정하는 것이 매우 중요합니다. 이렇게 하면 부트 소스 이미지가 openshift-virtualization-os-images 네임스페이스에서 VolumeSnapshots로 제공되어 오류 없이 효율적으로 VMs를 프로비저닝할 수 있습니다.

Trident CSI를 사용하는 또 다른 장점: . OpenShift Virtualization에서 VM의 Live migration을 수행하려면 VM의 부트 디스크에 RWX 액세스 모드가 있어야 합니다. Trident는 NAS 프로토콜뿐만 아니라 SAN 프로토콜을 사용하여 프로비저닝된 볼륨에 대해서도 **ReadWriteMany** 액세스 모드를 지원합니다(volumeMode가 FileSystem이 아닌 Block으로 설정된 경우). 부트 디스크에 두 프로토콜 중 하나를 사용할 수 있습니다. Storage profile 오브젝트는 StorageClass 설정을 기반으로 자동으로 구성되며, VM을 프로비저닝할 때 부트 디스크 액세스 모드가 적절하게 설정됩니다.

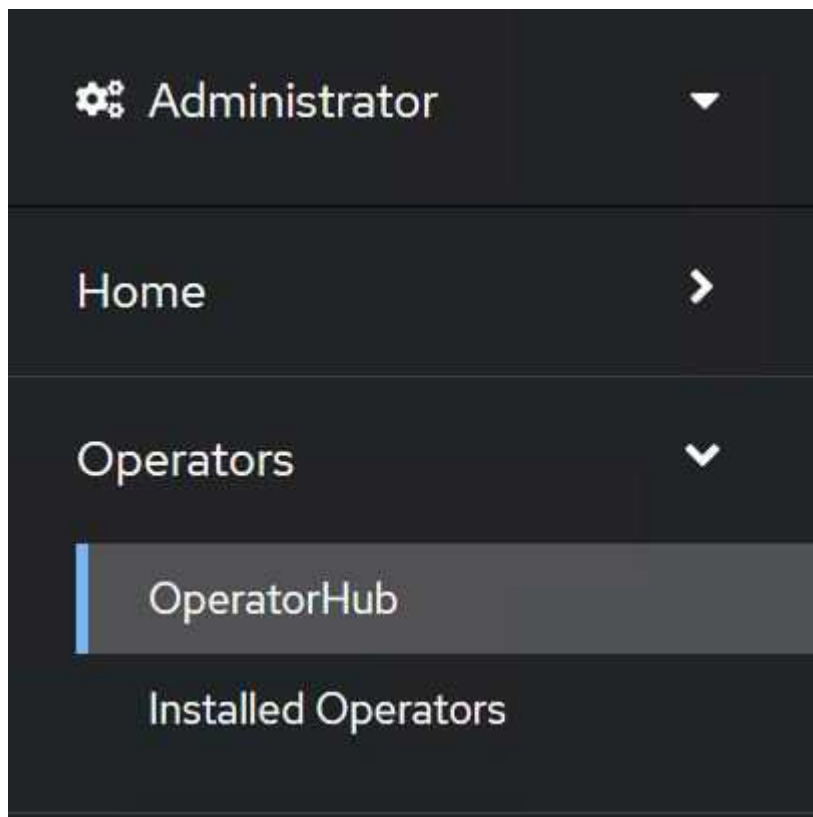
클러스터의 기본값으로 Trident StorageClass를 설정한 후에는 다른 매개변수를 사용하여 Trident용 추가

StorageClasses를 생성할 수 있으며, OpenShift Virtualization은 이러한 StorageClass를 자동으로 인식하고 적절한 스토리지 프로파일을 생성합니다. 그런 다음 VM 정의에서 VM의 부팅 디스크와 데이터 디스크에 사용할 StorageClass를 지정할 수 있습니다.

OpenShift 클러스터에 OpenShift Virtualization을 설치하는 것은 OpenShift OperatorHub에서 사용할 수 있는 OpenShift Virtualization 오퍼레이터를 사용하는 간단한 프로세스입니다. OpenShift Virtualization을 설치하는 방법에 대한 자세한 단계는 설명서 "[여기](#)"를 참조하십시오.

Red Hat OpenShift 베어 메탈 클러스터에 OpenShift Virtualization을 설치합니다. 이 절차에는 클러스터 관리자 액세스 권한으로 로그인하고, OperatorHub로 이동하고, OpenShift Virtualization 운영자를 설치하는 작업이 포함됩니다.

1. 클러스터 관리자 액세스 권한으로 Red Hat OpenShift 베어 메탈 클러스터에 로그인합니다.
2. Perspective 드롭다운에서 Administrator를 선택하세요.
3. Operators > OperatorHub로 이동하여 OpenShift Virtualization을 검색합니다.



4. OpenShift Virtualization 타일을 선택하고 설치를 클릭합니다.



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. 설치 운영자 화면에서 모든 기본 매개변수를 그대로 두고 설치를 클릭합니다.

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. 운영자 설치가 완료될 때까지 기다리세요.



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. 운영자를 설치한 후 HyperConverged 만들기를 클릭합니다.



OpenShift Virtualization
2.6.2 provided by Red Hat

Installed operator – operand required

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

HC HyperConverged **Required**

Creates and maintains an OpenShift Virtualization Deployment

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

8. HyperConverged 생성 화면에서 생성을 클릭하고 모든 기본 매개변수를 수락합니다. 이 단계에서는 OpenShift Virtualization 설치가 시작됩니다.

Name *

kubevirt-hyperconverged

Labels

app=frontend

Infra >

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

Workloads >

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

Bare Metal Platform

☒ true

BareMetalPlatform indicates whether the infrastructure is baremetal.

Feature Gates >

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

Local Storage Class Name

LocalStorageClassName the name of the local storage class.

