Red Hat OpenShift 클러스터에 Trident 설치하고 스토리지 객체를 생성합니다.

NetApp virtualization solutions

NetApp
August 18, 2025

This PDF was generated from https://docs.netapp.com/ko-kr/netapp-solutions-virtualization/openshift/osv-trident-install.html on August 18, 2025. Always check docs.netapp.com for the latest.

목차

Red Hat OpenShift 클러스터에 Trident 설치하고 스토리지 객체를 생성합니다. 1
비디오 데모 6
온프레미스 OpenShift 클러스터를 위한 Trident 구성 6
FSxN 스토리지를 사용한 ROSA 클러스터에 대한 Trident 구성 11
Trident 볼륨 스냅샷 클래스 생성 12
Trident Storage 및 Snapshot Class를 사용하여 기본값 설정 13

Red Hat OpenShift 클러스터에 Trident 설치하고 스토리지 객체를 생성합니다.

OpenShift 클러스터에 Red Hat Certified Trident Operator를 사용하여 Trident 설치하고 블록 액세스를 위해 작업자 노드를 준비합니다. ONTAP 및 FSxN 스토리지에 대한 Trident 백엔드 및 스토리지 클래스 객체를 생성하여 컨테이너 및 VM에 대한 동적 볼륨 프로비저닝을 활성화합니다.

> OpenShift Virtualization에서 VM을 생성해야 하는 경우 Trident 설치해야 하며 백엔드 개체와 스토리지 클래스 개체를 OpenShift Virtualization을 클러스터(온프레미스 및 ROSA)에 설치하기 전에 OpenShift 클러스터에 생성해야 합니다. 기본 스토리지 클래스와 기본 볼륨 스냅샷 클래스는 클러스터의 Trident 스토리지와 스냅샷 클래스로 설정해야 합니다. 이것이 구성된 경우에만 OpenShift Virtualization은 템플릿을 사용하여 VM을 생성하기 위해 골든 이미지를 로컬로 제공할 수 있습니다.

> Trident 설치하기 전에 OpenShift Virtualization operator가 설치되어 있는 경우 다음 명령을 사용하여 다른 스토리지 클래스를 사용하여 생성된 골든 이미지를 삭제한 다음, Trident 스토리지 및 볼륨 스냅샷 클래스 기본값이 설정되어 있는지 확인하여 OpenShift Virtualization이 Trident 스토리지 클래스를 사용하여 골든 이미지를 생성하도록 할 수 있습니다.

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

> ROSA 클러스터의 FSxN 스토리지에 대한 트라이던트 객체를 생성하기 위한 샘플 YAML 파일을 얻고, VolumeSnapshotClass에 대한 샘플 YAML 파일을 얻으려면 이 페이지를 아래로 스크롤하세요.

