



# NetApp ONTAP용 Amazon FSx Trident

NetApp  
February 02, 2026

# 목차

NetApp ONTAP용 Amazon FSx .....	1
Trident를 Amazon FSx for NetApp ONTAP와 함께 사용해 보십시오 .....	1
요구 사항 .....	1
고려 사항 .....	1
인증 .....	2
테스트된 아마존 머신 이미지(AMI) .....	2
자세한 내용을 확인하십시오 .....	3
IAM 역할 및 AWS Secret을 생성합니다 .....	3
AWS Secrets Manager 암호를 생성합니다 .....	4
IAM 정책을 생성합니다 .....	4
Trident를 설치합니다 .....	8
Helm을 통해 Trident를 설치합니다 .....	9
EKS 애드온을 통해 Trident를 설치합니다 .....	11
스토리지 백엔드를 구성합니다 .....	17
ONTAP SAN 및 NAS 드라이버 통합 .....	17
FSx for ONTAP 드라이버 세부 정보 .....	19
백엔드 고급 구성 및 예 .....	20
볼륨 프로비저닝을 위한 백엔드 구성 옵션 .....	24
SMB 볼륨 제공 .....	25
저장소 클래스 및 PVC를 구성합니다 .....	25
스토리지 클래스를 생성합니다 .....	25
PVC를 작성합니다 .....	27
Trident 특성 .....	29
샘플 응용 프로그램을 배포합니다 .....	30
EKS 클러스터에서 Trident EKS 애드온을 구성합니다 .....	31
필수 구성 요소 .....	31
단계 .....	31
CLI를 사용하여 Trident EKS 애드온을 설치/제거합니다 .....	34

# NetApp ONTAP용 Amazon FSx

## Trident를 Amazon FSx for NetApp ONTAP와 함께 사용해 보십시오

"[NetApp ONTAP용 Amazon FSx](#)" NetApp ONTAP 스토리지 운영 체제가 제공하는 파일 시스템을 실행하고 실행할 수 있도록 완벽하게 관리되는 AWS 서비스입니다. ONTAP용 FSx를 사용하면 익숙한 NetApp 기능, 성능 및 관리 기능을 활용하는 동시에, AWS에 데이터를 저장하는 데 따른 단순성, 민첩성, 보안, 확장성을 활용할 수 있습니다. ONTAP용 FSx는 ONTAP 파일 시스템 기능 및 관리 API를 지원합니다.

Amazon FSx for NetApp ONTAP 파일 시스템을 Trident와 통합하여 Amazon EKS(Elastic Kubernetes Service)에서 실행되는 Kubernetes 클러스터가 ONTAP에서 지원하는 블록 및 파일 영구 볼륨을 프로비저닝할 수 있도록 할 수 있습니다.

파일 시스템은 Amazon FSx의 주요 리소스이며, 이는 사내 ONTAP 클러스터와 유사합니다. 각 SVM 내에서 파일 시스템에 파일과 폴더를 저장하는 데이터 컨테이너인 하나 이상의 볼륨을 생성할 수 있습니다. Amazon FSx for NetApp ONTAP은 클라우드에서 관리형 파일 시스템으로 제공됩니다. 새로운 파일 시스템 유형을 \* NetApp ONTAP \* 라고 합니다.

Trident를 Amazon FSx for NetApp ONTAP와 함께 사용하면 Amazon EKS(Elastic Kubernetes Service)에서 실행되는 Kubernetes 클러스터가 ONTAP에서 지원하는 블록 및 파일 영구 볼륨을 프로비저닝할 수 있습니다.

### 요구 사항

"[Trident 요구사항](#)" FSx for ONTAP를 Trident와 통합하려면 다음이 필요합니다.

- kubctl이 설치된 기존 Amazon EKS 클러스터 또는 자체 관리 Kubernetes 클러스터
- 클러스터의 작업자 노드에서 연결할 수 있는 NetApp ONTAP 파일 시스템용 기존 Amazon FSx 및 SVM(Storage Virtual Machine).
- 에 대해 준비된 작업자 노드입니다 "[NFS 또는 iSCSI](#)".



Amazon Linux 및 Ubuntu에 필요한 노드 준비 단계를 따라야 합니다 "[Amazon Machine Images\(아마존 머신 이미지\)](#)" (AMI) EKS AMI 유형에 따라 다릅니다.

### 고려 사항

- SMB 볼륨:
  - SMB 볼륨은 를 사용하여 지원됩니다 `ontap-nas` 드라이버만 해당.
  - Trident EKS 애드온에서는 SMB 볼륨이 지원되지 않습니다.
  - Trident는 Windows 노드에서만 실행되는 Pod에 마운트된 SMB 볼륨을 지원합니다. 자세한 내용은 을 "[SMB 볼륨 프로비저닝을 위한 준비](#)" 참조하십시오.
- Trident 24.02 이전에는 자동 백업이 활성화된 Amazon FSx 파일 시스템에서 생성된 볼륨을 Trident에서 삭제할 수 없었습니다. Trident 24.02 이상에서 이 문제를 방지하려면 `fsxFilesystemID` AWS FSx for ONTAP의 백엔드 구성 파일에, AWS, `apikey` AWS `apiRegion` 및 AWS를 `secretKey` 지정합니다.



Trident에 IAM 역할을 지정하는 경우, `apiKey` 및 `secretKey` 필드를 명시적으로 Trident에 지정하지 않아도 됩니다 `apiRegion`. 자세한 내용은 ["ONTAP 구성 옵션 및 예제용 FSX"](#) 참조하십시오.

## Trident SAN/iSCSI 및 EBS-CSI 드라이버 동시 사용

AWS(EKS, ROSA, EC2 또는 기타 인스턴스)에서 `ontap-san` 드라이버(예: iSCSI)를 사용하려는 경우 노드에 필요한 다중 경로 구성이 Amazon Elastic Block Store(EBS) CSI 드라이버와 충돌할 수 있습니다. 동일한 노드에 있는 EBS 디스크를 방해하지 않고 멀티패스 기능을 보장하려면 멀티패스 설정에서 EBS를 제외해야 합니다. 이 예에서는 다음을 보여줍니다. `multipath.conf` EBS 디스크를 다중 경로에서 제외하면서 필수 Trident 설정을 포함하는 파일:

```
defaults {
    find_multipaths no
}
blacklist {
    device {
        vendor "NVME"
        product "Amazon Elastic Block Store"
    }
}
```

## 인증

Trident는 두 가지 인증 모드를 제공합니다.

- 자격 증명 기반(권장): 자격 증명을 AWS Secrets Manager에 안전하게 저장합니다. 파일 시스템 또는 SVM에 구성된 사용자를 사용할 수 있습니다 `fsxadmin` `vsadmin`.



Trident은 SVM 사용자로 실행하거나 동일한 역할을 가진 다른 이름의 사용자로 실행해야 `vsadmin` 합니다. Amazon FSx for NetApp ONTAP에는 `fsxadmin` ONTAP 클러스터 사용자를 제한적으로 대체하는 사용자가 `admin` 있습니다. Trident와 함께 를 사용하는 `vsadmin` 것이 좋습니다.

- 인증서 기반: Trident은 SVM에 설치된 인증서를 사용하여 FSx 파일 시스템의 SVM과 통신합니다.

인증 활성화에 대한 자세한 내용은 드라이버 유형에 대한 인증을 참조하십시오.

