



Trident 업그레이드

Trident

NetApp
March 04, 2026

목차

Trident 업그레이드	1
Trident 업그레이드	1
업그레이드 전 고려 사항	1
1단계: 버전 선택	1
2단계: 원래 설치 방법 확인	2
3단계: 업그레이드 방법 선택	2
운영자와 함께 업그레이드하세요	2
운영자 업그레이드 워크플로 이해	2
Trident Operator 또는 Helm을 사용하여 Trident 설치 업그레이드	3
tridentctl로 업그레이드하세요	7

Trident 업그레이드

Trident 업그레이드

Trident 24.02 릴리스부터 4개월 릴리스 주기를 따르며, 매년 3개의 주요 릴리스를 제공합니다. 각각의 새로운 릴리스는 이전 릴리스를 기반으로 하며 새로운 기능, 성능 향상, 버그 수정 및 개선 사항을 제공합니다. Trident의 새로운 기능을 활용하려면 최소한 1년에 한 번 업그레이드하는 것이 좋습니다.

업그레이드 전 고려 사항

Trident의 최신 릴리스로 업그레이드할 때 다음 사항을 고려하세요.

- 주어진 Kubernetes 클러스터의 모든 네임스페이스에 Trident 인스턴스는 하나만 설치되어야 합니다.
- Trident 23.07 이상에서는 v1 볼륨 스냅샷이 필요하며 더 이상 알파 또는 베타 스냅샷을 지원하지 않습니다.
- Google Cloud에 대한 Cloud Volumes Service 생성한 경우 "[CVS 서비스 유형](#)", 백엔드 구성을 업데이트하여 사용해야 합니다. `standardsw` 또는 `zoneredundantstandardsw` Trident 23.01에서 업그레이드할 때의 서비스 수준. 업데이트 실패 `serviceLevel` 백엔드에서는 볼륨이 실패할 수 있습니다. 참조하다 "[CVS 서비스 유형 샘플](#)" 자세한 내용은.
- 업그레이드할 때 다음을 제공하는 것이 중요합니다. `parameter.fsType` ~에 `StorageClasses` Trident에서 사용됨. 삭제하고 다시 만들 수 있습니다 `StorageClasses` 기존 볼륨을 방해하지 않고.
 - 이는 시행을 위한 요구사항입니다. "[보안 컨텍스트](#)" SAN 볼륨의 경우.
 - <https://github.com/NetApp/trident/tree/master/trident-installer/sample-input> [sample input^] 디렉토리에는 <https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-basic.yaml.template> 과 같은 예가 포함되어 있습니다.[storage-class-basic.yaml.template] 및 링크: `storage-class-bronze-default.yaml` .
 - 자세한 내용은 다음을 참조하세요. "[알려진 문제](#)" .

1단계: 버전 선택

Trident 버전은 날짜 기반을 따릅니다. YY.MM 명명 규칙에 따르면 "YY"는 연도의 마지막 두 자리이고 "MM"은 월입니다. Dot 릴리스는 다음을 따릅니다. YY.MM.X 여기서 "X"는 패치 수준입니다. 업그레이드할 버전은 업그레이드하려는 버전에 따라 선택됩니다.

- 설치된 버전의 4개 릴리스 창 내에 있는 모든 대상 릴리스로 직접 업그레이드를 수행할 수 있습니다. 예를 들어, 24.06(또는 모든 24.06 dot 릴리스)에서 25.06으로 직접 업그레이드할 수 있습니다.
- 4개 릴리스 창 이외의 릴리스에서 업그레이드하는 경우 여러 단계로 업그레이드하세요. 업그레이드 지침을 사용하세요. "[이전 버전](#)" 4개 릴리스 창에 맞는 최신 릴리스로 업그레이드합니다. 예를 들어, 23.07을 실행 중이고 25.06으로 업그레이드하려는 경우:
 - a. 23.07에서 24.06으로 첫 번째 업그레이드.
 - b. 그런 다음 24.06에서 25.06으로 업그레이드하세요.