Create **Cancel**

9. 모든 포드가 openshift-cnv 네임스페이스에서 실행 상태로 이동하고 OpenShift Virtualization 연산자가 성공 상태가 되면 연산자를 사용할 준비가 됩니다. 이제 OpenShift 클러스터에서 VM을 생성할 수 있습니다.

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

Red Hat OpenShift Virtualization을 사용하여 ONTAP 스토리지로 VM 생성

OpenShift 가상화를 사용하여 가상 머신(VM)을 생성합니다. 이 절차에는 운영 체제 템플릿 선택, StorageClasses 구성, 특정 요구 사항에 맞게 VM 매개변수를 사용자 지정하는 과정이 포함됩니다. 사전 요구 사항으로 Trident 백엔드, StorageClass 및 볼륨 스냅샷 클래스 객체를

이미 생성해야 합니다. 자세한 내용은 "[Trident 설치 섹션](#)"을 참조하십시오.

VM 생성

VM은 운영 체제와 데이터를 호스팅하기 위해 볼륨이 필요한 상태 저장 배포입니다. CNV를 사용하면 VM이 포드로 실행되고 VM은 Trident 통해 NetApp ONTAP에 호스팅된 PV에 의해 지원됩니다. 이러한 볼륨은 디스크로 연결되며 VM의 부팅 소스를 포함한 전체 파일 시스템을 저장합니다.



VM 디스크용으로 생성된 PVC가 Trident 백엔드를 사용하여 프로비저닝되도록 하려면 클러스터에서 Trident StorageClass를 기본 StorageClass로 설정해야 합니다. 클러스터에 여러 StorageClasses를 사용할 수 있지만 VM 부팅 디스크용 PVC 프로비저닝에는 기본 StorageClass를 사용해야 합니다. 필요한 경우 추가 디스크에는 다른 StorageClasses를 사용할 수 있습니다.



OpenShift 클러스터에서 가상 머신을 빠르게 생성하려면 다음 단계를 완료하세요.

1. 가상화 > 가상 머신으로 이동하여 만들기를 클릭합니다.
2. 템플릿에서 선택하세요.
3. 부팅 소스를 사용할 수 있는 원하는 운영 체제를 선택하세요.
4. VirtualMachine을 생성한 후 시작 확인란을 선택합니다.
5. 빠른 가상 머신 생성을 클릭합니다.

가상 머신이 생성되고 시작되어 실행 중 상태가 됩니다. 기본 StorageClass를 사용하여 부팅 디스크에 대한 PVC 및 해당 PV를 자동으로 생성합니다. 향후 가상 머신을 라이브 마이그레이션하려면 디스크에 사용된 StorageClass가

RWX 볼륨을 지원하는지 확인해야 합니다. 이는 라이브 마이그레이션의 필수 조건입니다. ontap-nas 및 ontap-san(iSCSI 및 NVMe/TCP 프로토콜용 volumeMode 블록)은 각각의 StorageClasses를 사용하여 생성된 볼륨에 대해 RWX 액세스 모드를 지원합니다.

클러스터에서 ontap-san StorageClass를 구성하려면 "[StorageClass 구성 섹션](#)"을 참조하십시오.



Quick create VirtualMachine을 클릭하면 기본 StorageClass를 사용하여 VM의 부팅 가능한 루트 디스크용 PVC 및 PV를 생성합니다. Customize VirtualMachine > Customize VirtualMachine parameters > Disks를 선택한 다음 필요한 StorageClass를 사용하도록 디스크를 편집하여 디스크에 다른 StorageClass를 선택할 수 있습니다.

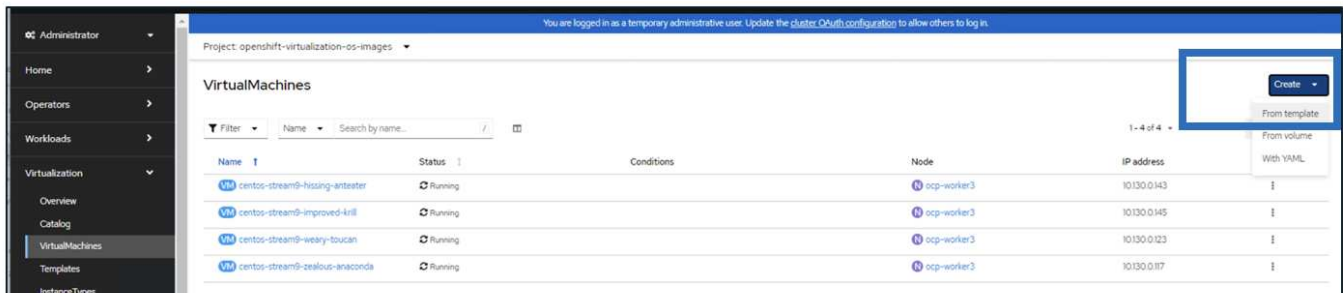
일반적으로 VM 디스크를 프로비저닝하는 동안 파일 시스템보다 블록 액세스 모드가 선호됩니다.

OS 템플릿을 선택한 후 가상 머신 생성을 사용자 지정하려면 빠른 생성 대신 가상 머신 사용자 지정을 클릭합니다.

1. 선택한 운영 체제에 부팅 소스가 구성되어 있으면 *VirtualMachine 매개변수 사용자 지정*을 클릭할 수 있습니다.
2. 선택한 운영 체제에 부팅 소스가 구성되어 있지 않으면 이를 구성해야 합니다. 표시된 절차에 대한 세부 정보를 볼 수 있습니다."[선적 서류 비치](#)".
3. 부팅 디스크를 구성한 후 *VirtualMachine 매개변수 사용자 지정*을 클릭할 수 있습니다.
4. 이 페이지의 탭에서 VM을 사용자 정의할 수 있습니다. 예를 들어, 디스크 탭을 클릭한 다음 *디스크 추가*를 클릭하여 VM에 다른 디스크를 추가합니다.
5. 가상 머신을 생성하려면 '가상 머신 생성'을 클릭합니다. 그러면 백그라운드에서 해당 포드가 실행됩니다.



