Trident를 업그레이드합니다

Trident

NetApp
October 07, 2025

This PDF was generated from https://docs.netapp.com/ko-kr/trident/trident-managing-k8s/upgrade-trident.html on October 07, 2025. Always check docs.netapp.com for the latest.

목차

Trident를 업그레이드합니다 . . . 1
Trident를 업그레이드합니다 . . . 1
업그레이드 전 고려 사항 . . . 1
1단계: 버전을 선택합니다 . . . 1
2단계: 원래 설치 방법을 결정합니다 . . . 2
3단계: 업그레이드 방법을 선택합니다 . . . 2
운영자와 함께 업그레이드하십시오 . . . 2
운영자 업그레이드 워크플로우를 이해합니다 . . . 2
Trident operator 또는 Helm을 사용하여 Trident 설치를 업그레이드합니다 . . . 3
tridentctl로 업그레이드하십시오 . . . 7

Trident를 업그레이드합니다

Trident를 업그레이드합니다

24.02 릴리즈부터 Trident는 4개월 릴리즈 주기를 따르며 매년 3개의 주요 릴리즈를 제공합니다. 각각의 새 릴리스는 이전 릴리스에서 빌드되며 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 릴리즈로 업그레이드할 때 다음 사항을 고려하십시오.

- 특정 Kubernetes 클러스터의 모든 네임스페이스에 Trident 인스턴스는 하나만 설치해야 합니다.
- Trident 23.07 이상에서는 v1 볼륨 스냅샷이 필요하며 알파 또는 베타 스냅샷을 더 이상 지원하지 않습니다.
- 에서 Google Cloud용 Cloud Volumes Service를 생성한 경우 Trident 23.01에서 업그레이드할 때 또는 `zoneredundantstandardsw` 서비스 수준을"CVS 서비스 유형입니다" 사용하도록 백엔드 구성을 업데이트해야 `standardsw` 합니다. 백엔드에서 을 업데이트하지 `serviceLevel` 못하면 볼륨이 실패할 수 있습니다. 자세한 내용은 을 "CVS 서비스 유형 샘플" 참조하십시오.
- 업그레이드할 때 Trident에서 사용되는 를 `StorageClasses` 제공하는 것이 중요합니다 `parameter.fsType`. 기존 볼륨을 중단하지 않고 삭제 및 재생성할 수 `StorageClasses` 있습니다.
  - 이것은 시행에 대한 ** 요구 사항입니다 "보안 컨텍스트" SAN 볼륨:
  - sample input 디렉토리에는 https://github.com/NetApp/trident/blob/master/trident-installer/sample-input/storage-class-samples/storage-class-basic.yaml.templ와 같은 예가 포함되어 있습니다[`storage-class-basic.yaml.templ`^] 및 링크: `storage-class-bronze-default.yaml`.
  - 자세한 내용은 을 참조하십시오 "알려진 문제".

1단계: 버전을 선택합니다

Trident 버전은 날짜 기반 `YY.MM` 명명 규칙을 따릅니다. 여기서 "YY"는 연도의 마지막 두 자리이고 "MM"은 월입니다. DOT 릴리스는 `YY.MM.X` 규칙을 따릅니다. 여기서 "X"는 패치 수준입니다. 업그레이드할 버전에 따라 업그레이드할 버전을 선택합니다.

- 설치된 버전의 4 릴리스 창 내에 있는 모든 대상 릴리스에 대해 직접 업그레이드를 수행할 수 있습니다. 예를 들어 24.06(또는 24.06 도트 릴리스)에서 25.06로 직접 업그레이드할 수 있습니다.
- 릴리스 4개가 아닌 다른 릴리스에서 업그레이드하는 경우 여러 단계로 업그레이드를 수행합니다. 에서 4개의 릴리스 윈도우에 맞는 가장 최신 릴리스로 업그레이드하려는 의 업그레이드 지침을 "이전 버전" 사용합니다. 예를 들어, 23.07을 실행하고 있고 25.06로 업그레이드하려는 경우:
  a. 23.07에서 24.06로 첫 번째 업그레이드.
  b. 그런 다음 24.06에서 25.06로 업그레이드합니다.

> OpenShift Container Platform에서 Trident 연산자를 사용하여 업그레이드할 때는 Trident 21.01.1 이상으로 업그레이드해야 합니다. 21.01.0으로 릴리스된 Trident 연산자에는 21.01.1에서 해결된 알려진 문제가 포함되어 있습니다. 자세한 내용은 를 참조하십시오 "GitHub에 대한 발행 세부 정보".

2단계: 원래 설치 방법을 결정합니다

Trident를 처음 설치할 때 사용한 버전을 확인하려면 다음을 수행합니다.

