



백엔드 구성 Astra Trident

NetApp
November 14, 2025

목차

백엔드 구성	1
Azure NetApp Files	1
Azure NetApp Files 백엔드를 구성합니다	1
Azure NetApp Files 백엔드를 구성할 준비를 합니다	1
Azure NetApp Files 백엔드 구성 옵션 및 예	4
Google Cloud용 Cloud Volumes Service 백엔드를 구성합니다	12
Cloud Volumes Service for Google Cloud를 위한 Astra Trident 지원에 대해 알아보십시오	12
백엔드 구성 옵션	13
볼륨 프로비저닝 옵션	15
CVS - 성능 서비스 유형의 예	15
CVS 서비스 유형 예	24
다음 단계	28
NetApp HCI 또는 SolidFire 백엔드를 구성합니다	28
백엔드 구성 옵션	29
예 1: 예 대한 백엔드 구성 solidfire-san 세 가지 볼륨 유형을 가진 드라이버	30
예 2: 예 대한 백엔드 및 스토리지 클래스 구성 solidfire-san 가상 풀이 있는 드라이버	31
자세한 내용을 확인하십시오	35
ONTAP SAN 드라이버를 사용하여 백엔드를 구성합니다	35
사용자 권한	35
ONTAP SAN 드라이버를 사용하여 백엔드를 구성할 준비를 합니다	35
ONTAP SAN 구성 옵션 및 예	42
ONTAP NAS 백엔드를 구성합니다	54
사용자 권한	54
ONTAP NAS 드라이버를 사용하여 백엔드를 구성할 준비를 합니다	54
ONTAP NAS 구성 옵션 및 예	62
NetApp ONTAP용 Amazon FSx	79
NetApp ONTAP용 Amazon FSx와 함께 Astra Trident를 사용하십시오	79
NetApp ONTAP용 Amazon FSx를 통합합니다	80
ONTAP 구성 옵션 및 예제용 FSX	83

백엔드 구성

백엔드는 Astra Trident와 스토리지 시스템 간의 관계를 정의합니다. Astra Trident가 스토리지 시스템과 통신하는 방법과 Astra Trident가 스토리지 시스템에서 볼륨을 프로비저닝하는 방법을 알려줍니다.

Astra Trident는 스토리지 클래스에 정의된 요구 사항과 일치하는 백엔드에서 스토리지 풀을 자동으로 제공합니다. 스토리지 시스템에 대한 백엔드를 구성하는 방법에 대해 알아보십시오.

- ["Azure NetApp Files 백엔드를 구성합니다"](#)
- ["Google Cloud Platform 백엔드에 Cloud Volumes Service를 구성합니다"](#)
- ["NetApp HCI 또는 SolidFire 백엔드를 구성합니다"](#)
- ["ONTAP 또는 Cloud Volumes ONTAP NAS 드라이버를 사용하여 백엔드를 구성합니다"](#)
- ["ONTAP 또는 Cloud Volumes ONTAP SAN 드라이버를 사용하여 백엔드를 구성합니다"](#)
- ["NetApp ONTAP용 Amazon FSx와 함께 Astra Trident를 사용하십시오"](#)

Azure NetApp Files

Azure NetApp Files 백엔드를 구성합니다

Astra Trident의 백엔드로 Azure NetApp Files(ANF)를 구성할 수 있습니다. ANF 백엔드를 사용하여 NFS 및 SMB 볼륨을 연결할 수 있습니다.

- ["준비"](#)
- ["구성 옵션 및 예"](#)

고려 사항

- Azure NetApp Files 서비스는 100GB 미만의 볼륨을 지원하지 않습니다. Astra Trident는 더 작은 볼륨을 요청하는 경우 100GB 볼륨을 자동으로 생성합니다.
- Astra Trident는 Windows 노드에서 실행되는 Pod에만 마운트된 SMB 볼륨을 지원합니다.
- Astra Trident는 Windows ARM 아키텍처를 지원하지 않습니다.

Azure NetApp Files 백엔드를 구성할 준비를 합니다

Azure NetApp Files 백엔드를 구성하기 전에 다음 요구 사항이 충족되는지 확인해야 합니다.



Azure NetApp Files를 처음 사용하거나 새 위치에서 사용하는 경우 Azure NetApp Files를 설정하고 NFS 볼륨을 생성하려면 몇 가지 초기 구성이 필요합니다. 을 참조하십시오 ["Azure: Azure NetApp Files를 설정하고 NFS 볼륨을 생성합니다"](#).

NFS 및 SMB 볼륨의 사전 요구 사항

를 구성하고 사용합니다 ["Azure NetApp Files"](#) 백엔드, 다음이 필요합니다.

- 용량 풀입니다. 을 참조하십시오 ["Microsoft: Azure NetApp Files에 대한 용량 풀을 생성합니다"](#).
- Azure NetApp Files에 위임된 서버넷. 을 참조하십시오 ["Microsoft: Azure NetApp Files에 서버넷을 위임합니다"](#).
- subscriptionID Azure NetApp Files가 활성화된 Azure 구독에서
- tenantID, clientID, 및 clientSecret 에서 ["앱 등록"](#) Azure NetApp Files 서비스에 대한 충분한 권한이 있는 Azure Active Directory에서 앱 등록에서는 다음 중 하나를 사용해야 합니다.
 - 소유자 또는 참가자 역할입니다 ["Azure에서 사전 정의"](#).
 - A ["사용자 지정 참가자 역할"](#) 구독 레벨입니다 (assignableScopes) 다음 사용 권한은 Astra Trident에 필요한 권한만 가집니다. 사용자 지정 역할을 만든 후 ["Azure 포털을 사용하여 역할을 할당합니다"](#).

```
{
  "id": "/subscriptions/<subscription-id>/providers/Microsoft.Authorization/roleDefinitions/<role-definition-id>",
  "properties": {
    "roleName": "custom-role-with-limited-perms",
    "description": "custom role providing limited permissions",
    "assignableScopes": [
      "/subscriptions/<subscription-id>"
    ],
    "permissions": [
      {
        "actions": [

          "Microsoft.NetApp/netAppAccounts/capacityPools/read",

          "Microsoft.NetApp/netAppAccounts/capacityPools/write",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/read",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/write",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/delete",

          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/read",
          ",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/write",
          "e",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/snapshots/delete",
          "te",
          "Microsoft.NetApp/netAppAccounts/capacityPools/volumes/subvolumes/read",
          "d",
```

```

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/subvolumes/write",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/subvolumes/delete",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/subvolumes/GetMetadata/action",

"Microsoft.NetApp/netAppAccounts/capacityPools/volumes/MountTargets/read",

    "Microsoft.Network/virtualNetworks/read",
    "Microsoft.Network/virtualNetworks/subnets/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/read",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/write",

"Microsoft.Features/featureProviders/subscriptionFeatureRegistrations/delete",

    "Microsoft.Features/features/read",
    "Microsoft.Features/operations/read",
    "Microsoft.Features/providers/features/read",

"Microsoft.Features/providers/features/register/action",

"Microsoft.Features/providers/features/unregister/action",

"Microsoft.Features/subscriptionFeatureRegistrations/read"
    ],
    "notActions": [],
    "dataActions": [],
    "notDataActions": []
  }
]
}
}

```

- Azure를 선택합니다 location 하나 이상의 항목이 포함되어 있습니다 "위임된 서브넷". Trident 22.01부터 location 매개 변수는 백엔드 구성 파일의 최상위 수준에 있는 필수 필드입니다. 가상 풀에 지정된 위치 값은 무시됩니다.

SMB 볼륨에 대한 추가 요구사항

SMB 볼륨을 생성하려면 다음이 있어야 합니다.

- Active Directory가 구성되어 Azure NetApp Files에 연결되었습니다. 을 참조하십시오 ["Microsoft: Azure NetApp Files에 대한 Active Directory 연결을 만들고 관리합니다"](#).
- Linux 컨트롤러 노드 및 Windows Server 2019를 실행하는 Windows 작업자 노드가 있는 Kubernetes 클러스터 Astra Trident는 Windows 노드에서 실행되는 Pod에만 마운트된 SMB 볼륨을 지원합니다.
- Azure NetApp Files가 Active Directory에 인증할 수 있도록 Active Directory 자격 증명이 포함된 Astra Trident 암호가 하나 이상 있어야 합니다. 암호를 생성합니다 `smbcreds`:

```
kubectl create secret generic smbcreds --from-literal username=user  
--from-literal password='password'
```

- Windows 서비스로 구성된 CSI 프록시. 를 구성합니다 `csi-proxy`를 참조하십시오 ["GitHub:CSI 프록시"](#) 또는 ["GitHub: Windows용 CSI 프록시"](#) Windows에서 실행되는 Kubernetes 노드의 경우:

Azure NetApp Files 백엔드 구성 옵션 및 예

ANF에 대한 NFS 및 SMB 백엔드 구성 옵션에 대해 알아보고 구성 예를 검토하십시오.

Astra Trident는 백엔드 구성(서브넷, 가상 네트워크, 서비스 수준 및 위치)을 사용하여 요청된 위치에서 사용할 수 있고 요청된 서비스 수준 및 서브넷과 일치하는 용량 풀에 ANF 볼륨을 생성합니다.



Astra Trident는 수동 QoS 용량 풀을 지원하지 않습니다.

백엔드 구성 옵션

ANF 백엔드는 이러한 구성 옵션을 제공합니다.

매개 변수	설명	기본값
version		항상 1
storageDriverName	스토리지 드라이버의 이름입니다	"Azure-NetApp-파일"
backendName	사용자 지정 이름 또는 스토리지 백엔드	드라이버 이름 + "_" + 임의 문자
subscriptionID	Azure 구독의 구독 ID입니다	
tenantID	앱 등록에서 테넌트 ID입니다	
clientID	앱 등록의 클라이언트 ID입니다	
clientSecret	앱 등록에서 클라이언트 암호	
serviceLevel	중 하나 Standard, Premium, 또는 Ultra	""(임의)
location	새 볼륨을 생성할 Azure 위치의 이름입니다	

매개 변수	설명	기본값
resourceGroups	검색된 자원을 필터링하기 위한 자원 그룹 목록입니다	[] (필터 없음)
netappAccounts	검색된 리소스를 필터링하기 위한 NetApp 계정의 목록입니다	[] (필터 없음)
capacityPools	검색된 리소스를 필터링하기 위한 용량 풀 목록입니다	[] (필터 없음, 임의)
virtualNetwork	위임된 서브넷이 있는 가상 네트워크의 이름입니다	""
subnet	위임된 서브넷의 이름입니다 Microsoft.Netapp/volumes	""
networkFeatures	볼륨에 대한 VNET 기능 집합은 일 수 있습니다 Basic 또는 Standard. 일부 지역에서는 네트워크 기능을 사용할 수 없으며 구독에서 활성화해야 할 수도 있습니다. 지정 networkFeatures 이 기능을 사용하지 않으면 볼륨 프로비저닝이 실패합니다.	""
nfsMountOptions	NFS 마운트 옵션에 대한 세밀한 제어 SMB 볼륨에 대해 무시됩니다. NFS 버전 4.1을 사용하여 볼륨을 마운트하려면 을 포함합니다 nfsvers=4 심표로 구분된 마운트 옵션 목록에서 NFS v4.1을 선택합니다. 스토리지 클래스 정의에 설정된 마운트 옵션은 백엔드 구성에 설정된 마운트 옵션을 재정의합니다.	"nfsvers=3"
limitVolumeSize	요청된 볼륨 크기가 이 값보다 큰 경우 용량 할당에 실패합니다	"" (기본적으로 적용되지 않음)
debugTraceFlags	문제 해결 시 사용할 디버그 플래그입니다. 예: \{"api": false, "method": true, "discovery": true\}. 문제 해결 중이 아니며 자세한 로그 덤프가 필요한 경우가 아니면 이 방법을 사용하지 마십시오.	null입니다
nasType	NFS 또는 SMB 볼륨 생성을 구성합니다. 옵션은 입니다 nfs, smb 또는 null입니다. Null로 설정하면 기본적으로 NFS 볼륨이 설정됩니다.	nfs



네트워크 기능에 대한 자세한 내용은 을 참조하십시오 ["Azure NetApp Files 볼륨에 대한 네트워크 기능을 구성합니다"](#).

필요한 권한 및 리소스

PVC를 생성할 때 "용량 풀을 찾을 수 없음" 오류가 발생하는 경우 앱 등록에 필요한 사용 권한 및 리소스(서브넷, 가상 네트워크, 용량 풀)가 연결되지 않은 것일 수 있습니다. DEBUG가 활성화된 경우 Astra Trident는 백엔드가 생성될 때 검색된 Azure 리소스를 기록합니다. 적절한 역할이 사용되고 있는지 확인합니다.

의 값 resourceGroups, netappAccounts, capacityPools, virtualNetwork, 및 subnet 간단한 이름 또는 정규화된 이름을 사용하여 지정할 수 있습니다. 이름이 같은 여러 리소스와 이름이 일치할 수 있으므로 대부분의 경우 정규화된 이름을 사용하는 것이 좋습니다.

를 클릭합니다 resourceGroups, netappAccounts, 및 capacityPools 값은 검색된 리소스 집합을 이 스토리지 백엔드에서 사용할 수 있는 리소스로 제한하는 필터이며, 이 둘을 조합하여 지정할 수 있습니다. 정규화된 이름은 다음 형식을 따릅니다.

유형	형식
리소스 그룹	리소스 그룹>
NetApp 계정	리소스 그룹>/<NetApp 계정>
용량 풀	리소스 그룹>/<NetApp 계정>/<용량 풀>
가상 네트워크	리소스 그룹>/<가상 네트워크>
서브넷	리소스 그룹>/<가상 네트워크>/<서브넷>

볼륨 프로비저닝

구성 파일의 특수 섹션에서 다음 옵션을 지정하여 기본 볼륨 프로비저닝을 제어할 수 있습니다. 을 참조하십시오 [예제 설정](#) 를 참조하십시오.

매개 변수	설명	기본값
exportRule	새 볼륨에 대한 익스포트 규칙 exportRule CIDR 표기법을 사용하여 IPv4 주소 또는 IPv4 서브넷의 조합을 쉼표로 구분해야 합니다. SMB 볼륨에 대해 무시됩니다.	"0.0.0.0/0"
snapshotDir	스냅샷 디렉터리의 표시 여부를 제어합니다	"거짓"
size	새 볼륨의 기본 크기입니다	"100G"
unixPermissions	새 볼륨의 UNIX 사용 권한(8진수 4자리) SMB 볼륨에 대해 무시됩니다.	""(미리보기 기능, 가입 시 화이트리스트 필요)

예제 설정

예 1: 최소 구성

이는 절대적인 최소 백엔드 구성입니다. 이 구성을 통해 Astra Trident는 구성된 위치에서 ANF에 위임된 모든 NetApp 계정, 용량 풀 및 서브넷을 검색하고 해당 풀 및 서브넷 중 하나에 무작위로 새 볼륨을 배치합니다. 왜냐하면 `nasType` 생략됩니다 `nfs` 기본값은 NFS 볼륨에 대해 백엔드가 프로비저닝됩니다.

이 구성은 ANF를 사용하여 지금 막 시작하는 데 이상적이지만, 실제로 용량 할당을 수행하는 볼륨에 대한 추가적인 범위 지정은 제공하려는 경우에 적합합니다.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
```

예 2: 용량 풀 필터를 사용하는 특정 서비스 수준 구성

이 백엔드 구성은 Azure에 볼륨을 배치합니다 `eastus` 의 위치 `Ultra` 용량 풀. Astra Trident는 해당 위치의 ANF에 위임된 모든 서브넷을 자동으로 검색하여 그 중 하나에 무작위로 새 볼륨을 배치합니다.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
- application-group-1/account-1/ultra-1
- application-group-1/account-1/ultra-2
```

예 3: 고급 구성

이 백엔드 구성은 단일 서브넷에 대한 볼륨 배치 범위를 더욱 줄여주고 일부 볼륨 프로비저닝 기본값도 수정합니다.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
serviceLevel: Ultra
capacityPools:
- application-group-1/account-1/ultra-1
- application-group-1/account-1/ultra-2
virtualNetwork: my-virtual-network
subnet: my-subnet
networkFeatures: Standard
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 500Gi
defaults:
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  snapshotDir: 'true'
  size: 200Gi
  unixPermissions: '0777'
```

예 4: 가상 풀 구성

이 백엔드 구성은 단일 파일에 여러 스토리지 풀을 정의합니다. 다양한 서비스 수준을 지원하는 여러 용량 풀이 있고 이를 나타내는 Kubernetes의 스토리지 클래스를 생성하려는 경우에 유용합니다. 가상 풀 레이블을 사용하여 에 따라 풀을 구분했습니다 performance.

```
---
version: 1
storageDriverName: azure-netapp-files
subscriptionID: 9f87c765-4774-fake-ae98-a721add45451
tenantID: 68e4f836-edc1-fake-bff9-b2d865ee56cf
clientID: dd043f63-bf8e-fake-8076-8de91e5713aa
clientSecret: SECRET
location: eastus
resourceGroups:
- application-group-1
networkFeatures: Basic
nfsMountOptions: vers=3,proto=tcp,timeo=600
labels:
  cloud: azure
storage:
- labels:
    performance: gold
    serviceLevel: Ultra
    capacityPools:
    - ultra-1
    - ultra-2
    networkFeatures: Standard
- labels:
    performance: silver
    serviceLevel: Premium
    capacityPools:
    - premium-1
- labels:
    performance: bronze
    serviceLevel: Standard
    capacityPools:
    - standard-1
    - standard-2
```

스토리지 클래스 정의

다음 사항을 참조하십시오 StorageClass 정의는 위의 스토리지 풀을 참조합니다.

을 사용한 정의 예 `parameter.selector` 필드에 입력합니다

사용 `parameter.selector` 각각에 대해 지정할 수 있습니다 `StorageClass` 볼륨을 호스팅하는 데 사용되는 가상 풀입니다. 볼륨은 선택한 풀에 정의된 측면을 갖습니다.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gold
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=gold"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: silver
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=silver"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: bronze
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=bronze"
allowVolumeExpansion: true
```

SMB 볼륨에 대한 정의의 예

사용 `nasType`, `node-stage-secret-name`, 및 `node-stage-secret-namespace`, **SMB** 볼륨을 지정하고 필요한 **Active Directory** 자격 증명을 제공할 수 있습니다.

