



# Trident Protect를 설치합니다

## Trident

NetApp  
July 01, 2026

# 목차

Trident Protect를 설치합니다	1
Trident Protect 요구 사항	1
Trident Protect Kubernetes 클러스터 호환성	1
Trident Protect 스토리지 백엔드 호환성	1
nas-economy 볼륨에 대한 요구 사항	2
KubeVirt VM으로 데이터 보호	2
SnapMirror 복제 요구 사항	3
Trident Protect를 설치하고 구성합니다	4
Trident Protect를 설치합니다	5
Trident Protect CLI 플러그인을 설치합니다	8
Trident Protect CLI 플러그인을 설치합니다	8
Trident CLI 플러그인 도움말 보기	10
명령 자동 완성 활성화	10
Trident Protect 설치 사용자 지정	12
Trident Protect 컨테이너 리소스 제한 지정	12
보안 컨텍스트 제약 조건 사용자 지정	13
Trident Protect 헬름 차트의 추가 설정을 구성합니다	14
Trident Protect Pod를 특정 노드로 제한합니다	16

# Trident Protect를 설치합니다

## Trident Protect 요구 사항

먼저 운영 환경, 애플리케이션 클러스터, 애플리케이션 및 라이선스의 준비 상태를 확인하십시오. Trident Protect를 배포하고 운영하기 위한 요구 사항을 환경이 충족하는지 확인하십시오.

### Trident Protect Kubernetes 클러스터 호환성

Trident Protect는 다음을 포함한 다양한 완전 관리형 및 자체 관리형 Kubernetes 제품과 호환됩니다.

- Amazon Elastic Kubernetes Service(EKS)
- Google Kubernetes Engine(GKE)
- Microsoft Azure Kubernetes Service(AKS)
- Red Hat OpenShift
- SUSE Harvester 1.7.0(ONTAP iSCSI)
- SUSE Rancher
- VMware Tanzu Portfolio
- 업스트림 Kubernetes



- Trident Protect 백업은 Linux 컴퓨팅 노드에서만 지원됩니다. Windows 컴퓨팅 노드에서는 백업 작업이 지원되지 않습니다.
- Trident Protect를 설치할 클러스터에 실행 중인 스냅샷 컨트롤러와 관련 CRD가 구성되어 있는지 확인하십시오. 스냅샷 컨트롤러를 설치하려면 "[이 지침](#)"을 참조하십시오.
- 하나 이상의 VolumeSnapshotClass가 존재하는지 확인하십시오. 자세한 내용은 "[VolumeSnapshotClass](#)"을 참조하십시오.
- Trident Protect를 설치하려면 Helm 4.x 이상이 필요합니다.

### Trident Protect 스토리지 백엔드 호환성

Trident Protect는 다음과 같은 스토리지 백엔드를 지원합니다.

- Amazon FSx for NetApp ONTAP
- Cloud Volumes ONTAP
- ONTAP 스토리지 어레이
- Google Cloud NetApp Volumes
- Azure NetApp Files

스토리지 백엔드가 다음 요구 사항을 충족하는지 확인하십시오.

- 클러스터에 연결된 NetApp 스토리지가 Trident 24.02 이상 버전(Trident 24.10 권장)을 사용하고 있는지 확인하십시오.

- NetApp ONTAP 스토리지 백엔드가 있는지 확인하십시오.
- 백업 저장을 위한 오브젝트 스토리지 버킷을 구성했는지 확인하십시오.
- 애플리케이션 또는 애플리케이션 데이터 관리 작업에 사용할 애플리케이션 네임스페이스를 생성하십시오. Trident Protect는 이러한 네임스페이스를 자동으로 생성하지 않으므로 사용자 지정 리소스에 존재하지 않는 네임스페이스를 지정하면 작업이 실패합니다.

## nas-economy 볼륨에 대한 요구 사항

Trident Protect는 nas-economy 볼륨에 대한 백업 및 복원 작업을 지원합니다. 스냅샷, 클론 및 SnapMirror 복제는 현재 nas-economy 볼륨에 대해 지원되지 않습니다. Trident Protect에서 사용할 각 nas-economy 볼륨에 대해 스냅샷 디렉터리를 활성화해야 합니다.