1. 사용 `kubectl get pods -n trident` 를 눌러 포드를 검사합니다.
    - 운영자 포드가 없으면 를 사용하여 Trident를 설치한 `tridentctl` 것입니다.
    - 운영자 포드가 있는 경우, Trident는 Trident 운영자를 사용하여 수동으로 또는 Helm을 사용하여 설치되었습니다.
2. 운영자 포드가 있는 경우 를 `kubectl describe torc` 사용하여 Trident가 Helm을 사용하여 설치되었는지 확인합니다.
    - Helm 레이블이 있는 경우 Helm을 사용하여 Trident를 설치한 것입니다.
    - Helm 레이블이 없는 경우 Trident 연산자를 사용하여 Trident를 수동으로 설치했습니다.

3단계: 업그레이드 방법을 선택합니다

일반적으로 초기 설치에 사용한 것과 동일한 방법으로 업그레이드해야 하지만, 가능한 경우"설치 방법 간에 이동합니다" Trident를 업그레이드할 수 있는 두 가지 옵션이 있습니다.

- "Trident 연산자를 사용하여 업그레이드합니다"

> 검토할 것을 제안합니다 "운영자 업그레이드 워크플로우를 이해합니다" 작업자와 업그레이드하기 전에

*

운영자와 함께 업그레이드하십시오

운영자 업그레이드 워크플로우를 이해합니다

Trident 연산자를 사용하여 Trident를 업그레이드하기 전에 업그레이드 중에 발생하는 백그라운드 프로세스를 이해해야 합니다. 여기에는 Trident 컨트롤러, 컨트롤러 Pod 및 노드 Pod, 롤링 업데이트를 사용하는 노드 DemonSet의 변경 사항이 포함됩니다.

Trident 운영자 업그레이드 처리

Trident를 설치하고 업그레이드할 수 있는 툴 중 "Trident 연산자를 사용할 때의 이점"하나는 기존의 마운트된 볼륨을 중단하지 않고 Trident 및 Kubernetes 개체를 자동으로 처리하는 것입니다. 이와 같이 Trident는 다운타임 없이 업그레이드를 지원할 수 있습니다. 또는"_롤링 업데이트 _" 특히, Trident 운영자는 Kubernetes 클러스터와 통신하여 다음을 수행합니다.

- Trident 컨트롤러 배포 및 노드 DemonSet을 삭제하고 다시 만듭니다.
- Trident 컨트롤러 Pod 및 Trident 노드 Pod를 새로운 버전으로 교체합니다.
    - 노드가 업데이트되지 않으면 나머지 노드가 업데이트됩니다.
    - 실행 중인 Trident 노드 Pod가 있는 노드에서만 볼륨을 마운트할 수 있습니다.

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

운영자 업그레이드 워크플로우

Trident 연산자를 사용하여 업그레이드를 시작하는 경우:

1. Trident 운영자 *:
   a. 현재 설치된 Trident 버전(version_n_)을 검색합니다.
   b. CRD, RBAC, Trident SVC를 포함한 모든 Kubernetes 오브젝트를 업데이트합니다.
   c. 버전_n_에 대한 Trident 컨트롤러 배포를 삭제합니다.
   d. 버전_n+1_에 대한 Trident 컨트롤러 배포를 생성합니다.
2. * Kubernetes * 는 _n+1_용 Trident 컨트롤러 포드를 생성합니다.
3. Trident 운영자 *:
   a. _n_에 대한 Trident 노드 데모 세트를 삭제합니다. 운용자는 Node Pod 종료를 기다리지 않는다.
   b. _n+1_에 대한 Trident 노드 데모 세트를 생성합니다.
4. * Kubernetes * 는 Trident 노드 Pod_n_을(를) 실행하지 않는 노드에서 Trident 노드 Pod를 생성합니다. 따라서 노드에 여러 버전의 Trident 노드 Pod가 둘 이상 있지 않도록 합니다.

Trident operator 또는 Helm을 사용하여 Trident 설치를 업그레이드합니다

Trident 연산자를 사용하여 수동으로 또는 Helm을 사용하여 Trident를 업그레이드할 수 있습니다. Trident 운영자 설치에서 다른 Trident 운영자 설치로 업그레이드하거나, 설치에서 Trident 운영자 버전으로 업그레이드할 수 `tridentctl` 있습니다. "업그레이드 방법을 선택합니다"Trident 운영자 설치를 업그레이드하기 전에 검토하십시오.

수동 설치를 업그레이드합니다

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 Trident에서 업그레이드하려면 Trident의 업그레이드 지침을 "설치된 버전"사용하십시오.

이 작업에 대해

Trident는 운영자를 설치하고 Kubernetes 버전에 대한 관련 오브젝트를 생성하는 데 사용할 수 있는 번들 파일을 제공합니다.

- Kubernetes 1.24를 실행하는 클러스터에는 "Bundle_PRE_1_25.YAML"를 사용합니다.
- Kubernetes 1.25 이상을 실행하는 클러스터의 경우 를 "Bundle_post_1_25.YAML"사용합니다.