- ["ONTAP NAS 인증"](#)
- ["ONTAP SAN 인증"](#)

## 테스트된 아마존 머신 이미지(AMI)

EKS 클러스터는 다양한 운영 체제를 지원하지만 AWS는 컨테이너 및 EKS에 대해 특정 AMI(Amazon Machine Images)를 최적화했습니다. 다음 AMI는 NetApp Trident 25.02에서 테스트되었습니다.

아미	NAS	NAS - 경제형	iSCSI	iSCSI 경제
----	-----	-----------	-------	----------

AL2023_x86_64_ST ANDARD를 참조하십시오	예	예	예	예
AL2_x86_64를 참조하십시오	예	예	예 *	예 *
BOTTLEROCKET_x 86_64를 참조하십시오	예**	예	해당 없음	해당 없음
AL2023_ARM_64_S TANDARD를 참조하십시오	예	예	예	예
AL2_ARM_64를 참조하십시오	예	예	예 *	예 *
BOTTLEROCKET_A RM_64를 참조하십시오	예**	예	해당 없음	해당 없음

- \* 노드를 재시작하지 않고는 PV를 삭제할 수 없습니다.
- \*\* Trident 버전 25.02에서는 NFSv3와 작동하지 않습니다.



원하는 AMI가 여기에 나열되지 않은 경우 지원되지 않는다는 의미는 아닙니다. 단순히 테스트를 거치지 않았음을 의미합니다. 이 목록은 AMI가 작동하는 방법에 대한 가이드 역할을 합니다.

- 다음을 사용하여 수행한 테스트:
- EKS 버전: 1.32
- 설치 방법: Helm 25.06 및 AWS 추가 기능 25.06
- NAS의 경우 NFSv3과 NFSv4.1이 모두 테스트되었습니다.
- SAN 전용 iSCSI는 테스트되었으며 NVMe-oF는 테스트되지 않았습니다.
- 수행된 테스트 \*:
- 생성 : 저장 클래스, PVC, POD
- 삭제: Pod, PVC(일반, qtree/LUN – 경제성, NAS와 AWS 백업)

자세한 내용을 확인하십시오

- ["NetApp ONTAP용 Amazon FSx 문서"](#)
- ["NetApp ONTAP용 Amazon FSx 블로그 게시물"](#)

## IAM 역할 및 AWS Secret을 생성합니다

AWS 자격 증명을 명시적으로 제공하는 대신 AWS IAM 역할로 인증하여 Kubernetes Pod를 구성하여 AWS 리소스에 액세스할 수 있습니다.



AWS IAM 역할을 사용하여 인증하려면 EKS를 사용하여 Kubernetes 클러스터를 구축해야 합니다.

## AWS Secrets Manager 암호를 생성합니다

Trident는 사용자를 위해 스토리지 관리를 위해 FSx 가상 서버에 대해 API를 발행하므로 이를 위해 자격 증명이 필요합니다. 이러한 자격 증명을 전달하는 안전한 방법은 AWS Secrets Manager 암호를 사용하는 것입니다. 따라서 아직 계정이 없는 경우 vsadmin 계정의 자격 증명이 포함된 AWS Secrets Manager 암호를 생성해야 합니다.

이 예에서는 Trident CSI 자격 증명을 저장하기 위한 AWS Secrets Manager 암호를 생성합니다.

```
aws secretsmanager create-secret --name trident-secret --description
"Trident CSI credentials"\
    --secret-string
"{\"username\": \"vsadmin\", \"password\": \"<svmpassword>\"}"
```

## IAM 정책을 생성합니다

Trident를 올바르게 실행하려면 AWS 권한도 필요합니다. 따라서 Trident에 필요한 사용 권한을 부여하는 정책을 만들어야 합니다.

다음 예에서는 AWS CLI를 사용하여 IAM 정책을 생성합니다.

```
aws iam create-policy --policy-name AmazonFSxNCSIDriverPolicy --policy
-document file://policy.json
    --description "This policy grants access to Trident CSI to FSxN and
Secrets manager"
```

- 정책 JSON 예 \*:

```

{
  "Statement": [
    {
      "Action": [
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes",
        "fsx:CreateVolume",
        "fsx:RestoreVolumeFromSnapshot",
        "fsx:DescribeStorageVirtualMachines",
        "fsx:UntagResource",
        "fsx:UpdateVolume",
        "fsx:TagResource",
        "fsx>DeleteVolume"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Effect": "Allow",
      "Resource": "arn:aws:secretsmanager:<aws-region>:<aws-account-id>:secret:<aws-secret-manager-name>*"
    }
  ],
  "Version": "2012-10-17"
}

```

서비스 계정 연결(IRSA)을 위한 **Pod ID** 또는 **IAM** 역할 생성

Kubernetes 서비스 계정이 EKS Pod Identity를 사용하는 AWS Identity and Access Management(IAM) 역할 또는 서비스 계정 연결(IRSA)을 위한 IAM 역할을 맡도록 구성할 수 있습니다. 서비스 계정을 사용하도록 구성된 모든 Pod는 해당 역할에 액세스 권한이 있는 모든 AWS 서비스에 액세스할 수 있습니다.

## 포드 아이덴티티

Amazon EKS Pod Identity 연결은 Amazon EC2 인스턴스 프로필이 Amazon EC2 인스턴스에 자격 증명을 제공하는 방식과 유사하게 애플리케이션의 자격 증명을 관리하는 기능을 제공합니다.

### EKS 클러스터에 Pod Identity 설치:

AWS 콘솔을 통해 또는 다음 AWS CLI 명령을 사용하여 Pod ID를 생성할 수 있습니다.

```
aws eks create-addon --cluster-name <EKS_CLUSTER_NAME> --addon-name
eks-pod-identity-agent
```

자세한 내용은 다음을 참조하세요. ["Amazon EKS Pod Identity Agent 설정"](#).

### trust-relationship.json을 생성합니다:

EKS 서비스 주체가 Pod Identity에 대한 이 역할을 수행할 수 있도록 trust-relationship.json 파일을 생성하세요. 그런 다음 다음 신뢰 정책을 사용하여 역할을 생성하세요.

```
aws iam create-role \
  --role-name fsxn-csi-role --assume-role-policy-document file://trust-
relationship.json \
  --description "fsxn csi pod identity role"
```

### trust-relationship.json 파일:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "pods.eks.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

### IAM 역할에 역할 정책 첨부:

이전 단계의 역할 정책을 생성된 IAM 역할에 연결합니다.



```
aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:111122223333:policy/fsxn-csi-policy \
  --role-name fsxn-csi-role
```

**포드 ID 연결 생성:**

IAM 역할과 Trident 서비스 계정(trident-controller) 간에 Pod ID 연결을 생성합니다.

```
aws eks create-pod-identity-association \
  --cluster-name <EKS_CLUSTER_NAME> \
  --role-arn arn:aws:iam::111122223333:role/fsxn-csi-role \
  --namespace trident --service-account trident-controller
```

서비스 계정 연결(IRSA)을 위한 IAM 역할

**AWS CLI 사용:**

```
aws iam create-role --role-name AmazonEKS_FSxN_CSI_DriverRole \
  --assume-role-policy-document file://trust-relationship.json
```

• trust-relationship.json 파일: \*

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::<account_id>:oidc-provider/<oidc_provider>"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "<oidc_provider>:aud": "sts.amazonaws.com",
          "<oidc_provider>:sub":
            "system:serviceaccount:trident:trident-controller"
        }
      }
    }
  ]
}
```

파일에 다음 값을 trust-relationship.json 업데이트합니다.

- \* <account\_id> \* - AWS 계정 ID
- \* <oidc\_provider> \* - EKS 클러스터의 OIDC. 다음을 실행하여 oidc\_provider를 가져올 수 있습니다.