URL 또는 레지스트리에서 템플릿이나 운영 체제에 대한 부팅 소스가 구성되면 PVC가 생성됩니다. openshift-virtualization-os-images 프로젝트를 실행하고 KVM 게스트 이미지를 PVC로 다운로드합니다. 해당 OS에 대한 KVM 게스트 이미지를 수용할 수 있을 만큼 템플릿 PVC에 충분한 프로비저닝 공간이 있는지 확인해야 합니다. 이러한 PVC는 해당 프로젝트의 해당 템플릿을 사용하여 생성될 때 복제되어 가상 머신에 루트디스크로서 연결됩니다.



Create new VirtualMachine

Select an option to create a VirtualMachine from.

Template catalog InstanceTypes

Template project All projects

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

Workload Desktop

CPU 1

Memory 4 GiB

Microsoft Windows 11 VM

windows11-desktop-medium

Project openshift

Boot source PVC

Workload Desktop

CPU 2

Memory 4 GiB

Microsoft Windows Server 2012 R2 VM

windows2k12r2-server-medium

Project openshift

Boot source PVC

Workload Server

CPU 1

Memory 4 GiB

VirtualMachines > VirtualMachine details

VM centos-stream9-zealous-anaconda Running

VAML Actions

Overview Details Metrics VAML Configuration Events Console Snapshots Diagnostics

Disks

Add disk

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

비디오 데모

다음 비디오에서는 iSCSI 스토리지를 사용하여 OpenShift Virtualization에서 VM을 만드는 방법을 보여줍니다.

[Block Storage를 사용하여 OpenShift Virtualization에서 VM 만들기](#)

Red Hat OpenShift Virtualization에서 대체 방법을 통해 ONTAP 스토리지를 사용하여 VM 생성

OpenShift Virtualization UI를 사용하여 VM을 생성하는 방법 외에도 OpenShift Virtualization에서 VM을 생성하는 다른 방법이 있습니다. 이 섹션에서는 이러한 방법 중 일부를 간략하게 설명합니다.

Red Hat OpenShift Virtualization을 사용하여 YAML 파일에서 VM 생성

VM은 `oc apply` 명령어를 사용하여 YAML 파일을 적용하는 방식으로 생성할 수 있습니다. 이 방법은 CLI 사용을 선호하거나 스크립트를 사용하여 VM 생성을 자동화하려는 사용자에게 유용합니다. 이 방법을 사용하여 VM을 생성하려면 VM과 VM 디스크용 PVC와 같은 관련 리소스를 정의하는 YAML 파일을 생성해야 합니다. 그런 다음 `oc apply -f <yaml-file>` 명령어를 사용하여 YAML 파일을 적용하면 OpenShift 클러스터에 VM이 생성됩니다.

예를 보여주세요

```
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

```

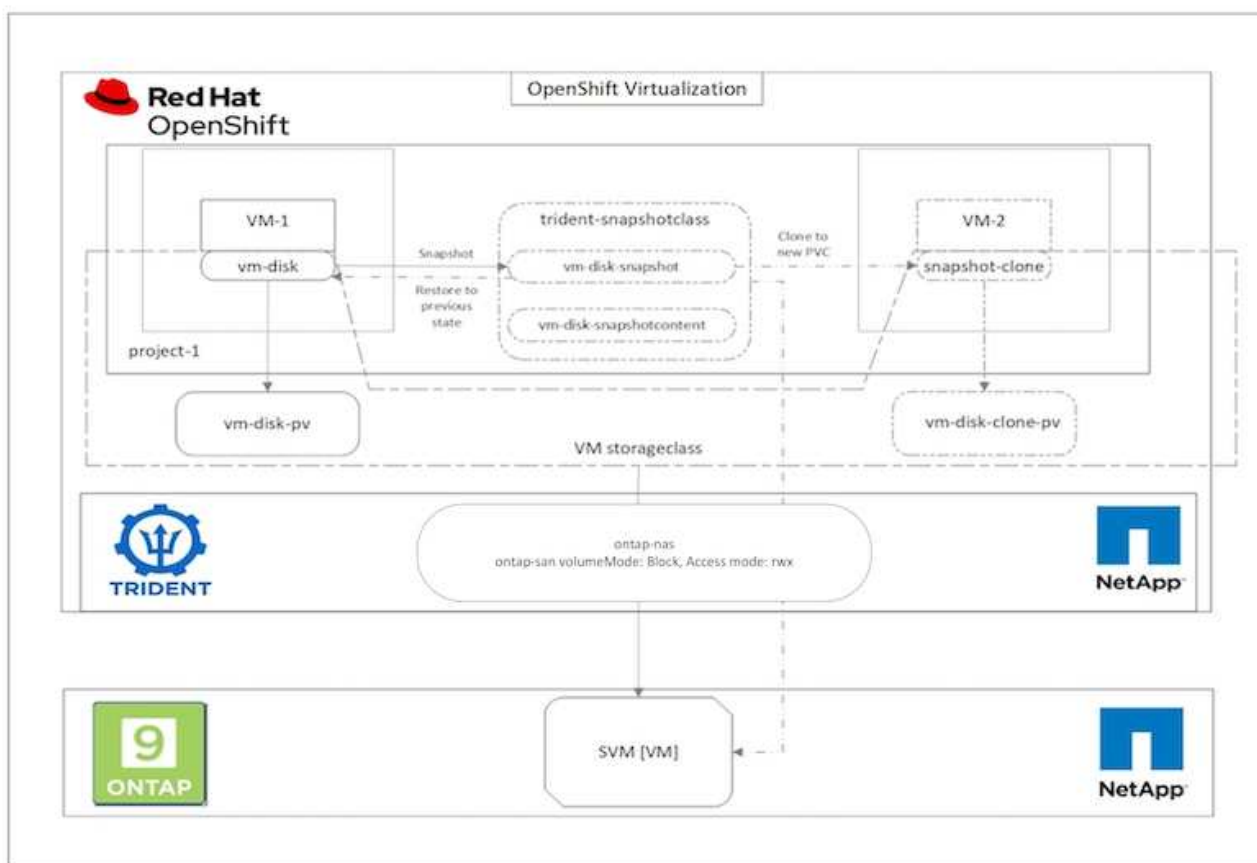
Red Hat OpenShift Virtualization을 사용하여 스냅샷 복사본에서 VM 만들기