예제 1: 기본 네임스페이스의 기본 구성

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

예제 2: 네임스페이스당 다른 암호 사용

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```

예 3: 볼륨별로 다른 암호 사용

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: anf-sc-smb
provisioner: csi.trident.netapp.io
parameters:
  backendType: "azure-netapp-files"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: ${pvc.name}
  csi.storage.k8s.io/node-stage-secret-namespace: ${pvc.namespace}
```



nasType: `smb` SMB 볼륨을 지원하는 풀에 대한 필터입니다. nasType: `nfs` 또는 nasType: `null` NFS 풀에 대한 필터입니다.

백엔드를 생성합니다

백엔드 구성 파일을 생성한 후 다음 명령을 실행합니다.

```
tridentctl create backend -f <backend-file>
```

백엔드 생성에 실패하면 백엔드 구성에 문제가 있는 것입니다. 다음 명령을 실행하여 로그를 보고 원인을 확인할 수 있습니다.

```
tridentctl logs
```

구성 파일의 문제를 확인하고 수정한 후 create 명령을 다시 실행할 수 있습니다.

Google Cloud용 Cloud Volumes Service 백엔드를 구성합니다

제공된 샘플 구성을 사용하여 Astra Trident 설치를 위한 백엔드로 NetApp Cloud Volumes Service for Google Cloud를 구성하는 방법을 알아보십시오.

Cloud Volumes Service for Google Cloud를 위한 Astra Trident 지원에 대해 알아보십시오

Astra Trident는 두 개 중 하나로 Cloud Volumes Service 볼륨을 생성할 수 있습니다 "[서비스 유형](#)":

- * CVS - 성능 *: 기본 Astra Trident 서비스 유형입니다. 이처럼 성능에 최적화된 서비스 유형은 성능을 중요시하는 운영 워크로드에 가장 적합합니다. CVS - 성능 서비스 유형은 최소 100GiB 크기의 볼륨을 지원하는 하드웨어 옵션입니다. 다음 중 하나를 선택할 수 있습니다 "[3가지 서비스 레벨](#)":
 - standard
 - premium
 - extreme
- CVS *: CVS 서비스 유형은 높은 조널 가용성을 제공하며, 성능은 중간 수준으로 제한됩니다. CVS 서비스 유형은 스토리지 풀을 사용하여 1GiB의 작은 볼륨을 지원하는 소프트웨어 옵션입니다. 스토리지 풀에는 최대 50개의 볼륨이 포함될 수 있으며 이 볼륨에서 풀의 용량과 성능을 공유할 수 있습니다. 다음 중 하나를 선택할 수 있습니다 "[서비스 레벨 2개](#)":
 - standardsw
 - zoneredundantstandardsw

필요한 것

를 구성하고 사용합니다 "[Google Cloud용 Cloud Volumes Service](#)" 백엔드, 다음이 필요합니다.

- NetApp Cloud Volumes Service로 구성된 Google Cloud 계정

- Google Cloud 계정의 프로젝트 번호입니다
- 예 Google Cloud 서비스 계정이 있습니다 `netappcloudvolumes.admin` 역할
- Cloud Volumes Service 계정에 대한 API 키 파일입니다

백엔드 구성 옵션

각 백엔드는 단일 Google Cloud 지역에 볼륨을 프로비저닝합니다. 다른 영역에 볼륨을 생성하려면 추가 백엔드를 정의할 수 있습니다.

매개 변수	설명	기본값
<code>version</code>		항상 1
<code>storageDriverName</code>	스토리지 드라이버의 이름입니다	"GCP-CV"
<code>backendName</code>	사용자 지정 이름 또는 스토리지 백엔드	드라이버 이름 + "_" + API 키의 일부
<code>storageClass</code>	CVS 서비스 유형을 지정하는 데 사용되는 선택적 매개 변수입니다. 사용 <code>software</code> CVS 서비스 유형을 선택합니다. 그렇지 않으면 Astra Trident가 CVS - 성능 서비스 유형을 가정합니다 (<code>hardware</code>)를 클릭합니다.	
<code>storagePools</code>	CVS 서비스 유형만 볼륨 생성을 위한 스토리지 풀을 지정하는 데 사용되는 선택적 매개 변수입니다.	
<code>projectNumber</code>	Google Cloud 계정 프로젝트 번호입니다. 이 값은 Google Cloud 포털 홈 페이지에서 확인할 수 있습니다.	
<code>hostProjectNumber</code>	공유 VPC 네트워크를 사용하는 경우 필요합니다. 이 시나리오에서는 <code>projectNumber</code> 은(는) 서비스 프로젝트입니다 <code>hostProjectNumber</code> 는 호스트 프로젝트입니다.	
<code>apiRegion</code>	Astra Trident가 Cloud Volumes Service 볼륨을 생성하는 Google 클라우드 영역 지역 간 Kubernetes 클러스터를 생성할 때 에서 생성된 볼륨입니다 <code>apiRegion</code> 여러 Google Cloud 지역의 노드에 예약된 워크로드에 사용할 수 있습니다. 지역 간 트래픽에는 추가 비용이 발생합니다.	

매개 변수	설명	기본값
apiKey	를 사용하여 Google Cloud 서비스 계정에 대한 API 키입니다 netappcloudvolumes.admin 역할. 여기에는 Google Cloud 서비스 계정의 개인 키 파일(백엔드 구성 파일에 verbatim 복사)의 JSON 형식 콘텐츠가 포함됩니다.	
proxyURL	프록시 서버가 CVS 계정에 연결해야 하는 경우 프록시 URL입니다. 프록시 서버는 HTTP 프록시 또는 HTTPS 프록시일 수 있습니다. HTTPS 프록시의 경우 프록시 서버에서 자체 서명된 인증서를 사용할 수 있도록 인증서 유효성 검사를 건너뛰십시오. 인증이 활성화된 프록시 서버는 지원되지 않습니다.	
nfsMountOptions	NFS 마운트 옵션에 대한 세밀한 제어	"nfsvers=3"
limitVolumeSize	요청된 볼륨 크기가 이 값보다 큰 경우 용량 할당에 실패합니다.	""(기본적으로 적용되지 않음)
serviceLevel	새 볼륨에 대한 CVS - 성능 또는 CVS 서비스 수준 CV - 성능 값은 입니다 standard, premium, 또는 extreme. CV 값은 입니다 standardsw 또는 zoneredundantstandardsw.	CV - 성능 기본값은 "표준"입니다. CV 기본값은 "standardsw"입니다.
network	Cloud Volumes Service 볼륨에 사용되는 Google Cloud 네트워크	"기본값"
debugTraceFlags	문제 해결 시 사용할 디버그 플래그입니다. 예: \{"api":false, "method":true}. 문제 해결 중이 아니며 자세한 로그 덤프가 필요한 경우가 아니면 이 방법을 사용하지 마십시오.	null입니다
allowedTopologies	지역 간 액세스를 설정하려면 에 대한 StorageClass 정의를 사용합니다 allowedTopologies 모든 지역을 포함해야 합니다. 예를 들면 다음과 같습니다. - key: topology.kubernetes.io/region values: - us-east1 - europe-west1	

볼륨 프로비저닝 옵션

에서 기본 볼륨 프로비저닝을 제어할 수 있습니다 `defaults` 구성 파일의 섹션입니다.

매개 변수	설명	기본값
<code>exportRule</code>	새 볼륨의 내보내기 규칙. CIDR 표기법을 사용하여 IPv4 주소 또는 IPv4 서브넷의 조합을 심표로 구분해야 합니다.	"0.0.0.0/0"
<code>snapshotDir</code>	에 액세스합니다 <code>.snapshot</code> 디렉토리	"거짓"
<code>snapshotReserve</code>	스냅숏용으로 예약된 볼륨의 백분율입니다	""(CVS 기본값 0 허용)
<code>size</code>	새 볼륨의 크기입니다. CVS - 최소 성능은 100GiB입니다. CV 최소값은 1GiB입니다.	CVS - 성능 서비스 유형의 기본값은 "100GiB"입니다. CVS 서비스 유형은 기본값을 설정하지 않지만 최소 1GiB가 필요합니다.

CVS - 성능 서비스 유형의 예

다음 예에서는 CVS - 성능 서비스 유형에 대한 샘플 구성을 제공합니다.

예 1: 최소 구성

기본 CVS - 성능 서비스 유형과 기본 "표준" 서비스 수준을 사용하는 최소 백엔드 구성입니다.

[illegible]

```
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
```

예 2: 서비스 수준 구성

이 샘플에서는 서비스 수준 및 볼륨 기본값을 포함한 백엔드 구성 옵션을 보여 줍니다.

```
--  
version: 1  
storageDriverName: gcp-cvs  
projectNumber: '012345678901'  
apiRegion: us-west2  
apiKey:  
  type: service_account  
  project_id: my-gcp-project  
  private_key_id: "<id_value>"  
  private_key: |  
    -----BEGIN PRIVATE KEY-----  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8 zX5ojY9m  
    XsYg6gyxy4zq7OlwWgLwGa==  
    -----END PRIVATE KEY-----  
  
  client_email: cloudvolumes-admin-sa@my-gcp-  
project.iam.gserviceaccount.com  
  client_id: '123456789012345678901'  
  auth uri: https://accounts.google.com/o/oauth2/auth
```

```
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
proxyURL: http://proxy-server-hostname/
nfsMountOptions: vers=3,proto=tcp,timeo=600
limitVolumeSize: 10Ti
serviceLevel: premium
defaults:
  snapshotDir: 'true'
  snapshotReserve: '5'
  exportRule: 10.0.0.0/24,10.0.1.0/24,10.0.2.100
  size: 5Ti
```

예 3: 가상 풀 구성

이 샘플은 `를` 사용합니다 `storage` 가상 풀 및 `를` 구성합니다 `StorageClasses` 다시 언급한다는 것입니다. 을 참조하십시오 [스토리지 클래스 정의](#) 스토리지 클래스를 정의한 방법을 확인합니다.

이 경우 를 설정하는 모든 가상 풀에 대해 특정 기본값이 설정됩니다 snapshotReserve 5% 및 에서 exportRule 를 0.0.0.0/0으로 설정합니다. 가상 풀은 에 정의되어 있습니다 storage 섹션을 참조하십시오. 각 개별 가상 풀은 고유한 가상 풀을 정의합니다 serviceLevel 그리고 일부 풀은 기본값을 덮어씁니다. 가상 풀 레이블을 사용하여 에 따라 풀을 구분했습니다 performance 및 protection.

[illegible]

```

XsYg6gyxy4zq70lwWgLwGa==
-----END PRIVATE KEY-----
client_email: cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com
client_id: '123456789012345678901'
auth_uri: https://accounts.google.com/o/oauth2/auth
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
nfsMountOptions: vers=3,proto=tcp,timeo=600
defaults:
  snapshotReserve: '5'
  exportRule: 0.0.0.0/0
labels:
  cloud: gcp
region: us-west2
storage:
- labels:
  performance: extreme
  protection: extra
  serviceLevel: extreme
  defaults:
    snapshotDir: 'true'
    snapshotReserve: '10'
    exportRule: 10.0.0.0/24
- labels:
  performance: extreme
  protection: standard
  serviceLevel: extreme
- labels:
  performance: premium
  protection: extra
  serviceLevel: premium
  defaults:
    snapshotDir: 'true'
    snapshotReserve: '10'
- labels:
  performance: premium
  protection: standard
  serviceLevel: premium
- labels:
  performance: standard
  serviceLevel: standard

```

스토리지 클래스 정의

다음 StorageClass 정의는 가상 풀 구성 예에 적용됩니다. 사용 ``parameters.selector`` 볼륨을 호스팅하는 데 사용되는 가상 풀을 각 StorageClass에 대해 지정할 수 있습니다. 볼륨은 선택한 풀에 정의된 측면을 갖습니다.


```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extreme-extra-protection
provisioner: netapp.io/trident
parameters:
  selector: "performance=extreme; protection=extra"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extreme-standard-protection
provisioner: netapp.io/trident
parameters:
  selector: "performance=premium; protection=standard"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium-extra-protection
provisioner: netapp.io/trident
parameters:
  selector: "performance=premium; protection=extra"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-premium
provisioner: netapp.io/trident
parameters:
  selector: "performance=premium; protection=standard"
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-standard
provisioner: netapp.io/trident
parameters:
  selector: "performance=standard"

```

```
allowVolumeExpansion: true
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: cvs-extra-protection
provisioner: netapp.io/trident
parameters:
  selector: "protection=extra"
allowVolumeExpansion: true
```

- 첫 번째 StorageClass입니다 (cvs-extreme-extra-protection)가 첫 번째 가상 풀에 매핑됩니다. 이 풀은 스냅샷 예약 공간이 10%인 최고 성능을 제공하는 유일한 풀입니다.
- 마지막 StorageClass입니다 (cvs-extra-protection) 10%의 스냅샷 예약 공간을 제공하는 스토리지 풀을 호출합니다. Astra Trident는 선택된 가상 풀을 결정하고 스냅샷 예약 요구 사항이 충족되는지 확인합니다.

CVS 서비스 유형 예

다음 예에서는 CVS 서비스 유형에 대한 샘플 구성을 제공합니다.

예 1: 최소 구성

을 사용하는 최소 백엔드 구성입니다 `storageClass` CVS 서비스 유형과 기본값을 지정합니다 `standardsw` 서비스 레벨:

```
--  
version: 1  
storageDriverName: gcp-cvs  
projectNumber: '012345678901'  
storageClass: software  
apiRegion: us-east4  
apiKey:  
  type: service_account  
  project_id: my-gcp-project  
  private_key_id: "<id_value>"  
  private_key: |  
    -----BEGIN PRIVATE KEY-----  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
znHczZsr rtHisIsAbOguSaPIKeyAZNchRAGz lzZE4jK3bl /qp8B4Kws8zX5ojY9m  
XsYg6gyxy4zq7OlwWgLwGa==  
    -----END PRIVATE KEY-----  
  
client_email: cloudvolumes-admin-sa@my-gcp-  
project.iam.gserviceaccount.com
```

```
client_id: '123456789012345678901'
auth_uri: https://accounts.google.com/o/oauth2/auth
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com
serviceLevel: standardsw
```

예 2: 스토리지 풀 구성

이 백엔드 구성은 를 사용합니다 storagePools 스토리지 풀을 구성하려면 다음을 수행합니다.

```
---
version: 1
storageDriverName: gcp-cvs
backendName: gcp-std-so-with-pool
projectNumber: '531265380079'
apiRegion: europe-west1
apiKey:
  type: service_account
  project_id: cloud-native-data
  private_key_id: "<id_value>"
  private_key: |-
    -----BEGIN PRIVATE KEY-----
    MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQDaT+Oui9FBAw19
    L1AGEkrYU5xd9K5NlO5jMkIFND5wCD+Nv+jd1GvtFRLaLK5RvXyF5wzvztmODNS+
    qtScpQ+5cFpQkuGtv9U9+N6qtuVYYO3b504Kp5CtqVPJCgMJaK2j8pZTIqUiMum/
    5/Y9oTbZrjAHSMgJm2nHzFq2X0rqVMAHghI6ATm4DOuWx8XGWKTGIPlc0qPqJlqS
    LLaWOH4VIZQZCAyW5IU9p9CmwqHgdG0uhFNfCgMmED6PBUvVLsLvcq86X+QSWR9k
    ETqElj/sGCenPF7ti1DhGBFafd9hPnxg9PZY29ArEZwY9G/ZjZQX7WPgs0VvxiNR
    DxZRC3GXAgMBAAECggEACn5c59bG/qnVEVI1CwMAalM5M2z09JFh1L1ljKwntNPj
    Vilw2eTW2+UE7HbJru/S7KQgA5Dnn9kvCraEahPRuddUMrD0vG4kTl/IODV6uFuk
    Y0sZfbqd4jMUQ21smvGsqFzwloYWS5qzO1W83ivXH/HW/iqkmY2eW+EPRS/hwSSu
    SscR+SojI7PB0BWSJhlV4yqYf3vcD/D95e12CVHfRCkL85DKumeZ+yHENpiXGZAE
    t8xSs4a50OPm6NHhevCw2a/UQ95/foXNUR450HtbjieJo5o+FF6EYZQGfU2ZHZO8
    37FBKuaJkdGW5xqaI9TL7aqkGkFMF4F2qvOZM+vy8QKBgQD4oVuOkJD1hkTHP86W
    esFlw1kpWyJR9ZA7LI0g/rVpslnX+XdDq0WQf4umDLNau5hYEH9LU6ZSGs1Xk3/B
    NHwR6OXFuqEKNiu83d0zSlHhTy7PZpOZdj5a/vVvQfPDMz7OvsqLRd7YCAbdzuQ0
    +Ahq0Ztwvg0HQ64hdW0ukpYRRwKBgQDgyHj98oqswoYuIa+pPlyS0pPwLmjwKyNm
    /HayzCp+Qjiyy7Tzg8AUqlH1Ou83XbV428jvg7kDhO7PCCKFq+mMmfqHmTpb0Maq
    KpKnZg4ipsqPlyHNNEoRmcailXbwIhCLewMqMrggUiLOmCw4PscL5nK+4GKu2XE1
    jLqjWAZFMQKBgFhkQ9XXRAJ1kR3XpGHoGN890pZOkCVSrqu6aUef/5KY1Fct8ew
    F/+aIXM2iQSVmWQYOvVCnhuY/F2GfAQ7d0om3decuwIOCX/xy7PjHMkLXa2uaZs4
    WR17sLduj62RqXRLX0c0QkwBiNFyHbRcpdkZJQujbYMhBa+7j7SxT4BtAoGAWMWT
    UucocRXZm/pdvz9wteNH3YDwnJLMxm1KC06qMXbBoYrliY4sm3ywJWMC+iCd/H8A
    Gecxd/xVu5mA2L2N3KMq18Zhz8Th0G5DwKyDRJgOQ0Q46yuNXOoYEjlo4Wjyk8Me
    +t1Q8iK98E0UmZnhTgfSpSNElbz2AqnzQ3MN9uECgYAqdvdpVnKGfvdT2ZDjyMoJ
    E89UIC41WjjJGmHsd8W65+3X0RwMzKMT6aZc5tK9J5dHvmWIETnbM+lTImdBbFga
    NWOC6f3r2xbGXHhaWSl+nobpTuvlo56ZRJVvVk7lFMsidzMuHH8pxfgNJemwA4P
    ThDHCEjv035NNV6KyoO0tA==
    -----END PRIVATE KEY-----
  client_email: cloudvolumes-admin-sa@cloud-native-
  data.iam.gserviceaccount.com
  client_id: '107071413297115343396'
```

```
auth_uri: https://accounts.google.com/o/oauth2/auth
token_uri: https://oauth2.googleapis.com/token
auth_provider_x509_cert_url:
https://www.googleapis.com/oauth2/v1/certs
client_x509_cert_url:
https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40cloud-native-data.iam.gserviceaccount.com
storageClass: software
zone: europe-west1-b
network: default
storagePools:
- 1bc7f380-3314-6005-45e9-c7dc8c2d7509
serviceLevel: Standardsw
```

다음 단계

백엔드 구성 파일을 생성한 후 다음 명령을 실행합니다.