일부 애플리케이션은 스냅샷 디렉터리를 사용하는 볼륨과 호환되지 않습니다. 이러한 애플리케이션의 경우 ONTAP 스토리지 시스템에서 다음 명령을 실행하여 스냅샷 디렉터를 숨겨야 합니다.

```
nfs modify -vserver <svm> -v3-hide-snapshot enabled
```

각 nas-economy 볼륨에 대해 다음 명령을 실행하여 스냅샷 디렉터를 활성화할 수 있습니다. ``<volume-UUID>``을(를) 변경하려는 볼륨의 UUID로 바꾸십시오.

```
tridentctl update volume <volume-UUID> --snapshot-dir=true --pool-level=true -n trident
```



Trident 백엔드 구성 옵션 `snapshotDir`을 `true`로 설정하여 새 볼륨에 대해 스냅샷 디렉터를 기본적으로 활성화할 수 있습니다. 기존 볼륨에는 영향을 미치지 않습니다.

## KubeVirt VM으로 데이터 보호

Trident Protect는 데이터 보호 작업 중 KubeVirt 가상 머신에 대한 파일 시스템 동결 및 동결 해제 기능을 제공하여 데이터 일관성을 보장합니다. VM 동결 작업에 대한 구성 방법과 기본 동작은 Trident Protect 버전에 따라 다르며, 최신 릴리스에서는 Helm 차트 매개변수를 통해 간소화된 구성을 제공합니다.



복원 작업 중에는 가상 머신(VM)에 대해 생성된 `VirtualMachineSnapshots`는 복원되지 않습니다.

### Trident Protect 25.10 이상

Trident Protect는 데이터 보호 작업 중에 일관성을 보장하기 위해 KubeVirt 파일 시스템을 자동으로 동결 및 해제합니다. Trident Protect 25.10부터는 Helm 차트 설치 시 `vm.freeze` 매개변수를 사용하여 이 동작을 비활성화할 수 있습니다. 이 매개변수는 기본적으로 활성화되어 있습니다.

```
helm install ... --set vm.freeze=false ...
```

## Trident Protect 24.10.1 ~ 25.06

Trident Protect 24.10.1부터 Trident Protect는 데이터 보호 작업 중에 KubeVirt 파일 시스템을 자동으로 동결 및 해제합니다. 필요에 따라 다음 명령을 사용하여 이 자동 동작을 비활성화할 수 있습니다:

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=false -n trident-protect
```

## Trident Protect 24.10

Trident Protect 24.10은 데이터 보호 작업 중에 KubeVirt VM 파일 시스템의 일관된 상태를 자동으로 보장하지 않습니다. Trident Protect 24.10을 사용하여 KubeVirt VM 데이터를 보호하려면 데이터 보호 작업을 수행하기 전에 파일 시스템의 동결/해제 기능을 수동으로 활성화해야 합니다. 이렇게 하면 파일 시스템이 일관된 상태를 유지하게 됩니다.

Trident Protect 24.10을 구성하여 "가상화 구성"을(를) 사용한 다음 다음 명령을 사용하여 데이터 보호 작업 중 VM 파일 시스템의 동결 및 해제를 관리할 수 있습니다.