- Trident 설치**

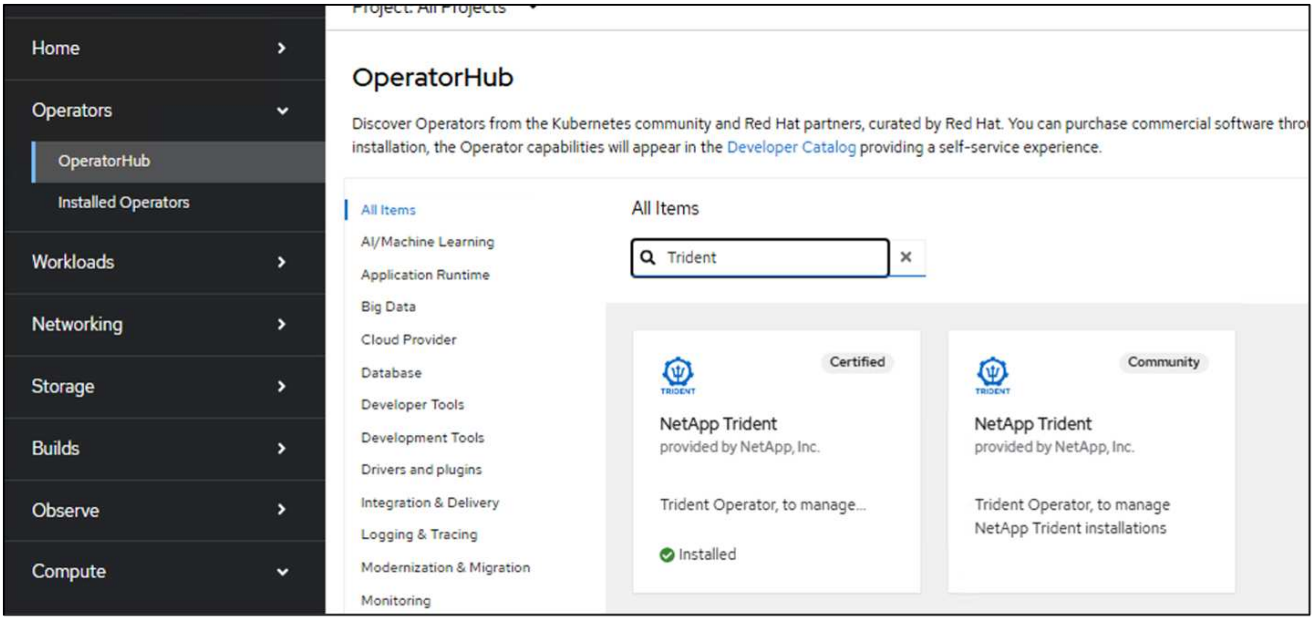
Red Hat Certified Operator를 사용하여 **Trident** 설치

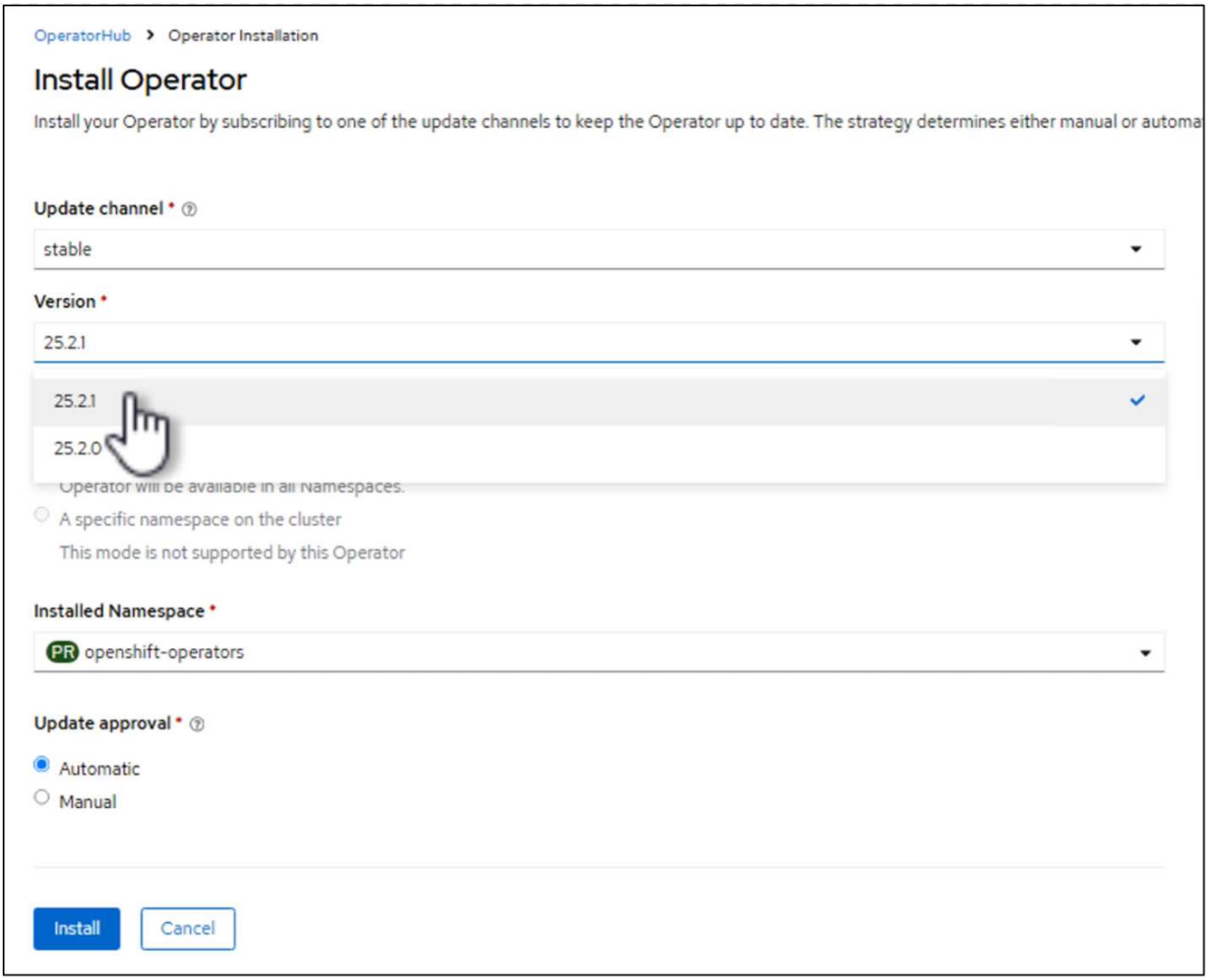
이 섹션에서는 Red Hat Certified Trident Operator를 사용하여 Trident 설치하는 방법에 대한 세부 정보를 제공합니다."Trident 문서를 참조하세요" Trident 설치하는 다른 방법은 여기를 참조하세요. Trident 25.02가 출시됨에 따라 온프레미스 및 클라우드의 Red Hat OpenShift에서 Trident 사용하는 사용자와 AWS의 Red Hat OpenShift Service와 같은 관리형 서비스를 사용하는 사용자는 이제 Operator Hub의 Trident Certified Operator를 사용하여 Trident 설치할 수 있습니다. Trident 이전에는 커뮤니티 운영자로만 제공되었기 때문에 이는 OpenShift 사용자 커뮤니티에 중요한 의미를 갖습니다.