```
aws eks describe-cluster --name my-cluster --query
"cluster.identity.oidc.issuer"\
--output text | sed -e "s/^https:\\/\\/\\/"
```

- IAM 정책에 IAM 역할 연결 \*:

역할이 생성되면 다음 명령을 사용하여 정책(위 단계에서 만든 정책)을 역할에 연결합니다.

```
aws iam attach-role-policy --role-name my-role --policy-arn <IAM policy
ARN>
```

- OICD 공급자가 연결되었는지 확인 \*:

OICD 공급자가 클러스터와 연결되어 있는지 확인합니다. 다음 명령을 사용하여 확인할 수 있습니다.

```
aws iam list-open-id-connect-providers | grep $oidc_id | cut -d "/" -f4
```

출력이 비어 있는 경우 다음 명령을 사용하여 IAM OIDC를 클러스터에 연결합니다.

```
eksctl utils associate-iam-oidc-provider --cluster $cluster_name
--approve
```

eksctl을 사용하는 경우 다음 예를 사용하여 EKS의 서비스 계정에 대한 IAM 역할을 생성하세요.

```
eksctl create iamserviceaccount --name trident-controller --namespace
trident \
--cluster <my-cluster> --role-name AmazonEKS_FSxN_CSI_DriverRole
--role-only \
--attach-policy-arn <IAM-Policy ARN> --approve
```

## Trident를 설치합니다

Trident은 Kubernetes에서 Amazon FSx for NetApp ONTAP 스토리지 관리를 간소화하여 개발자와 관리자가 애플리케이션 구축에 집중할 수 있도록 지원합니다.

다음 방법 중 하나를 사용하여 Trident를 설치할 수 있습니다.

- 헬름
- EKS 추가 기능

스냅샷 기능을 사용하려면 CSI 스냅샷 컨트롤러 애드온을 설치하십시오. 자세한 내용은 ["CSI 볼륨에 대해 스냅샷 기능을 활성화합니다"](#) 참조하십시오.

**Helm**을 통해 **Trident**를 설치합니다

## 포드 아이덴티티

### 1. Trident Helm 저장소 추가:

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

### 2. 다음 예를 사용하여 Trident를 설치하세요.

```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 --namespace trident --create-namespace
```

명령을 사용하여 이름, 네임스페이스, 차트, 상태, 앱 버전 및 수정 번호와 같은 설치 세부 정보를 검토할 수 `helm list` 있습니다.

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14
14:31:22.463122 +0300 IDT		deployed	trident-operator-
100.2502.0	25.02.0		

## 서비스 계정 연결(IRSA)

### 1. Trident Helm 저장소 추가:

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

### 2. 클라우드 공급자 및 \*클라우드 ID\*에 대한 값을 설정합니다.

```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 \  
--set cloudProvider="AWS" \  
--set cloudIdentity="'eks.amazonaws.com/role-arn:  
arn:aws:iam::<accountID>:role/<AmazonEKS_FSxN_CSI_DriverRole>' " \  
--namespace trident \  
--create-namespace
```

명령을 사용하여 이름, 네임스페이스, 차트, 상태, 앱 버전 및 수정 번호와 같은 설치 세부 정보를 검토할 수 `helm list` 있습니다.

```
helm list -n trident
```

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART		APP VERSION
trident-operator	trident	1	2024-10-14
14:31:22.463122 +0300	IDT	deployed	trident-operator-
100.2510.0	25.10.0		

iSCSI를 사용하려면 클라이언트 머신에서 iSCSI가 활성화되어 있는지 확인하세요. AL2023 Worker 노드 OS를 사용하는 경우, helm 설치 시 node prep 매개변수를 추가하여 iSCSI 클라이언트 설치를 자동화할 수 있습니다.



```
helm install trident-operator netapp-trident/trident-operator  
--version 100.2502.1 --namespace trident --create-namespace --  
set nodePrep={iscsi}
```

## EKS 애드온을 통해 Trident를 설치합니다

Trident EKS 애드온에는 최신 보안 패치 및 버그 수정이 포함되어 있으며 AWS에서 Amazon EKS와 함께 사용할 수 있다는 것이 검증되었습니다. EKS 애드온을 사용하면 Amazon EKS 클러스터의 보안과 안정성을 지속적으로 보장하고 애드온을 설치, 구성 및 업데이트하는 데 필요한 작업량을 줄일 수 있습니다.

### 필수 구성 요소

AWS EKS용 Trident 애드온을 구성하기 전에 다음 사항이 있는지 확인하십시오.

- 애드온 가입이 있는 Amazon EKS 클러스터 계정입니다
- AWS 마켓플레이스에 대한 AWS 권한:  
"aws-marketplace:ViewSubscriptions",  
"aws-marketplace:Subscribe",  
"aws-marketplace:Unsubscribe"
- AMI 유형: Amazon Linux 2 (AL2\_x86\_64) 또는 Amazon Linux 2 Arm (AL2\_ARM\_64)
- 노드 유형: AMD 또는 ARM
- 기존 Amazon FSx for NetApp ONTAP 파일 시스템

**AWS**에 대해 **Trident** 애드온을 활성화합니다

관리 콘솔과 직접 연결되어 있습니다

1. 에서 Amazon EKS 콘솔을 엽니다 <https://console.aws.amazon.com/eks/home#/clusters>.
2. 왼쪽 탐색 창에서 \* 클러스터 \* 를 선택합니다.
3. NetApp Trident CSI 추가 기능을 구성할 클러스터의 이름을 선택합니다.
4. Add-ons \* 를 선택한 다음 \* Get more add-ons \* 를 선택합니다.
5. 추가 기능을 선택하려면 다음 단계를 따르세요.
  - a. **AWS Marketplace** 추가 기능 섹션까지 아래로 스크롤하여 검색 상자에 **"Trident"**를 입력합니다.
  - b. Trident by NetApp 상자의 오른쪽 상단에 있는 확인란을 선택하세요.
  - c. 다음 \* 을 선택합니다.
6. 선택한 추가 기능 구성 \* 설정 페이지에서 다음을 수행합니다.



**Pod Identity** 연결을 사용하는 경우 이 단계를 건너뛰니다.

- a. 사용할 \* 버전 \* 을 선택합니다.
- b. IRSA 인증을 사용하는 경우 선택적 구성 설정에서 사용 가능한 구성 값을 설정해야 합니다.
  - 사용할 \* 버전 \* 을 선택합니다.
  - 추가 기능 구성 스키마\*를 따르고 \*구성 값 섹션의 **configurationValues** 매개변수를 이전 단계에서 만든 role-arn으로 설정합니다(값은 다음 형식이어야 함).

```
{  
  
  "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",  
  "cloudProvider": "AWS"  
  
}
```

+

충돌 해결 방법으로 재정의 를 선택한 경우 기존 애드온에 대한 하나 이상의 설정을 Amazon EKS 애드온 설정으로 덮어쓸 수 있습니다. 이 옵션을 사용하지 않고 기존 설정과 충돌하는 경우 작업이 실패합니다. 결과 오류 메시지를 사용하여 충돌 문제를 해결할 수 있습니다. 이 옵션을 선택하기 전에 Amazon EKS 추가 기능이 자체 관리해야 하는 설정을 관리하지 않는지 확인하십시오.

7. 다음 \* 을 선택합니다.
8. 검토 및 추가 \* 페이지에서 \* 만들기 \* 를 선택합니다.

추가 기능 설치가 완료되면 설치된 추가 기능이 표시됩니다.

**AWS CLI**를 참조하십시오

- 1. 생성하다 add-on.json 파일\*:

**Pod Identity**의 경우 다음 형식을 사용하세요:



다음을 사용하세요

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
}
```

IRSA 인증의 경우 다음 형식을 사용하세요:

```
{
  "clusterName": "<eks-cluster>",
  "addonName": "netapp_trident-operator",
  "addonVersion": "v25.6.0-eksbuild.1",
  "serviceAccountRoleArn": "<role ARN>",
  "configurationValues": {
    "cloudIdentity": "'eks.amazonaws.com/role-arn: <role ARN>'",
    "cloudProvider": "AWS"
  }
}
```



`<role ARN>`이전 단계에서 생성한 역할의 ARN으로 바꿉니다.