```
tridentctl create backend -f <backend-file>
```

백엔드 생성에 실패하면 백엔드 구성에 문제가 있는 것입니다. 다음 명령을 실행하여 로그를 보고 원인을 확인할 수 있습니다.

```
tridentctl logs
```

구성 파일의 문제를 확인하고 수정한 후 create 명령을 다시 실행할 수 있습니다.

NetApp HCI 또는 SolidFire 백엔드를 구성합니다

Astra Trident 설치와 함께 Element 백엔드를 생성하고 사용하는 방법에 대해 알아보십시오.

필요한 것

- Element 소프트웨어를 실행하는 지원되는 스토리지 시스템
- 볼륨을 관리할 수 있는 NetApp HCI/SolidFire 클러스터 관리자 또는 테넌트 사용자에게 대한 자격 증명
- 모든 Kubernetes 작업자 노드에 적절한 iSCSI 톨이 설치되어 있어야 합니다. 을 참조하십시오 ["작업자 노드 준비 정보"](#).

알아야 할 내용

를 클릭합니다 solidfire-san 스토리지 드라이버는 파일 및 블록의 볼륨 모드를 모두 지원합니다. 의 경우 Filesystem 볼륨 코드, Astra Trident가 볼륨을 생성하고 파일 시스템을 생성합니다. 파일 시스템 유형은 StorageClass에 의해 지정됩니다.

드라이버	프로토콜	볼륨 모드	액세스 모드가 지원됩니다	지원되는 파일 시스템
solidfire-san	iSCSI	블록	RWO, ROX, rwx	파일 시스템이 없습니다. 원시 블록 장치.
solidfire-san	iSCSI	블록	RWO, ROX, rwx	파일 시스템이 없습니다. 원시 블록 장치.
solidfire-san	iSCSI	파일 시스템	RWO, ROX	xfs, ext3, ext4
solidfire-san	iSCSI	파일 시스템	RWO, ROX	xfs, ext3, ext4



Astra Trident는 향상된 CSI Provisioner로 작동할 때 CHAP를 사용합니다. CHAP(CSI의 기본값)를 사용하는 경우 추가 준비가 필요하지 않습니다. 를 명시적으로 설정하는 것이 좋습니다 UseCHAP 비 CSI Trident와 CHAP를 사용하는 옵션입니다. 그렇지 않으면 를 참조하십시오 ["여기"](#).



블록 액세스 그룹은 Astra Trident의 기존 비 CSI 프레임워크에서만 지원됩니다. CSI 모드에서 작동하도록 구성된 경우 Astra Trident는 CHAP를 사용합니다.

둘 다 해당되지 않는 경우 AccessGroups 또는 UseCHAP 설정된 경우 다음 규칙 중 하나가 적용됩니다.

- 기본값 trident 액세스 그룹이 감지되고 액세스 그룹이 사용됩니다.
- 액세스 그룹이 감지되지 않고 Kubernetes 버전이 1.7 이상인 경우 CHAP가 사용됩니다.

백엔드 구성 옵션

백엔드 구성 옵션은 다음 표를 참조하십시오.

매개 변수	설명	기본값
version		항상 1
storageDriverName	스토리지 드라이버의 이름입니다	항상 "solidfire-san"
backendName	사용자 지정 이름 또는 스토리지 백엔드	"SolidFire_" + 스토리지(iSCSI) IP 주소입니다
Endpoint	테넌트 자격 증명이 있는 SolidFire 클러스터의 MVIP입니다	
SVIP	스토리지(iSCSI) IP 주소 및 포트	
labels	볼륨에 적용할 임의의 JSON 형식 레이블 세트입니다.	""
TenantName	사용할 테넌트 이름(찾을 수 없는 경우 생성됨)	

매개 변수	설명	기본값
InitiatorIFace	iSCSI 트래픽을 특정 호스트 인터페이스로 제한합니다	“기본값”
UseCHAP	CHAP를 사용하여 iSCSI를 인증합니다	참
AccessGroups	사용할 액세스 그룹 ID 목록입니다	"트리덴트"라는 액세스 그룹의 ID를 찾습니다.
Types	QoS 사양	
limitVolumeSize	요청된 볼륨 크기가 이 값보다 큰 경우 용량 할당에 실패합니다	“(기본적으로 적용되지 않음)
debugTraceFlags	문제 해결 시 사용할 디버그 플래그입니다. 예: {“api”:false, “method”:true}	null입니다



사용하지 마십시오 debugTraceFlags 문제 해결 및 자세한 로그 덤프가 필요한 경우를 제외하고

예 1: 에 대한 백엔드 구성 solidfire-san 세 가지 볼륨 유형을 가진 드라이버

이 예에서는 CHAP 인증을 사용하는 백엔드 파일을 보여 주고 특정 QoS 보장을 포함하는 세 가지 볼륨 유형을 모델링합니다. 그런 다음 예에서 각 스토리지 클래스를 사용할 스토리지 클래스를 정의할 가능성이 높습니다 IOPS 스토리지 클래스 매개 변수입니다.


```

---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0
SVIP: "<svip>:3260"
TenantName: "<tenant>"
labels:
  k8scluster: dev1
  backend: dev1-element-cluster
UseCHAP: true
Types:
- Type: Bronze
  Qos:
    minIOPS: 1000
    maxIOPS: 2000
    burstIOPS: 4000
- Type: Silver
  Qos:
    minIOPS: 4000
    maxIOPS: 6000
    burstIOPS: 8000
- Type: Gold
  Qos:
    minIOPS: 6000
    maxIOPS: 8000
    burstIOPS: 10000

```

예 2: 에 대한 백엔드 및 스토리지 클래스 구성 solidfire-san 가상 풀이 있는 드라이버

이 예에서는 가상 풀과 이를 다시 참조하는 StorageClasses와 함께 구성된 백엔드 정의 파일을 보여 줍니다.

Astra Trident는 스토리지 풀에 있는 레이블을 프로비저닝할 때 백엔드 스토리지 LUN에 복사합니다. 편의를 위해 스토리지 관리자는 가상 풀 및 그룹 볼륨별로 레이블을 레이블별로 정의할 수 있습니다.

아래 표시된 샘플 백엔드 정의 파일에서 특정 기본값은 를 설정하는 모든 스토리지 풀에 대해 설정됩니다 type 실버. 가상 풀은 에 정의되어 있습니다 storage 섹션을 참조하십시오. 이 예에서는 일부 스토리지 풀이 자체 유형을 설정하고 일부 풀은 위에 설정된 기본값을 덮어씁니다.

```

---
version: 1
storageDriverName: solidfire-san
Endpoint: https://<user>:<password>@<mvip>/json-rpc/8.0
SVIP: "<svip>:3260"
TenantName: "<tenant>"
UseCHAP: true

```

```

Types:
- Type: Bronze
  Qos:
    minIOPS: 1000
    maxIOPS: 2000
    burstIOPS: 4000
- Type: Silver
  Qos:
    minIOPS: 4000
    maxIOPS: 6000
    burstIOPS: 8000
- Type: Gold
  Qos:
    minIOPS: 6000
    maxIOPS: 8000
    burstIOPS: 10000
type: Silver
labels:
  store: solidfire
  k8scluster: dev-1-cluster
region: us-east-1
storage:
- labels:
    performance: gold
    cost: '4'
  zone: us-east-1a
  type: Gold
- labels:
    performance: silver
    cost: '3'
  zone: us-east-1b
  type: Silver
- labels:
    performance: bronze
    cost: '2'
  zone: us-east-1c
  type: Bronze
- labels:
    performance: silver
    cost: '1'
  zone: us-east-1d

```

다음 StorageClass 정의는 위의 가상 풀을 참조합니다. 를 사용합니다 parameters.selector 필드에서 각 StorageClass는 볼륨을 호스팅하는 데 사용할 수 있는 가상 풀을 호출합니다. 선택한 가상 풀에 볼륨이 정의되어 있습니다.

첫 번째 StorageClass입니다 (solidfire-gold-four)는 첫 번째 가상 풀에 매핑합니다. 이 수영장은 골드 성능을 제공하는 유일한 수영장입니다 Volume Type QoS 않습니다. 마지막 StorageClass입니다 (solidfire-silver)은 뛰어난 성능을 제공하는 스토리지 풀을 호출합니다. Astra Trident가 선택한 가상 풀을 결정하고 스토리지 요구 사항을 충족시킵니다.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-gold-four
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=gold; cost=4"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-three
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=silver; cost=3"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-bronze-two
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=bronze; cost=2"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver-one
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=silver; cost=1"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: solidfire-silver
provisioner: csi.trident.netapp.io
parameters:
  selector: "performance=silver"
  fsType: "ext4"

```

자세한 내용을 확인하십시오

- "볼륨 액세스 그룹"

ONTAP SAN 드라이버를 사용하여 백엔드를 구성합니다

ONTAP 및 Cloud Volumes ONTAP SAN 드라이버를 사용하여 ONTAP 백엔드를 구성하는 방법에 대해 알아보십시오.

- "준비"
- "구성 및 예"

Astra Control은 로 생성한 볼륨을 위해 원활한 보호, 재해 복구, 이동성(Kubernetes 클러스터 간에 볼륨 이동)을 제공합니다 `ontap-nas`, `ontap-nas-flexgroup`, 및 `ontap-san` 드라이버. 을 참조하십시오 ["Astra Control 복제 사전 요구 사항"](#) 을 참조하십시오.



- 를 사용해야 합니다 `ontap-nas` 데이터 보호, 재해 복구 및 이동성을 필요로 하는 운영 워크로드에 적합합니다.
- 사용 `ontap-san-economy` 예상되는 볼륨 사용량이 ONTAP가 지원하는 것보다 훨씬 많을 것으로 예상됩니다.
- 사용 `ontap-nas-economy` 예상되는 볼륨 사용량이 ONTAP에서 지원하는 것보다 훨씬 많을 것으로 예상되는 경우에만 및 입니다 `ontap-san-economy` 드라이버를 사용할 수 없습니다.
- 사용하지 마십시오 `ontap-nas-economy` 데이터 보호, 재해 복구 또는 이동성이 필요할 것으로 예상되는 경우

사용자 권한

Astra Trident는 일반적으로 를 사용하여 ONTAP 또는 SVM 관리자로 실행될 것으로 예상합니다 `admin` 클러스터 사용자 또는 입니다 `vsadmin` SVM 사용자 또는 동일한 역할을 가진 다른 이름의 사용자 NetApp ONTAP 구축을 위한 Amazon FSx의 경우, Astra Trident는 클러스터를 사용하여 ONTAP 또는 SVM 관리자로 실행될 것으로 예상합니다 `fsxadmin` 사용자 또는 `a vsadmin` SVM 사용자 또는 동일한 역할을 가진 다른 이름의 사용자 를 클릭합니다 `fsxadmin` 사용자는 클러스터 관리자를 제한적으로 대체합니다.



를 사용하는 경우 `limitAggregateUsage` 매개 변수, 클러스터 관리자 권한이 필요합니다. Astra Trident와 함께 NetApp ONTAP에 Amazon FSx를 사용하는 경우, 를 참조하십시오 `limitAggregateUsage` 매개 변수는 에서 작동하지 않습니다 `vsadmin` 및 `fsxadmin` 사용자 계정. 이 매개 변수를 지정하면 구성 작업이 실패합니다.

ONTAP 내에서 Trident 드라이버가 사용할 수 있는 보다 제한적인 역할을 만들 수 있지만 권장하지 않습니다. Trident의 대부분의 새로운 릴리즈에서는 추가 API를 호출하므로 업그레이드가 어렵고 오류가 발생하기 쉽습니다.

ONTAP SAN 드라이버를 사용하여 백엔드를 구성할 준비를 합니다

ONTAP SAN 드라이버를 사용하여 ONTAP 백엔드를 구성하는 방법에 대해 알아보십시오. 모든 ONTAP 백엔드의 경우, Astra Trident는 SVM에 하나 이상의 Aggregate가 할당되어 있어야 합니다.

또한 둘 이상의 드라이버를 실행하고 둘 중 하나를 가리키는 스토리지 클래스를 생성할 수도 있습니다. 예를 들어, 을 구성할 수 있습니다 `san-dev` 를 사용하는 클래스입니다 `ontap-san` 드라이버 및 `A san-default` 를 사용하는

클래스입니다 `ontap-san-economy` 1개.

모든 Kubernetes 작업자 노드에는 적절한 iSCSI 톨이 설치되어 있어야 합니다. 을 참조하십시오 ["여기"](#) 를 참조하십시오.

인증

Astra Trident는 ONTAP 백엔드를 인증하는 두 가지 모드를 제공합니다.

- 자격 증명 기반: 필요한 권한이 있는 ONTAP 사용자의 사용자 이름 및 암호입니다. 과 같이 미리 정의된 보안 로그인 역할을 사용하는 것이 좋습니다 `admin` 또는 `vsadmin` ONTAP 버전과의 호환성을 최대한 보장하기 위해
- 인증서 기반: Astra Trident는 백엔드에 설치된 인증서를 사용하여 ONTAP 클러스터와 통신할 수도 있습니다. 이 경우 백엔드 정의에는 클라이언트 인증서, 키 및 사용할 경우 신뢰할 수 있는 CA 인증서의 Base64로 인코딩된 값이 있어야 합니다(권장).

자격 증명 기반 방법과 인증서 기반 방법 간에 이동하기 위해 기존 백엔드를 업데이트할 수 있습니다. 그러나 한 번에 하나의 인증 방법만 지원됩니다. 다른 인증 방법으로 전환하려면 백엔드 구성에서 기존 방법을 제거해야 합니다.



자격 증명과 인증서 * 를 모두 제공하려고 하면 구성 파일에 둘 이상의 인증 방법이 제공된다는 오류가 발생하여 백엔드 생성이 실패합니다.

자격 증명 기반 인증을 사용합니다

Astra Trident는 SVM 범위/클러스터 범위 관리자에게 ONTAP 백엔드와 통신하기 위한 자격 증명을 요구합니다. 과 같이 미리 정의된 표준 역할을 사용하는 것이 좋습니다 `admin` 또는 `vsadmin`. 이를 통해 향후 Astra Trident 릴리스에서 사용할 기능 API를 노출할 수 있는 향후 ONTAP 릴리스와 향후 호환성이 보장됩니다. 사용자 지정 보안 로그인 역할은 Astra Trident와 함께 생성 및 사용할 수 있지만 권장되지 않습니다.

백엔드 정의의 예는 다음과 같습니다.

YAML

버전: 1 backendName: ExampleBackend storageDriverName: ONTAP-SAN 관리 LIF: 10.0.0.1
svm:svm_nfs 사용자 이름: vsadmin 암호: password

JSON을 참조하십시오

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password"
}
```

백엔드 정의는 자격 증명이 일반 텍스트로 저장되는 유일한 위치라는 점에 유의하십시오. 백엔드가 생성된 후 사용자 이름/암호는 Base64로 인코딩되어 Kubernetes 암호로 저장됩니다. 백엔드의 생성 또는 업데이트는 자격 증명에 대한 지식이 필요한 유일한 단계입니다. 따라서 Kubernetes/스토리지 관리자가 수행할 수 있는 관리 전용 작업입니다.

인증서 기반 인증을 사용합니다

신규 및 기존 백엔드는 인증서를 사용하여 ONTAP 백엔드와 통신할 수 있습니다. 백엔드 정의에는 세 가지 매개 변수가 필요합니다.

- `clientCertificate`: Base64로 인코딩된 클라이언트 인증서 값입니다.
- `clientPrivateKey`: Base64 - 연결된 개인 키의 인코딩된 값입니다.
- `TrustedCACertificate`: 신뢰할 수 있는 CA 인증서의 Base64 인코딩 값입니다. 신뢰할 수 있는 CA를 사용하는 경우 이 매개 변수를 제공해야 합니다. 신뢰할 수 있는 CA가 사용되지 않으면 이 작업을 무시할 수 있습니다.

일반적인 워크플로에는 다음 단계가 포함됩니다.

단계

1. 클라이언트 인증서 및 키를 생성합니다. 생성 시 CN(일반 이름)을 ONTAP 사용자로 설정하여 인증하십시오.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key  
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=admin"
```

2. 신뢰할 수 있는 CA 인증서를 ONTAP 클러스터에 추가합니다. 이는 스토리지 관리자가 이미 처리한 것일 수 있습니다. 트러스트된 CA가 사용되지 않으면 무시합니다.

```
security certificate install -type server -cert-name <trusted-ca-cert-name> -vserver <vserver-name>  
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled  
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca  
<cert-authority>
```

3. ONTAP 클러스터에 클라이언트 인증서 및 키(1단계)를 설치합니다.

```
security certificate install -type client-ca -cert-name <certificate-name> -vserver <vserver-name>  
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. ONTAP 보안 로그인 역할이 지원되는지 확인합니다 cert 인증 방법.

```
security login create -user-or-group-name admin -application ontapi  
-authentication-method cert  
security login create -user-or-group-name admin -application http  
-authentication-method cert
```

5. 생성된 인증서를 사용하여 인증을 테스트합니다. ONTAP 관리 LIF> 및 <SVM 이름>을 관리 LIF IP 및 SVM 이름으로 바꿉니다.

```
curl -X POST -Lk https://<ONTAP-Management-
LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp
xmlns="http://www.netapp.com/filer/admin" version="1.21"
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Base64로 인증서, 키 및 신뢰할 수 있는 CA 인증서를 인코딩합니다.

```
base64 -w 0 k8senv.pem >> cert_base64
base64 -w 0 k8senv.key >> key_base64
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. 이전 단계에서 얻은 값을 사용하여 백엔드를 생성합니다.

