



# **tridentctl**을 사용하여 설치합니다

## Astra Trident

NetApp  
November 14, 2025

# 목차

tridentctl을 사용하여 설치합니다	1
tridentctl을 사용하여 설치합니다	1
Astra Trident 23.04에 대한 중요 정보	1
를 사용하여 Astra Trident를 설치합니다 tridentctl	1
1단계: Trident 설치 프로그램 패키지를 다운로드합니다	2
2단계: Astra Trident 설치	2
설치를 확인합니다	3
샘플 구성	4
다음 단계	5
tridentctl 설치를 사용자 지정합니다	5
설치 프로그램에 대해 알아보십시오	5

# tridentctl을 사용하여 설치합니다

## tridentctl을 사용하여 설치합니다

을 사용하여 Astra Trident를 설치할 수 있습니다 `tridentctl`. 이 프로세스는 Astra Trident에 필요한 컨테이너 이미지가 개인 레지스트리에 저장되거나 저장되지 않은 설치에 적용됩니다. 를 사용자 지정합니다 `tridentctl` 구축 방법은 을 참조하십시오 "[tridentctl 배포를 사용자 지정합니다](#)".

### Astra Trident 23.04에 대한 중요 정보

- Astra Trident \* 에 대한 다음 중요 정보를 읽어야 합니다

#### <strong> 중요 정보 Astra Trident </strong>

- 이제 Trident에서 Kubernetes 1.27이 지원됩니다. Kubernetes를 업그레이드하기 전에 Trident를 업그레이드하십시오.
- Astra Trident는 SAN 환경에서 다중 경로 구성을 엄격하게 사용하며 권장 값은 입니다  
`find_multipaths: no` 다중 경로 .conf 파일

비 경로 다중화 구성 또는 의 사용 `find_multipaths: yes` 또는 `find_multipaths: smart` `multipath.conf` 파일의 값으로 인해 마운트 오류가 발생합니다. Trident에서 의 사용을 권장했습니다 `find_multipaths: no` 21.07 릴리스 이후.

### 를 사용하여 Astra Trident를 설치합니다 `tridentctl`

검토 "[설치 개요](#)" 설치 사전 요구 사항을 충족하고 환경에 맞는 올바른 설치 옵션을 선택했는지 확인합니다.

시작하기 전에

설치를 시작하기 전에 Linux 호스트에 로그인하여 작업 관리 여부를 확인합니다. "[지원되는 Kubernetes 클러스터](#)" 필요한 권한이 있어야 합니다.



OpenShift에서는 을 사용합니다 `oc` 대신 `kubectl` 다음 모든 예에서 를 실행하여 먼저 \* `system:admin` \* 으로 로그인합니다 `oc login -u system:admin` 또는 `oc login -u kube-admin`.

### 1. Kubernetes 버전 확인:

```
kubectl version
```

### 2. 클러스터 관리자 권한 확인:

```
kubectl auth can-i '*' '*' --all-namespaces
```

### 3. Docker Hub의 이미지를 사용하는 Pod를 시작하고 Pod 네트워크를 통해 스토리지 시스템에 연결할 수 있는지 확인합니다.

```
kubectl run -i --tty ping --image=busybox --restart=Never --rm -- \
ping <management IP>
```

## 1단계: Trident 설치 프로그램 패키지를 다운로드합니다

Astra Trident 설치 프로그램 패키지는 Trident Pod를 생성하고 상태를 유지하는 데 사용되는 CRD 객체를 구성하며, CSI 사이드카를 초기화하여 클러스터 호스트에 볼륨 프로비저닝 및 연결과 같은 작업을 수행합니다. 에서 최신 버전의 Trident 설치 프로그램을 다운로드하고 압축을 풉니다 "[GitHub의 Assets\\_섹션](#)". [선택한 Astra Trident 버전을 사용하여 예제에서](#) `update<trident-installer-XX.XX.X.tar.gz>`를 선택합니다.

```
wget https://github.com/NetApp/trident/releases/download/v23.04.0/trident-
installer-23.04.0.tar.gz
tar -xf trident-installer-23.04.0.tar.gz
cd trident-installer
```

## 2단계: Astra Trident 설치

를 실행하여 원하는 네임스페이스에 Astra Trident를 설치합니다 `tridentctl install` 명령. 추가 인수를 추가하여 이미지 레지스트리 위치를 지정할 수 있습니다.