- 2. Trident EKS 애드온을 설치하세요.\*

```
aws eks create-addon --cli-input-json file://add-on.json
```

**eksctl**입니다

다음 명령 예에서는 Trident EKS 추가 기능을 설치합니다.

```
eksctl create addon --name netapp_trident-operator --cluster
<cluster_name> --force
```

**Trident EKS** 추가 기능을 업데이트합니다



관리 콘솔과 직접 연결되어 있습니다

1. Amazon EKS 콘솔을 <https://console.aws.amazon.com/eks/home#/clusters>입니다.
2. 왼쪽 탐색 창에서 \* 클러스터 \* 를 선택합니다.
3. NetApp Trident CSI 애드온을 업데이트할 클러스터의 이름을 선택합니다.
4. Add-ons \* 탭을 선택합니다.
5. Trident by NetApp \* 를 선택한 다음 \* 편집 \* 을 선택합니다.
6. Trident by NetApp \* 구성 페이지에서 다음을 수행합니다.
  - a. 사용할 \* 버전 \* 을 선택합니다.
  - b. 선택적 구성 설정 \* 을 확장하고 필요에 따라 수정합니다.
  - c. 변경 내용 저장 \* 을 선택합니다.

#### AWS CLI를 참조하십시오

다음 예에서는 EKS 추가 기능을 업데이트합니다.

```
aws eks update-addon --cluster-name <eks_cluster_name> --addon-name
netapp_trident-operator --addon-version v25.6.0-eksbuild.1 \
--service-account-role-arn <role-ARN> --resolve-conflict preserve \
--configuration-values "{\"cloudIdentity\":
\"'eks.amazonaws.com/role-arn: <role ARN>'\"}"
```

#### eksctl입니다

- FSxN Trident CSI 추가 기능의 현재 버전을 확인합니다. 클러스터 이름으로 교체합니다 my-cluster.

```
eksctl get addon --name netapp_trident-operator --cluster my-cluster
```

- 출력 예: \*

NAME	VERSION	STATUS	ISSUES
IAMROLE	UPDATE AVAILABLE	CONFIGURATION VALUES	
netapp_trident-operator	v25.6.0-eksbuild.1	ACTIVE	0
{\"cloudIdentity\":\"'eks.amazonaws.com/role-arn: arn:aws:iam::139763910815:role/AmazonEKS_FSXN_CSI_DriverRole'\"}			

- 이전 단계의 출력에서 사용할 수 있는 업데이트 아래에 반환된 버전으로 추가 기능을 업데이트합니다.

```
eksctl update addon --name netapp_trident-operator --version
v25.6.0-eksbuild.1 --cluster my-cluster --force
```

옵션을 제거하고 Amazon EKS 추가 기능 설정이 기존 설정과 충돌하는 경우 `--force` Amazon EKS 추가 기능 업데이트가 실패하고 충돌 문제를 해결하는 데 도움이 되는 오류 메시지가 표시됩니다. 이 옵션을 지정하기 전에 Amazon EKS 애드온이 관리해야 하는 설정을 관리하지 않는지 확인하십시오. 이러한 설정은 이 옵션으로 덮어쓰이기 때문입니다. 이 설정의 다른 옵션에 대한 자세한 내용은 을 참조하십시오 "[추가 기능](#)". Amazon EKS Kubernetes 필드 관리에 대한 자세한 내용은 를 참조하십시오 "[Kubernetes 현장 관리](#)".

## Trident EKS 추가 기능을 제거/제거합니다

Amazon EKS 애드온을 제거하는 두 가지 옵션이 있습니다.

- \* 클러스터에 애드온 소프트웨어 유지 \* – 이 옵션은 모든 설정의 Amazon EKS 관리를 제거합니다. 또한 업데이트를 시작한 후 Amazon EKS에서 업데이트를 알리고 Amazon EKS 애드온을 자동으로 업데이트하는 기능도 제거합니다. 하지만 클러스터에 애드온 소프트웨어가 보존됩니다. 이 옵션을 사용하면 Amazon EKS 애드온이 아닌 자가 관리형 설치가 됩니다. 이 옵션을 사용하면 애드온에 대한 다운타임이 없습니다. `--preserve` 명령의 옵션을 유지하여 추가 기능을 유지합니다.
- \* 클러스터에서 애드온 소프트웨어 완전히 제거 \* – NetApp는 클러스터에 종속된 리소스가 없는 경우에만 클러스터에서 Amazon EKS 애드온을 제거할 것을 권장합니다. `--preserve` 추가 기능을 제거하려면 명령에서 옵션을 `delete` 제거하십시오.



애드온에 IAM 계정이 연결되어 있으면 IAM 계정이 제거되지 않습니다.

관리 콘솔과 직접 연결되어 있습니다

1. 에서 Amazon EKS 콘솔을 엽니다 <https://console.aws.amazon.com/eks/home#/clusters>.
2. 왼쪽 탐색 창에서 \* 클러스터 \* 를 선택합니다.
3. NetApp Trident CSI 추가 기능을 제거할 클러스터의 이름을 선택합니다.
4. 추가 기능 \* 탭을 선택한 다음 \* Trident by NetApp \* . \* 를 선택합니다
5. 제거 \* 를 선택합니다.
6. Remove netapp\_trident-operator confirmation \* 대화 상자에서 다음을 수행합니다.
  - a. Amazon EKS가 애드온에 대한 설정 관리를 중지하도록 하려면 \* 클러스터에서 유지 \* 를 선택합니다. 추가 기능의 모든 설정을 직접 관리할 수 있도록 클러스터에 추가 소프트웨어를 유지하려는 경우 이 작업을 수행합니다.
  - b. netapp\_trident-operator \* 를 입력합니다.
  - c. 제거 \* 를 선택합니다.

#### AWS CLI를 참조하십시오

클러스터 이름으로 바꾸고 my-cluster 다음 명령을 실행합니다.

```
aws eks delete-addon --cluster-name my-cluster --addon-name  
netapp_trident-operator --preserve
```

#### eksctl입니다

다음 명령을 실행하면 Trident EKS 추가 기능이 제거됩니다.

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

## 스토리지 백엔드를 구성합니다

### ONTAP SAN 및 NAS 드라이버 통합

스토리지 백엔드를 생성하려면 JSON 또는 YAML 형식으로 구성 파일을 만들어야 합니다. 파일에서 원하는 스토리지 유형(NAS 또는 SAN), 파일 시스템, SVM을 가져와 인증 방법을 지정해야 합니다. 다음 예제는 NAS 기반 스토리지를 정의하고 AWS 암호를 사용하여 사용하려는 SVM에 자격 증명을 저장하는 방법을 보여줍니다.