```
cat cert-backend.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "SanBackend",
  "managementLIF": "1.2.3.4",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuuuueeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "trustedCACertificate": "QNFinfO...SiqOyN",
  "storagePrefix": "myPrefix_"
}

tridentctl create backend -f cert-backend.json -n trident
+-----+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |                      UUID                      |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |         0 |
+-----+-----+-----+-----+
+-----+-----+
```


인증 방법을 업데이트하거나 자격 증명을 회전합니다

다른 인증 방법을 사용하거나 자격 증명을 회전하도록 기존 백엔드를 업데이트할 수 있습니다. 이렇게 하면 사용자 이름/암호를 사용하는 백엔드를 인증서를 사용하도록 업데이트할 수 있고 인증서를 사용하는 백엔드는 사용자 이름/암호 기반으로 업데이트할 수 있습니다. 이렇게 하려면 기존 인증 방법을 제거하고 새 인증 방법을 추가해야 합니다. 그런 다음 실행할 필수 매개 변수가 포함된 업데이트된 backend.json 파일을 사용합니다 tridentctl backend update.

```
cat cert-backend-updated.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "SanBackend",
  "managementLIF": "1.2.3.4",
  "svm": "vserver_test",
  "username": "vsadmin",
  "password": "password",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend SanBackend -f cert-backend-updated.json -n
trident
+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |                                UUID                                |
STATE  | VOLUMES  |
+-----+-----+-----+
+-----+-----+
| SanBackend | ontap-san      | 586b1cd5-8cf8-428d-a76c-2872713612c1 |
online |          9 |
+-----+-----+-----+
+-----+-----+-----+
+-----+-----+

```



암호를 회전할 때 스토리지 관리자는 먼저 ONTAP에서 사용자의 암호를 업데이트해야 합니다. 그 다음에는 백엔드 업데이트가 있습니다. 인증서를 회전할 때 여러 인증서를 사용자에게 추가할 수 있습니다. 그런 다음 백엔드가 업데이트되어 새 인증서를 사용합니다. 그러면 ONTAP 클러스터에서 이전 인증서를 삭제할 수 있습니다.

백엔드를 업데이트해도 이미 생성된 볼륨에 대한 액세스가 중단되거나 이후에 생성된 볼륨 연결에 영향을 미치지 않습니다. 백엔드 업데이트가 성공적이면 Astra Trident가 ONTAP 백엔드와 통신하고 향후 볼륨 작업을 처리할 수 있음을 나타냅니다.

Igroup을 지정합니다

Astra Trident에서 igroup을 사용하여 프로비저닝하는 볼륨(LUN)에 대한 액세스를 제어합니다. 관리자는 백엔드에 대한 igroup을 지정할 때 다음 두 가지 옵션을 사용할 수 있습니다.

- Astra Trident는 백엔드에 따라 igroup을 자동으로 생성하고 관리할 수 있습니다. If(경우 igroupName 백엔드 정의에 포함되지 않은 Astra Trident는 이름을 가진 igroup을 생성합니다 trident-<backend-UUID> SVM에서. 그러면 각 백엔드에 전용 igroup이 있고 Kubernetes 노드 IQN의 자동 추가/삭제를 처리합니다.
- 또는 미리 생성된 igroup을 백엔드 정의로 제공할 수도 있습니다. 이 작업은 를 사용하여 수행할 수 있습니다 igroupName 구성 매개 변수입니다. Astra Trident가 기존 igroup에 Kubernetes 노드 IQN을 추가/삭제합니다.

을(를) 가진 백엔드의 경우 igroupName 정의, 입니다 igroupName 을 사용하여 삭제할 수 있습니다 tridentctl backend update Astra Trident에서 igroup 자동 처리 이 경우 워크로드에 이미 연결된 볼륨에 대한 액세스가 중단되지 않습니다. 생성된 igroup Astra Trident를 사용하여 향후 연결을 처리할 것입니다.



Astra Trident의 각 고유 인스턴스에 대해 igroup을 할당하는 것은 Kubernetes 관리자 및 스토리지 관리자에게 유용한 모범 사례입니다. CSI Trident는 클러스터 노드 IQN을 igroup에 추가 및 제거하여 관리를 크게 단순화합니다. 전용 igroup을 사용하여 Kubernetes 환경(및 Astra Trident 설치)에서 동일한 SVM을 사용할 경우 한 Kubernetes 클러스터의 변경 사항이 다른 Kubernetes 클러스터와 관련된 igroup에 영향을 미치지 않도록 합니다. 또한 Kubernetes 클러스터의 각 노드에 고유한 IQN이 있는지 확인하는 것도 중요합니다. 위에 언급한 바와 같이, Astra Trident는 IQN의 추가 및 제거를 자동으로 처리합니다. 호스트 간에 IQN을 재사용하면 호스트가 서로 잘못 인식되어 LUN에 대한 액세스가 거부되는 바람직하지 않은 시나리오가 발생할 수 있습니다.

Astra Trident가 CSI Provisioner로 작동하도록 구성된 경우 Kubernetes 노드 IQN이 igroup에 자동으로 추가/제거됩니다. Kubernetes 클러스터에 노드를 추가할 경우, trident-csi DemonSet가 포드를 전개합니다 (trident-csi-xxxxxx 23.01 또는 이전 버전 trident-node<operating system>-xxxxx 23.01 이상에서)를 새로 추가한 노드에 추가하고 볼륨을 연결할 수 있는 새 노드를 등록합니다. 노드 IQN도 백엔드의 igroup에 추가됩니다. 이와 유사한 일련의 단계에서는 Kubernetes에서 노드에 코드로닝, 드레이닝 및 삭제가 발생하는 경우 IQN 제거를 처리합니다.

Astra Trident가 CSI Provisioner로 실행되지 않을 경우, Kubernetes 클러스터의 모든 작업자 노드에서 iSCSI IQN을 포함하도록 igroup을 수동으로 업데이트해야 합니다. Kubernetes 클러스터에 참여하는 노드의 IQN을 igroup에 추가해야 합니다. 마찬가지로, Kubernetes 클러스터에서 제거된 노드의 IQN을 igroup에서 제거해야 합니다.

양방향 **CHAP**를 사용하여 연결을 인증합니다

Astra Trident는 의 양방향 CHAP를 사용하여 iSCSI 세션을 인증할 수 있습니다 ontap-san 및 ontap-san-economy 드라이버. 이를 위해서는 가 활성화되어야 합니다 useCHAP 백엔드 정의에서 선택할 수 있습니다. 를 로 설정한 경우 true, Astra Trident는 SVM의 기본 이니시에이터 보안을 양방향 CHAP로 구성하고 백엔드 파일의 사용자 이름과 암호를 설정합니다. 양방향 CHAP를 사용하여 연결을 인증하는 것이 좋습니다. 다음 샘플 구성을 참조하십시오.

```

---
version: 1
storageDriverName: ontap-san
backendName: ontap_san_chap
managementLIF: 192.168.0.135
svm: ontap_iscsi_svm
useCHAP: true
username: vsadmin
password: password
igroupName: trident
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz

```



를 클릭합니다 useCHAP 매개 변수는 한 번만 구성할 수 있는 부울 옵션입니다. 기본적으로 false로 설정되어 있습니다. true 로 설정한 후에는 false 로 설정할 수 없습니다.

또한 useCHAP=true, chapInitiatorSecret, chapTargetInitiatorSecret, chapTargetUsername, 및 chapUsername 필드는 백엔드 정의에 포함되어야 합니다. 을 실행하여 백엔드를 생성한 후 암호를 변경할 수 있습니다 tridentctl update.

작동 방식

설정을 통해 useCHAP 스토리지 관리자는 스토리지 백엔드에서 CHAP를 구성하도록 Astra Trident에 지시합니다. 여기에는 다음이 포함됩니다.

- SVM에서 CHAP 설정:
 - SVM의 기본 이니시에이터 보안 유형이 없음(기본값 설정) * 이고 * 볼륨에 이미 기존 LUN이 없으면 Astra Trident가 기본 보안 유형을 로 설정합니다 CHAP CHAP 이니시에이터와 타겟 사용자 이름 및 암호 구성 을 진행합니다.
 - SVM에 LUN이 포함된 경우 Astra Trident는 SVM에서 CHAP를 활성화하지 않습니다. 따라서 SVM에 이미 있는 LUN에 대한 액세스가 제한되지 않습니다.
- CHAP 이니시에이터 및 타겟 사용자 이름과 암호를 구성합니다. 이러한 옵션은 백엔드 구성에 지정해야 합니다(위 참조).
- 에 이니시에이터 추가 관리 igroupName 백엔드에서 제공됩니다. 지정되지 않은 경우 이 기본값은 입니다 trident.

백엔드가 생성된 후 Astra Trident가 해당 을 생성합니다 tridentbackend CHAP 암호 및 사용자 이름을 Kubernetes 비밀로 CRD 및 저장합니다. 이 백엔드에서 Astra Trident에 의해 생성된 모든 PVS는 CHAP를 통해 마운트되고 연결됩니다.

자격 증명을 회전하고 백엔드를 업데이트합니다

에서 CHAP 매개 변수를 업데이트하여 CHAP 자격 증명을 업데이트할 수 있습니다 backend.json 파일. CHAP 암호를 업데이트하고 를 사용해야 합니다 tridentctl update 명령을 사용하여 이러한 변경 사항을 반영합니다.



백엔드의 CHAP 암호를 업데이트할 때 를 사용해야 합니다 tridentctl 백엔드를 업데이트합니다. Astra Trident에서 변경 사항을 선택할 수 없으므로 CLI/ONTAP UI를 통해 스토리지 클러스터의 자격 증명을 업데이트하지 마십시오.

```
cat backend-san.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "ontap_san_chap",
  "managementLIF": "192.168.0.135",
  "svm": "ontap_iscsi_svm",
  "useCHAP": true,
  "username": "vsadmin",
  "password": "password",
  "igroupName": "trident",
  "chapInitiatorSecret": "cl9qxUpDaTeD",
  "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
}
```

```
./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+-----+-----+-----+-----+
+-----+-----+
|  NAME          | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbeb5c |
online |      7 |
+-----+-----+-----+-----+
+-----+-----+
```

기존 연결은 영향을 받지 않습니다. SVM에서 Astra Trident가 자격 증명을 업데이트하면 활성 상태로 유지됩니다. 새 연결은 업데이트된 자격 증명을 사용하며 기존 연결은 계속 활성 상태로 유지됩니다. 기존 PVS를 연결 해제하고 다시 연결하면 업데이트된 자격 증명을 사용하게 됩니다.

ONTAP SAN 구성 옵션 및 예

Astra Trident 설치를 통해 ONTAP SAN 드라이버를 생성하고 사용하는 방법에 대해 알아보십시오. 이 섹션에서는 백엔드 구성 예제 및 백엔드를 StorageClasses에 매핑하는 방법에 대한 세부 정보를 제공합니다.

백엔드 구성 옵션

백엔드 구성 옵션은 다음 표를 참조하십시오.

매개 변수	설명	기본값
version		항상 1
storageDriverName	스토리지 드라이버의 이름입니다	"ONTAP-NAS", "ONTAP-NAS-이코노미", "ONTAP-NAS-Flexgroup", "ONTAP-SAN", "ONTAP-SAN-이코노미"
backendName	사용자 지정 이름 또는 스토리지 백엔드	드라이버 이름 + "_" + dataLIF
managementLIF	클러스터 또는 SVM 관리 LIF의 IP 주소 MetroCluster의 원활한 전환을 위해 SVM 관리 LIF를 지정해야 합니다. FQDN(정규화된 도메인 이름)을 지정할 수 있습니다. 을 사용하여 Astra Trident가 설치된 경우 IPv6 주소를 사용하도록 설정할 수 있습니다 --use-ipv6 깃발. IPv6 주소는 대괄호로 묶어야 합니다(예: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]).	"10.0.0.1", "[2001:1234:ABCD::fee]"
dataLIF	프로토콜 LIF의 IP 주소입니다. * iSCSI에 대해서는 지정하지 마십시오. * Astra Trident가 사용합니다 "ONTAP 선택적 LUN 맵" 다중 경로 세션을 설정하는 데 필요한 iSCSI LIF를 검색하려면 경고 발생 시 dataLIF 명시적으로 정의됩니다.	SVM에서 파생됩니다
useCHAP	CHAP를 사용하여 ONTAP SAN 드라이버에 대한 iSCSI 인증 [Boolean]. 를 로 설정합니다 true Astra Trident에서 백엔드에 제공된 SVM에 대한 기본 인증으로 양방향 CHAP를 구성하고 사용합니다. 을 참조하십시오 "ONTAP SAN 드라이버를 사용하여 백엔드를 구성할 준비를 합니다" 를 참조하십시오.	거짓
chapInitiatorSecret	CHAP 이니시에이터 암호입니다. 필요한 경우 useCHAP=true	""
labels	볼륨에 적용할 임의의 JSON 형식 레이블 세트입니다	""
chapTargetInitiatorSecret	CHAP 타겟 이니시에이터 암호입니다. 필요한 경우 useCHAP=true	""
chapUsername	인바운드 사용자 이름입니다. 필요한 경우 useCHAP=true	""
chapTargetUsername	대상 사용자 이름입니다. 필요한 경우 useCHAP=true	""

매개 변수	설명	기본값
clientCertificate	Base64 - 클라이언트 인증서의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다	""
clientPrivateKey	Base64 - 클라이언트 개인 키의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다	""
trustedCACertificate	Base64 - 신뢰할 수 있는 CA 인증서의 인코딩된 값입니다. 선택 사항. 인증서 기반 인증에 사용됩니다.	""
username	ONTAP 클러스터와 통신하는 데 필요한 사용자 이름입니다. 자격 증명 기반 인증에 사용됩니다.	""
password	ONTAP 클러스터와 통신하는 데 필요한 암호입니다. 자격 증명 기반 인증에 사용됩니다.	""
svm	사용할 스토리지 가상 머신입니다	SVM에서 파생된 경우 managementLIF 이(가) 지정되었습니다
igroupName	사용할 SAN 볼륨에 대한 igroup의 이름입니다. 을 참조하십시오 를 참조하십시오.	"삼중 - <backend-UUID>"
storagePrefix	SVM에서 새 볼륨을 프로비저닝할 때 사용되는 접두사 나중에서 수정할 수 없습니다. 이 매개 변수를 업데이트하려면 새 백엔드를 생성해야 합니다.	"삼중류"
limitAggregateUsage	사용량이 이 비율을 초과하면 프로비저닝이 실패합니다. NetApp ONTAP 백엔드에 Amazon FSx를 사용하는 경우 를 지정하지 마십시오 limitAggregateUsage. 제공 fsxadmin 및 vsadmin 애그리게이트 사용을 검색하고 Astra Trident를 사용하여 제한하는 데 필요한 권한이 포함되어 있지 않습니다.	""(기본적으로 적용되지 않음)
limitVolumeSize	요청된 볼륨 크기가 이 값보다 큰 경우 용량 할당에 실패합니다. 또한 qtree 및 LUN에 대해 관리하는 볼륨의 최대 크기도 제한합니다.	""(기본적으로 적용되지 않음)
lunsPerFlexvol	FlexVol당 최대 LUN 수는 범위[50, 200]에 있어야 합니다.	"100"

매개 변수	설명	기본값
debugTraceFlags	문제 해결 시 사용할 디버그 플래그입니다. 예: {"api":false, "method":true} 문제 해결 및 자세한 로그 덤프가 필요한 경우가 아니면 사용하지 마십시오.	null입니다
useREST	ONTAP REST API를 사용하는 부울 매개 변수입니다. * 기술 미리 보기 * useREST 프로덕션 작업 부하가 아닌 테스트 환경에 권장되는 기술 미리 보기로 제공됩니다. 를 로 설정한 경우 true, Astra Trident는 ONTAP REST API를 사용하여 백엔드와 통신합니다. 이 기능을 사용하려면 ONTAP 9.11.1 이상이 필요합니다. 또한 사용되는 ONTAP 로그인 역할에 대한 액세스 권한이 있어야 합니다 ontap 응용 프로그램. 이는 사전 정의된 에 의해 충족됩니다 vsadmin 및 cluster-admin 역할. useREST 는 MetroCluster에서 지원되지 않습니다.	거짓

에 대한 세부 정보 igroupName

igroupName ONTAP 클러스터에서 이미 생성된 igroup으로 설정할 수 있습니다. 지정되지 않은 경우 Astra Trident가 이름이 인 igroup을 자동으로 생성합니다 trident-<backend-UUID>.

미리 정의된 횃수 이름을 제공하는 경우 환경 간에 SVM을 공유하려면 Kubernetes 클러스터 당 하나의 igroup을 사용하는 것이 좋습니다. 이는 Astra Trident가 IQN 추가 및 삭제를 자동으로 유지 관리하는 데 필요합니다.

- igroupName Astra Trident 외부의 SVM에서 생성 및 관리되는 새로운 igroup을 가리키도록 업데이트할 수 있습니다.
- igroupName 생략할 수 있습니다. 이 경우 Astra Trident가 이름을 가진 igroup을 생성하고 관리합니다 trident-<backend-UUID> 자동으로.

두 경우 모두 볼륨 첨부 파일에 계속 액세스할 수 있습니다. 향후 볼륨 첨부 파일은 업데이트된 igroup을 사용합니다. 이 업데이트는 백엔드에 있는 볼륨에 대한 액세스를 방해하지 않습니다.

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
spaceAllocation	LUN에 대한 공간 할당	"참"
spaceReserve	공간 예약 모드, "없음"(썬) 또는 "볼륨"(일반)	"없음"
snapshotPolicy	사용할 스냅샷 정책입니다	"없음"

매개 변수	설명	기본값
qosPolicy	생성된 볼륨에 할당할 QoS 정책 그룹입니다. 스토리지 풀/백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다. Astra Trident와 함께 QoS 정책 그룹을 사용하려면 ONTAP 9.8 이상이 필요합니다. 비공유 QoS 정책 그룹을 사용하고 정책 그룹이 각 구성요소별로 적용되도록 하는 것이 좋습니다. 공유 QoS 정책 그룹은 모든 워크로드의 총 처리량에 대해 상한을 적용합니다.	""
adaptiveQosPolicy	생성된 볼륨에 할당할 적응형 QoS 정책 그룹입니다. 스토리지 풀/백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다	""
snapshotReserve	스냅샷 "0"에 예약된 볼륨의 백분율	If(경우 snapshotPolicy "없음"이고, 그렇지 않으면 ""입니다.
splitOnClone	생성 시 상위 클론에서 클론을 분할합니다	"거짓"
encryption	새 볼륨에 NVE(NetApp Volume Encryption)를 활성화합니다. 기본값은 입니다 false. 이 옵션을 사용하려면 NVE 라이선스가 클러스터에서 활성화되어 있어야 합니다. 백엔드에서 NAE가 활성화된 경우 Astra Trident에 프로비저닝된 모든 볼륨은 NAE가 활성화됩니다. 자세한 내용은 다음을 참조하십시오. "Astra Trident가 NVE 및 NAE와 연동되는 방식" .	"거짓"
luksEncryption	LUKS 암호화를 사용합니다. 을 참조하십시오 "LUKS(Linux Unified Key Setup) 사용" .	""
securityStyle	새로운 볼륨에 대한 보안 스타일	unix
tieringPolicy	"없음"을 사용하는 계층화 정책	ONTAP 9.5 이전 SVM-DR 구성의 경우 "스냅샷 전용

볼륨 프로비저닝의 예

다음은 기본값이 정의된 예입니다.


