



Trident 업그레이드

Trident

NetApp
July 01, 2026

목차

| | |
|---|---|
| Trident 업그레이드 | 1 |
| Trident 업그레이드 | 1 |
| 업그레이드 전 고려 사항 | 1 |
| 1단계: 버전 선택 | 1 |
| 2단계: 원래 설치 방법을 확인합니다 | 1 |
| 3단계: 업그레이드 방법 선택 | 2 |
| 운영자를 통해 업그레이드 | 2 |
| 운영자 업그레이드 워크플로 이해 | 2 |
| Trident Operator 또는 Helm을 사용하여 Trident 설치를 업그레이드하세요 | 3 |
| tridentctl로 업그레이드합니다 | 7 |

Trident 업그레이드

Trident 업그레이드

24.02 릴리스부터 Trident는 4개월 주기로 릴리스를 진행하며, 매년 세 번의 주요 릴리스를 제공합니다. 각 새 릴리스는 이전 릴리스를 기반으로 새로운 기능, 성능 향상, 버그 수정 및 개선 사항을 제공합니다. Trident의 새로운 기능을 활용하려면 1년에 한 번 이상 업그레이드하는 것이 좋습니다.

업그레이드 전 고려 사항

Trident의 최신 릴리스로 업그레이드할 때 다음 사항을 고려하십시오.

- 주어진 Kubernetes 클러스터 내의 모든 네임스페이스에는 하나의 Trident 인스턴스만 설치되어야 합니다.
- Trident 23.07 이상에서는 v1 볼륨 스냅샷이 필요하며 알파 또는 베타 스냅샷은 더 이상 지원되지 않습니다.
- 업그레이드할 때, Trident에서 사용하는 `parameter.fsType`에 `StorageClasses`를 제공하는 것이 중요합니다. 기존 볼륨에 영향을 주지 않고 `StorageClasses`를 삭제하고 다시 생성할 수 있습니다.
 - 이는 SAN 볼륨에 대한 ["보안 컨텍스트"](#) 적용을 위한 필수 요건입니다.
 - [샘플 입력](#) 디렉토리에는 `storage-class-basic.yaml.template1` 및 `storage-class-bronze-default.yaml`와 같은 예제가 포함되어 있습니다.
 - 자세한 내용은 ["알려진 문제"](#)를 참조하십시오.

1단계: 버전 선택

Trident 버전은 날짜 기반 YY.MM 명명 규칙을 따르며, 여기서 "YY"는 연도의 마지막 두 자리 숫자이고 "MM"은 월을 나타냅니다. 도트 릴리스는 YY.MM.X 규칙을 따르며, 여기서 "X"는 패치 레벨을 나타냅니다. 업그레이드할 버전은 업그레이드하려는 현재 버전을 기준으로 선택합니다.

- 설치된 버전에서 4개 릴리스 이내의 대상 릴리스로 직접 업그레이드할 수 있습니다. 예를 들어 24.06(또는 24.06의 모든 마이너 버전)에서 25.06으로 직접 업그레이드할 수 있습니다.
- 4개 릴리스 기간 외의 버전에서 업그레이드하는 경우 여러 단계를 거쳐 업그레이드하십시오. ["이전 버전"](#)에서 업그레이드하는 경우 해당 버전의 업그레이드 지침을 사용하여 4개 릴리스 기간에 맞는 최신 릴리스로 업그레이드하십시오. 예를 들어 23.07 버전을 실행 중이고 25.06으로 업그레이드하려는 경우:
 - a. 먼저 23.07에서 24.06으로 업그레이드하십시오.
 - b. 그다음 24.06에서 25.06으로 업그레이드하세요.



OpenShift Container Platform에서 Trident 오퍼레이터를 사용하여 업그레이드할 경우 Trident 21.01.1 이상으로 업그레이드해야 합니다. 21.01.0과 함께 릴리스된 Trident 오퍼레이터에는 알려진 문제가 있으며 이 문제는 21.01.1에서 수정되었습니다. 자세한 내용은 ["GitHub의 문제 세부 정보"](#)를 참조하십시오.