## YAML

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  backendName: tbc-ontap-nas
  svm: svm-name
  aws:
    fsxFilesystemID: fs-xxxxxxxxxx
  credentials:
    name: "arn:aws:secretsmanager:us-west-2:xxxxxxx:secret:secret-
name"
    type: awsarn
```

## JSON을 참조하십시오

```
{
  "apiVersion": "trident.netapp.io/v1",
  "kind": "TridentBackendConfig",
  "metadata": {
    "name": "backend-tbc-ontap-nas"
    "namespace": "trident"
  },
  "spec": {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "tbc-ontap-nas",
    "svm": "svm-name",
    "aws": {
      "fsxFilesystemID": "fs-xxxxxxxxxx"
    },
    "managementLIF": null,
    "credentials": {
      "name": "arn:aws:secretsmanager:us-west-2:xxxxxxx:secret:secret-
name",
      "type": "awsarn"
    }
  }
}
```

다음 명령을 실행하여 Trident TBC(백엔드 구성)를 생성하고 검증합니다.

- YAML 파일에서 TBC(Trident 백엔드 구성)를 생성하고 다음 명령을 실행합니다.

```
kubectl create -f backendconfig.yaml -n trident
```

```
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-nas created
```

- Trident 백엔드 구성(TBC)이 성공적으로 생성되었는지 확인:

```
Kubectl get tbc -n trident
```

NAME	BACKEND NAME	BACKEND UUID
PHASE	STATUS	
backend-tbc-ontap-nas	tbc-ontap-nas	933e0071-66ce-4324-
b9ff-f96d916ac5e9	Bound	Success

## FSx for ONTAP 드라이버 세부 정보

다음 드라이버를 사용하여 Trident를 Amazon FSx for NetApp ONTAP와 통합할 수 있습니다.

- **ontap-san**: 프로비저닝된 각 PV는 자체 Amazon FSx for NetApp ONTAP 볼륨 내의 LUN입니다. 블록 스토리지에 권장됩니다.
- **ontap-nas**: 프로비저닝된 각 PV는 전체 Amazon FSx for NetApp ONTAP 볼륨입니다. NFS 및 SMB에 권장됩니다.
- **'ONTAP-SAN-이코노미'**: 프로비저닝되는 각 PV는 NetApp ONTAP 볼륨에 대해 Amazon FSx당 구성 가능한 LUN 수를 가진 LUN입니다.
- **'ONTAP-NAS-E경제적인'**: 각 PV 프로비저닝은 qtree이며, NetApp ONTAP 볼륨에 대해 Amazon FSx당 qtree를 구성할 수 있습니다.
- **'ONTAP-NAS-flexgroup'**: 프로비저닝되는 각 PV는 NetApp ONTAP FlexGroup 볼륨에 대한 전체 Amazon FSx입니다.

드라이버에 대한 자세한 내용은 을 참조하십시오 ["NAS 드라이버"](#) 및 ["SAN 드라이버"](#).

구성 파일이 생성되면 다음 명령을 실행하여 EKS 내에 구성 파일을 생성합니다.

```
kubectl create -f configuration_file
```

상태를 확인하려면 다음 명령을 실행합니다.