Red Hat Certified Trident 운영자의 장점은 OpenShift와 함께 사용할 경우(온프레미스, 클라우드 또는 ROSA를 통한 관리형 서비스로 사용) NetApp 에서 운영자와 컨테이너의 기반을 완벽하게 지원한다는 것입니다. 또한 NetApp Trident 고객에게 무료로 제공되므로 Red Hat OpenShift와 원활하게 작동하도록 검증된 인증 운영자를 사용하여 설치하기만 하면 되며 간편한 수명 주기 관리를 위해 패키지로 제공됩니다.

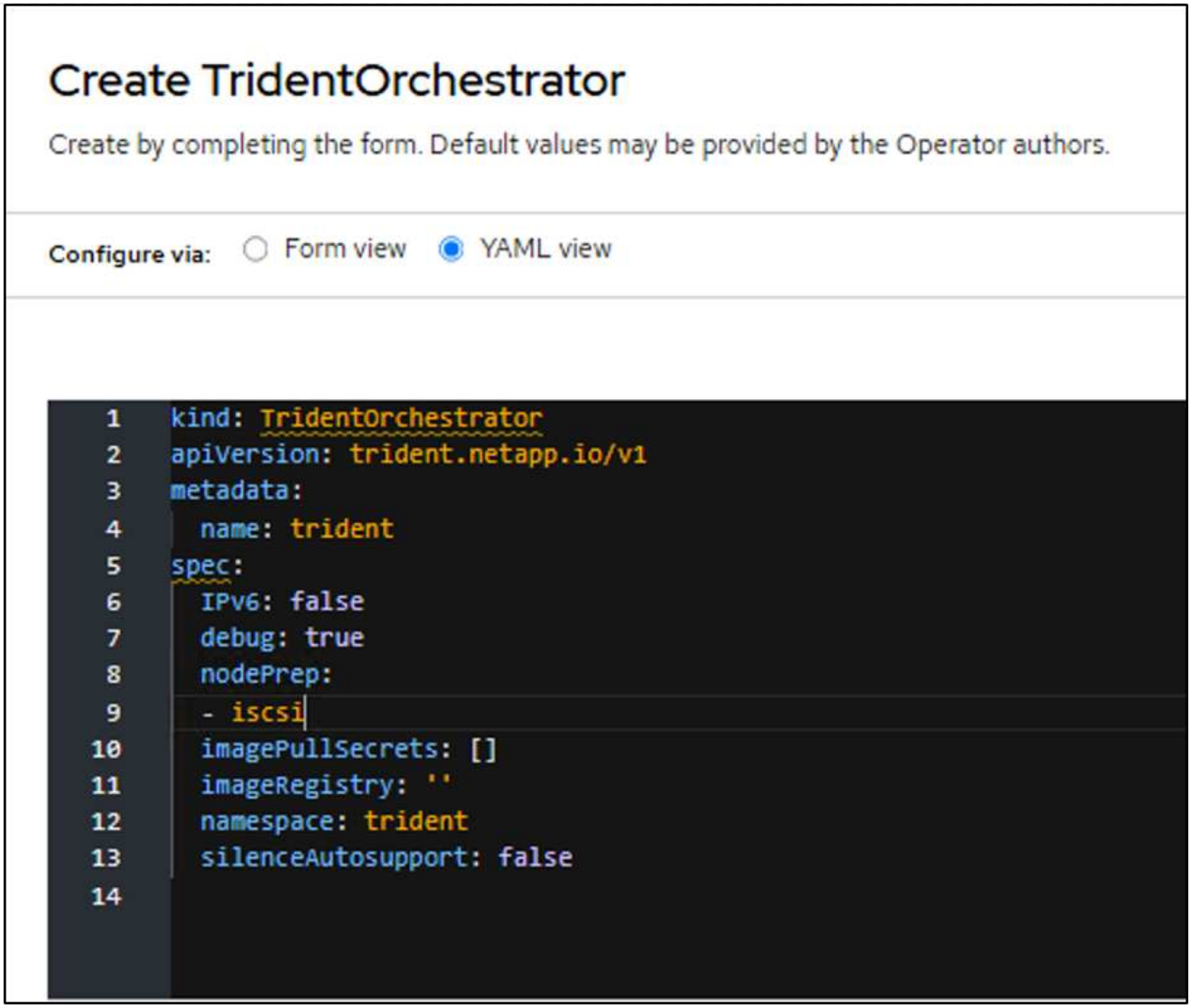
또한, Trident 25.02 운영자(및 이후 버전)는 iSCSI를 위한 작업자 노드를 준비하는 선택적 이점을 제공합니다. 특히 ROSA 클러스터에 워크로드를 배포하고 FSxN과 함께 iSCSI 프로토콜을 사용하려는 경우 이 기능이 매우 유용합니다. 특히 OpenShift Virtualization VM 워크로드에 유용합니다. FSxN을 사용하여 ROSA 클러스터에서 iSCSI를 위한 워커 노드를 준비하는 과제는 클러스터에 Trident 설치하면 이 기능으로 완화됩니다.

운영자를 사용하는 설치 단계는 온프레미스 클러스터에 설치하든 ROSA에 설치하든 동일합니다. Operator를 사용하여 Trident 설치하려면 Operator 허브를 클릭하고 Certified NetApp Trident 선택하세요. 설치 페이지에서는 기본적으로 최신 버전이 선택되어 있습니다. 설치를 클릭하세요
.





운영자가 설치되면 운영자 보기를 클릭한 다음 Trident Orchestrator의 인스턴스를 만듭니다. iSCSI 스토리지 액세스를 위해 작업자 노드를 준비하려면 YAML 보기로 이동하여 iscsi를 추가하여 nodePrep 매개변수를 수정합니다.