OpenShift Container Platform에서 Trident 연산자를 사용하여 업그레이드하는 경우 Trident 21.01.1 이상으로 업그레이드해야 합니다. 21.01.0과 함께 출시된 Trident 연산자에는 21.01.1에서 수정된 알려진 문제가 포함되어 있습니다. 자세한 내용은 다음을 참조하세요. "[GitHub의 이슈 세부 정보](#)".

2단계: 원래 설치 방법 확인

Trident 처음 설치하는 데 사용한 버전을 확인하려면 다음을 수행하세요.

1. 사용 `kubectl get pods -n trident` 꼬투리를 조사하기 위해서.
 - 운영자 포드가 없는 경우 Trident 설치되었습니다. `tridentctl`.
 - 운영자 포드가 있는 경우, Trident 운영자를 수동으로 사용하거나 Helm을 사용하여 Trident 설치했습니다.
2. 운영자 포드가 있는 경우 사용하세요 `kubectl describe torc` Helm을 사용하여 Trident 설치되었는지 확인합니다.
 - Helm 레이블이 있으면 Helm을 사용하여 Trident 설치되었다는 의미입니다.
 - Helm 레이블이 없으면 Trident 연산자를 사용하여 Trident 수동으로 설치되었습니다.

3단계: 업그레이드 방법 선택

일반적으로 초기 설치에 사용한 것과 동일한 방법을 사용하여 업그레이드해야 하지만 "[설치 방법 간 이동](#)". Trident 업그레이드하는 데는 두 가지 옵션이 있습니다.

- "[Trident 연산자를 사용하여 업그레이드](#)"



검토해 보시기 바랍니다 "[운영자 업그레이드 워크플로 이해](#)" 운영자와 함께 업그레이드하기 전에.

*

운영자와 함께 업그레이드하세요

운영자 업그레이드 워크플로 이해

Trident 운영자를 사용하여 Trident 업그레이드하기 전에 업그레이드 중에 발생하는 백그라운드 프로세스를 이해해야 합니다. 여기에는 롤링 업데이트를 활성화하는 Trident 컨트롤러, 컨트롤러 Pod 및 노드 Pod, 노드 DaemonSet에 대한 변경 사항이 포함됩니다.

Trident 운영자 업그레이드 처리

많은 것 중 하나 "[Trident 연산자 사용의 이점](#)" Trident 설치하고 업그레이드하면 기존에 마운트된 볼륨을 방해하지 않고 Trident 와 Kubernetes 객체를 자동으로 처리할 수 있습니다. 이러한 방식으로 Trident 다운타임 없이 업그레이드를 지원할 수 있습니다. "[롤링 업데이트](#)". 특히, Trident 운영자는 Kubernetes 클러스터와 통신하여 다음을 수행합니다.

- Trident Controller 배포 및 노드 DaemonSet을 삭제하고 다시 생성합니다.
- Trident Controller Pod와 Trident Node Pod를 새 버전으로 교체하세요.
 - 노드가 업데이트되지 않더라도 나머지 노드가 업데이트되는 것은 방해가 되지 않습니다.

- Trident Node Pod가 실행 중인 노드만 볼륨을 마운트할 수 있습니다.



Kubernetes 클러스터의 Trident 아키텍처에 대한 자세한 내용은 다음을 참조하세요. "[Trident 아키텍처](#)"

운영자 업그레이드 워크플로

Trident 연산자를 사용하여 업그레이드를 시작하는 경우:

1. * Trident 연산자*:
 - a. 현재 설치된 Trident 버전(버전 n)을 감지합니다.
 - b. CRD, RBAC, 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 연산자*:
 - a. $_n$ 에 대한 Trident Node DaemonSet을 삭제합니다. 운영자는 Node Pod 종료를 기다리지 않습니다.
 - b. $_n+1$ 에 대한 Trident Node Daemonset을 생성합니다.
4. *Kubernetes*는 Trident Node Pod $_n$ 을 실행하지 않는 노드에 Trident Node Pod를 생성합니다. 이렇게 하면 어떤 버전의 Trident Node Pod가 노드에 두 개 이상 존재하지 않게 됩니다.