```
kubectl get tbc -n trident
```

NAME	BACKEND NAME	BACKEND UUID
PHASE      STATUS		
backend-fsx-ontap-nas	backend-fsx-ontap-nas	7a551921-997c-4c37-a1d1-f2f4c87fa629
Bound	Success	

## 백엔드 고급 구성 및 예

백엔드 구성 옵션은 다음 표를 참조하십시오.

매개 변수	설명	예
'내전'		항상 1
'storageDriverName'입니다	스토리지 드라이버의 이름입니다	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, ontap-san-economy
백엔드이름	사용자 지정 이름 또는 스토리지 백엔드	드라이버 이름 + "_" + dataLIF
마나멘타LIF	클러스터 또는 SVM 관리 LIF의 IP 주소 정규화된 도메인 이름(FQDN)을 지정할 수 있습니다. IPv6 플래그를 사용하여 Trident가 설치된 경우 IPv6 주소를 사용하도록 설정할 수 있습니다. IPv6 주소는 대괄호로 묶어야 합니다(예: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]). aws` 현장에서 을 제공하는 경우 `fsxFilesystemID, Trident이 AWS에서 SVM 정보를 검색하기 때문에 를 제공할 필요가 없습니다 managementLIF. managementLIF 따라서 SVM에서 사용자에게 대한 자격 증명(예: vsadmin)을 제공해야 하며 사용자에게 역할이 있어야 합니다. vsadmin	"10.0.0.1", [2001:1234:ABCD::fefe]"

매개 변수	설명	예
다타LIF	<p>프로토콜 LIF의 IP 주소입니다. *</p> <p>ONTAP NAS 드라이버 *: NetApp는 dataLIF를 지정할 것을 권장합니다. 제공되지 않는 경우 Trident는 SVM에서 데이터 LIF를 가져옵니다. NFS 마운트 작업에 사용할 FQDN(정규화된 도메인 이름)을 지정할 수 있습니다. 이렇게 하면 라운드 로빈 DNS를 생성하여 여러 데이터 LIF의 로드 밸런싱을 수행할 수 있습니다. 초기 설정 후에 변경할 수 있습니다. 을 참조하십시오. * ONTAP SAN 드라이버 *: iSCSI에 대해 지정하지 마십시오. Trident는 ONTAP 선택적 LUN 맵을 사용하여 다중 경로 세션을 설정하는 데 필요한 iSCSI LIF를 검색합니다. 데이터 LIF가 명시적으로 정의되어 있으면 경고가 생성됩니다. IPv6 플래그를 사용하여 Trident가 설치된 경우 IPv6 주소를 사용하도록 설정할 수 있습니다. IPv6 주소는 대괄호로 묶어야 합니다(예: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]).</p>	
자동 내보내기 정책	<p>자동 익스포트 정책 생성 및 업데이트 [Boolean] 활성화 Trident는 및 autoExportCIDRs 옵션을 사용하여 autoExportPolicy 익스포트 정책을 자동으로 관리할 수 있습니다.</p>	거짓입니다
자동 내보내기	<p>이 설정된 경우에 대해 Kubernetes 노드 IP를 필터링하는 CIDR autoExportPolicy 목록입니다. Trident는 및 autoExportCIDRs 옵션을 사용하여 autoExportPolicy 익스포트 정책을 자동으로 관리할 수 있습니다.</p>	"["0.0.0.0/0",":/0"]"
'라벨'	<p>볼륨에 적용할 임의의 JSON 형식 레이블 세트입니다</p>	""
'고객증명서'	<p>Base64 - 클라이언트 인증서의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다</p>	""
'clientPrivateKey'입니다	<p>Base64 - 클라이언트 개인 키의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다</p>	""
신택인증서다	<p>Base64 - 신뢰할 수 있는 CA 인증서의 인코딩된 값입니다. 선택 사항. 인증서 기반 인증에 사용됩니다.</p>	""

매개 변수	설명	예
'사용자 이름'	클러스터 또는 SVM에 연결할 사용자 이름입니다. 자격 증명 기반 인증에 사용됩니다. 예: vsadmin.	
"암호"	클러스터 또는 SVM에 연결하는 암호 자격 증명 기반 인증에 사용됩니다.	
'VM'입니다	사용할 스토리지 가상 머신입니다	SVM 관리 LIF가 지정된 경우에 파생됩니다.
'트러스트Prefix'	SVM에서 새 볼륨을 프로비저닝할 때 사용되는 접두어 생성 후에는 수정할 수 없습니다. 이 매개 변수를 업데이트하려면 새 백엔드를 생성해야 합니다.	trident
제한선택사용법	* NetApp ONTAP * 용 아마존 FSx에 대해서는 지정하지 마십시오 제공된 및 vsadmin에는 fsxadmin 애그리게이트 사용량을 검색하고 Trident를 사용하여 제한하는 데 필요한 권한이 포함되어 있지 않습니다.	사용하지 마십시오.
LimitVolumeSize	요청된 볼륨 크기가 이 값보다 큰 경우 용량 할당에 실패합니다. 또한 qtree 및 LUN에서 관리하는 볼륨의 최대 크기를 제한하고, qtreesPerFlexvol 옵션을 통해 FlexVol volume당 최대 qtree 수를 사용자 지정할 수 있습니다	""(기본적으로 적용되지 않음)
'오만유연한'	FlexVol volume당 최대 LUN은 [50, 200] 범위에 있어야 합니다. SAN만 해당.	""100""
debugTraceFlags를 선택합니다	문제 해결 시 사용할 디버그 플래그입니다. 예: {"api":false, "method":true}  사용하지 마십시오 debugTraceFlags 문제 해결 및 자세한 로그 덤프가 필요한 경우를 제외하고	null입니다



매개 변수	설명	예
nfsMountOptions를 선택합니다	심표로 구분된 NFS 마운트 옵션 목록입니다. Kubernetes 영구 볼륨의 마운트 옵션은 일반적으로 스토리지 클래스에 지정되어 있지만, 스토리지 클래스에 마운트 옵션이 지정되지 않은 경우 Trident는 스토리지 백엔드의 구성 파일에 지정된 마운트 옵션을 사용하도록 폴백합니다. 스토리지 클래스 또는 구성 파일에 마운트 옵션이 지정되지 않은 경우 Trident는 연결된 영구 볼륨에 마운트 옵션을 설정하지 않습니다.	""
nasType	NFS 또는 SMB 볼륨 생성을 구성합니다. 옵션은 nfs, smb 또는 null입니다. * 를 로 설정해야 합니다 `smb` SMB 볼륨의 경우. * null로 설정하면 기본적으로 NFS 볼륨이 설정됩니다.	nfs
"케트리스퍼플렉스볼륨"	FlexVol volume당 최대 qtree, 범위 [50, 300] 내에 있어야 함	"200"
smbShare	Microsoft 관리 콘솔 또는 ONTAP CLI를 사용하여 생성된 SMB 공유의 이름 또는 Trident에서 SMB 공유를 생성하도록 허용하는 이름을 지정할 수 있습니다. 이 매개변수는 ONTAP 백엔드에 대한 Amazon FSx에 필요합니다.	smb-share
'useREST'	ONTAP REST API를 사용하는 부울 매개 변수입니다. 로 true 설정하면 Trident에서 ONTAP REST API를 사용하여 백엔드와 통신합니다. 이 기능을 사용하려면 ONTAP 9.11.1 이상이 필요합니다. 또한 사용되는 ONTAP 로그인 역할에는 애플리케이션에 대한 액세스 권한이 있어야 ontap 합니다. 이는 미리 정의된 역할과 역할에 의해 충족됩니다. vsadmin cluster-admin	거짓입니다
aws	AWS FSx for ONTAP의 구성 파일에서 다음을 지정할 수 있습니다. - fsxFilesystemID: AWS FSx 파일 시스템의 ID를 지정합니다. - apiRegion: AWS API 지역 이름입니다. - apikey: AWS API 키입니다. - secretKey: AWS 비밀 키입니다.	"" "" ""

매개 변수	설명	예
credentials	AWS Secrets Manager에 저장할 FSx SVM 자격 증명을 지정합니다. - name: SVM의 자격 증명에 포함된 비밀의 ARN(Amazon Resource Name). type-: 로 `awsarn` 설정합니다. 자세한 내용은 <a href="#">"AWS Secrets Manager 암호를 생성합니다"</a> 참조하십시오.	

## 볼륨 프로비저닝을 위한 백엔드 구성 옵션

에서 이러한 옵션을 사용하여 기본 프로비저닝을 제어할 수 있습니다 defaults 섹션을 참조하십시오. 예를 들어, 아래 구성 예제를 참조하십시오.

매개 변수	설명	기본값
'팩시배부'	LUN에 대한 공간 할당	"참"입니다
'예비공간'	공간 예약 모드, "없음"(싹) 또는 "볼륨"(일반)	"없음"
냅샷정책	사용할 스냅샷 정책입니다	"없음"
"qosPolicy"	생성된 볼륨에 할당할 QoS 정책 그룹입니다. 스토리지 풀 또는 백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다. Trident에서 QoS 정책 그룹을 사용하려면 ONTAP 9.8 이상이 필요합니다. 비공유 QoS 정책 그룹을 사용하고 정책 그룹이 각 구성 요소에 개별적으로 적용되도록 해야 합니다. 공유 QoS 정책 그룹은 모든 워크로드의 총 처리량에 대한 제한을 적용합니다.	""
적응성 QosPolicy	생성된 볼륨에 할당할 적응형 QoS 정책 그룹입니다. 스토리지 풀 또는 백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다. ONTAP에서 지원되지 않음 - NAS - 이코노미	""
안산예비역	스냅샷 "0"에 대해 예약된 볼륨의 백분율	snapshotPolicy `none` 있다면, `else` ""
'plitOnClone'을 선택합니다	생성 시 상위 클론에서 클론을 분할합니다	거짓입니다

매개 변수	설명	기본값
암호화	새 볼륨에서 NetApp 볼륨 암호화(NVE)를 활성화하고, 기본값은 false 이 옵션을 사용하려면 NVE 라이선스가 클러스터에서 활성화되어 있어야 합니다. 백엔드에서 NAE가 활성화된 경우 Trident에서 프로비저닝된 모든 볼륨은 NAE가 사용됩니다. 자세한 내용은 다음을 " <a href="#">Trident가 NVE 및 NAE와 작동하는 방법</a> "참조하십시오.	거짓입니다
luksEncryption	LUKS 암호화를 사용합니다. 을 참조하십시오 " <a href="#">LUKS(Linux Unified Key Setup) 사용</a> ". SAN만 해당.	""
'계층화 정책'	사용할 계층화 정책 none	
유니크권한	모드를 선택합니다. * SMB 볼륨의 경우 비워 둡니다. *	""
'생태성 스타일'을 참조하십시오	새로운 볼륨에 대한 보안 스타일 NFS를 지원합니다 mixed 및 unix 보안 스타일. SMB 지원 mixed 및 ntfs 보안 스타일.	NFS 기본값은 입니다 unix. SMB 기본값은 입니다 ntfs.

## SMB 볼륨 제공

다음을 사용하여 SMB 볼륨을 프로비저닝할 수 있습니다. `ontap-nas` 운전자. 완료하기 전에 [ONTAP SAN 및 NAS 드라이버 통합](#) 다음 단계를 완료하세요. "[SMB 볼륨 프로비저닝을 위한 준비](#)".

## 저장소 클래스 및 PVC를 구성합니다

Kubernetes StorageClass 개체를 구성하고 스토리지 클래스를 생성하여 Trident에 볼륨 프로비저닝 방법을 지시합니다. 구성된 Kubernetes StorageClass를 사용하여 PV에 대한 액세스를 요청하는 PersistentVolumeClaim(PVC)을 생성합니다. 그런 다음 PV를 포드에 장착할 수 있습니다.

### 스토리지 클래스를 생성합니다

#### Kubernetes StorageClass 개체를 구성합니다

그만큼 "[Kubernetes StorageClass 객체](#)" 객체는 Trident 해당 클래스에 사용되는 프로비저너로 식별하고 Trident 볼륨을 프로비저닝하는 방법을 지시합니다. NFS를 사용하여 볼륨에 대한 Storageclass를 설정하려면 이 예제를 사용하세요(전체 속성 목록은 아래의 Trident 속성 섹션을 참조하세요).

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  provisioningType: "thin"
  snapshots: "true"