이제 클러스터에서 모든 트라이던트 포드가 실행 중이어야 합니다

.

```
[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-mkmmp              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 ~]#
```

OpenShift 클러스터의 작업자 노드에서 iSCSI 도구가 활성화되었는지 확인하려면 작업자 노드에 로그인하여 표시된 대로 iscsid, multipathd active 및 multipath.conf 파일의 항목을 확인하세요.

```
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#
```

```
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#
```

비디오 데모

다음 비디오는 Red Hat Certified Trident Operator를 사용하여 Trident 설치하는 방법을 보여줍니다.

OpenShift에서 인증된 Trident Operator를 사용하여 Trident 25.02.1 설치

온프레미스 **OpenShift** 클러스터를 위한 **Trident** 구성

NAS용 Trident 백엔드 및 스토리지 클래스

```
cat 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: <cluster management lif>
  backendName: tbc-nas
  svm: zoneb
  storagePrefix: testzoneb
  defaults:
    nameTemplate: "{{ .config.StoragePrefix }}_{{ .volume.Namespace
}}_{{ .volume.RequestName }}"
  credentials:
    name: tbc-nas-secret
```

```
cat 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
```

iSCSI를 위한 Trident 백엔드 및 스토리지 클래스

```yaml
# cat 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
```

```yaml
# cat 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/TCP를 위한 Trident 백엔드 및 스토리지 클래스

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

```yaml
# cat 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
```

FC용 Trident 백엔드 및 스토리지 클래스

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

```
# cat 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
```

FSxN 스토리지를 사용한 ROSA 클러스터에 대한 Trident 구성

FSxN NAS를 위한 Trident 백엔드 및 스토리지 클래스

```yaml
#cat tbc-fsx-nas.yaml
apiVersion: v1
kind: Secret
metadata:
  name: backend-fsx-ontap-nas-secret
  namespace: trident
type: Opaque
stringData:
  username: <cluster admin lif>
  password: <cluster admin passwd>
---
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-fsx-ontap-nas
  namespace: trident
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
```

```yaml
# cat sc-fsx-nas.yaml
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
```

FSxN iSCSI를 위한 Trident 백엔드 및 스토리지 클래스

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

```
# cat 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
```

Trident 볼륨 스냅샷 클래스 생성

Trident 볼륨 스냅샷 클래스

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

백엔드 구성, 스토리지 클래스 구성 및 스냅샷 구성에 필요한 yaml 파일을 준비한 후 다음 명령을 사용하여 trident 백엔드, 스토리지 클래스 및 스냅샷 클래스 객체를 생성할 수 있습니다.

```
oc create -f <backend-filename.yaml> -n trident
oc create -f < storageclass-filename.yaml>
oc create -f <snapshotclass-filename.yaml>
```

Trident Storage 및 Snapshot Class를 사용하여 기본값 설정

Trident Storage 및 Snapshot Class를 사용하여 기본값 설정

이제 OpenShift 클러스터에서 필수 Trident 스토리지 클래스와 볼륨 스냅샷 클래스를 기본값으로 만들 수 있습니다. 앞서 언급했듯이 기본 스토리지 클래스와 볼륨 스냅샷 클래스를 설정해야 OpenShift Virtualization에서 골든 이미지 소스를 사용하여 기본 템플릿에서 VM을 생성할 수 있습니다.

콘솔에서 주석을 편집하거나 다음 명령어를 사용하여 명령줄에서 패치를 적용하여 Trident 스토리지 클래스와 스냅샷 클래스를 기본값으로 설정할 수 있습니다.

```
storageclass.kubernetes.io/is-default-class:true
or
kubectl patch storageclass standard -p '{"metadata": {"annotations":{
"storageclass.kubernetes.io/is-default-class":"true"}}}'

storageclass.kubevirt.io/is-default-virt-class: true
or
kubectl patch storageclass standard -p '{"metadata": {"annotations":{
"storageclass.kubevirt.io/is-default-virt-class": "true"}}}'
```

이것이 설정되면 다음 명령을 사용하여 기존 dv 및 VolumeSnapShot 개체를 삭제할 수 있습니다.

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

저작권 정보

Copyright © 2025 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 라이센스와 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 라이센스도 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 라이센스를 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 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

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