```

---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: trident_svm
username: admin
password: password
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
igroupName: custom
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'

```



를 사용하여 생성된 모든 볼륨에 대해 ontap-san 드라이버, Astra Trident는 FlexVol에 10%의 용량을 추가하여 LUN 메타데이터를 수용합니다. LUN은 사용자가 PVC에서 요청하는 정확한 크기로 프로비저닝됩니다. Astra Trident가 FlexVol에 10%를 더합니다(ONTAP에서 사용 가능한 크기로 표시). 이제 사용자가 요청한 가용 용량을 얻을 수 있습니다. 또한 이 변경으로 인해 사용 가능한 공간이 완전히 활용되지 않는 한 LUN이 읽기 전용이 되는 것을 방지할 수 있습니다. ONTAP-SAN-경제에는 적용되지 않습니다.

을 정의하는 백엔드의 경우 snapshotReserve, Astra Trident는 다음과 같이 볼륨의 크기를 계산합니다.

```

Total volume size = [(PVC requested size) / (1 - (snapshotReserve
percentage) / 100)] * 1.1

```

1.1은 LUN 메타데이터를 수용하도록 FlexVol에 추가된 10%의 Astra Trident입니다. 용 snapshotReserve = 5%, PVC 요청 = 5GiB인 경우 총 볼륨 크기는 5.79GiB이고 사용 가능한 크기는 5.5GiB입니다. 를 클릭합니다 volume show 명령은 이 예제와 유사한 결과를 표시해야 합니다.

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
	_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4		online	RW	10GB	5.00GB	0%
	_pvc_e42ec6fe_3baa_4af6_996d_134adbbb8e6d		online	RW	5.79GB	5.50GB	0%
	_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba		online	RW	1GB	511.8MB	0%

3 entries were displayed.

현재 기존 볼륨에 대해 새 계산을 사용하는 유일한 방법은 크기 조정입니다.

최소 구성의 예

다음 예에서는 대부분의 매개 변수를 기본값으로 두는 기본 구성을 보여 줍니다. 이는 백엔드를 정의하는 가장 쉬운 방법입니다.



Astra Trident가 있는 NetApp ONTAP에서 Amazon FSx를 사용하는 경우 IP 주소 대신 LIF에 대한 DNS 이름을 지정하는 것이 좋습니다.

ontap-san 인증서 기반 인증을 사용하는 드라이버

이는 최소 백엔드 구성의 예입니다. `clientCertificate`, `clientPrivateKey`, 및 `trustedCACertificate` (신뢰할 수 있는 CA를 사용하는 경우 선택 사항)는 에 채워집니다 `backend.json` 그리고 각각 클라이언트 인증서, 개인 키 및 신뢰할 수 있는 CA 인증서의 base64로 인코딩된 값을 사용합니다.

```
---
version: 1
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
igroupName: trident
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

ontap-san 양방향 **CHAP**가 있는 드라이버

이는 최소 백엔드 구성의 예입니다. 이 기본 구성은 를 생성합니다 `ontap-san` 을(를) 이용한 백엔드 `useCHAP` 를 로 설정합니다 `true`.

```

---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
igroupName: trident
username: vsadmin
password: password

```

ontap-san-economy 드라이버

```

---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
igroupName: trident
username: vsadmin
password: password

```

가상 풀의 백엔드 예

아래 표시된 샘플 백엔드 정의 파일에서와 같은 모든 스토리지 풀에 대한 특정 기본값이 설정됩니다. spaceReserve 없음, spaceAllocation 거짓일 경우, 및 encryption 거짓일 때. 가상 풀은 스토리지 섹션에 정의됩니다.

Astra Trident는 "Comments" 필드에 제공 레이블을 설정합니다. FlexVol에 주석이 설정됩니다. Astra Trident는 프로비저닝할 때 가상 풀에 있는 모든 레이블을 스토리지 볼륨에 복사합니다. 편의를 위해 스토리지 관리자는 가상 풀 및 그룹 볼륨별로 레이블을 레이블별로 정의할 수 있습니다.

이 예에서는 일부 스토리지 풀이 자체적으로 설정됩니다. spaceReserve, spaceAllocation, 및 encryption 값 및 일부 풀이 위에 설정된 기본값을 덮어씁니다.

```

---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
igroupName: trident
username: vsadmin
password: password
defaults:
  spaceAllocation: 'false'
  encryption: 'false'
  qosPolicy: standard
labels:
  store: san_store
  kubernetes-cluster: prod-cluster-1
region: us_east_1
storage:
- labels:
  protection: gold
  creditpoints: '40000'
  zone: us_east_1a
  defaults:
    spaceAllocation: 'true'
    encryption: 'true'
    adaptiveQosPolicy: adaptive-extreme
- labels:
  protection: silver
  creditpoints: '20000'
  zone: us_east_1b
  defaults:
    spaceAllocation: 'false'
    encryption: 'true'
    qosPolicy: premium
- labels:
  protection: bronze
  creditpoints: '5000'
  zone: us_east_1c
  defaults:
    spaceAllocation: 'true'
    encryption: 'false'

```

iSCSI의 예는 다음과 같습니다 ontap-san-economy 드라이버:

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
igroupName: trident
username: vsadmin
password: password
defaults:
  spaceAllocation: 'false'
  encryption: 'false'
labels:
  store: san_economy_store
region: us_east_1
storage:
- labels:
  app: oracledb
  cost: '30'
  zone: us_east_1a
  defaults:
    spaceAllocation: 'true'
    encryption: 'true'
- labels:
  app: postgresdb
  cost: '20'
  zone: us_east_1b
  defaults:
    spaceAllocation: 'false'
    encryption: 'true'
- labels:
  app: mysqldb
  cost: '10'
  zone: us_east_1c
  defaults:
    spaceAllocation: 'true'
    encryption: 'false'
```

백엔드를 **StorageClasses**에 매핑합니다

다음 StorageClass 정의는 위의 가상 풀을 참조합니다. 를 사용합니다 `parameters.selector` 필드에서 각 StorageClass는 볼륨을 호스팅하는 데 사용할 수 있는 가상 풀을 호출합니다. 선택한 가상 풀에 볼륨이 정의되어 있습니다.

- 첫 번째 StorageClass입니다 (`protection-gold`)는 의 첫 번째, 두 번째 가상 풀에 매핑됩니다 `ontap-nas-flexgroup` 백엔드 및 의 첫 번째 가상 풀 `ontap-san` 백엔드. 골드 레벨 보호 기능을 제공하는 유일한 풀입니다.
- 두 번째 StorageClass입니다 (`protection-not-gold`)는 에서 세 번째, 네 번째 가상 풀에 매핑됩니다 `ontap-nas-flexgroup` 백엔드 및 에서 세 번째 가상 풀입니다 `ontap-san` 백엔드. 금 이외의 보호 수준을 제공하는 유일한 풀입니다.
- 세 번째 StorageClass입니다 (`app-mysqldb`)는 에서 네 번째 가상 풀에 매핑됩니다 `ontap-nas` 에 있는 백엔드 및 세 번째 가상 풀입니다 `ontap-san-economy` 백엔드. `mysqldb` 유형 앱에 대한 스토리지 풀 구성을 제공하는 유일한 풀입니다.
- 네 번째 StorageClass입니다 (`protection-silver-creditpoints-20k`)는 의 세 번째 가상 풀에 매핑됩니다 `ontap-nas-flexgroup` 의 백엔드 및 두 번째 가상 풀입니다 `ontap-san` 백엔드. 20000 크레딧 포인트에서 골드 레벨 보호 기능을 제공하는 유일한 풀입니다.
- 다섯 번째 StorageClass입니다 (`creditpoints-5k`)는 의 두 번째 가상 풀에 매핑됩니다 `ontap-nas-economy` 에 있는 백엔드 및 세 번째 가상 풀입니다 `ontap-san` 백엔드. 5000 크레딧 포인트에 있는 유일한 풀 서비스입니다.

Astra Trident가 선택한 가상 풀을 결정하고 스토리지 요구 사항을 충족시킵니다.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: netapp.io/trident
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```

ONTAP NAS 백엔드를 구성합니다

ONTAP 및 Cloud Volumes ONTAP NAS 드라이버를 사용하여 ONTAP 백엔드를 구성하는 방법에 대해 알아보십시오.

- "준비"
- "구성 및 예"

Astra Control은 로 생성한 볼륨을 위해 원활한 보호, 재해 복구, 이동성(Kubernetes 클러스터 간에 볼륨 이동)을 제공합니다 `ontap-nas`, `ontap-nas-flexgroup`, 및 `ontap-san` 드라이버. 을 참조하십시오 "Astra Control 복제 사전 요구 사항" 를 참조하십시오.



- 를 사용해야 합니다 `ontap-nas` 데이터 보호, 재해 복구 및 이동성을 필요로 하는 운영 워크로드에 적합합니다.
- 사용 `ontap-san-economy` 예상되는 볼륨 사용량이 ONTAP가 지원하는 것보다 훨씬 많을 것으로 예상됩니다.
- 사용 `ontap-nas-economy` 예상되는 볼륨 사용량이 ONTAP에서 지원하는 것보다 훨씬 많을 것으로 예상되는 경우에만 및 입니다 `ontap-san-economy` 드라이버를 사용할 수 없습니다.
- 사용하지 마십시오 `ontap-nas-economy` 데이터 보호, 재해 복구 또는 이동성이 필요할 것으로 예상되는 경우

사용자 권한

Astra Trident는 일반적으로 를 사용하여 ONTAP 또는 SVM 관리자로 실행될 것으로 예상합니다 `admin` 클러스터 사용자 또는 입니다 `vsadmin` SVM 사용자 또는 동일한 역할을 가진 다른 이름의 사용자 NetApp ONTAP 구축을 위한 Amazon FSx의 경우, Astra Trident는 클러스터를 사용하여 ONTAP 또는 SVM 관리자로 실행될 것으로 예상합니다 `fsxadmin` 사용자 또는 `a vsadmin` SVM 사용자 또는 동일한 역할을 가진 다른 이름의 사용자 를 클릭합니다 `fsxadmin` 사용자는 클러스터 관리자를 제한적으로 대체합니다.



를 사용하는 경우 `limitAggregateUsage` 매개 변수, 클러스터 관리자 권한이 필요합니다. Astra Trident와 함께 NetApp ONTAP에 Amazon FSx를 사용하는 경우, 를 참조하십시오 `limitAggregateUsage` 매개 변수는 에서 작동하지 않습니다 `vsadmin` 및 `fsxadmin` 사용자 계정. 이 매개 변수를 지정하면 구성 작업이 실패합니다.

ONTAP 내에서 Trident 드라이버가 사용할 수 있는 보다 제한적인 역할을 만들 수 있지만 권장하지 않습니다. Trident의 대부분의 새로운 릴리즈에서는 추가 API를 호출하므로 업그레이드가 어렵고 오류가 발생하기 쉽습니다.

ONTAP NAS 드라이버를 사용하여 백엔드를 구성할 준비를 합니다

ONTAP NAS 드라이버를 사용하여 ONTAP 백엔드를 구성하는 방법에 대해 알아보십시오. 모든 ONTAP 백엔드의 경우, Astra Trident는 SVM에 하나 이상의 Aggregate가 할당되어 있어야 합니다.

모든 ONTAP 백엔드의 경우, Astra Trident는 SVM에 하나 이상의 Aggregate가 할당되어 있어야 합니다.

또한 둘 이상의 드라이버를 실행하고 둘 중 하나를 가리키는 스토리지 클래스를 생성할 수도 있습니다. 예를 들어, 을 사용하는 Gold 클래스를 구성할 수 있습니다 `ontap-nas` 드라이버 및 를 사용하는 Bronze 클래스 `ontap-nas-economy` 1개.

모든 Kubernetes 작업자 노드에 적절한 NFS 툴이 설치되어 있어야 합니다. 을 참조하십시오 ["여기"](#) 를 참조하십시오.

인증

Astra Trident는 ONTAP 백엔드를 인증하는 두 가지 모드를 제공합니다.

- 자격 증명 기반: 필요한 권한이 있는 ONTAP 사용자의 사용자 이름 및 암호입니다. 과 같이 미리 정의된 보안 로그인 역할을 사용하는 것이 좋습니다 `admin` 또는 `vsadmin` ONTAP 버전과의 호환성을 최대한 보장하기 위해
- 인증서 기반: Astra Trident는 백엔드에 설치된 인증서를 사용하여 ONTAP 클러스터와 통신할 수도 있습니다. 이 경우 백엔드 정의에는 클라이언트 인증서, 키 및 사용할 경우 신뢰할 수 있는 CA 인증서의 Base64로 인코딩된 값이 있어야 합니다(권장).

자격 증명 기반 방법과 인증서 기반 방법 간에 이동하기 위해 기존 백엔드를 업데이트할 수 있습니다. 그러나 한 번에 하나의 인증 방법만 지원됩니다. 다른 인증 방법으로 전환하려면 백엔드 구성에서 기존 방법을 제거해야 합니다.



자격 증명과 인증서 * 를 모두 제공하려고 하면 구성 파일에 둘 이상의 인증 방법이 제공된다는 오류가 발생하여 백엔드 생성이 실패합니다.

자격 증명 기반 인증을 사용합니다

Astra Trident는 SVM 범위/클러스터 범위 관리자에게 ONTAP 백엔드와 통신하기 위한 자격 증명을 요구합니다. 과 같이 미리 정의된 표준 역할을 사용하는 것이 좋습니다 `admin` 또는 `vsadmin`. 이를 통해 향후 Astra Trident 릴리스에서 사용할 기능 API를 노출할 수 있는 향후 ONTAP 릴리스와 향후 호환성이 보장됩니다. 사용자 지정 보안 로그인 역할은 Astra Trident와 함께 생성 및 사용할 수 있지만 권장되지 않습니다.

백엔드 정의의 예는 다음과 같습니다.

YAML

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

JSON을 참조하십시오

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "username": "vsadmin",
  "password": "password"
}
```

백엔드 정의는 자격 증명에 일반 텍스트로 저장되는 유일한 위치라는 점에 유의하십시오. 백엔드가 생성된 후 사용자 이름/암호는 Base64로 인코딩되어 Kubernetes 암호로 저장됩니다. 백엔드의 생성/업데이트는 자격 증명에 대한 지식이 필요한 유일한 단계입니다. 따라서 Kubernetes/스토리지 관리자가 수행할 수 있는 관리 전용 작업입니다.

인증서 기반 인증을 사용합니다

신규 및 기존 백엔드는 인증서를 사용하여 ONTAP 백엔드와 통신할 수 있습니다. 백엔드 정의에는 세 가지 매개 변수가 필요합니다.

- `clientCertificate`: Base64로 인코딩된 클라이언트 인증서 값입니다.
- `clientPrivateKey`: Base64 - 연결된 개인 키의 인코딩된 값입니다.
- `TrustedCACertificate`: 신뢰할 수 있는 CA 인증서의 Base64 인코딩 값입니다. 신뢰할 수 있는 CA를 사용하는 경우 이 매개 변수를 제공해야 합니다. 신뢰할 수 있는 CA가 사용되지 않으면 이 작업을 무시할 수 있습니다.

일반적인 워크플로에는 다음 단계가 포함됩니다.