OpenShift Virtualization을 사용하면 스냅샷에서 가상 머신(VM)을 생성할 수 있습니다. 이 절차에는 VolumeSnapshotClass 생성, VM의 영구 볼륨 클레임(PVC) 스냅샷 생성, 새 PVC에 스냅샷 복원, 그리고 복원된 PVC를 루트 디스크로 사용하는 새 VM 배포가 포함됩니다.

Trident 와 Red Hat OpenShift를 사용하면 사용자는 해당 볼륨에서 프로비저닝된 스토리지 클래스의 영구 볼륨에 대한 스냅샷을 만들 수 있습니다. 이 기능을 사용하면 사용자는 볼륨의 특정 시점 복사본을 만들어 새 볼륨을 만들거나 동일한 볼륨을 이전 상태로 복원할 수 있습니다. 이를 통해 롤백, 복제, 데이터 복원 등 다양한 사용 사례가 가능해지고 지원됩니다.

OpenShift에서 스냅샷 작업을 수행하려면 VolumeSnapshotClass, VolumeSnapshot 및 VolumeSnapshotContent 리소스를 정의해야 합니다.

- VolumeSnapshotContent는 클러스터의 볼륨에서 찍은 실제 스냅샷입니다. 이는 저장소를 위한 PersistentVolume과 유사한 클러스터 전체 리소스입니다.
- VolumeSnapshot은 볼륨의 스냅샷을 생성하기 위한 요청입니다. 이는 PersistentVolumeClaim과 유사합니다.
- VolumeSnapshotClass를 사용하면 관리자가 VolumeSnapshot에 대해 다양한 속성을 지정할 수 있습니다. 동일한 볼륨에서 찍은 다양한 스냅샷에 대해 서로 다른 속성을 가질 수 있습니다.



예를 보여주세요

필수 조건

PVC의 스냅샷을 생성하려면 먼저 Trident Storage Class에 프로비저닝된 PVC가 연결된 기존 VM이 있어야 하며, VM이 실행 중인 상태여야 합니다. VolumeSnapshot을 생성하는 데 사용할 수 있는

VolumeSnapshotClass가 생성되어 있어야 합니다. 이는 OpenShift Virtualization을 설치하기 위한 사전 요구사항의 일부로 생성되었을 것입니다. 자세한 내용은 "[VolumeSnapshotClass 구성을 생성하는 3단계](#)"를 참조하십시오.

VM의 스냅샷을 생성하려면 다음 단계를 완료하십시오.

1. 소스 VM에 연결된 PVC를 식별한 다음 해당 PVC의 스냅샷을 만듭니다. 로 이동 `Storage > VolumeSnapshots` 볼륨 스냅샷 만들기를 클릭하세요.
2. 스냅샷을 만들려는 PVC를 선택하고 스냅샷 이름을 입력하거나 기본값을 사용하고 적절한 VolumeSnapshotClass를 선택합니다. 그런 다음 만들기를 클릭합니다.
3. 이렇게 하면 해당 시점의 PVC 스냅샷이 생성됩니다.

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

스냅샷에서 새 **VM** 생성

1. 먼저 스냅샷을 새로운 PVC로 복원합니다. 저장소 > 볼륨 스냅샷으로 이동하여 복원하려는 스냅샷 옆에 있는 줄임표를 클릭하고 새 PVC로 복원을 클릭합니다.
2. 새 PVC의 세부 정보를 입력하고 복원을 클릭합니다. 이렇게 하면 새로운 PVC가 생성됩니다.

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. 다음으로, 이 PVC에서 새로운 VM을 만듭니다. 가상화 > 가상 머신으로 이동하여 만들기 > YAML 사용을 클릭합니다.
2. 사양 > 템플릿 > 사양 > 볼륨 섹션에서 컨테이너 디스크 대신 스냅샷에서 생성된 새 PVC를 지정합니다. 귀하의 요구 사항에 따라 새 VM에 대한 다른 모든 세부 정보를 제공합니다.

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

1. 새로운 VM을 만들려면 만들기를 클릭하세요.
2. VM이 성공적으로 생성된 후, 스냅샷이 생성된 시점에 PVC를 사용하여 스냅샷을 생성한 VM과 동일한 상태인지 액세스하여 확인합니다.