```
kubectl set env deployment/trident-protect-controller-manager
NEPTUNE_VM_FREEZE=true -n trident-protect
```

## SnapMirror 복제 요구 사항

NetApp SnapMirror 복제는 다음 ONTAP 솔루션에서 Trident Protect와 함께 사용할 수 있습니다.

- 온프레미스 NetApp FAS, AFF 및 ASA 시스템. SnapMirror Trident Protect를 사용한 복제는 현재 ASA r2 시스템에서 지원되지 않습니다.
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP
- Amazon FSx for NetApp ONTAP

## ONTAP 클러스터 요구 사항 SnapMirror 복제

SnapMirror 복제를 사용하려는 경우 ONTAP 클러스터가 다음 요구 사항을 충족하는지 확인하십시오.

- **NetApp Trident:** NetApp Trident는 ONTAP를 백엔드로 사용하는 소스 및 타겟 Kubernetes 클러스터 모두에 존재해야 합니다. Trident Protect는 다음 드라이버를 기반으로 하는 스토리지 클래스를 사용하여 NetApp SnapMirror 기술을 통한 복제를 지원합니다:
  - ontap-nas: NFS
  - ontap-san: iSCSI
  - ontap-san: FC
  - ontap-san: NVMe/TCP(최소 ONTAP 버전 9.15.1 필요)

- 라이선스: 데이터 보호 번들을 사용하는 ONTAP SnapMirror 비동기 라이선스는 소스 및 타겟 ONTAP 클러스터 모두에서 활성화되어야 합니다. 자세한 내용은 "[SnapMirror ONTAP 라이선싱 개요](#)"를 참조하십시오.

ONTAP 9.10.1부터 모든 라이선스는 여러 기능을 활성화하는 단일 파일인 NetApp 라이선스 파일(NLF)로 제공됩니다. 자세한 내용은 "[ONTAP One에 포함된 라이선스](#)"를 참조하십시오.



SnapMirror 비동기 보호 기능만 지원됩니다.

### SnapMirror 복제를 위한 피어링 고려 사항

스토리지 백엔드 피어링을 사용하려는 경우 환경이 다음 요구 사항을 충족하는지 확인하십시오.

- 클러스터 및 **SVM**: ONTAP 스토리지 백엔드는 피어링되어야 합니다. 자세한 내용은 "[클러스터 및 SVM 피어링 개요](#)"를 참조하십시오.



두 ONTAP 클러스터 간의 복제 관계에 사용되는 SVM 이름이 고유한지 확인하십시오.

- **NetApp Trident** 및 **SVM**: 피어링된 원격 SVM은 타겟 클러스터의 NetApp Trident에서 사용할 수 있어야 합니다.
- 관리형 백엔드: 복제 관계를 생성하려면 Trident Protect에 ONTAP 스토리지 백엔드를 추가하고 관리해야 합니다.

### SnapMirror 복제를 위한 Trident/ONTAP 구성

Trident Protect를 사용하려면 소스 클러스터와 타겟 클러스터 모두에 대해 복제를 지원하는 스토리지 백엔드를 하나 이상 구성해야 합니다. 소스 클러스터와 타겟 클러스터가 동일한 경우, 최상의 복원력을 위해 타겟 애플리케이션은 소스 애플리케이션과 다른 스토리지 백엔드를 사용해야 합니다.

### SnapMirror 복제를 위한 Kubernetes 클러스터 요구 사항

Kubernetes 클러스터가 다음 요구 사항을 충족하는지 확인하십시오.

- **AppVault** 접근성: 소스 및 타겟 클러스터 모두 애플리케이션 객체 복제를 위해 AppVault에서 읽고 쓰기 위한 네트워크 액세스가 필요합니다.
- 네트워크 연결: 방화벽 규칙, 버킷 권한 및 IP 허용 목록을 구성하여 두 클러스터와 AppVault 간의 WAN 통신을 활성화합니다.



많은 기업 환경에서는 WAN 연결 전반에 걸쳐 엄격한 방화벽 정책을 시행합니다. 복제를 구성하기 전에 인프라 팀과 함께 이러한 네트워크 요구 사항을 확인하십시오.

## Trident Protect를 설치하고 구성합니다.

사용 환경이 Trident Protect의 요구 사항을 충족하는 경우 다음 단계에 따라 클러스터에 Trident Protect를 설치할 수 있습니다. Trident Protect는 NetApp에서 얻거나 자체 프라이빗 레지스트리에서 설치할 수 있습니다. 클러스터가 인터넷에 액세스할 수 없는 경우 프라이빗 레지스트리에서 설치하는 것이 유용합니다.

**Trident Protect**를 설치합니다

## NetApp에서 Trident Protect 설치

### 단계

1. Trident Helm 리포지토리 추가:

```
helm repo add netapp-trident-protect
https://netapp.github.io/trident-protect-helm-chart
```

2. Helm을 사용하여 Trident Protect를 설치합니다. ``<name-of-cluster>``를 클러스터 이름으로 바꾸세요. 이 이름은 클러스터에 할당되어 클러스터의 백업 데이터 및 스냅샷을 식별하는 데 사용됩니다.

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --version 100.2602.0 --create
--namespace --namespace trident-protect
```

3. 선택적으로 디버그 로깅을 활성화하려면(문제 해결 권장) 다음을 사용하십시오.