Trident Operator 또는 Helm을 사용하여 Trident 설치 업그레이드

Trident 운영자를 사용하여 수동으로 또는 Helm을 사용하여 Trident 업그레이드할 수 있습니다. Trident 운영자 설치에서 다른 Trident 운영자 설치로 업그레이드하거나 다음에서 업그레이드할 수 있습니다. `tridentctl` Trident 운영자 버전에 설치. 검토 "[업그레이드 방법을 선택하세요](#)" Trident 운영자 설치를 업그레이드하기 전에.

수동 설치 업그레이드

클러스터 범위 Trident Operator 설치에서 다른 클러스터 범위 Trident Operator 설치로 업그레이드할 수 있습니다. 모든 Trident 버전은 클러스터 범위 연산자를 사용합니다.



네임스페이스 범위 연산자(버전 20.07~20.10)를 사용하여 설치된 Trident 에서 업그레이드하려면 다음 업그레이드 지침을 사용하세요. "[설치된 버전](#)" Trident 의.

이 작업에 관하여

Trident Kubernetes 버전에 대한 연산자를 설치하고 관련 객체를 생성하는 데 사용할 수 있는 번들 파일을 제공합니다.

- Kubernetes 1.24를 실행하는 클러스터의 경우 다음을 사용하세요. "[bundle_pre_1_25.yaml](#)".
- Kubernetes 1.25 이상을 실행하는 클러스터의 경우 다음을 사용하세요. "[bundle_post_1_25.yaml](#)".

시작하기 전에

Kubernetes 클러스터를 실행 중인지 확인하세요. "[지원되는 Kubernetes 버전](#)".

단계

1. Trident 버전을 확인하세요:

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

2. 업데이트 `operator.yaml`, `tridentorchestrator_cr.yaml`, 그리고 `post_1_25_bundle.yaml` 업그레이드하려는 버전(예: 25.06)에 대한 레지스트리와 이미지 경로, 그리고 올바른 비밀번호를 사용합니다.
3. 현재 Trident 인스턴스를 설치하는 데 사용된 Trident 연산자를 삭제합니다. 예를 들어, 25.02에서 업그레이드하는 경우 다음 명령을 실행합니다.

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

4. 초기 설치를 사용자 정의한 경우 `TridentOrchestrator` 속성을 편집할 수 있습니다. `TridentOrchestrator` 설치 매개변수를 수정하는 객체입니다. 여기에는 오프라인 모드에 대한 미러링된 Trident 및 CSI 이미지 레지스트리를 지정하고, 디버그 로그를 활성화하거나, 이미지 풀 비밀을 지정하기 위해 변경한 내용이 포함될 수 있습니다.
5. `<bundle.yaml>` 이 있는 환경에 맞는 올바른 번들 YAML 파일을 사용하여 Trident 설치하세요. `bundle_pre_1_25.yaml` 또는 `bundle_post_1_25.yaml` Kubernetes 버전에 따라 다릅니다. 예를 들어, Trident 25.06.0을 설치하는 경우 다음 명령을 실행합니다.

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

6. 트라이던트 토크를 편집하여 이미지 25.06.0을 포함시킵니다.

Helm 설치 업그레이드

Trident 헬름 설치를 업그레이드할 수 있습니다.



Trident 설치된 Kubernetes 클러스터를 1.24에서 1.25 이상으로 업그레이드하는 경우 `values.yaml`을 업데이트하여 설정해야 합니다. `excludePodSecurityPolicy` 에게 `true` 또는 추가 `--set excludePodSecurityPolicy=true` 에게 `helm upgrade` 클러스터를 업그레이드하기 전에 명령을 실행하세요.

Trident helm을 업그레이드하지 않고 이미 Kubernetes 클러스터를 1.24에서 1.25로 업그레이드한 경우 `helm upgrade`가 실패합니다. 헬름 업그레이드를 진행하려면 다음 단계를 필수 조건으로 수행하세요.