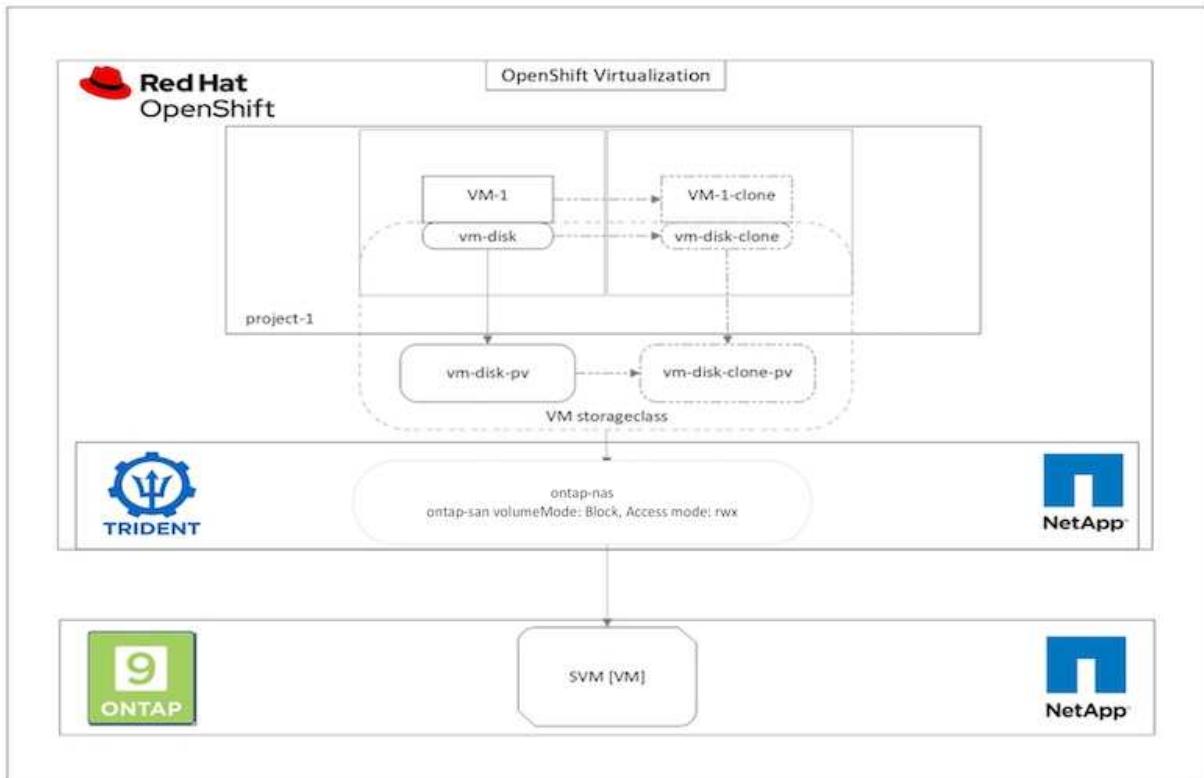
OpenShift Virtualization에서 Trident를 사용하여 VM 복제

이 절차에는 Trident CSI 볼륨 복제 기능을 활용하는 것이 포함되어 있어 원본 VM을 종료하여 새 VM을 생성하거나 원본 VM을 계속 실행 상태로 유지할 수 있습니다.

예를 보여주세요

VM 복제

OpenShift에서 기존 VM을 복제하는 작업은 Trident의 Volume CSI 복제 기능을 통해 수행됩니다. CSI 볼륨 클로닝을 사용하면 기존 PVC를 데이터 소스로 사용하여 PV를 복제하여 새로운 PVC를 생성할 수 있습니다. 새로운 PVC가 생성된 후에는 별도의 엔터티로 작동하며 소스 PVC에 대한 링크나 종속성이 없습니다.



CSI 볼륨 클로닝에는 고려해야 할 몇 가지 제한 사항이 있습니다.

1. 소스 PVC와 대상 PVC는 동일한 프로젝트에 있어야 합니다.
2. 동일한 스토리지 클래스 내에서 복제가 지원됩니다.
3. 소스 볼륨과 대상 볼륨이 동일한 VolumeMode 설정을 사용하는 경우에만 복제를 수행할 수 있습니다. 예를 들어, 블록 볼륨은 다른 블록 볼륨에만 복제할 수 있습니다.

OpenShift 클러스터의 VM은 두 가지 방법으로 복제할 수 있습니다.

1. 소스 VM을 종료하여
2. 소스 VM을 활성 상태로 유지하여

소스 **VM**을 종료하여

VM을 종료하여 기존 VM을 복제하는 것은 Trident 의 지원을 받아 구현된 기본 OpenShift 기능입니다. VM을 복제하려면 다음 단계를 완료하세요.

1. 워크로드 > 가상화 > 가상 머신으로 이동한 다음 복제하려는 가상 머신 옆에 있는 줄임표를 클릭합니다.

2. 가상 머신 복제를 클릭하고 새 VM에 대한 세부 정보를 제공합니다.

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. 가상 머신 복제를 클릭합니다. 그러면 소스 VM이 종료되고 복제 VM 생성이 시작됩니다.

4. 이 단계가 완료되면 복제된 VM의 콘텐츠에 액세스하여 확인할 수 있습니다.

원본 **VM**을 계속 실행 상태로 유지함으로써

소스 VM의 기존 PVC를 복제한 다음 복제된 PVC를 사용하여 새 VM을 만드는 방식으로 기존 VM을 복제할 수도 있습니다. 이 방법을 사용하면 소스 VM을 종료할 필요가 없습니다. VM을 종료하지 않고 복제하려면 다음 단계를 완료하세요.

1. 저장소 > 영구 볼륨 클레임으로 이동한 다음 소스 VM에 연결된 PVC 옆에 있는 줄임표를 클릭합니다.
2. PVC 복제를 클릭하고 새 PVC에 대한 세부 정보를 제공합니다.

Clone

Name *

rhel8-short-frog-rootdisk-28dvvb-clone

Access Mode *

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

Size *

20

GiB

PVC details

Namespace

 default

Requested capacity

20 GiB

Access mode

Shared Access (RWX)

Storage Class

 basic

Used capacity

2.2 GiB

Volume mode

Filesystem

Cancel

Clone

3. 그런 다음 복제를 클릭합니다. 이렇게 하면 새 VM에 대한 PVC가 생성됩니다.
4. 워크로드 > 가상화 > 가상 머신으로 이동한 후 만들기 > YAML 사용을 클릭합니다.
5. 사양 > 템플릿 > 사양 > 볼륨 섹션에서 컨테이너 디스크 대신 복제된 PVC를 첨부합니다. 귀하의 요구 사항에 따라 새 VM에 대한 다른 모든 세부 정보를 제공합니다.

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

1. 새로운 VM을 만들려면 만들기를 클릭하세요.
2. VM이 성공적으로 생성된 후 새 VM에 액세스하여 소스 VM의 복제본인지 확인합니다.