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --set logLevel=debug --version
100.2602.0 --create-namespace --namespace trident-protect
```

디버그 로깅은 NetApp 지원팀이 로그 레벨 변경이나 문제 재현 없이 문제를 해결하는 데 도움이 됩니다.

### 개인 레지스트리에서 Trident Protect를 설치하세요

Kubernetes 클러스터가 인터넷에 액세스할 수 없는 경우 개인 이미지 레지스트리에서 Trident Protect를 설치할 수 있습니다. 다음 예에서 괄호 안의 값은 사용자 환경에 맞는 정보로 바꾸십시오.

### 단계

1. 다음 이미지를 로컬 시스템으로 가져오고 태그를 업데이트한 다음 프라이빗 레지스트리로 푸시합니다.

```
docker.io/netapp/controller:26.02.0
docker.io/netapp/restic:26.02.0
docker.io/netapp/kopia:26.02.0
docker.io/netapp/kopiablockrestore:26.02.0
docker.io/netapp/trident-autosupport:26.02.0
docker.io/netapp/exehook:26.02.0
docker.io/netapp/resourcebackup:26.02.0
docker.io/netapp/resourcerestore:26.02.0
docker.io/netapp/resourcedelete:26.02.0
docker.io/netapp/trident-protect-utils:v1.0.0
```

예를 들면 다음과 같습니다.

```
docker pull docker.io/netapp/controller:26.02.0
```

```
docker tag docker.io/netapp/controller:26.02.0 <private-registry-url>/controller:26.02.0
```

```
docker push <private-registry-url>/controller:26.02.0
```



Helm 차트를 얻으려면 먼저 인터넷에 연결된 컴퓨터에서 `helm pull trident-protect --version 100.2602.0 --repo https://netapp.github.io/trident-protect-helm-chart``를 사용하여 Helm 차트를 다운로드한 다음, 생성된 ``trident-protect-100.2602.0.tgz`` 파일을 오프라인 환경으로 복사하고 마지막 단계에서 저장소 참조 대신 ``helm install trident-protect ./trident-protect-100.2602.0.tgz``를 사용하여 설치하십시오.

## 2. Trident Protect 시스템 네임스페이스를 생성합니다.

```
kubectl create ns trident-protect
```

## 3. 레지스트리에 로그인합니다.

```
helm registry login <private-registry-url> -u <account-id> -p <api-token>
```

## 4. 프라이빗 레지스트리 인증에 사용할 풀 시크릿을 생성합니다.

```
kubectl create secret docker-registry regcred --docker-username=<registry-username> --docker-password=<api-token> -n trident-protect --docker-server=<private-registry-url>
```

## 5. Trident Helm 리포지토리 추가:

```
helm repo add netapp-trident-protect https://netapp.github.io/trident-protect-helm-chart
```

## 6. ``protectValues.yaml``라는 이름의 파일을 생성하세요. 해당 파일에 다음 Trident Protect 설정이 포함되어 있는지 확인하세요.

---

```
imageRegistry: <private-registry-url>
imagePullSecrets:
  - name: regcred
```



imageRegistry 및 imagePullSecrets 값은 resourcebackup 및 resourcerestore`를 포함한 모든 구성 요소 이미지에 적용됩니다. 레지스트리 내의 특정 저장소 경로(예: `example.com:443/my-repo)에 이미지를 푸시하는 경우 레지스트리 필드에 전체 경로를 포함하십시오. 이렇게 하면 모든 이미지가 `

7. Helm을 사용하여 Trident Protect를 설치합니다. ``를 클러스터 이름으로 바꾸세요. 이 이름은 클러스터에 할당되어 클러스터의 백업 데이터 및 스냅샷을 식별하는 데 사용됩니다.

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name_of_cluster> --version 100.2602.0 --create
--namespace --namespace trident-protect -f protectValues.yaml
```

8. 선택적으로 디버그 로깅을 활성화하려면(문제 해결 권장) 다음을 사용하십시오.

```
helm install trident-protect netapp-trident-protect/trident-protect
--set clusterName=<name-of-cluster> --set logLevel=debug --version
100.2602.0 --create-namespace --namespace trident-protect -f
protectValues.yaml
```

디버그 로깅은 NetApp 지원팀이 로그 레벨 변경이나 문제 재현 없이 문제를 해결하는 데 도움이 됩니다.



