



# バックエンドを設定

## Astra Trident

NetApp  
November 14, 2025

# 目次

バックエンドを設定	1
バックエンドを設定	1
Azure NetApp Files の特長	1
Azure NetApp Files バックエンドを設定します	1
Azure NetApp Files バックエンドを設定する準備をします	1
Azure NetApp Files バックエンド構成のオプションと例	4
Google Cloudバックエンド用にCloud Volumes Service を設定します	13
Cloud Volumes Service for Google Cloudに対するAstra Tridentサポートの詳細をご確認ください	13
バックエンド構成オプション	14
ボリュームのプロビジョニングオプション	16
CVS -パフォーマンスサービスの種類の例	17
CVSサービスタイプの例	26
次の手順	30
NetApp HCI または SolidFire バックエンドを設定します	30
作業を開始する前に	30
ボリュームモード	31
バックエンド構成オプション	31
例1：のバックエンド構成 solidfire-san 3種類のボリュームを備えたドライバ	32
例2：のバックエンドとストレージクラスの設定 solidfire-san 仮想プールを備えたドライバ	33
詳細については、こちらをご覧ください	37
ONTAP SANドライバ	37
ONTAP SANドライバの概要	37
バックエンドにONTAP SANドライバを設定する準備をします	37
ONTAP のSAN構成オプションと例	44
ONTAP NAS ドライバ	59
ONTAP NASドライバの概要	59
ONTAP NASドライバを使用してバックエンドを設定する準備をします	60
ONTAP NASの設定オプションと例	70
NetApp ONTAP 対応の Amazon FSX	89
Amazon FSX for NetApp ONTAP で Astra Trident を使用	89
NetApp ONTAP 向けAmazon FSXを統合します	91
FSX（ONTAP の構成オプションと例）	94

# バックエンドを設定

## バックエンドを設定

バックエンドは、Astra Trident とストレージシステムの関係性を定義します。Trident がストレージシステムとの通信方法を Trident から指示し、Astra Trident がボリュームをプロビジョニングする方法も解説します。

Astra Tridentは、ストレージクラスによって定義された要件に一致するストレージプールをバックエンドから自動的に提供します。ストレージシステムにバックエンドを設定する方法について説明します。

- ["Azure NetApp Files バックエンドを設定します"](#)
- ["Cloud Volumes Service for Google Cloud Platform バックエンドを設定します"](#)
- ["NetApp HCI または SolidFire バックエンドを設定します"](#)
- ["ONTAPまたはCloud Volumes ONTAP NASドライバを使用したバックエンドの設定"](#)
- ["バックエンドに ONTAP または Cloud Volumes ONTAP SAN ドライバを設定します"](#)
- ["Amazon FSX for NetApp ONTAP で Astra Trident を使用"](#)

## Azure NetApp Files の特長

### Azure NetApp Files バックエンドを設定します

Azure NetApp Files（ANF）をAstra Tridentのバックエンドとして設定できます。ANFバックエンドを使用してNFSボリュームとSMBボリュームを接続できます。

#### 考慮事項

- Azure NetApp Files サービスでは、100GB未満のボリュームはサポートされません。100 GB のボリュームが小さい場合は、Trident が自動的に作成します。
- Astra Tridentは、Windowsノードで実行されているポッドにマウントされたSMBボリュームのみをサポート

### Azure NetApp Files バックエンドを設定する準備をします

Azure NetApp Files バックエンドを設定する前に、次の要件を満たしていることを確認する必要があります。

#### NFSボリュームとSMBボリュームの前提条件



Azure NetApp Files を初めてまたは新しい場所で使用する場合は、Azure NetApp Files をセットアップしてNFSボリュームを作成するためにいくつかの初期設定が必要です。を参照してください ["Azure : Azure NetApp Files をセットアップし、NFSボリュームを作成します"](#)。

を設定して使用します ["Azure NetApp Files の特長"](#) バックエンドには次のものがが必要です。

- 容量プール。を参照してください ["Microsoft：Azure NetApp Files 用の容量プールを作成します"](#)。
- Azure NetApp Files に委任されたサブネット。を参照してください ["Microsoft：サブネットをAzure NetApp Files に委任します"](#)。
- subscriptionID Azure NetApp Files を有効にしたAzureサブスクリプションから選択します。
- tenantID、clientID`および `clientSecret から ["アプリケーション登録"](#) Azure Active Directory で、Azure NetApp Files サービスに対する十分な権限がある。アプリケーション登録では、次のいずれかを使用します。
  - オーナーまたは寄与者のロール ["Azureで事前定義"](#)。
  - A ["カスタム投稿者ロール"](#) をサブスクリプションレベルで選択します (assignableScopes)以下のアクセス許可は、Astra Tridentが必要とするものに限定されます。カスタムロールを作成したあと、["Azureポータルを使用してロールを割り当てます"](#)。

```
{
  "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",

",

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

",


```

```

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

"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": []
}
]
}
}