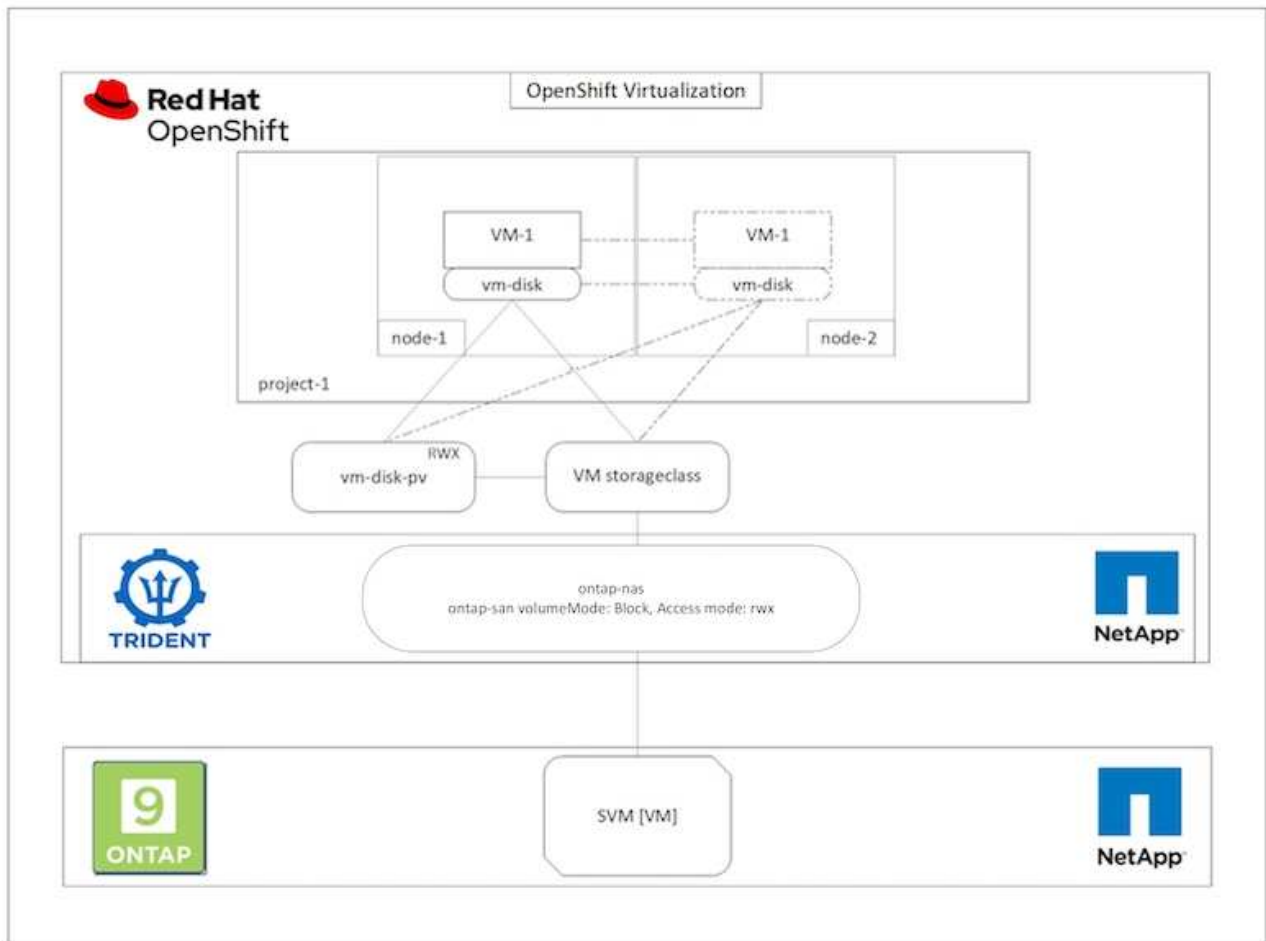
Red Hat OpenShift 클러스터의 두 노드 간에 VM 마이그레이션

가동 중지 시간 없이 클러스터의 두 노드 간에 OpenShift Virtualization에서 VM을 마이그레이션합니다. 이 절차에는 디스크가 RWX 호환 스토리지 클래스를 사용하는지 확인하고, 마이그레이션을 시작하고, 진행 상황을 모니터링하는 작업이 포함됩니다.

VM 라이브 마이그레이션

라이브 마이그레이션은 OpenShift 클러스터에서 가동 중지 시간 없이 VM 인스턴스를 한 노드에서 다른 노드로 마이그레이션하는 프로세스입니다. OpenShift 클러스터에서 라이브 마이그레이션이 작동하려면 VM이 공유 ReadWriteMany 액세스 모드를 사용하여 PVC에 바인딩되어야 합니다. `ontap-nas` 드라이버를 사용하여 구성된 Trident 백엔드는 파일 시스템 프로토콜 `nfs` 및 `smb`에 대한 RWX 액세스 모드를 지원합니다. 문서를 참조하세요 ["여기"](#). `ontap-san` 드라이버를 사용하여 구성된 Trident 백엔드는 iSCSI 및 NVMe/TCP 프로토콜의 블록 볼륨 모드에 대한 RWX 액세스 모드를 지원합니다. 문서를 참조하세요 ["여기"](#).

따라서 라이브 마이그레이션이 성공하려면 VM에 `ontap-nas` 또는 `ontap-san(volumeMode: Block)` 스토리지 클래스를 사용하여 PVC가 있는 디스크(부팅 디스크 및 추가 핫 플러그 디스크)를 프로비저닝해야 합니다. PVC가 생성되면 Trident NFS 또는 iSCSI가 활성화된 SVM에 ONTAP 볼륨을 생성합니다.