AutoSupport 설정 및 네임스페이스 필터링을 포함한 추가 Helm 차트 구성 옵션은 "[Trident Protect 설치 사용자 지정](#)"을 참조하십시오.

## Trident Protect CLI 플러그인을 설치합니다

Trident `tridentctl` 유틸리티의 확장 기능인 Trident Protect 명령줄 플러그인을 사용하여 Trident Protect 사용자 정의 리소스(CR)를 생성하고 상호 작용할 수 있습니다.

### Trident Protect CLI 플러그인을 설치합니다

명령줄 유틸리티를 사용하기 전에 클러스터에 액세스하는 데 사용하는 시스템에 유틸리티를 설치해야 합니다. 시스템에서 x64 또는 ARM CPU를 사용하는지 여부에 따라 다음 단계를 수행합니다.

### Linux AMD64 CPU용 플러그인 다운로드

단계

1. Trident Protect CLI 플러그인을 다운로드합니다.

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-linux-amd64
```

### Linux ARM64 CPU용 플러그인 다운로드

단계

1. Trident Protect CLI 플러그인을 다운로드합니다.

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-linux-arm64
```

### Mac AMD64 CPU용 플러그인 다운로드

단계

1. Trident Protect CLI 플러그인을 다운로드합니다.

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-macos-amd64
```

### Mac ARM64 CPU용 플러그인 다운로드

단계

1. Trident Protect CLI 플러그인을 다운로드합니다.

```
curl -L -o tridentctl-protect https://github.com/NetApp/tridentctl-protect/releases/download/26.02.0/tridentctl-protect-macos-arm64
```

1. 플러그인 바이너리에 대한 실행 권한을 활성화합니다.

```
chmod +x tridentctl-protect
```

2. PATH 변수에 정의된 위치에 플러그인 바이너리를 복사합니다. 예를 들어 /usr/bin 또는 /usr/local/bin (상승된 권한이 필요할 수 있음):

```
cp ./tridentctl-protect /usr/local/bin/
```

3. 선택적으로 플러그인 바이너리를 홈 디렉토리의 위치에 복사할 수 있습니다. 이 경우 해당 위치가 PATH 변수의 일부인지 확인하는 것이 좋습니다:

```
cp ./tridentctl-protect ~/bin/
```



PATH 변수의 위치에 플러그인을 복사하면 어느 위치에서나 `tridentctl-protect` 또는 ``tridentctl protect``을 입력하여 플러그인을 사용할 수 있습니다.

## Trident CLI 플러그인 도움말 보기

내장된 플러그인 도움말 기능을 사용하여 플러그인의 기능에 대한 자세한 도움말을 얻을 수 있습니다.

단계

1. 도움말 기능을 사용하여 사용 지침을 확인합니다.

```
tridentctl-protect help
```

## 명령 자동 완성 활성화

Trident Protect CLI 플러그인을 설치한 후 특정 명령에 대해 자동 완성을 활성화할 수 있습니다.

## Bash 셸에 대한 자동 완성 활성화

### 단계

1. 완료 스크립트를 생성합니다.

```
tridentctl-protect completion bash > tridentctl-completion.bash
```

2. 스크립트를 포함할 새 디렉토리를 홈 디렉토리에 만듭니다.

```
mkdir -p ~/.bash/completions
```

3. 다운로드한 스크립트를 ~/.bash/completions 디렉토리로 이동합니다.

```
mv tridentctl-completion.bash ~/.bash/completions/
```

4. 홈 디렉토리의 ~/.bashrc 파일에 다음 줄을 추가합니다.

```
source ~/.bash/completions/tridentctl-completion.bash
```

## Z 셸에 대한 자동 완성 활성화

### 단계

1. 완료 스크립트를 생성합니다.

```
tridentctl-protect completion zsh > tridentctl-completion.zsh
```

2. 스크립트를 포함할 새 디렉토리를 홈 디렉토리에 만듭니다.

```
mkdir -p ~/.zsh/completions
```

3. 다운로드한 스크립트를 ~/.zsh/completions 디렉토리로 이동합니다.