단계

1. 클라이언트 인증서 및 키를 생성합니다. 생성 시 CN(일반 이름)을 ONTAP 사용자로 설정하여 인증하십시오.

```
openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout k8senv.key  
-out k8senv.pem -subj "/C=US/ST=NC/L=RTP/O=NetApp/CN=vsadmin"
```

2. 신뢰할 수 있는 CA 인증서를 ONTAP 클러스터에 추가합니다. 이는 스토리지 관리자가 이미 처리한 것일 수 있습니다. 트러스트된 CA가 사용되지 않으면 무시합니다.

```
security certificate install -type server -cert-name <trusted-ca-cert-name> -vserver <vserver-name>  
ssl modify -vserver <vserver-name> -server-enabled true -client-enabled  
true -common-name <common-name> -serial <SN-from-trusted-CA-cert> -ca  
<cert-authority>
```

3. ONTAP 클러스터에 클라이언트 인증서 및 키(1단계)를 설치합니다.

```
security certificate install -type client-ca -cert-name <certificate-name> -vserver <vserver-name>  
security ssl modify -vserver <vserver-name> -client-enabled true
```

4. ONTAP 보안 로그인 역할이 지원되는지 확인합니다 cert 인증 방법.

```
security login create -user-or-group-name vsadmin -application ontapi  
-authentication-method cert -vserver <vserver-name>  
security login create -user-or-group-name vsadmin -application http  
-authentication-method cert -vserver <vserver-name>
```

5. 생성된 인증서를 사용하여 인증을 테스트합니다. ONTAP 관리 LIF> 및 <SVM 이름>을 관리 LIF IP 및 SVM 이름으로 바꿉니다. LIF의 서비스 정책이 으로 설정되어 있는지 확인해야 합니다 default-data-management.

```
curl -X POST -Lk https://<ONTAP-Management-LIF>/servlets/netapp.servlets.admin.XMLrequest_filer --key k8senv.key  
--cert ~/k8senv.pem -d '<?xml version="1.0" encoding="UTF-8"?><netapp  
xmlns="http://www.netapp.com/filer/admin" version="1.21"  
vfiler="<vserver-name>"><vserver-get></vserver-get></netapp>'
```

6. Base64로 인증서, 키 및 신뢰할 수 있는 CA 인증서를 인코딩합니다.

```
base64 -w 0 k8senv.pem >> cert_base64  
base64 -w 0 k8senv.key >> key_base64  
base64 -w 0 trustedca.pem >> trustedca_base64
```

7. 이전 단계에서 얻은 값을 사용하여 백엔드를 생성합니다.

```
cat cert-backend-updated.json
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "NasBackend",
  "managementLIF": "1.2.3.4",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuuuueeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident
+-----+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |                      UUID                      |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |          9 |
+-----+-----+-----+-----+
+-----+-----+
```

인증 방법을 업데이트하거나 자격 증명을 회전합니다

다른 인증 방법을 사용하거나 자격 증명을 회전하도록 기존 백엔드를 업데이트할 수 있습니다. 이렇게 하면 사용자 이름/암호를 사용하는 백엔드를 인증서를 사용하도록 업데이트할 수 있고 인증서를 사용하는 백엔드는 사용자 이름/암호 기반으로 업데이트할 수 있습니다. 이렇게 하려면 기존 인증 방법을 제거하고 새 인증 방법을 추가해야 합니다. 그런 다음 실행할 필수 매개 변수가 포함된 업데이트된 backend.json 파일을 사용합니다 tridentctl update backend.

```
cat cert-backend-updated.json
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "NasBackend",
  "managementLIF": "1.2.3.4",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "username": "vsadmin",
  "password": "password",
  "storagePrefix": "myPrefix_"
}

#Update backend with tridentctl
tridentctl update backend NasBackend -f cert-backend-updated.json -n
trident

+-----+-----+-----+
+-----+-----+
|      NAME      | STORAGE DRIVER |          UUID          |
STATE | VOLUMES |
+-----+-----+-----+
+-----+-----+
| NasBackend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |          9 |
+-----+-----+-----+
+-----+-----+
```



암호를 회전할 때 스토리지 관리자는 먼저 ONTAP에서 사용자의 암호를 업데이트해야 합니다. 그 다음에는 백엔드 업데이트가 있습니다. 인증서를 회전할 때 여러 인증서를 사용자에게 추가할 수 있습니다. 그런 다음 백엔드가 업데이트되어 새 인증서를 사용합니다. 그러면 ONTAP 클러스터에서 이전 인증서를 삭제할 수 있습니다.

백엔드를 업데이트해도 이미 생성된 볼륨에 대한 액세스가 중단되거나 이후에 생성된 볼륨 연결에 영향을 미치지 않습니다. 백엔드 업데이트가 성공적이면 Astra Trident가 ONTAP 백엔드와 통신하고 향후 볼륨 작업을 처리할 수 있음을 나타냅니다.

NFS 익스포트 정책을 관리합니다

Astra Trident는 NFS 익스포트 정책을 사용하여 프로비저닝하는 볼륨에 대한 액세스를 제어합니다.

Astra Trident는 익스포트 정책을 사용할 때 다음 두 가지 옵션을 제공합니다.

- Astra Trident는 익스포트 정책 자체를 동적으로 관리할 수 있습니다. 이 운영 모드에서 스토리지 관리자는 허용할 수 있는 IP 주소를 나타내는 CIDR 블록 목록을 지정합니다. Astra Trident는 이러한 범위에 속하는 노드 IP를 익스포트 정책에 자동으로 추가합니다. 또는 CIDR을 지정하지 않으면 노드에서 발견된 글로벌 범위의 유니캐스트 IP가 내보내기 정책에 추가됩니다.

- 스토리지 관리자는 익스포트 정책을 생성하고 규칙을 수동으로 추가할 수 있습니다. Astra Trident는 구성에 다른 익스포트 정책 이름을 지정하지 않는 한 기본 익스포트 정책을 사용합니다.

익스포트 정책을 동적으로 관리

CSI Trident의 20.04 릴리스는 ONTAP 백엔드에 대한 익스포트 정책을 동적으로 관리할 수 있는 기능을 제공합니다. 따라서 스토리지 관리자는 명시적 규칙을 수동으로 정의하는 대신 작업자 노드 IP에 허용되는 주소 공간을 지정할 수 있습니다. 익스포트 정책 관리를 크게 간소화하므로, 익스포트 정책을 수정하면 더 이상 스토리지 클러스터에 대한 수동 작업이 필요하지 않습니다. 또한 스토리지 클러스터에 대한 액세스를 지정된 범위의 IP가 있는 작업자 노드에만 제한함으로써 세분화된 자동 관리를 지원합니다.



CSI Trident에만 내보내기 정책의 동적 관리를 사용할 수 있습니다. 작업자 노드가 NATED가 아닌지 확인하는 것이 중요합니다.

예

두 가지 구성 옵션을 사용해야 합니다. 다음은 백엔드 정의의 예입니다.

```
---
version: 1
storageDriverName: ontap-nas
backendName: ontap_nas_auto_export
managementLIF: 192.168.0.135
svm: svm1
username: vsadmin
password: password
autoExportCIDRs:
- 192.168.0.0/24
autoExportPolicy: true
```



이 기능을 사용할 때는 SVM의 루트 교차점에 노드 CIDR 블록(예: 기본 익스포트 정책)을 허용하는 익스포트 규칙과 함께 이전에 생성된 익스포트 정책이 있는지 확인해야 합니다. Astra Trident를 위한 SVM 전용 NetApp의 권장 모범 사례를 항상 따르십시오.

다음은 위의 예를 사용하여 이 기능이 작동하는 방식에 대한 설명입니다.

- `autoExportPolicy` 가 로 설정되어 있습니다 `true`. 이는 Astra Trident가 에 대한 익스포트 정책을 생성한다는 것을 나타냅니다 `svm1` 를 사용하여 규칙 추가 및 삭제를 처리합니다 `autoExportCIDRs` 주소 블록. 예를 들어 UUID 403b5326-8482-40dB-96d0-d83fb3f4daec 및 가 있는 백엔드를 사용할 수 있습니다 `autoExportPolicy` 를 로 설정합니다 `true` 이라는 익스포트 정책을 생성합니다 `trident-403b5326-8482-40db-96d0-d83fb3f4daec` SVM에서.
- `autoExportCIDRs` 주소 블록 목록이 포함되어 있습니다. 이 필드는 선택 사항이며 기본적으로 `["0.0.0.0/0", ":/0"]`입니다. 정의되지 않은 경우 Astra Trident는 작업자 노드에 있는 모든 전역 범위의 유니캐스트 주소를 추가합니다.

이 예에서 는 입니다 192.168.0.0/24 주소 공간이 제공됩니다. 이 주소 범위에 속하는 Kubernetes 노드 IP가 Astra Trident가 생성하는 익스포트 정책에 추가됨을 나타냅니다. Astra Trident가 실행되는 노드를 등록하면 노드의 IP 주소를 검색하여 에 제공된 주소 블록과 대조하여 확인합니다 `autoExportCIDRs`. IP를 필터링한 후 Astra Trident는

검색된 클라이언트 IP에 대한 익스포트 정책 규칙을 생성하며, 식별하는 각 노드에 대해 하나의 규칙을 사용합니다.

업데이트할 수 있습니다 `autoExportPolicy` 및 `autoExportCIDRs` 백엔드는 만든 후에 사용합니다. 기존 CIDR을 자동으로 관리하거나 삭제하는 백엔드에 새 CIDR을 추가할 수 있습니다. CIDR을 삭제할 때는 기존 연결이 끊어지지 않도록 주의해야 합니다. 를 비활성화하도록 선택할 수도 있습니다 `autoExportPolicy` 백엔드의 경우 수동으로 생성된 내보내기 정책으로 돌아갑니다. 이렇게 하려면 을 설정해야 합니다 `exportPolicy` 백엔드 구성의 매개 변수입니다.

Astra Trident가 백엔드를 생성하거나 업데이트한 후 을 사용하여 백엔드를 확인할 수 있습니다 `tridentctl` 또는 해당 `tridentbackend` CRD:

```
./tridentctl get backends ontap_nas_auto_export -n trident -o yaml
items:
- backendUUID: 403b5326-8482-40db-96d0-d83fb3f4daec
  config:
    aggregate: ""
    autoExportCIDRs:
    - 192.168.0.0/24
    autoExportPolicy: true
    backendName: ontap_nas_auto_export
    chapInitiatorSecret: ""
    chapTargetInitiatorSecret: ""
    chapTargetUsername: ""
    chapUsername: ""
    dataLIF: 192.168.0.135
    debug: false
    debugTraceFlags: null
    defaults:
      encryption: "false"
      exportPolicy: <automatic>
      fileType: ext4
```

노드가 Kubernetes 클러스터에 추가되고 Astra Trident 컨트롤러에 등록되면 기존 백엔드의 내보내기 정책이 업데이트됩니다(에 지정된 주소 범위에 속하면 됨) `autoExportCIDRs` 백 엔드용).

노드가 제거되면 Astra Trident는 온라인 상태인 모든 백엔드를 검사하여 노드에 대한 액세스 규칙을 제거합니다. Astra Trident는 관리되는 백엔드의 내보내기 정책에서 이 노드 IP를 제거하여 불량 마운트를 방지합니다. 단, 클러스터의 새 노드에서 이 IP를 다시 사용하지 않는 한 마찬가지입니다.

기존 백엔드의 경우 백엔드를 로 업데이트합니다 `tridentctl update backend` Astra Trident가 익스포트 정책을 자동으로 관리하는지 확인합니다. 그러면 백엔드의 UUID 뒤에 이름이 지정된 새 내보내기 정책이 생성되고 백엔드에 있는 볼륨은 새로 생성된 내보내기 정책을 다시 마운트할 때 사용합니다.



자동 관리되는 내보내기 정책이 있는 백엔드를 삭제하면 동적으로 생성된 내보내기 정책이 삭제됩니다. 백엔드가 다시 생성되면 백엔드가 새 백엔드로 처리되어 새 익스포트 정책이 생성됩니다.

라이브 노드의 IP 주소가 업데이트되면 노드에서 Astra Trident POD를 다시 시작해야 합니다. 그런 다음 Astra Trident가 이 IP 변경 사항을 반영하도록 관리하는 백엔드에 대한 익스포트 정책을 업데이트합니다.

ONTAP NAS 구성 옵션 및 예

Astra Trident 설치를 통해 ONTAP NAS 드라이버를 생성하고 사용하는 방법에 대해 알아보십시오. 이 섹션에서는 백엔드 구성 예제 및 백엔드를 StorageClasses에 매핑하는 방법에 대한 세부 정보를 제공합니다.

백엔드 구성 옵션

백엔드 구성 옵션은 다음 표를 참조하십시오.

매개 변수	설명	기본값
version		항상 1
storageDriverName	스토리지 드라이버의 이름입니다	"ONTAP-NAS", "ONTAP-NAS-이코노미", "ONTAP-NAS-Flexgroup", "ONTAP-SAN", "ONTAP-SAN-이코노미"
backendName	사용자 지정 이름 또는 스토리지 백엔드	드라이버 이름 + "_" + dataLIF
managementLIF	클러스터 또는 SVM 관리 LIF의 IP 주소 MetroCluster의 원활한 전환을 위해 SVM 관리 LIF를 지정해야 합니다. FQDN(정규화된 도메인 이름)을 지정할 수 있습니다. 을 사용하여 Astra Trident가 설치된 경우 IPv6 주소를 사용하도록 설정할 수 있습니다 --use-ipv6 깃발. IPv6 주소는 대괄호로 묶어야 합니다(예: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]).	"10.0.0.1", "[2001:1234:ABCD::fee]"
dataLIF	프로토콜 LIF의 IP 주소입니다. 지정할 것을 권장합니다 dataLIF. 제공되지 않는 경우 Astra Trident는 SVM에서 데이터 LIF를 가져옵니다. NFS 마운트 작업에 사용할 FQDN(정규화된 도메인 이름)을 지정하면 여러 데이터 LIF에서 로드 밸런싱을 위해 라운드 로빈 DNS를 생성할 수 있습니다. 초기 설정 후에 변경할 수 있습니다. 을 참조하십시오 . 을 사용하여 Astra Trident가 설치된 경우 IPv6 주소를 사용하도록 설정할 수 있습니다 --use-ipv6 깃발. IPv6 주소는 대괄호로 묶어야 합니다(예: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]).	지정되지 않은 경우 SVM에서 지정 주소 또는 파생(권장하지 않음)
autoExportPolicy	자동 익스포트 정책 생성 및 업데이트 [Boolean] 활성화 를 사용합니다 autoExportPolicy 및 autoExportCIDRs 옵션, Astra Trident는 익스포트 정책을 자동으로 관리할 수 있습니다.	거짓

매개 변수	설명	기본값
autoExportCIDRs	Kubernetes 노드 IP를 언제 기준으로 필터링하는 CIDR 목록입니다 autoExportPolicy 가 활성화됩니다. 를 사용합니다 autoExportPolicy 및 autoExportCIDRs 옵션, Astra Trident는 익스포트 정책을 자동으로 관리할 수 있습니다.	["0.0.0.0/0", ":::/0"]
labels	볼륨에 적용할 임의의 JSON 형식 레이블 세트입니다	""
clientCertificate	Base64 - 클라이언트 인증서의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다	""
clientPrivateKey	Base64 - 클라이언트 개인 키의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다	""
trustedCACertificate	Base64 - 신뢰할 수 있는 CA 인증서의 인코딩된 값입니다. 선택 사항. 인증서 기반 인증에 사용됩니다	""
username	클러스터/SVM에 연결할 사용자 이름입니다. 자격 증명 기반 인증에 사용됩니다	
password	클러스터/SVM에 연결하는 암호 자격 증명 기반 인증에 사용됩니다	
svm	사용할 스토리지 가상 머신입니다	SVM에서 파생된 경우 managementLIF 이(가) 지정되었습니다
storagePrefix	SVM에서 새 볼륨을 프로비저닝할 때 사용되는 접두사 설정한 후에는 업데이트할 수 없습니다	"삼중류"
limitAggregateUsage	사용량이 이 비율을 초과하면 프로비저닝이 실패합니다. ONTAP * 용 아마존 FSx에는 * 가 적용되지 않습니다	""(기본적으로 적용되지 않음)
limitVolumeSize	요청된 볼륨 크기가 이 값보다 큰 경우 용량 할당에 실패합니다.	""(기본적으로 적용되지 않음)
limitVolumeSize	요청된 볼륨 크기가 이 값보다 큰 경우 용량 할당에 실패합니다. 또한 qtree 및 LUN에 대해 관리하는 볼륨의 최대 크기 및 을 제한합니다 qtreesPerFlexvol 옵션을 사용하면 FlexVol당 최대 qtree 수를 사용자 지정할 수 있습니다.	""(기본적으로 적용되지 않음)
lunsPerFlexvol	FlexVol당 최대 LUN 수는 범위[50, 200]에 있어야 합니다.	"100"

매개 변수	설명	기본값
debugTraceFlags	문제 해결 시 사용할 디버그 플래그입니다. 예: {"api":false, "method":true} 사용하지 마십시오 debugTraceFlags 문제 해결 및 자세한 로그 덤프가 필요한 경우를 제외하고	null입니다
nfsMountOptions	심표로 구분된 NFS 마운트 옵션 목록입니다. Kubernetes 영구 볼륨의 마운트 옵션은 일반적으로 스토리지 클래스에서 지정되지만, 스토리지 클래스에 마운트 옵션을 지정하지 않으면 Astra Trident가 스토리지 백엔드의 구성 파일에 지정된 마운트 옵션을 사용하여 로 돌아갑니다. 스토리지 클래스 또는 구성 파일에 마운트 옵션이 지정되지 않은 경우 Astra Trident는 연결된 영구 볼륨에 마운트 옵션을 설정하지 않습니다.	""
qtreesPerFlexvol	FlexVol당 최대 qtree, 범위 [50, 300]에 있어야 함	"200"
useREST	ONTAP REST API를 사용하는 부울 매개 변수입니다. * 기술 미리 보기 * useREST 프로덕션 작업 부하가 아닌 테스트 환경에 권장되는 기술 미리 보기로 제공됩니다. 를 로 설정한 경우 true, Astra Trident는 ONTAP REST API를 사용하여 백엔드와 통신합니다. 이 기능을 사용하려면 ONTAP 9.11.1 이상이 필요합니다. 또한 사용되는 ONTAP 로그인 역할에 대한 액세스 권한이 있어야 합니다 ontap 응용 프로그램. 이는 사전 정의된 에 의해 충족됩니다 vsadmin 및 cluster-admin 역할. useREST 는 MetroCluster에서 지원되지 않습니다.	거짓

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
spaceAllocation	LUN에 대한 공간 할당	"참"
spaceReserve	공간 예약 모드, "없음"(썬) 또는 "볼륨"(일반)	"없음"
snapshotPolicy	사용할 스냅샷 정책입니다	"없음"

매개 변수	설명	기본값
qosPolicy	생성된 볼륨에 할당할 QoS 정책 그룹입니다. 스토리지 풀/백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다	“”
adaptiveQosPolicy	생성된 볼륨에 할당할 적응형 QoS 정책 그룹입니다. 스토리지 풀/백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다. ONTAP에서 지원되지 않음 - NAS - 이코노미	“”
snapshotReserve	스냅샷 "0"에 예약된 볼륨의 백분율	If(경우 snapshotPolicy "없음"이고, 그렇지 않으면 ""입니다.
splitOnClone	생성 시 상위 클론에서 클론을 분할합니다	"거짓"
encryption	새 볼륨에 NVE(NetApp Volume Encryption)를 활성화합니다. 기본값은 입니다 false. 이 옵션을 사용하려면 NVE 라이선스가 클러스터에서 활성화되어 있어야 합니다. 백엔드에서 NAE가 활성화된 경우 Astra Trident에 프로비저닝된 모든 볼륨은 NAE가 활성화됩니다. 자세한 내용은 다음을 참조하십시오. "Astra Trident가 NVE 및 NAE와 연동되는 방식" .	"거짓"
tieringPolicy	"없음"을 사용하는 계층화 정책	ONTAP 9.5 이전 SVM-DR 구성의 경우 "스냅샷 전용
unixPermissions	모드를 선택합니다	NFS 볼륨의 경우 "777", SMB 볼륨의 경우 비어 있음(해당 없음
snapshotDir	의 표시 여부를 제어합니다 .snapshot 디렉토리	"거짓"
exportPolicy	사용할 익스포트 정책	“기본값”
securityStyle	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.