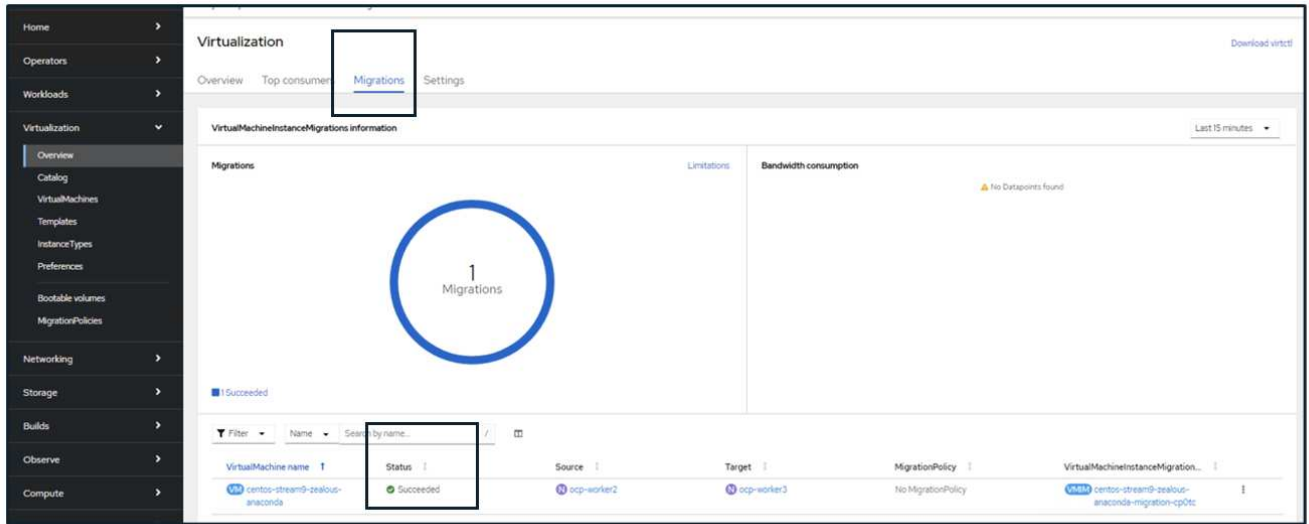
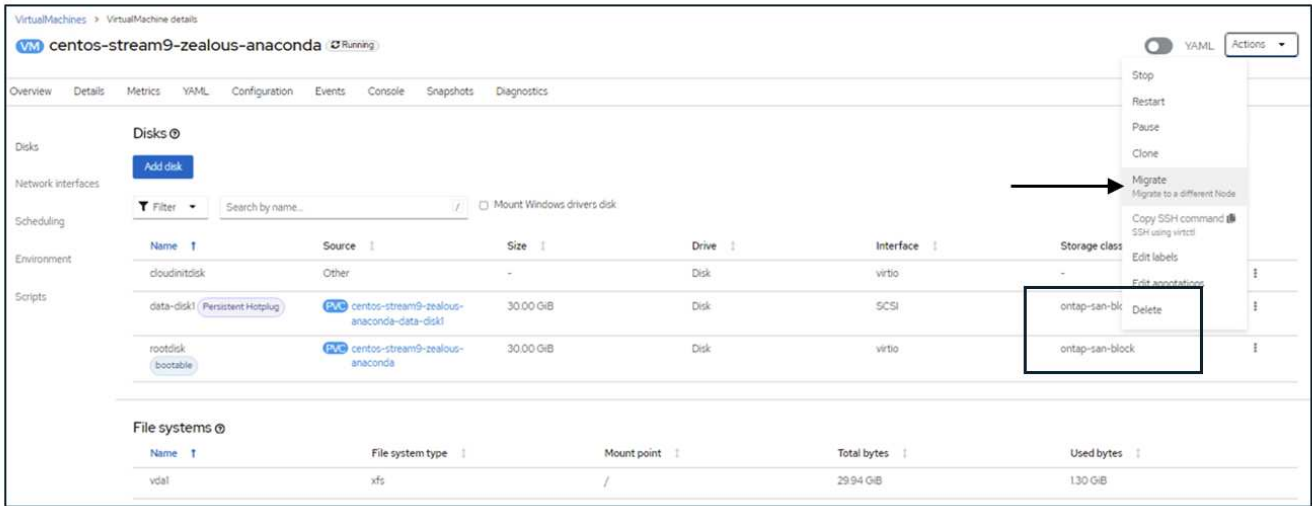


이전에 생성되어 실행 중인 VM의 라이브 마이그레이션을 수행하려면 다음 단계를 수행하세요.

1. 라이브 마이그레이션할 VM을 선택하세요.
2. 구성 탭을 클릭합니다.
3. 모든 VM 디스크가 RWX 액세스 모드를 지원할 수 있는 스토리지 클래스를 사용하여 생성되었는지 확인하세요.
4. 오른쪽 모서리에 있는 *작업*을 클릭한 다음 *마이그레이션*을 선택합니다.
5. 마이그레이션 진행 상황을 보려면 왼쪽 메뉴에서 가상화 > 개요로 이동한 다음 마이그레이션 탭을 클릭하세요. VM 마이그레이션은 *보류*에서 *예약*으로, *성공*으로 전환됩니다.



evictionStrategy가 LiveMigrate로 설정된 경우, 원래 노드가 유지 관리 모드로 전환되면 OpenShift 클러스터의 VM 인스턴스가 자동으로 다른 노드로 마이그레이션됩니다.



OpenShift Virtualization에서 스토리지 클래스 간 VM 스토리지 마이그레이션

OpenShift Virtualization에서는 가상 머신(VM)을 삭제하지 않고 웹 콘솔을 통해 VM의 디스크를 다른 스토리지 클래스로 직접 마이그레이션할 수 있습니다. 이 프로세스를 통해 전체 VM 스토리지 또는 특정 볼륨을 이동할 수 있습니다.