2단계: 원래 설치 방법을 확인합니다

Trident를 원래 설치하는 데 사용한 버전을 확인하려면:

1. `kubectl get pods -n trident`를 사용하여 Pod를 검사합니다.
 - 오퍼레이터 포드가 없는 경우 Trident가 `tridentctl`을 사용하여 설치되었습니다.
 - 오퍼레이터 포드가 있는 경우, Trident는 수동으로 또는 Helm을 사용하여 Trident 오퍼레이터를 통해 설치되었습니다.
2. 운영자 포드가 있는 경우 `kubectl describe torc`를 사용하여 Helm을 통해 Trident가 설치되었는지 확인합니다.
 - Helm 레이블이 있는 경우 Trident는 Helm을 사용하여 설치되었습니다.
 - Helm 레이블이 없으면 Trident 운영자를 사용하여 Trident를 수동으로 설치한 것입니다.

3단계: 업그레이드 방법 선택

일반적으로 초기 설치 시 사용했던 방법과 동일한 방법으로 업그레이드하는 것이 좋지만, "설치 방법 간 이동". Trident를 업그레이드하는 두 가지 옵션이 있습니다.

- "Trident 운영자를 사용하여 업그레이드하세요"



운영자와 함께 업그레이드하기 전에 "운영자 업그레이드 워크플로 이해"을(를) 검토하는 것이 좋습니다.

*

운영자를 통해 업그레이드

운영자 업그레이드 워크플로 이해

Trident 오퍼레이터를 사용하여 Trident를 업그레이드하기 전에 업그레이드 중에 발생하는 백그라운드 프로세스를 이해해야 합니다. 여기에는 롤링 업데이트를 가능하게 하는 Trident 컨트롤러, 컨트롤러 Pod 및 노드 Pod, 노드 DaemonSet의 변경 사항이 포함됩니다.

Trident 운영자 업그레이드 처리

Trident를 설치 및 업그레이드하는 많은 "Trident 운영자 사용의 이점" 중 하나는 기존에 마운트된 볼륨에 영향을 주지 않고 Trident 및 Kubernetes 객체를 자동으로 처리한다는 점입니다. 이러한 방식으로 Trident는 다운타임 없이 업그레이드를 지원할 수 있습니다("롤링 업데이트"). 특히, Trident 오퍼레이터는 다음을 위해 Kubernetes 클러스터와 통신합니다.

- Trident Controller 배포 및 노드 DaemonSet을 삭제하고 다시 생성합니다.
- Trident Controller Pod와 Trident Node Pod를 새 버전으로 교체하십시오.
 - 노드가 업데이트되지 않더라도 나머지 노드의 업데이트가 차단되지 않습니다.
 - 실행 중인 Trident Node Pod가 있는 노드만 볼륨을 마운트할 수 있습니다.



Kubernetes 클러스터의 Trident 아키텍처에 대한 자세한 내용은 "Trident 아키텍처"을 참조하십시오.

Operator 업그레이드 워크플로

Trident 운영자를 사용하여 업그레이드를 시작할 때:

1. **Trident** 운영자:
 - a. 현재 설치된 Trident 버전(n)을 감지합니다.
 - b. CRD, RBAC 및 Trident SVC를 포함한 모든 Kubernetes 객체를 업데이트합니다.
 - c. 버전 $_n$ 에 대한 Trident Controller 배포를 삭제합니다.
 - d. 버전 $_n+1$ 에 대한 Trident Controller 배포를 생성합니다.
2. *Kubernetes*는 $_n+1$ 에 대한 Trident Controller Pod를 생성합니다.
3. **Trident** 운영자:
 - a. $_n$ 에 대한 Trident 노드 DaemonSet을 삭제합니다. 운영자는 노드 Pod 종료를 기다리지 않습니다.
 - b. $_n+1$ 에 대한 Trident 노드 데몬셋을 생성합니다.
4. *Kubernetes*는 Trident Node Pod $_n$ 이 실행 중이지 않은 노드에 Trident Node Pod을 생성합니다. 이를 통해 노드에는 어떤 버전이든 Trident Node Pod이 두 개 이상 존재하지 않게 됩니다.

Trident Operator 또는 Helm을 사용하여 Trident 설치를 업그레이드하세요

Trident 운영자를 사용하여 수동으로 또는 Helm을 통해 Trident를 업그레이드할 수 있습니다. Trident 운영자 설치에서 다른 Trident 운영자 설치로 업그레이드하거나 `tridentctl` 설치에서 Trident 운영자 버전으로 업그레이드할 수 있습니다. Trident 운영자 설치를 업그레이드하기 전에 ["업그레이드 방법을 선택하세요"](#)를 검토하십시오.