1. `helm-mapkubeapis` 플러그인을 설치하세요 <https://github.com/helm/helm-mapkubeapis>.
2. Trident 설치된 네임스페이스에서 Trident 릴리스에 대한 테스트 실행을 수행합니다. 여기에는 정리될 리소스가 나열됩니다.

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

3. 헬름을 타고 전체 주행을 한 후 청소를 실시합니다.

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

단계

1. 만약 당신이라면 "[Helm을 사용하여 Trident 설치했습니다.](#)", 사용할 수 있습니다 `helm upgrade trident netapp-trident/trident-operator --version 100.2506.0` 한 단계로 업그레이드하세요. Helm 저장소를 추가하지 않았거나 이를 사용하여 업그레이드할 수 없는 경우:

- a. 최신 Trident 릴리스를 다운로드하세요 "[GitHub의 Assets 섹션](#)".
- b. 사용하다 `helm upgrade` 명령은 어디에 `trident-operator-25.06.0.tgz` 업그레이드하려는 버전을 나타냅니다.

```
helm upgrade <name> trident-operator-25.06.0.tgz
```



초기 설치 중에 사용자 정의 옵션을 설정한 경우(예: Trident 및 CSI 이미지에 대한 개인 미러링 레지스트리 지정) 다음을 추가합니다. `helm upgrade` 명령을 사용하여 `--set` 해당 옵션이 업그레이드 명령에 포함되어 있는지 확인하세요. 그렇지 않으면 값이 기본값으로 재설정됩니다.

2. 달리다 `helm list` 차트와 앱 버전이 모두 업그레이드되었는지 확인하세요. 달리다 `tridentctl logs` 디버그 메시지를 검토합니다.

업그레이드 `tridentctl` Trident 운영자에 설치

Trident 운영자의 최신 릴리스로 업그레이드할 수 있습니다. `tridentctl` 설치. 기존 백엔드와 PVC는 자동으로 사용 가능합니다.



설치 방법을 전환하기 전에 다음을 검토하세요. "[설치 방법 간 이동](#)".

단계

1. 최신 Trident 릴리스를 다운로드하세요.

```
# Download the release required [25.06.0]
mkdir 25.06.0
cd 25.06.0
wget
https://github.com/NetApp/trident/releases/download/v25.06.0/trident-
installer-25.06.0.tar.gz
tar -xf trident-installer-25.06.0.tar.gz
cd trident-installer
```

2. 생성하다 `tridentorchestrator` 매니페스트의 CRD.

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

3. 동일한 네임스페이스에 클러스터 범위 연산자를 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. 생성하다 TridentOrchestrator Trident 설치를 위한 CR.

```
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. Trident 의도한 버전으로 업그레이드되었는지 확인하세요.

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

```
Message:          Trident installed
Namespace:        trident
Status:           Installed
Version:          v25.06.0
```

tridentctl로 업그레이드하세요

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

이 작업에 관하여

Trident 제거하고 다시 설치하면 업그레이드가 수행됩니다. Trident 제거해도 Trident 배포에 사용되는 영구 볼륨 클레임(PVC) 및 영구 볼륨(PV)은 삭제되지 않습니다. 이미 프로비저닝된 PV는 Trident 가 오프라인인 동안에도 사용 가능하며, Trident 온라인 상태로 돌아온 후 그 사이에 생성된 모든 PVC에 대한 볼륨을 프로비저닝합니다.

시작하기 전에

검토"[업그레이드 방법을 선택하세요](#)" 업그레이드하기 전에 다음을 사용하세요. `tridentctl`.

단계

1. 제거 명령을 실행하세요 `tridentctl` CRD 및 관련 객체를 제외한 Trident 와 연관된 모든 리소스를 제거합니다.

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

2. Trident 다시 설치하세요. 참조하다"[tridentctl을 사용하여 Trident 설치](#)".



업그레이드 과정을 중단하지 마세요. 설치 프로그램이 완전히 실행되었는지 확인하세요.

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.