```

iSCSI를 사용하여 볼륨에 대한 Storageclass를 설정하려면 다음 예를 사용하세요.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
  provisioningType: "thin"
  snapshots: "true"

```

AWS Bottlerocket에서 NFSv3 볼륨을 프로비저닝하려면 필요한 `mountOptions` 스토리지 클래스에 추가합니다.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  media: "ssd"
  provisioningType: "thin"
  snapshots: "true"
mountOptions:
  - nfsvers=3
  - nolock

```

Trident에서 볼륨을 프로비저닝하는 방법을 제어하는 데 사용되는 몇 개 변수와 스토리지 클래스가 상호 작용하는 방법에 대한 자세한 PersistentVolumeClaim 내용은 ["Kubernetes 및 Trident 오브젝트"](#) 참조하십시오.

스토리지 클래스를 생성합니다

단계

1. Kubernetes 오브젝트이므로 를 사용하십시오 kubectl Kubernetes에서 생성해야 합니다.

```
kubectl create -f storage-class-ontapas.yaml
```

2. 이제 Kubernetes와 Trident 모두에서 \* BASIC-CSI \* 스토리지 클래스가 표시되고 Trident는 백엔드에서 풀을 검색해야 합니다.

```
kubectl get sc basic-csi
```

NAME	PROVISIONER	AGE
basic-csi	csi.trident.netapp.io	15h

## PVC를 작성합니다

A "[\*PersistentVolumeClaim\*](#)"(PVC)는 클러스터의 PersistentVolume에 대한 액세스 요청입니다.

PVC는 특정 크기 또는 액세스 모드의 저장을 요청하도록 구성할 수 있습니다. 클러스터 관리자는 연결된 StorageClass를 사용하여 PersistentVolume 크기 및 액세스 모드(예: 성능 또는 서비스 수준)를 제어할 수 있습니다.

PVC를 생성한 후 포드에 볼륨을 장착할 수 있습니다.

샘플 매니페스트

## PersistentVolumeClaim 샘플 매니페스트

이러한 예는 기본적인 PVC 구성 옵션을 보여줍니다.

### RWX 액세스 PVC

이 예에서는 이름이 인 StorageClass와 연결된 rwx 액세스 권한이 있는 기본 PVC를 보여 `basic-csi` 줍니다.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-storage
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-gold
```

### iSCSI를 사용한 PVC 예제

이 예에서는 RWO 액세스가 있는 iSCSI용 기본 PVC를 보여줍니다. 이 PVC는 StorageClass와 연결되어 있습니다. protection-gold.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-san
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: protection-gold
```

## PVC 생성

단계

1. PVC를 작성합니다.

```
kubectl create -f pvc.yaml
```

## 2. PVC 상태를 확인합니다.

```
kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
pvc-storage	Bound	pv-name	2Gi	RWO		5m

Trident에서 볼륨을 프로비저닝하는 방법을 제어하는 데 사용되는 몇 가지 매개 변수와 스토리지 클래스가 상호 작용하는 방법에 대한 자세한 PersistentVolumeClaim 내용은 ["Kubernetes 및 Trident 오브젝트"](#) 참조하십시오.

## Trident 특성

이러한 매개 변수는 지정된 유형의 볼륨을 프로비저닝하는 데 사용해야 하는 Trident 관리 스토리지 풀을 결정합니다.

속성	유형	값	제공합니다	요청하십시오	에 의해 지원됩니다
미디어 <sup>1</sup>	문자열	HDD, 하이브리드, SSD	풀에는 이 유형의 미디어가 포함되어 있으며, 하이브리드는 둘 모두를 의미합니다	지정된 미디어 유형입니다	ONTAP-NAS, ONTAP-NAS-이코노미, ONTAP-NAS-Flexgroup, ONTAP-SAN, solidfire-SAN
프로비저닝 유형	문자열	얇고 두껍습니다	풀은 이 프로비저닝 방법을 지원합니다	프로비저닝 방법이 지정되었습니다	Thick: All ONTAP; Thin: All ONTAP & solidfire-SAN
백엔드 유형	문자열	ontap-nas, ontap-nas-economy, ontap-nas-flexgroup, ontap-san, solidfire-san, azure-netapp-files, ontap-san-economy	풀이 이 백엔드 유형에 속합니다	백엔드가 지정되었습니다	모든 드라이버
스냅샷 수	불입니다	참, 거짓	풀은 스냅샷이 있는 볼륨을 지원합니다	스냅샷이 활성화된 볼륨	온타프나스, 온타프씨, 솔리드파이어씨
복제	불입니다	참, 거짓	풀은 볼륨 클론을 지원합니다	클론이 활성화된 볼륨	온타프나스, 온타프씨, 솔리드파이어씨

속성	유형	값	제공합니다	요청하십시오	예 의해 지원됩니다
암호화	불입니다	참, 거짓	풀은 암호화된 볼륨을 지원합니다	암호화가 활성화된 볼륨입니다	ONTAP-NAS, ONTAP-NAS- 이코노미, ONTAP-NAS- Flexgroups, ONTAP-SAN
IOPS	내부	양의 정수입니다	풀은 이 범위에서 IOPS를 보장할 수 있습니다	볼륨은 이러한 IOPS를 보장합니다	solidfire-SAN

<sup>1</sup>: ONTAP Select 시스템에서 지원되지 않습니다

## 샘플 응용 프로그램을 배포합니다

저장 클래스 및 PVC가 생성되면 PV를 포드에 장착할 수 있습니다. 이 섹션에는 PV를 Pod에 연결하기 위한 명령 및 구성의 예가 나와 있습니다.

단계

1. 볼륨을 Pod에 마운트합니다.

```
kubectl create -f pv-pod.yaml
```

다음 예는 PVC를 포드에 부착하기 위한 기본 구성을 보여줍니다. \* 기본 구성 \*:

```
kind: Pod
apiVersion: v1
metadata:
  name: pv-pod
spec:
  volumes:
    - name: pv-storage
      persistentVolumeClaim:
        claimName: basic
  containers:
    - name: pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/my/mount/path"
          name: pv-storage
```





을 사용하여 진행 상황을 모니터링할 수 있습니다 `kubectl get pod --watch`.

2. 볼륨이 에 마운트되어 있는지 확인합니다 `/my/mount/path`.

```
kubectl exec -it pv-pod -- df -h /my/mount/path
```

```
Filesystem                                Size
Used Avail Use% Mounted on
192.168.188.78:/trident_pvc_ae45ed05_3ace_4e7c_9080_d2a83ae03d06 1.1G
320K 1.0G 1% /my/mount/path
```

이제 Pod를 삭제할 수 있습니다. Pod 응용 프로그램은 더 이상 존재하지 않지만 볼륨은 유지됩니다.