수동 설치 업그레이드

클러스터 범위 Trident 운영자 설치에서 다른 클러스터 범위 Trident 운영자 설치로 업그레이드할 수 있습니다. 모든 Trident 버전은 클러스터 범위 운영자를 사용합니다.



네임스페이스 범위 연산자를 사용하여 설치된 Trident(버전 20.07~20.10)에서 업그레이드하려면 ["설치된 버전"](#) Trident의 업그레이드 지침을 사용하십시오.

이 작업 정보

Trident는 Kubernetes 버전에 맞는 오퍼레이터를 설치하고 관련 객체를 생성하는 데 사용할 수 있는 번들 파일을 제공합니다.

- Kubernetes 1.25 이상 버전을 실행하는 클러스터의 경우 ["bundle_post_1_25.yaml"](#)를 사용하십시오.

시작하기 전에

["지원되는 Kubernetes 버전"](#)을(를) 실행하는 Kubernetes 클러스터를 사용하고 있는지 확인하십시오.

단계

1. Trident 버전을 확인하세요:

```
./tridentctl -n trident version
```

- operator.yaml, tridentorchestrator_cr.yaml, `post_1_25_bundle.yaml`를 업그레이드할 버전(예: 25.06)에 맞는 레지스트리 및 이미지 경로와 올바른 시크릿으로 업데이트하십시오.
- 현재 Trident 인스턴스를 설치하는 데 사용된 Trident 오퍼레이터를 삭제하십시오. 예를 들어 25.02에서 업그레이드하는 경우 다음 명령을 실행하십시오.

```
kubectl delete -f 25.02.0/trident-installer/deploy/<bundle.yaml> -n trident
```

- TridentOrchestrator 속성을 사용하여 초기 설치를 사용자 지정한 경우 TridentOrchestrator 객체를 편집하여 설치 매개 변수를 수정할 수 있습니다. 여기에는 오프라인 모드용 미러링된 Trident 및 CSI 이미지 레지스트리 지정, 디버그 로그 활성화 또는 이미지 풀 시크릿 지정을 위한 변경 사항이 포함될 수 있습니다.
- Kubernetes 버전에 따라 _<bundle.yaml>_가 bundle_pre_1_25.yaml 또는 `bundle_post_1_25.yaml`인 환경에 맞는 번들 YAML 파일을 사용하여 Trident를 설치합니다. 예를 들어 Trident 25.06.0을 설치하는 경우 다음 명령을 실행합니다.

```
kubectl create -f 25.06.0/trident-installer/deploy/<bundle.yaml> -n trident
```

- Trident torc를 수정하여 이미지 25.06.0을 포함시키세요.

Helm 설치 업그레이드

Trident Helm 설치를 업그레이드할 수 있습니다.



Trident가 설치된 Kubernetes 클러스터를 1.24에서 1.25 이상 버전으로 업그레이드할 때는 클러스터를 업그레이드하기 전에 values.yaml을 업데이트하여 excludePodSecurityPolicy`을 `true`로 설정하거나 `--set excludePodSecurityPolicy=true`을 `helm upgrade` 명령에 추가해야 합니다.

Kubernetes 클러스터를 1.24에서 1.25로 업그레이드했지만 Trident helm을 업그레이드하지 않은 경우 helm 업그레이드가 실패합니다. helm 업그레이드를 성공적으로 완료하려면 다음 단계를 사전 수행해야 합니다.

- <https://github.com/helm/helm-mapkubeapis>에서 helm-mapkubeapis 플러그인을 설치합니다.
- Trident가 설치된 네임스페이스에서 Trident 릴리스에 대한 사전 실행(dry run)을 수행합니다. 이렇게 하면 정리될 리소스 목록이 표시됩니다.

```
helm mapkubeapis --dry-run trident --namespace trident
```

- 정리 작업을 위해 helm을 사용하여 전체 실행을 수행하십시오.

```
helm mapkubeapis trident --namespace trident
```