시작하기 전에

실행 중인 Kubernetes 클러스터를 사용하고 있는지 확인합니다 "지원되는 Kubernetes 버전".

단계

1. Trident 버전 확인:

```
./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. 를 사용하여 초기 설치를 사용자 지정한 경우 `TridentOrchestrator` 속성을 편집할 수 있습니다 `TridentOrchestrator` 설치 매개 변수를 수정하는 개체입니다. 여기에는 오프라인 모드에 대해 미러링된 Trident 및 CSI 이미지 레지스트리를 지정하는 변경 사항, 디버그 로그 활성화 또는 이미지 풀 비밀을 지정하는 변경 사항이 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 설치를 업그레이드합니다

Trident Helm 설치를 업그레이드할 수 있습니다.

> Trident가 설치된 Kubernetes 클러스터를 1.24에서 1.25 이상으로 업그레이드할 `true` `helm upgrade` 경우 클러스터를 업그레이드하기 전에 value.yaml을 명령으로 설정하거나 `--set excludePodSecurityPolicy=true` 명령에 추가해야 `excludePodSecurityPolicy` 합니다.

Trident Helm을 업그레이드하지 않고 이미 Kubernetes 클러스터를 1.24에서 1.25로 업그레이드한 경우에는 Helm 업그레이드가 실패합니다. Helm 업그레이드를 진행하려면 다음 단계를 전제 조건으로 수행하십시오.

1. 에서 helm-mapkubeapis 플러그인을 https://github.com/helm/helm-mapkubeapis 설치합니다.
2. Trident가 설치된 네임스페이스에서 Trident 릴리스에 대해 건식 실행을 수행합니다. 정리할 리소스가 나열됩니다.

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

3. 정리 작업을 수행하려면 Helm을 사용하여 전체 실행을 수행합니다.

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

단계

1. "Helm을 사용하여 Trident를 설치했습니다"를 사용하여 한 단계로 업그레이드할 수 `helm upgrade trident netapp-trident/trident-operator --version 100.2506.0` 있습니다. Helm repo를 추가하지 않았거나 업그레이드에 사용할 수 없는 경우:
   a. 에서 최신 Trident 릴리스를 "GitHub의 _Assets_섹션"다운로드하십시오.
   b. `helm upgrade`에서 업그레이드할 버전을 반영하는 명령을 `trident-operator-25.06.0.tgz` 사용합니다.

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

   > 초기 설치 중에 사용자 정의 옵션을 설정하는 경우(예: Trident 및 CSI 이미지에 대한 전용 미러링된 레지스트리 지정) 를 추가합니다 `helm upgrade` 명령을 사용합니다 `--set` 이러한 옵션이 업그레이드 명령에 포함되도록 하려면 값이 기본값으로 재설정됩니다.

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

에서 업그레이드 `tridentctl` **Trident** 운영자에 설치

에서 Trident 운영자의 최신 릴리즈로 업그레이드할 수 있습니다 `tridentctl` 설치: 기존 백엔드 및 PVC를 자동으로 사용할 수 있습니다.

> 설치 방법 간에 전환하기 전에 를 참조하십시오 "설치 방법 간 이동".

단계

1. 최신 Trident 릴리스를 다운로드합니다.

   ```
   # 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. 매니페스트에서 트라이디오케스트레이터 CRD를 만듭니다.

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

3. 클러스터 범위 연산자를 같은 네임스페이스에 구현합니다.

```
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. `TridentOrchestrator`Trident 설치를 위한 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 확인합니다.

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

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

tridentctl로 업그레이드하십시오

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

이 작업에 대해

Trident 제거 및 재설치는 업그레이드로 작동합니다. Trident를 제거하면 Trident 배포에 사용되는 PVC(영구 볼륨 클레임) 및 PV(영구 볼륨)가 삭제되지 않습니다. 이미 프로비저닝된 PVS는 Trident가 오프라인 상태인 동안 사용 가능한 상태로 유지되며, Trident는 다시 온라인 상태가 된 후 그 사이에 생성된 모든 PVC에 대한 볼륨을 프로비저닝합니다.

시작하기 전에

검토 "업그레이드 방법을 선택합니다" 를 사용하여 업그레이드하기 전에 `tridentctl`.

단계

1. 에서 uninstall 명령을 `tridentctl` 실행하여 CRD 및 관련 객체를 제외한 Trident와 연결된 모든 리소스를 제거합니다.

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

2. Trident를 다시 설치합니다. 을 "tridentctl을 사용하여 Trident를 설치합니다"참조하십시오.

> 업그레이드 프로세스를 중단하지 마십시오. 설치 프로그램이 완료될 때까지 실행되는지 확인합니다.

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.