Astra Trident와 함께 QoS 정책 그룹을 사용하려면 ONTAP 9.8 이상이 필요합니다. 비공유 QoS 정책 그룹을 사용하고 정책 그룹이 각 구성요소별로 적용되었는지 확인하는 것이 좋습니다. 공유 QoS 정책 그룹은 모든 워크로드의 총 처리량에 대해 상한을 적용합니다.

볼륨 프로비저닝의 예

다음은 기본값이 정의된 예입니다.

```

---
version: 1
storageDriverName: ontap-nas
backendName: customBackendName
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
labels:
  k8scluster: dev1
  backend: dev1-nasbackend
svm: trident_svm
username: cluster-admin
password: password
limitAggregateUsage: 80%
limitVolumeSize: 50Gi
nfsMountOptions: nfsvers=4
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: premium
  exportPolicy: myk8scluster
  snapshotPolicy: default
  snapshotReserve: '10'

```

용 ontap-nas 및 ontap-nas-flexgroups`이제 Astra Trident가 새로운 계산을 사용하여 snapshotReserve Percentage 및 PVC로 FlexVol의 크기를 올바르게 지정합니다. 사용자가 PVC를 요청하면 Astra Trident는 새 계산을 사용하여 더 많은 공간을 가진 원본 FlexVol를 생성합니다. 이 계산을 통해 사용자는 PVC에서 요청한 쓰기 가능 공간을 확보할 수 있으며 요청된 공간보다 적은 공간을 확보할 수 있습니다. v21.07 이전에는 사용자가 스냅샷 보존 공간을 50%로 하여 PVC(예: 5GiB)를 요청할 때 쓰기 가능한 공간은 2.5GiB에 불과합니다. 사용자가 요청한 것은 전체 볼륨과 이기 때문입니다 `snapshotReserve`이 백분율에 포함됩니다. Trident 21.07을 사용하면 사용자가 요청하는 것이 쓰기 가능한 공간이고 Astra Trident가 이를 정의합니다 snapshotReserve 전체 볼륨의 백분율로 표시됩니다. 예는 적용되지 않습니다 ontap-nas-economy. 이 작동 방식을 보려면 다음 예를 참조하십시오.

계산은 다음과 같습니다.

```

Total volume size = (PVC requested size) / (1 - (snapshotReserve
percentage) / 100)

```

snapshotReserve = 50%, PVC request = 5GiB의 경우, 총 볼륨 크기는 $2/5 = 10\text{GiB}$ 이고 사용 가능한 크기는 5GiB입니다. 이는 사용자가 PVC 요청에서 요청한 것입니다. 를 클릭합니다 volume show 명령은 이 예제와 유사한 결과를 표시해야 합니다.

Vserver	Volume	Aggregate	State	Type	Size	Available	Used%
		_pvc_89f1c156_3801_4de4_9f9d_034d54c395f4	online	RW	10GB	5.00GB	0%
		_pvc_e8372153_9ad9_474a_951a_08ae15e1c0ba	online	RW	1GB	511.8MB	0%

2 entries were displayed.

이전 설치에서 기존 백엔드는 Astra Trident를 업그레이드할 때 위에서 설명한 대로 볼륨을 프로비저닝합니다. 업그레이드하기 전에 생성한 볼륨의 경우 변경 사항을 관찰하기 위해 볼륨의 크기를 조정해야 합니다. 예를 들어, 2GiB PVC가 인 경우 `snapshotReserve=50` 그 결과, 쓰기 가능한 공간 1GiB를 제공하는 볼륨이 탄생했습니다. 예를 들어, 볼륨을 3GiB로 조정하면 애플리케이션에 6GiB 볼륨의 쓰기 가능 공간이 3GiB로 표시됩니다.

예

최소 구성의 예

다음 예에서는 대부분의 매개 변수를 기본값으로 두는 기본 구성을 보여 줍니다. 이는 백엔드를 정의하는 가장 쉬운 방법입니다.



Trident가 있는 NetApp ONTAP에서 Amazon FSx를 사용하는 경우 IP 주소 대신 LIF에 대한 DNS 이름을 지정하는 것이 좋습니다.

의 기본 옵션 `ontap-nas-economy`

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

인증서 기반 인증입니다

이는 최소 백엔드 구성의 예입니다. `clientCertificate`, `clientPrivateKey`, 및 `trustedCACertificate` (신뢰할 수 있는 CA를 사용하는 경우 선택 사항)는 에 채워집니다 `backend.json` 그리고 각각 클라이언트 인증서, 개인 키 및 신뢰할 수 있는 CA 인증서의 base64로 인코딩된 값을 사용합니다.

```
---
version: 1
backendName: DefaultNASBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.15
svm: nfs_svm
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```

다음 예에서는 Astra Trident가 동적 익스포트 정책을 사용하여 익스포트 정책을 자동으로 생성하고 관리하도록 지시하는 방법을 보여 줍니다. 이 기능은 예 대해서도 동일하게 작동합니다. `ontap-nas-economy` 및 `ontap-nas-flexgroup` 드라이버.

ONTAP - NAS 드라이버

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-nasbackend
autoExportPolicy: true
autoExportCIDRs:
- 10.0.0.0/24
username: admin
password: password
nfsMountOptions: nfsvers=4
```

`ontap-nas-flexgroup` 드라이버

```
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
labels:
  k8scluster: test-cluster-east-1b
  backend: test1-ontap-cluster
svm: svm_nfs
username: vsadmin
password: password
```

IPv6 주소 사용

이 예에서는 를 보여 줍니다 managementLIF IPv6 주소 사용.

```
---
version: 1
storageDriverName: ontap-nas
backendName: nas_ipv6_backend
managementLIF: "[5c5d:5edf:8f:7657:bef8:109b:1b41:d491]"
labels:
  k8scluster: test-cluster-east-1a
  backend: test1-ontap-ipv6
svm: nas_ipv6_svm
username: vsadmin
password: password
```

ontap-nas-economy 드라이버

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```

ontap-nas **SMB** 볼륨을 사용하는 **ONTAP용 Amazon FSx** 드라이버입니다

```
---
version: 1
backendName: SMBBackend
storageDriverName: ontap-nas
managementLIF: example.mgmt.fqdn.aws.com
nasType: smb
dataLIF: 10.0.0.15
svm: nfs_svm
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
storagePrefix: myPrefix_
```


가상 풀의 백엔드 예

아래 표시된 샘플 백엔드 정의 파일에서와 같은 모든 스토리지 풀에 대한 특정 기본값이 설정됩니다 `spaceReserve` 없음, `spaceAllocation` 거짓일 경우, 및 `encryption` 거짓일 때. 가상 풀은 스토리지 섹션에 정의됩니다.

Astra Trident는 "Comments" 필드에 제공 레이블을 설정합니다. 설명은 FlexVol for에서 설정됩니다 `ontap-nas` 또는 `FlexGroup for ontap-nas-flexgroup`. Astra Trident는 프로비저닝할 때 가상 풀에 있는 모든 레이블을 스토리지 볼륨에 복사합니다. 편의를 위해 스토리지 관리자는 가상 풀 및 그룹 볼륨별로 레이블을 레이블별로 정의할 수 있습니다.

이 예에서는 일부 스토리지 풀이 자체적으로 설정됩니다 `spaceReserve`, `spaceAllocation`, 및 `encryption` 값 및 일부 풀이 위에 설정된 기본값을 덮어씁니다.

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: admin
password: password
nfsMountOptions: nfsvers=4
defaults:
  spaceReserve: none
  encryption: 'false'
  qosPolicy: standard
labels:
  store: nas_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
- labels:
  app: msoffice
  cost: '100'
  zone: us_east_1a
  defaults:
    spaceReserve: volume
    encryption: 'true'
    unixPermissions: '0755'
    adaptiveQosPolicy: adaptive-premium
- labels:
  app: slack
  cost: '75'
  zone: us_east_1b
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  app: wordpress
  cost: '50'
  zone: us_east_1c
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0775'
```

```
- labels:  
  app: mysqlldb  
  cost: '25'  
  zone: us_east_1d  
  defaults:  
    spaceReserve: volume  
    encryption: 'false'  
    unixPermissions: '0775'
```

```
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
defaults:
  spaceReserve: none
  encryption: 'false'
labels:
  store: flexgroup_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
- labels:
  protection: gold
  creditpoints: '50000'
  zone: us_east_1a
  defaults:
    spaceReserve: volume
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  protection: gold
  creditpoints: '30000'
  zone: us_east_1b
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  protection: silver
  creditpoints: '20000'
  zone: us_east_1c
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0775'
- labels:
  protection: bronze
  creditpoints: '10000'
```

```
zone: us_east_1d
defaults:
  spaceReserve: volume
  encryption: 'false'
  unixPermissions: '0775'
```

```
---
version: 1
storageDriverName: ontap-nas-economy
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
defaults:
  spaceReserve: none
  encryption: 'false'
labels:
  store: nas_economy_store
region: us_east_1
storage:
- labels:
  department: finance
  creditpoints: '6000'
  zone: us_east_1a
  defaults:
    spaceReserve: volume
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  department: legal
  creditpoints: '5000'
  zone: us_east_1b
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0755'
- labels:
  department: engineering
  creditpoints: '3000'
  zone: us_east_1c
  defaults:
    spaceReserve: none
    encryption: 'true'
    unixPermissions: '0775'
- labels:
  department: humanresource
  creditpoints: '2000'
  zone: us_east_1d
```

```
defaults:
  spaceReserve: volume
  encryption: 'false'
  unixPermissions: '0775'
```

업데이트 dataLIF 초기 구성 후

다음 명령을 실행하여 초기 구성 후에 데이터 LIF를 변경할 수 있으며, 업데이트된 데이터 LIF가 포함된 새 백엔드 JSON 파일을 제공할 수 있습니다.

```
tridentctl update backend <backend-name> -f <path-to-backend-json-file-
with-updated-dataLIF>
```



PVC가 하나 이상의 포드에 연결된 경우 해당 포드를 모두 내린 다음 다시 불러와서 새 데이터 LIF가 적용되도록 해야 합니다.

백엔드를 **StorageClasses**에 매핑합니다

다음 StorageClass 정의는 위의 가상 풀을 참조합니다. 를 사용합니다 `parameters.selector` 필드에서 각 StorageClass는 볼륨을 호스팅하는 데 사용할 수 있는 가상 풀을 호출합니다. 선택한 가상 풀에 볼륨이 정의되어 있습니다.

- 첫 번째 StorageClass입니다 (protection-gold)는 의 첫 번째, 두 번째 가상 풀에 매핑됩니다 `ontap-nas-flexgroup` 백엔드 및 의 첫 번째 가상 풀 `ontap-san` 백엔드. 골드 레벨 보호 기능을 제공하는 유일한 풀입니다.
- 두 번째 StorageClass입니다 (protection-not-gold)는 에서 세 번째, 네 번째 가상 풀에 매핑됩니다 `ontap-nas-flexgroup` 백엔드 및 에서 세 번째 가상 풀입니다 `ontap-san` 백엔드. 금 이외의 보호 수준을 제공하는 유일한 풀입니다.
- 세 번째 StorageClass입니다 (app-mysqldb)는 에서 네 번째 가상 풀에 매핑됩니다 `ontap-nas` 에 있는 백엔드 및 세 번째 가상 풀입니다 `ontap-san-economy` 백엔드. `mysqldb` 유형 앱에 대한 스토리지 풀 구성을 제공하는 유일한 풀입니다.
- 네 번째 StorageClass입니다 (protection-silver-creditpoints-20k)는 의 세 번째 가상 풀에 매핑됩니다 `ontap-nas-flexgroup` 의 백엔드 및 두 번째 가상 풀입니다 `ontap-san` 백엔드. 20000 크레딧 포인트에서 골드 레벨 보호 기능을 제공하는 유일한 풀입니다.
- 다섯 번째 StorageClass입니다 (creditpoints-5k)는 의 두 번째 가상 풀에 매핑됩니다 `ontap-nas-economy` 에 있는 백엔드 및 세 번째 가상 풀입니다 `ontap-san` 백엔드. 5000 크레딧 포인트에 있는 유일한 풀 서비스입니다.

Astra Trident가 선택한 가상 풀을 결정하고 스토리지 요구 사항을 충족시킵니다.

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: netapp.io/trident
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```


NetApp ONTAP용 Amazon FSx

NetApp ONTAP용 Amazon FSx와 함께 Astra Trident를 사용하십시오

"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.

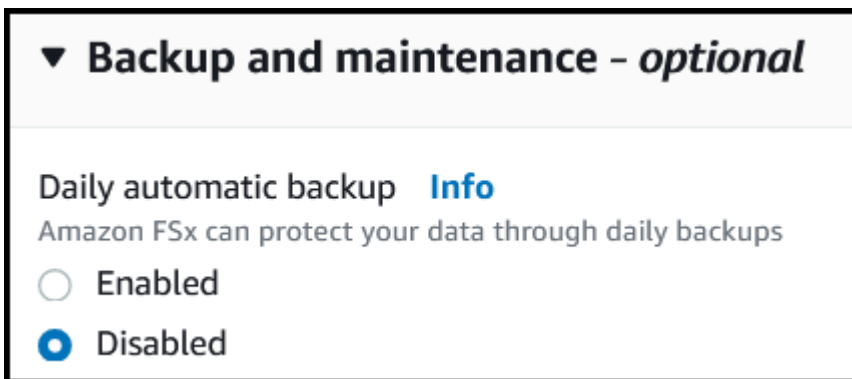
파일 시스템은 Amazon FSx의 주요 리소스이며, 이는 사내 ONTAP 클러스터와 유사합니다. 각 SVM 내에서 파일 시스템에 파일과 폴더를 저장하는 데이터 컨테이너인 하나 이상의 볼륨을 생성할 수 있습니다. NetApp ONTAP용 Amazon FSx를 사용하면 클라우드에서 Data ONTAP가 관리형 파일 시스템으로 제공됩니다. 새로운 파일 시스템 유형을 * NetApp ONTAP * 라고 합니다.

NetApp ONTAP용 Amazon FSx와 Astra Trident를 사용하면 Amazon EKS(Elastic Kubernetes Service)에서 실행되는 Kubernetes 클러스터가 ONTAP에서 지원하는 블록 및 파일 영구 볼륨을 프로비저닝할 수 있습니다.

Amazon FSx for NetApp ONTAP은 "FabricPool" 스토리지 계층을 관리하는 데 사용합니다. 이 기능을 사용하면 데이터의 액세스 빈도 여부에 따라 데이터를 계층에 저장할 수 있습니다.

고려 사항

- SMB 볼륨:
 - SMB 볼륨은 를 사용하여 지원됩니다 `ontap-nas` 드라이버만 해당.
 - Astra Trident는 Windows 노드에서 실행되는 Pod에만 마운트된 SMB 볼륨을 지원합니다.
 - Astra Trident는 Windows ARM 아키텍처를 지원하지 않습니다.
- 자동 백업이 설정된 Amazon FSx 파일 시스템에서 생성된 볼륨은 Trident에서 삭제할 수 없습니다. PVC를 삭제하려면 ONTAP 체적에 대한 PV 및 FSx를 수동으로 삭제해야 합니다. 이 문제를 방지하려면:
 - **Quick create**를 사용하여 ONTAP 파일 시스템용 FSx를 생성하지 마십시오. 빠른 생성 워크플로에서는 자동 백업을 사용할 수 있으며 수신 거부 옵션은 제공하지 않습니다.
 - 표준 생성 을 사용하는 경우 자동 백업을 비활성화합니다. 자동 백업을 비활성화하면 Trident는 추가적인 수동 개입 없이 볼륨을 성공적으로 삭제할 수 있습니다.



드라이버

다음 드라이버를 사용하여 Astra Trident를 NetApp ONTAP용 Amazon FSx와 통합할 수 있습니다.

- `ontap-san`: 각 PV 프로비저닝은 고유한 Amazon FSx for NetApp ONTAP 볼륨 내에 있는 LUN입니다.
- `ontap-san-economy`: 각 PV 프로비저닝은 NetApp ONTAP 볼륨에 대해 Amazon FSx당 구성 가능한 LUN 수를 가진 LUN입니다.
- `ontap-nas`: 각 PV 프로비저닝은 NetApp ONTAP 볼륨에 대한 전체 Amazon FSx입니다.
- `ontap-nas-economy`: 각 PV 프로비저닝은 qtree이며, NetApp ONTAP 볼륨용 Amazon FSx당 구성 가능한 Qtree 수가 있습니다.
- `ontap-nas-flexgroup`: 각 PV 프로비저닝은 NetApp ONTAP FlexGroup 볼륨에 대한 전체 Amazon FSx입니다.

드라이버 세부 정보는 를 참조하십시오 ["ONTAP 드라이버"](#).

인증

Astra Trident는 두 가지 인증 모드를 제공합니다.

- 인증서 기반: Astra Trident는 SVM에 설치된 인증서를 사용하여 FSx 파일 시스템의 SVM과 통신합니다.
- 자격 증명 기반: 을 사용할 수 있습니다 `fsxadmin` 파일 시스템 또는 의 사용자입니다 `vsadmin` SVM을 위해 사용자가 구성됨



Astra Trident는 한 대로 실행될 것으로 예상합니다 `vsadmin` SVM 사용자 또는 동일한 역할을 가진 다른 이름의 사용자 NetApp ONTAP용 Amazon FSx에는 가 있습니다 `fsxadmin` ONTAP의 제한된 교체인 사용자입니다 `admin` 클러스터 사용자. 을 사용하는 것이 좋습니다 `vsadmin` Astra Trident와 함께.

자격 증명 기반 방법과 인증서 기반 방법 간에 이동하도록 백엔드를 업데이트할 수 있습니다. 그러나 * 자격 증명 및 인증서 * 를 제공하려고 하면 백엔드 생성이 실패합니다. 다른 인증 방법으로 전환하려면 백엔드 구성에서 기존 방법을 제거해야 합니다.

인증 활성화에 대한 자세한 내용은 드라이버 유형에 대한 인증을 참조하십시오.

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

자세한 내용을 확인하십시오

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

NetApp ONTAP용 Amazon FSx를 통합합니다

Amazon Elastic Kubernetes Service(EKS)에서 실행되는 Kubernetes 클러스터가 ONTAP에서 지원하는 블록 및 파일 영구 볼륨을 프로비저닝할 수 있도록 NetApp ONTAP 파일 시스템용 Amazon FSx를 Astra Trident와 통합할 수 있습니다.

시작하기 전에

또한 "[Astra Trident 요구사항](#)", ONTAP용 FSx와 Astra Trident를 통합하려면 다음이 필요합니다.

- 기존 Amazon EKS 클러스터 또는 자체 관리형 Kubernetes 클러스터 `kubectl` 설치되어 있습니다.
- 클러스터의 작업자 노드에서 연결할 수 있는 NetApp ONTAP 파일 시스템용 기존 Amazon FSx 및 SVM(스토리지 가상 시스템)입니다.
- 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.

SMB 볼륨에 대한 추가 요구사항

- Linux 컨트롤러 노드 및 Windows Server 2019를 실행하는 Windows 작업자 노드가 있는 Kubernetes 클러스터 Astra Trident는 Windows 노드에서 실행되는 Pod에만 마운트된 SMB 볼륨을 지원합니다.
- Active Directory 자격 증명이 포함된 Astra Trident 암호가 하나 이상 있어야 합니다. 암호를 생성합니다 `smbcreds`:

```
kubectl create secret generic smbcreds --from-literal username=user  
--from-literal password='password'
```

- Windows 서비스로 구성된 CSI 프록시. 를 구성합니다 `csi-proxy`를 참조하십시오 "[GitHub:CSI 프록시](#)" 또는 "[GitHub: Windows용 CSI 프록시](#)" Windows에서 실행되는 Kubernetes 노드의 경우:

ONTAP SAN 및 NAS 드라이버 통합



SMB 볼륨에 대해 를 구성하는 경우 을 읽어야 합니다 [SMB 볼륨 프로비저닝을 위한 준비](#) 백엔드를 생성하기 전에

단계

1. 중 하나를 사용하여 Astra Trident를 배포합니다 "[배포 방법](#)".
2. SVM 관리 LIF DNS 이름을 수집합니다. 예를 들어, AWS CLI를 사용하여 를 찾습니다 `DNSName` 에 입력 `Endpoints` → `Management` 다음 명령을 실행한 후:

```
aws fsx describe-storage-virtual-machines --region <file system region>
```

3. 에 대한 인증서를 만들고 설치합니다 "[NAS 백엔드 인증](#)" 또는 "[SAN 백엔드 인증](#)".



파일 시스템에 연결할 수 있는 모든 위치에서 SSH를 사용하여 파일 시스템(예: 인증서 설치)에 로그인할 수 있습니다. 를 사용합니다 `fsxadmin` 사용자, 파일 시스템을 생성할 때 구성한 암호 및 에서 관리 DNS 이름 `aws fsx describe-file-systems`.

4. 아래 예에 표시된 대로 인증서와 관리 LIF의 DNS 이름을 사용하여 백엔드 파일을 생성합니다.

YAML

```
---
version: 1
storageDriverName: ontap-san
backendName: customBackendName
managementLIF: svm-XXXXXXXXXXXXXXXXXX.fs-XXXXXXXXXXXXXXXXXX.fsx.us-
east-2.aws.internal
svm: svm01
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

JSON을 참조하십시오

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "customBackendName",
  "managementLIF": "svm-XXXXXXXXXXXXXXXXXX.fs-
XXXXXXXXXXXXXXXXXX.fsx.us-east-2.aws.internal",
  "svm": "svm01",
  "clientCertificate": "ZXR0ZXJwYXB...ICMgJ3BhcGVyc2",
  "clientPrivateKey": "vciwKIyAgZG...0cnksIGRlc2NyaX",
  "trustedCACertificate": "zcyBbaG...b3Igb3duIGNsYXNz"
}
```

백엔드 만들기에 대한 자세한 내용은 다음 링크를 참조하십시오.

- ["ONTAP NAS 드라이버를 사용하여 백엔드를 구성합니다"](#)
- ["ONTAP SAN 드라이버를 사용하여 백엔드를 구성합니다"](#)

결과

배포 후에는 을 만들 수 있습니다 ["스토리지 클래스, 볼륨 프로비저닝 및 POD에 볼륨 마운트"](#).

SMB 볼륨 프로비저닝을 위한 준비

를 사용하여 SMB 볼륨을 프로비저닝할 수 있습니다 ontap-nas 드라이버. 를 누릅니다 [ONTAP SAN 및 NAS 드라이버 통합](#) 다음 단계를 완료합니다.

단계

1. SMB 공유를 생성합니다. 다음 두 가지 방법 중 하나로 SMB 관리자 공유를 생성할 수 있습니다 ["Microsoft 관리 콘솔"](#) 공유 폴더 스냅인 또는 ONTAP CLI 사용 ONTAP CLI를 사용하여 SMB 공유를 생성하려면 다음을 따르십시오.

a. 필요한 경우 공유에 대한 디렉토리 경로 구조를 생성합니다.

를 클릭합니다 `vserver cifs share create` 명령은 공유를 생성하는 동안 `-path` 옵션에 지정된 경로를 확인합니다. 지정한 경로가 없으면 명령이 실패합니다.

b. 지정된 SVM과 연결된 SMB 공유를 생성합니다.

```
vserver cifs share create -vserver vserver_name -share-name
share_name -path path [-share-properties share_properties,...]
[other_attributes] [-comment text]
```

c. 공유가 생성되었는지 확인합니다.

```
vserver cifs share show -share-name share_name
```



을 참조하십시오 ["SMB 공유를 생성합니다"](#) 를 참조하십시오.

2. 백엔드를 생성할 때 SMB 볼륨을 지정하려면 다음을 구성해야 합니다. 모든 ONTAP 백엔드 구성 옵션에 대한 자세한 내용은 을 참조하십시오 ["ONTAP 구성 옵션 및 예제용 FSX"](#).

매개 변수	설명	예
smbShare	공유 폴더 Microsoft 관리 콘솔을 사용하여 생성한 SMB 공유의 이름입니다. 예: "smb-share". * SMB 볼륨에 필요합니다. *	smb-share
nasType	* 를 로 설정해야 합니다 smb. * null인 경우 기본값은 로 설정됩니다 nfs.	smb
securityStyle	새로운 볼륨에 대한 보안 스타일 * 를 로 설정해야 합니다 ntfs 또는 mixed SMB 볼륨용. *	ntfs 또는 mixed SMB 볼륨용
unixPermissions	모드를 선택합니다. SMB 볼륨에 대해서는 * 를 비워 두어야 합니다. *	""

ONTAP 구성 옵션 및 예제용 FSX

Amazon FSx for ONTAP의 백엔드 구성 옵션에 대해 알아보십시오. 이 섹션에서는 백엔드 구성 예를 제공합니다.

백엔드 구성 옵션

백엔드 구성 옵션은 다음 표를 참조하십시오.

매개 변수	설명	예
version		항상 1
storageDriverName	스토리지 드라이버의 이름입니다	"ONTAP-NAS", "ONTAP-NAS-이코노미", "ONTAP-NAS-Flexgroup", "ONTAP-SAN", "ONTAP-SAN-이코노미"
backendName	사용자 지정 이름 또는 스토리지 백엔드	드라이버 이름 + "_" + dataLIF
managementLIF	클러스터 또는 SVM 관리 LIF의 IP 주소 MetroCluster의 원활한 전환을 위해 SVM 관리 LIF를 지정해야 합니다. FQDN(정규화된 도메인 이름)을 지정할 수 있습니다. 을 사용하여 Astra Trident가 설치된 경우 IPv6 주소를 사용하도록 설정할 수 있습니다 --use-ipv6 깃발. IPv6 주소는 대괄호로 묶어야 합니다(예: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]).	"10.0.0.1", "[2001:1234:ABCD::fee]"
dataLIF	프로토콜 LIF의 IP 주소입니다. * ONTAP NAS 드라이버 *: 데이터 LIF를 지정하는 것이 좋습니다. 제공되지 않는 경우 Astra Trident는 SVM에서 데이터 LIF를 가져옵니다. NFS 마운트 작업에 사용할 FQDN(정규화된 도메인 이름)을 지정하면 여러 데이터 LIF에서 로드 밸런싱을 위해 라운드 로빈 DNS를 생성할 수 있습니다. 초기 설정 후에 변경할 수 있습니다. 을 참조하십시오. * ONTAP SAN 드라이버 *: iSCSI에 대해 지정하지 마십시오. Astra Trident는 ONTAP 선택적 LUN 맵을 사용하여 다중 경로 세션을 설정하는데 필요한 iSCSI LIF를 검색합니다. 데이터 LIF가 명시적으로 정의되어 있으면 경고가 생성됩니다. 을 사용하여 Astra Trident가 설치된 경우 IPv6 주소를 사용하도록 설정할 수 있습니다 --use-ipv6 깃발. IPv6 주소는 대괄호로 묶어야 합니다(예: [28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]).	
autoExportPolicy	자동 익스포트 정책 생성 및 업데이트 [Boolean] 활성화 를 사용합니다 autoExportPolicy 및 autoExportCIDRs 옵션, Astra Trident는 익스포트 정책을 자동으로 관리할 수 있습니다.	"거짓"

매개 변수	설명	예
autoExportCIDRs	Kubernetes 노드 IP를 언제 기준으로 필터링하는 CIDR 목록입니다 autoExportPolicy 가 활성화됩니다. 를 사용합니다 autoExportPolicy 및 autoExportCIDRs 옵션, Astra Trident는 익스포트 정책을 자동으로 관리할 수 있습니다.	"["0.0.0.0/0", "::/0"]"
labels	볼륨에 적용할 임의의 JSON 형식 레이블 세트입니다	""
clientCertificate	Base64 - 클라이언트 인증서의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다	""
clientPrivateKey	Base64 - 클라이언트 개인 키의 인코딩된 값입니다. 인증서 기반 인증에 사용됩니다	""
trustedCACertificate	Base64 - 신뢰할 수 있는 CA 인증서의 인코딩된 값입니다. 선택 사항. 인증서 기반 인증에 사용됩니다.	""
username	클러스터 또는 SVM에 연결할 사용자 이름입니다. 자격 증명 기반 인증에 사용됩니다. 예: vsadmin.	
password	클러스터 또는 SVM에 연결하는 암호 자격 증명 기반 인증에 사용됩니다.	
svm	사용할 스토리지 가상 머신입니다	SVM 관리 LIF가 지정된 경우에 파생됩니다.
igroupName	사용할 SAN 볼륨에 대한 igroup의 이름입니다. 을 참조하십시오 .	"삼중 - <backend-UUID>"
storagePrefix	SVM에서 새 볼륨을 프로비저닝할 때 사용되는 접두사 생성 후에는 수정할 수 없습니다. 이 매개 변수를 업데이트하려면 새 백엔드를 생성해야 합니다.	"삼중류"
limitAggregateUsage	* NetApp ONTAP * 용 Amazon FSx는 지정하지 마십시오 fsxadmin 및 vsadmin 애그리게이트 사용을 검색하고 Astra Trident를 사용하여 제한하는 데 필요한 권한이 포함되어 있지 않습니다.	사용하지 마십시오.

매개 변수	설명	예
limitVolumeSize	요청된 볼륨 크기가 이 값보다 큰 경우 용량 할당에 실패합니다. 또한 qtree 및 LUN에 대해 관리하는 볼륨의 최대 크기 및 을 제한합니다 qtreesPerFlexvol 옵션을 사용하면 FlexVol당 최대 qtree 수를 사용자 지정할 수 있습니다.	""(기본적으로 적용되지 않음)
lunsPerFlexvol	FlexVol당 최대 LUN 수는 [50, 200] 범위 내에 있어야 합니다. SAN만 해당.	"100"
debugTraceFlags	문제 해결 시 사용할 디버그 플래그입니다. 예: {"api":false, "method":true} 사용하지 마십시오 debugTraceFlags 문제 해결 및 자세한 로그 덤프가 필요한 경우를 제외하고	null입니다
nfsMountOptions	심표로 구분된 NFS 마운트 옵션 목록입니다. Kubernetes 영구 볼륨의 마운트 옵션은 일반적으로 스토리지 클래스에서 지정되지만, 스토리지 클래스에 마운트 옵션을 지정하지 않으면 Astra Trident가 스토리지 백엔드의 구성 파일에 지정된 마운트 옵션을 사용하여 로 돌아갑니다. 스토리지 클래스 또는 구성 파일에 마운트 옵션이 지정되지 않은 경우 Astra Trident는 연결된 영구 볼륨에 마운트 옵션을 설정하지 않습니다.	""
nasType	NFS 또는 SMB 볼륨 생성을 구성합니다. 옵션은 입니다 nfs, smb 또는 null입니다. * 를 로 설정해야 합니다 `smb` SMB 볼륨의 경우. * null로 설정하면 기본적으로 NFS 볼륨이 설정됩니다.	"NFS"
qtreesPerFlexvol	FlexVol당 최대 qtree, 범위 [50, 300]에 있어야 함	"200"
smbShare	공유 폴더 Microsoft 관리 콘솔을 사용하여 생성한 SMB 공유의 이름입니다. * SMB 볼륨에 필요합니다. *	"SMB 공유"

매개 변수	설명	예
useREST	ONTAP REST API를 사용하는 부울 매개 변수입니다. * 기술 미리 보기 * useREST 프로덕션 작업 부하가 아닌 테스트 환경에 권장되는 기술 미리 보기로 제공됩니다. 를 로 설정한 경우 true, Astra Trident는 ONTAP REST API를 사용하여 백엔드와 통신합니다. 이 기능을 사용하려면 ONTAP 9.11.1 이상이 필요합니다. 또한 사용되는 ONTAP 로그인 역할에 대한 액세스 권한이 있어야 합니다 ontap 응용 프로그램. 이는 사전 정의된 에 의해 충족됩니다 vsadmin 및 cluster-admin 역할.	"거짓"

에 대한 세부 정보 igroupName

igroupName ONTAP 클러스터에서 이미 생성된 igroup으로 설정할 수 있습니다. 지정되지 않은 경우 Astra Trident가 이름이 인 igroup을 자동으로 생성합니다 trident-<backend-UUID>.

미리 정의된 횃수 이름을 제공하는 경우 환경 간에 SVM을 공유하려면 Kubernetes 클러스터 당 하나의 igroup을 사용하는 것이 좋습니다. 이는 Astra Trident가 IQN 추가 및 삭제를 자동으로 유지 관리하는 데 필요합니다.

- igroupName Astra Trident 외부의 SVM에서 생성 및 관리되는 새로운 igroup을 가리키도록 업데이트할 수 있습니다.
- igroupName 생략할 수 있습니다. 이 경우 Astra Trident가 이름을 가진 igroup을 생성하고 관리합니다 trident-<backend-UUID> 자동으로.

두 경우 모두 볼륨 첨부 파일에 계속 액세스할 수 있습니다. 향후 볼륨 첨부 파일은 업데이트된 igroup을 사용합니다. 이 업데이트는 백엔드에 있는 볼륨에 대한 액세스를 방해하지 않습니다.

업데이트 dataLIF 초기 구성 후

다음 명령을 실행하여 초기 구성 후에 데이터 LIF를 변경할 수 있으며, 업데이트된 데이터 LIF가 포함된 새 백엔드 JSON 파일을 제공할 수 있습니다.

```
tridentctl update backend <backend-name> -f <path-to-backend-json-file-with-updated-dataLIF>
```



PVC가 하나 이상의 포드에 연결된 경우 해당 포드를 모두 내린 다음 다시 불러와서 새 데이터 LIF가 적용되도록 해야 합니다.

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
spaceAllocation	LUN에 대한 공간 할당	"참"
spaceReserve	공간 예약 모드, "없음"(싌) 또는 "볼륨"(일반)	"없음"
snapshotPolicy	사용할 스냅샷 정책입니다	"없음"
qosPolicy	생성된 볼륨에 할당할 QoS 정책 그룹입니다. 스토리지 풀 또는 백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다. Astra Trident와 함께 QoS 정책 그룹을 사용하려면 ONTAP 9.8 이상이 필요합니다. 비공유 QoS 정책 그룹을 사용하고 정책 그룹이 각 구성요소별로 적용되도록 하는 것이 좋습니다. 공유 QoS 정책 그룹은 모든 워크로드의 총 처리량에 대해 상한을 적용합니다.	""
adaptiveQosPolicy	생성된 볼륨에 할당할 적응형 QoS 정책 그룹입니다. 스토리지 풀 또는 백엔드에서 qosPolicy 또는 adapativeQosPolicy 중 하나를 선택합니다. ONTAP에서 지원되지 않음 - NAS - 이코노미	""
snapshotReserve	스냅샷 "0"에 예약된 볼륨의 백분율	If(경우 snapshotPolicy "없음"이고, 그렇지 않으면 ""입니다.
splitOnClone	생성 시 상위 클론에서 클론을 분할합니다	"거짓"
encryption	새 볼륨에 NVE(NetApp Volume Encryption)를 활성화합니다. 기본값은 입니다 false. 이 옵션을 사용하려면 NVE 라이선스가 클러스터에서 활성화되어 있어야 합니다. 백엔드에서 NAE가 활성화된 경우 Astra Trident에 프로비저닝된 모든 볼륨은 NAE가 활성화됩니다. 자세한 내용은 다음을 참조하십시오. "Astra Trident가 NVE 및 NAE와 연동되는 방식" .	"거짓"
luksEncryption	LUKS 암호화를 사용합니다. 을 참조하십시오 "LUKS(Linux Unified Key Setup) 사용" . SAN만 해당.	""
tieringPolicy	"없음"을 사용하는 계층화 정책	ONTAP 9.5 이전 SVM-DR 구성의 경우 "스냅샷 전용
unixPermissions	모드를 선택합니다. * SMB 볼륨의 경우 비워 둡니다. *	""

매개 변수	설명	기본값
securityStyle	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.

예

사용 nasType, node-stage-secret-name, 및 node-stage-secret-namespace, SMB 볼륨을 지정하고 필요한 Active Directory 자격 증명을 제공할 수 있습니다. SMB 볼륨은 를 사용하여 지원됩니다 ontap-nas 드라이버만 해당.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: nas-smb-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
  trident.netapp.io/nasType: "smb"
  csi.storage.k8s.io/node-stage-secret-name: "smbcreds"
  csi.storage.k8s.io/node-stage-secret-namespace: "default"
```

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.