단계

1. "Helm을 사용하여 Trident를 설치했습니다"한 경우 `helm upgrade trident netapp-trident/trident-operator --version 100.2602.0`를 사용하여 한 단계로 업그레이드할 수 있습니다. Helm 리포지토리를 추가하지 않았거나 업그레이드에 사용할 수 없는 경우:
 - a. 최신 Trident 릴리스를 "GitHub의 Assets 섹션"에서 다운로드하십시오.
 - b. `helm upgrade` 명령을 사용합니다. 여기서 `trident-operator-26.02.0.tgz`는 업그레이드할 버전을 나타냅니다.

```
helm upgrade <name> trident-operator-26.02.0.tgz
```



초기 설치 중에 사용자 지정 옵션(예: Trident 및 CSI 이미지에 대한 개인 미러링 레지스트리 지정)을 설정한 경우 `helm upgrade` 명령을 `--set`를 사용하여 추가하여 해당 옵션이 업그레이드 명령에 포함되도록 해야 합니다. 그렇지 않으면 값이 기본값으로 재설정됩니다.

2. `helm list`를 실행하여 차트와 앱 버전이 모두 업그레이드되었는지 확인합니다. `tridentctl logs`를 실행하여 디버그 메시지를 검토합니다.

tridentctl 설치에서 Trident operator로 업그레이드

`tridentctl` 설치에서 Trident 운영자의 최신 릴리스로 업그레이드할 수 있습니다. 기존 백엔드와 PVC는 자동으로 사용할 수 있습니다.



설치 방법을 전환하기 전에 "설치 방법 간 이동"을(를) 검토하십시오.

단계

1. 최신 Trident 릴리스를 다운로드하세요.

```
# Download the release required [26.02.0]
mkdir 26.02.0
cd 26.02.0
wget
https://github.com/NetApp/trident/releases/download/v26.02.0/trident-
installer-26.02.0.tar.gz
tar -xf trident-installer-26.02.0.tar.gz
cd trident-installer
```

2. 매니페스트에서 `tridentorchestrator` CRD를 생성합니다.

```
kubectl create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
```

3. 클러스터 범위 오퍼레이터를 동일한 네임스페이스에 배포합니다.

```
kubectl create -f deploy/<bundle-name.yaml>

serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-controller-79df798bdc-m79dc 6/6     Running   0           150d
trident-node-linux-xrst8             2/2     Running   0           150d
trident-operator-5574dbbc68-nthjv    1/1     Running   0           1m30s
```

4. Trident 설치를 위한 TridentOrchestrator CR을 생성합니다.

```
cat deploy/crds/tridentorchestrator_cr.yaml
apiVersion: trident.netapp.io/v1
kind: TridentOrchestrator
metadata:
  name: trident
spec:
  debug: true
  namespace: trident

kubectl create -f deploy/crds/tridentorchestrator_cr.yaml

#Examine the pods in the Trident namespace
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-79df798bdc-m79dc        6/6     Running   0           1m
trident-csi-xrst8                   2/2     Running   0           1m
trident-operator-5574dbbc68-nthjv    1/1     Running   0           5m41s
```

5. Trident가 의도한 버전으로 업그레이드되었는지 확인하십시오.

```
kubectl describe torc trident | grep Message -A 3
```

```
Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v26.02.0
```

tridentctl로 업그레이드합니다

`tridentctl`를 사용하여 기존 Trident 설치를 쉽게 업그레이드할 수 있습니다.

이 작업 정보

Trident를 제거했다가 다시 설치하는 것은 업그레이드와 같습니다. Trident를 제거해도 Trident 배포에서 사용되는 영구 볼륨 클레임(PVC) 및 영구 볼륨(PV)은 삭제되지 않습니다. 이미 프로비저닝된 PV는 Trident가 오프라인 상태인 동안에도 계속 사용 가능하며, Trident는 온라인 상태로 돌아온 후 그 사이에 생성된 모든 PVC에 대해 볼륨을 프로비저닝합니다.

시작하기 전에

업그레이드하기 전에 "[업그레이드 방법을 선택하세요](#)"를 검토한 후 `tridentctl`를 사용하십시오.

단계

1. `tridentctl`에서 제거 명령을 실행하여 CRD 및 관련 개체를 제외한 Trident와 관련된 모든 리소스를 제거합니다.

```
./tridentctl uninstall -n <namespace>
```

2. Trident를 다시 설치합니다. 다음을 참조하십시오. "[tridentctl을 사용하여 Trident를 설치합니다](#)".



업그레이드 프로세스를 중단하지 마십시오. 설치 프로그램이 완료될 때까지 실행되는지 확인하십시오.

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.