```
mv tridentctl-completion.zsh ~/.zsh/completions/
```

4. 홈 디렉토리의 ~/.zprofile 파일에 다음 줄을 추가합니다.

```
source ~/.zsh/completions/tridentctl-completion.zsh
```

결과

다음 셸 로그인 시 `tridentctl-protect` 플러그인으로 명령 자동 완성을 사용할 수 있습니다.

## Trident Protect 설치 사용자 지정

사용자 환경의 특정 요구 사항을 충족하도록 Trident Protect의 기본 구성을 사용자 지정할 수 있습니다.

### Trident Protect 컨테이너 리소스 제한 지정

Trident Protect를 설치한 후 구성 파일을 사용하여 Trident Protect 컨테이너에 대한 리소스 제한을 지정할 수 있습니다. 리소스 제한을 설정하면 Trident Protect 작업에서 소비하는 클러스터 리소스의 양을 제어할 수 있습니다.

단계

1. `resourceLimits.yaml` 라는 이름의 파일을 생성하세요.
2. 환경의 요구 사항에 따라 Trident Protect 컨테이너에 대한 리소스 제한 옵션으로 파일을 채웁니다.

다음 예제 구성 파일에는 사용 가능한 설정이 표시되어 있으며 각 리소스 제한의 기본값이 포함되어 있습니다.

```
---
jobResources:
  defaults:
    limits:
      cpu: 8000m
      memory: 10000Mi
      ephemeralStorage: ""
    requests:
      cpu: 100m
      memory: 100Mi
      ephemeralStorage: ""
  resticVolumeBackup:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
  resticVolumeRestore:
    limits:
      cpu: ""
      memory: ""
      ephemeralStorage: ""
    requests:
```

```

    cpu: ""
    memory: ""
    ephemeralStorage: ""
kopiaVolumeBackup:
  limits:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
  requests:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
kopiaVolumeRestore:
  limits:
    cpu: ""
    memory: ""
    ephemeralStorage: ""
  requests:
    cpu: ""
    memory: ""
    ephemeralStorage: ""

```

3. resourceLimits.yaml 파일의 값을 적용합니다.

```

helm upgrade trident-protect -n trident-protect netapp-trident-protect/trident-protect -f resourceLimits.yaml --reuse-values

```

## 보안 컨텍스트 제약 조건 사용자 지정

구성 파일을 사용하여 Trident Protect 설치 후 Trident Protect 컨테이너에 대한 OpenShift 보안 컨텍스트 제약 조건(SCC)을 수정할 수 있습니다. 이러한 제약 조건은 Red Hat OpenShift 클러스터의 Pod에 대한 보안 제한을 정의합니다.

단계

1. `sccconfig.yaml`라는 이름의 파일을 생성하세요.
2. 파일에 SCC 옵션을 추가하고 환경 요구 사항에 따라 매개 변수를 수정합니다.

다음 예는 SCC 옵션의 매개변수 기본값을 보여줍니다.

```

scc:
  create: true
  name: trident-protect-job
  priority: 1

```

이 표에서는 SCC 옵션의 매개 변수에 대해 설명합니다.

매개변수	설명	기본값
생성	SCC 리소스를 생성할 수 있는지 여부를 결정합니다. `scc.create`이 `true`로 설정되어 있고 Helm 설치 프로세스가 OpenShift 환경을 식별하는 경우에만 SCC 리소스가 생성됩니다. OpenShift에서 작동하지 않거나 `scc.create`이 `false`로 설정되어 있는 경우 SCC 리소스가 생성되지 않습니다.	true
이름	SCC의 이름을 지정합니다.	trident-protect-job
우선순위	SCC의 우선순위를 정의합니다. 우선순위 값이 높은 SCC는 값이 낮은 SCC보다 먼저 평가됩니다.	1

### 3. sccconfig.yaml 파일의 값을 적용합니다.