다음 단계는 가상 머신의 스토리지 클래스를 마이그레이션하는 방법을 설명합니다.

1. 대상 스토리지 클래스가 OpenShift 클러스터에서 올바르게 구성되어 있고 사용 가능한지 확인하십시오. 여기에는 필요한 스토리지 백엔드(예: NetApp ONTAP)가 설정되어 있고 액세스 가능한지 확인하는 것이 포함됩니다.

예를 보여주세요

```
[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. OpenShift Container Platform 웹 콘솔에서 가상화 관점으로 이동하여 마이그레이션할 VM을 선택합니다.

예를 보여주세요

VirtualMachines

VirtualMachine details

VM

rhel9-sc-migration

Running

Actions

Overview

Metrics

YAML

Configuration

Events

Console

Snapshots

Diagnostics

Details

Name

rhel9-sc-migration

Status

Running

Created

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

Operating system

Red Hat Enterprise Linux 9.7 (Plow)

CPU | Memory

1 CPU | 2 GiB Memory

Time zone

EDT

Template

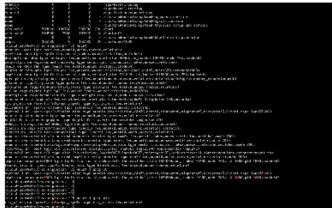
rhel9-server-small

Hostname

rhel9-sc-migration

VNC console

Open web console



Alerts (0)

General

Namespace

test-sc-migration

Node

00-50-56-b5-8d-fa

VirtualMachineInstance

rhel9-sc-migration

Pod

virt-launcher-rhel9-sc-migration-5jkw

Owner

No owner

Snapshots (0)

Take snapshot

VirtualMachines

VirtualMachine details

VM

rhel9-sc-migration

Running

YAML

Overview

Metrics

YAML

Configuration

Events

Console

Snapshots

Diagnostics

q

Details

Storage

Network

Scheduling

SSH

Initial run

Metadata

Disks

Add disk

Filter

Search by name...

Mount Windows drivers disk

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-bb64342f-c01f-453e-864d-a8f72e710a2f  53Gi        RWX              sc-nas          <unset>                  4m19s
persistentvolumeclaim/rhel9-migration                                Bound    pvc-46a34903-942f-4ddc-8e6d-974008f05d09  34144990004  RWX              sc-nas          <unset>                  4m19s
[root@00-50-56-b5-2b-9b ~]#
```

3. [작업] > [마이그레이션] > [스토리지 마이그레이션]을 클릭합니다. 마이그레이션할 볼륨과 대상 스토리지 클래스를 선택한 다음 [마이그레이션]을 클릭합니다. 마이그레이션 프로세스가 시작되며 VM의 이벤트 탭에서 진행 상황을 모니터링할 수 있습니다. 마이그레이션이 완료되면 VM의 디스크가 새 스토리지 클래스를 사용하게 됩니다.

예를 보여주세요

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
Migrate VirtualMachine to a different Node

Namespace Storage
Migrate VirtualMachine storage to a different StorageClass

Node VirtualMachineInstance
VM **rhel9-sc-migration**

Pod
virt-launcher-rhel9-sc-migration-Sjkdw

Owner
No owner

Snapshots (0) [Take snapshot](#)

Actions

- Stop
- Restart
- Pause
- Clone
- Take snapshot
- Migration >
- Copy SSH command SSH using virtctl
- Delete
The VirtualMachine is running

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. 마이그레이션이 완료되면 웹 콘솔에서 VM 세부 정보를 확인하거나 OpenShift CLI를 사용하여 VM 디스크가 새 스토리지 클래스를 사용하고 있는지 확인할 수 있습니다.

예를 보여주세요

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

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

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

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

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

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

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

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

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

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

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

```

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

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

```

```

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

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

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

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

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

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

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

```

```

[root@00-50-56-b5-2b-9b ~]# oc describe pv -n test-sc-migration pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
Name:          pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
Labels:        <none>
Annotations:   pv.kubernetes.io/provisioned-by: csi.trident.netapp.io
               volume.kubernetes.io/provisioner-deletion-secret-name:
               volume.kubernetes.io/provisioner-deletion-secret-namespace:
Finalizers:    [external-provisioner.volume.kubernetes.io/finalizer kubernetes.io/pv-protection external-attacher/csi-trident-netapp-io]
StorageClass:  sc-nas-economy
Status:        Bound
Claim:         test-sc-migration/dv-rhel9-migration-disk-emerald-wren-75-d88usc-mig-839zfb
Reclaim Policy: Delete
Access Modes:  Rwx
Volume Mode:   Filesystem
Capacity:       60322815673
Node Affinity: <none>
Message:
Source:
  Type:          CSI (a Container Storage Interface (CSI) volume source)
  Driver:         csi.trident.netapp.io
  FSType:         ext4
  VolumeHandle:   pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
  ReadOnly:       false
  VolumeAttributes:
    backendUUID=a59e8053-bb00-4410-9b12-1dc4d283928b
    internalID=/svm/oc-automation-svm/flexvol/trident_qtree_pool_oc_automation_OGBGMAIOLG/qtree/oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546...
    internalName=oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546dbdb3b
    name=pvc-75a885f0-824e-407c-ab29-8cc546dbdb3b
    protocol=file
    storage.kubernetes.io/csiProvisionerIdentity=1778505459781-8296-csi.trident.netapp.io
Events:        <none>

```

Qtrees

+ Add

🔍 ⬇️ 🗃️ ☰

☐	Name	Storage VM	Volume name	Mount path
	oc_automation_pvc_75a885f0_824e_407c_ab29_8ccf			
▼	oc_automation_pvc_75a885f0_824e_407c_ab29_8cc546dbdb3b	oc-automation-svm	trident_qtree_pool_oc_automation_OGBGMIAOLG	/trident_qtree_pool_oc_automation_OGBGMIAOLG/oc_...

저작권 정보

Copyright © 2026 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.