## 표준 모드

```
./tridentctl install -n trident
```

## 하나의 레지스트리에 있는 이미지

```
./tridentctl install -n trident --image-registry <your-registry>  
--autosupport-image <your-registry>/trident-autosupport:23.04 --trident  
-image <your-registry>/trident:23.04.0
```

## 다른 레지스트리의 이미지

추가해야 합니다 sig-storage 를 누릅니다 imageRegistry 다른 레지스트리 위치를 사용합니다.

```
./tridentctl install -n trident --image-registry <your-registry>/sig-  
storage --autosupport-image <your-registry>/netapp/trident-  
autosupport:23.04 --trident-image <your-  
registry>/netapp/trident:23.04.0
```

설치 상태는 다음과 같습니다.

```
.....  
INFO Starting Trident installation.                namespace=trident  
INFO Created service account.  
INFO Created cluster role.  
INFO Created cluster role binding.  
INFO Added finalizers to custom resource definitions.  
INFO Created Trident service.  
INFO Created Trident secret.  
INFO Created Trident deployment.  
INFO Created Trident daemonset.  
INFO Waiting for Trident pod to start.  
INFO Trident pod started.                          namespace=trident  
pod=trident-controller-679648bd45-cv2mx  
INFO Waiting for Trident REST interface.  
INFO Trident REST interface is up.                version=23.04.0  
INFO Trident installation succeeded.  
.....
```

## 설치를 확인합니다

POD 생성 상태 또는 를 사용하여 설치를 확인할 수 있습니다 tridentctl.

## POD 생성 상태 사용

생성된 Pod의 상태를 검토하여 Astra Trident 설치가 완료되었는지 확인할 수 있습니다.

```
kubectl get pods -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
trident-controller-679648bd45-cv2mx	6/6	Running	0	5m29s
trident-node-linux-vgc8n	2/2	Running	0	5m29s



설치 프로그램이 성공적으로 완료되지 않거나 `trident-controller-<generated id>` (`trident-csi-<generated id>` 23.01 이전 버전에서는 \*Running\* 상태가 없으며 플랫폼이 설치되지 않았습니다. 사용 `-d` 를 선택합니다 "[디버그 모드를 켭니다](#)" 문제를 해결합니다.

사용 `tridentctl`

을 사용할 수 있습니다 `tridentctl` 설치된 Astra Trident의 버전을 확인합니다.

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

```
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 23.04.0       | 23.04.0       |
+-----+-----+
```

## 샘플 구성

예 1: **Windows** 노드에서 **Astra Trident**를 실행하도록 설정합니다

Astra Trident를 Windows 노드에서 실행하려면 다음을 수행합니다.

```
tridentctl install --windows -n trident
```

예 2: 강제 분리 사용

강제 분리에 대한 자세한 내용은 을 참조하십시오 "[Trident 운영자 설치를 사용자 지정합니다](#)".

```
tridentctl install --enable-force-detach=true -n trident
```

## 다음 단계

이제 가능합니다 "백엔드 및 스토리지 클래스를 생성하고, 볼륨을 프로비저닝하고, POD에 볼륨을 마운트합니다".

## tridentctl 설치를 사용자 지정합니다

Astra Trident 설치 프로그램을 사용하여 설치를 사용자 지정할 수 있습니다.

### 설치 프로그램에 대해 알아보십시오

Astra Trident 설치 프로그램을 사용하여 특성을 사용자 지정할 수 있습니다. 예를 들어, Trident 이미지를 개인 저장소에 복사한 경우 를 사용하여 이미지 이름을 지정할 수 있습니다 `--trident-image`. Trident 이미지와 필요한 CSI 사이드카 이미지를 개인 저장소에 복사한 경우 를 사용하여 해당 저장소의 위치를 지정하는 것이 좋습니다 `--image-registry` 스위치를 누릅니다 `<registry FQDN>[:port]`.

Kubernetes 배포를 사용 중인 경우 kubelet 일반적인 경로 이외의 경로에 데이터를 보관합니다 `/var/lib/kubelet`, 을 사용하여 대체 경로를 지정할 수 있습니다 `--kubelet-dir`.

설치 관리자의 인수 이외에 설치를 사용자 지정해야 하는 경우 배포 파일을 사용자 지정할 수도 있습니다. 를 사용합니다 `--generate-custom-yaml` 매개 변수는 설치 관리자의 에 다음 YAML 파일을 생성합니다 `setup` 디렉터리:

- `trident-clusterrolebinding.yaml`
- `trident-deployment.yaml`
- `trident-crds.yaml`
- `trident-clusterrole.yaml`
- `trident-daemonset.yaml`
- `trident-service.yaml`
- `trident-namespace.yaml`
- `trident-serviceaccount.yaml`
- `trident-resourcequota.yaml`

이러한 파일을 생성한 후 필요에 따라 수정한 다음 을 사용할 수 있습니다 `--use-custom-yaml` 사용자 지정 배포를 설치합니다.

```
./tridentctl install -n trident --use-custom-yaml
```

## 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 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.