```
helm upgrade trident-protect -n trident-protect netapp-trident-protect/trident-protect -f sccconfig.yaml --reuse-values
```

이렇게 하면 기본값이 sccconfig.yaml 파일에 지정된 값으로 대체됩니다.

## Trident Protect 헬름 차트의 추가 설정을 구성합니다

AutoSupport 설정 및 네임스페이스 필터링을 사용자 지정하여 특정 요구 사항을 충족할 수 있습니다. 다음 표는 사용 가능한 구성 매개변수를 설명합니다.

매개변수	유형	설명
autoSupport.프록시	문자열	NetApp AutoSupport 연결을 위한 프록시 URL을 구성합니다. 이 설정을 사용하여 지원 번들 업로드를 프록시 서버를 통해 라우팅합니다. 예: <a href="http://my.proxy.url">http://my.proxy.url</a> .
autoSupport.insecure	boolean	AutoSupport 프록시 연결에 대해 `true`로 설정하면 TLS 검증을 건너뜁니다. 안전하지 않은 프록시 연결에만 사용하십시오. (기본값: `false`)

매개변수	유형	설명
autoSupport.enabled	boolean	Trident Protect AutoSupport 번들 일일 업로드 기능을 활성화 또는 비활성화합니다. <code>false</code> 로 설정하면 예약된 일일 업로드가 비활성화되지만 수동으로 지원 번들을 생성할 수는 있습니다. (기본값: <code>true</code> )
restoreSkipNamespaceAnnotations	문자열	백업 및 복원 작업에서 제외할 네임스페이스 주석의 심표로 구분된 목록입니다. 주석을 기반으로 네임스페이스를 필터링할 수 있습니다.
restoreSkipNamespaceLabels	문자열	백업 및 복원 작업에서 제외할 네임스페이스 레이블을 심표로 구분하여 나열합니다. 레이블을 기준으로 네임스페이스를 필터링할 수 있습니다.

YAML 구성 파일 또는 명령줄 플래그를 사용하여 이러한 옵션을 구성할 수 있습니다.

## YAML 파일 사용

### 단계

1. 구성 파일을 생성하고 이름을 지정하세요 `values.yaml`.
2. 생성한 파일에 사용자 지정하려는 구성 옵션을 추가하세요.

```
autoSupport:
  enabled: false
  proxy: http://my.proxy.url
  insecure: true
restoreSkipNamespaceAnnotations: "annotation1,annotation2"
restoreSkipNamespaceLabels: "label1,label2"
```

3. `values.yaml` 파일에 올바른 값을 입력한 후 구성 파일을 적용합니다.

```
helm upgrade trident-protect -n trident-protect netapp-trident-protect/trident-protect -f values.yaml --reuse-values
```

## CLI 플래그 사용

### 단계

1. `--set` 플래그와 함께 다음 명령을 사용하여 개별 매개변수를 지정하십시오.

```
helm upgrade trident-protect -n trident-protect netapp-trident-protect/trident-protect \
  --set autoSupport.enabled=false \
  --set autoSupport.proxy=http://my.proxy.url \
  --set-string
restoreSkipNamespaceAnnotations="{annotation1,annotation2}" \
  --set-string restoreSkipNamespaceLabels="{label1,label2}" \
  --reuse-values
```

## Trident Protect Pod를 특정 노드로 제한합니다

Kubernetes `nodeSelector` 노드 선택 제약 조건을 사용하여 노드 레이블을 기반으로 Trident Protect Pod를 실행할 수 있는 노드를 제어할 수 있습니다. 기본적으로 Trident Protect는 Linux를 실행하는 노드로 제한됩니다. 필요에 따라 이러한 제약 조건을 추가로 사용자 지정할 수 있습니다.

### 단계

1. `nodeSelectorConfig.yaml` 라는 이름의 파일을 생성하세요.
2. 파일에 `nodeSelector` 옵션을 추가하고 환경 요구 사항에 따라 제한하도록 노드 레이블을 추가하거나 변경하도록 파일을 수정합니다. 예를 들어, 다음 파일에는 기본 OS 제한이 포함되어 있지만 특정 지역 및 앱 이름도 대상으로

지정합니다.

```
nodeSelector:  
  kubernetes.io/os: linux  
  region: us-west  
  app.kubernetes.io/name: mysql
```

3. nodeSelectorConfig.yaml 파일의 값을 적용합니다.

```
helm upgrade trident-protect -n trident-protect netapp-trident-  
protect/trident-protect -f nodeSelectorConfig.yaml --reuse-values
```

이렇게 하면 기본 제한 사항이 nodeSelectorConfig.yaml 파일에 지정한 제한 사항으로 대체됩니다.

## 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.