```
kubectl delete pod pv-pod
```

## EKS 클러스터에서 Trident EKS 애드온을 구성합니다

NetApp Trident은 Kubernetes에서 Amazon FSx for NetApp ONTAP 스토리지 관리를 간소화하여 개발자와 관리자가 애플리케이션 구축에 집중할 수 있도록 지원합니다. NetApp Trident EKS 애드온에는 최신 보안 패치 및 버그 수정이 포함되어 있으며 AWS에서 Amazon EKS와 함께 사용할 수 있다는 것이 검증되었습니다. EKS 애드온을 사용하면 Amazon EKS 클러스터의 보안과 안정성을 지속적으로 보장하고 애드온을 설치, 구성 및 업데이트하는 데 필요한 작업량을 줄일 수 있습니다.

### 필수 구성 요소

AWS EKS용 Trident 애드온을 구성하기 전에 다음 사항이 있는지 확인하십시오.

- 추가 기능을 사용할 수 있는 권한이 있는 Amazon EKS 클러스터 계정입니다. 을 ["Amazon EKS 애드온"](#) 참조하십시오.
- AWS 마켓플레이스에 대한 AWS 권한:
 

```
"aws-marketplace:ViewSubscriptions",
"aws-marketplace:Subscribe",
"aws-marketplace:Unsubscribe
```
- AMI 유형: Amazon Linux 2 (AL2\_x86\_64) 또는 Amazon Linux 2 Arm (AL2\_ARM\_64)
- 노드 유형: AMD 또는 ARM
- 기존 Amazon FSx for NetApp ONTAP 파일 시스템

### 단계

1. EKS Pod에서 AWS 리소스에 액세스할 수 있도록 IAM 역할 및 AWS 암호를 생성하십시오. 자세한 내용은 을 ["IAM"](#)

역할 및 AWS Secret을 생성합니다"참조하십시오.

2. EKS Kubernetes 클러스터에서 \* Add-ons \* 탭으로 이동합니다.

tri-env-eks

End of standard support for Kubernetes version 1.30 is July 28, 2025. On that date, your cluster will enter the extended support period with additional fees. For more information, see the [pricing page](#).

Cluster info

- Status: Active
- Kubernetes version: 1.30
- Support period: Standard support until July 28, 2025
- Provider: EKS
- Cluster health issues: 0
- Upgrade insights: 0

Overview | Resources | Compute | Networking | **Add-ons 1** | Access | Observability | Update history | Tags

New versions are available for 1 add-on.

Add-ons (3)

Find add-on

Any category... Any status 3 matches

3. AWS Marketplace 애드온 \* 으로 이동하여 \_STORAGE\_CATEGORY를 선택합니다.

AWS Marketplace add-ons (1)

Discover, subscribe to and configure EKS add-ons to enhance your EKS clusters.

Find add-on

Filtering options

Any category NetApp, Inc. Any pricing model Clear filters

NetApp, Inc.

NetApp Trident

NetApp Trident streamlines Amazon FSx for NetApp ONTAP storage management in Kubernetes to let your developers and administrators focus on application deployment. FSx for ONTAP flexibility, scalability, and integration capabilities make it the ideal choice for organizations seeking efficient containerized storage workflows. [Product details](#)

Standard Contract

Category storage	Listed by <a href="#">NetApp, Inc.</a>	Supported versions 1.31, 1.30, 1.29, 1.28, 1.27, 1.26, 1.25, 1.24, 1.23	Pricing starting at <a href="#">View pricing details</a>
---------------------	---	--	---

Cancel Next

4. NetApp Trident \* 를 찾아 Trident 애드온의 확인란을 선택하고 \* 다음 \* 을 클릭합니다.

5. 원하는 추가 기능 버전을 선택합니다.

## Configure selected add-ons settings

Configure the add-ons for your cluster by selecting settings.

**NetApp Trident**

Remove add-on

Listed by

Category

Status

NetApp

storage

Ready to install

You're subscribed to this software

View subscription

You can view the terms and pricing details for this product or choose another offer if one is available.

Version

Select the version for this add-on.

v25.6.0-eksbuild.1

Optional configuration settings

Cancel

Previous

Next

6. 필요한 추가 기능 설정을 구성합니다.

## Review and add

### Step 1: Select add-ons

[Edit](#)

**Selected add-ons (1)**

Find add-on

< 1 >

Add-on name	Type	Status
netapp_trident-operator	storage	Ready to install

### Step 2: Configure selected add-ons settings

[Edit](#)

**Selected add-ons version (1)**

< 1 >

Add-on name	Version	IAM role for service account (IRSA)
netapp_trident-operator	v24.10.0-eksbuild.1	Not set

**EKS Pod Identity (0)**

< 1 >

Add-on name	IAM role	Service account
No Pod Identity associations None of the selected add-on(s) have Pod Identity associations.		

Cancel

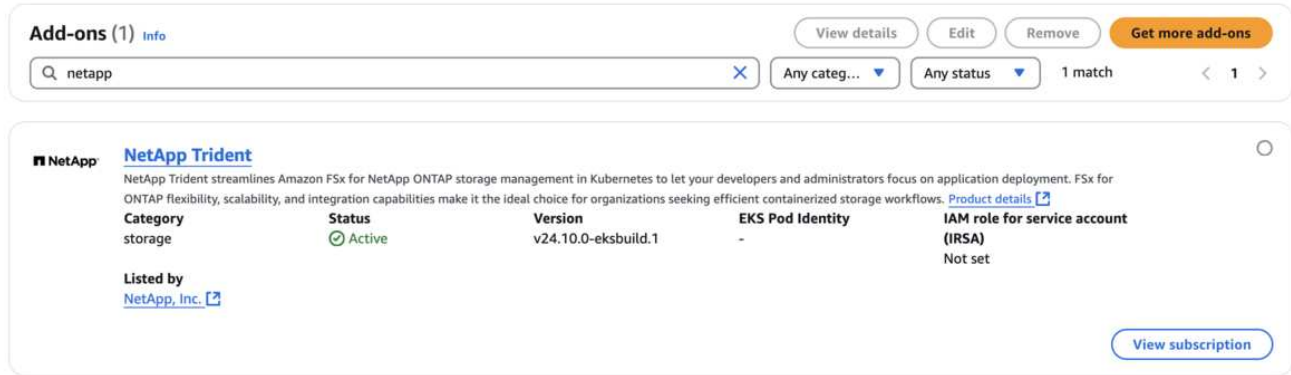
Previous

Create

7. IRSA(서비스 계정에 대한 IAM 역할을) 사용하는 경우 추가 구성 단계를 참조하세요."여기".

8. Create \* 를 선택합니다.

9. 애드온의 상태가 `_Active_` 인지 확인합니다.



10. 다음 명령을 실행하여 Trident가 클러스터에 올바르게 설치되어 있는지 확인합니다.

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

11. 설치를 계속하고 스토리지 백엔드를 구성합니다. 자세한 내용은 을 "[스토리지 백엔드를 구성합니다](#)"참조하십시오.

## CLI를 사용하여 Trident EKS 애드온을 설치/제거합니다

**CLI**를 사용하여 **NetApp Trident EKS** 추가 기능을 설치합니다.

다음 예제 명령은 Trident EKS 추가 기능을 설치합니다.

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.0-eksbuild.1 (전용 버전 포함)
```

다음 예시 명령은 Trident EKS 애드온 버전 25.6.1을 설치합니다:

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.1-eksbuild.1 (전용 버전 포함)
```

다음 예시 명령은 Trident EKS 애드온 버전 25.6.2를 설치합니다:

```
eksctl create addon --cluster clusterName --name netapp_trident-operator  
--version v25.6.2-eksbuild.1 (전용 버전 포함)
```

**CLI**를 사용하여 **NetApp Trident EKS** 애드온을 제거합니다.

다음 명령을 실행하면 Trident EKS 추가 기능이 제거됩니다.

```
eksctl delete addon --cluster K8s-arm --name netapp_trident-operator
```

## 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.