```

- Azureがサポートされます location を1つ以上含むデータセンターを展開します **"委任されたサブネット"**。Trident 22.01の時点では location パラメータは、バックエンド構成ファイルの最上位にある必須フィールドです。仮想プールで指定された場所の値は無視されます。

## SMBボリュームに関するその他の要件

SMBボリュームを作成するには、以下が必要です。

- Active Directoryが設定され、Azure NetApp Files に接続されています。を参照してください "[Microsoft : Azure NetApp Files のActive Directory接続を作成および管理します](#)"。
- Linuxコントローラノードと少なくとも1つのWindowsワーカーノードでWindows Server 2019を実行しているKubernetesクラスター。Astra Tridentは、Windowsノードで実行されているポッドにマウントされたSMBボリュームのみをサポート
- Azure NetApp Files がActive Directoryに対して認証できるように、Active Directoryクレデンシャルを含むAstra Tridentのシークレットが少なくとも1つ含まれています。シークレットを生成します `smbcreds` :

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

- Windowsサービスとして設定されたCSIプロキシ。を設定します `csi-proxy` を参照してください "[GitHub: CSIプロキシ](#)" または "[GitHub: Windows向けCSIプロキシ](#)" Windowsで実行されているKubernetesノードの場合。

## Azure NetApp Files バックエンド構成のオプションと例

ANF用のNFSとSMBのバックエンド構成オプションについて説明し、設定例を確認してください。

### バックエンド構成オプション

Astra Tridentは、バックエンド構成（サブネット、仮想ネットワーク、サービスレベル、場所）を使用して、要求された場所で利用可能で、要求されたサービスレベルとサブネットに一致する容量プールにANFボリュームを作成します。



Astra Trident は、手動 QoS 容量プールをサポートしていません。

ANFバックエンドには次の設定オプションがあります。

パラメータ	説明	デフォルト
version		常に 1
storageDriverName	ストレージドライバの名前	「 azure-NetApp-files 」
backendName	カスタム名またはストレージバックエンド	ドライバ名 + "_" + ランダムな文字
subscriptionID	Azure サブスクリプションのサブスクリプション ID	
tenantID	アプリケーション登録からのテナント ID	
clientID	アプリケーション登録からのクライアント ID	

パラメータ	説明	デフォルト
clientSecret	アプリケーション登録からのクライアントシークレット	
serviceLevel	の1つ Standard、 Premium`または `Ultra	"" (ランダム)
location	新しいボリュームを作成する Azure の場所の名前	
resourceGroups	検出されたリソースをフィルタリングするためのリソースグループのリスト	[] (フィルタなし)
netappAccounts	検出されたリソースをフィルタリングするためのネットアップアカウントのリスト	[] (フィルタなし)
capacityPools	検出されたリソースをフィルタリングする容量プールのリスト	[] (フィルタなし、ランダム)
virtualNetwork	委任されたサブネットを持つ仮想ネットワークの名前	""
subnet	に委任されたサブネットの名前 Microsoft.Netapp/volumes	""
networkFeatures	ボリューム用のVNet機能のセットです。の場合もあります Basic または Standard。  ネットワーク機能は一部の地域では使用できず、サブスクリプションで有効にする必要がある場合があります。を指定します networkFeatures この機能を有効にしないと、ボリュームのプロビジョニングが失敗します。	""
nfsMountOptions	NFS マウントオプションのきめ細かな制御。  SMBボリュームでは無視されます。  NFSバージョン4.1を使用してボリュームをマウントするには、を参照してください nfsvers=4 カンマで区切って複数のマウントオプションリストを指定し、NFS v4.1 を選択します。  ストレージクラス定義で設定されたマウントオプションは、バックエンド構成で設定されたマウントオプションよりも優先されます。	"nfsvers=3 "

パラメータ	説明	デフォルト
limitVolumeSize	要求されたボリュームサイズがこの値を超えている場合はプロビジョニングが失敗します	""（デフォルトでは適用されません）
debugTraceFlags	トラブルシューティング時に使用するデバッグフラグ。例： \{"api": false, "method": true, "discovery": true\}。 トラブルシューティングを行って詳細なログダンプが必要な場合を除き、このオプションは使用しないでください。	null
nasType	NFSボリュームまたはSMBボリュームの作成を設定  オプションはです nfs、 smb または null。nullに設定すると、デフォルトでNFSボリュームが使用されます。	nfs



ネットワーク機能の詳細については、を参照してください ["Azure NetApp Files ボリュームのネットワーク機能を設定します"](#)。

#### 必要な権限とリソース

PVCの作成時に「No capacity pools found」エラーが発生した場合、アプリケーション登録に必要な権限とリソース（サブネット、仮想ネットワーク、容量プール）が関連付けられていない可能性があります。デバッグが有効になっている場合、Astra Tridentはバックエンドの作成時に検出されたAzureリソースをログに記録します。適切なロールが使用されていることを確認します。

の値 resourceGroups、netappAccounts、capacityPools、virtualNetwork および subnet 短縮名または完全修飾名を使用して指定できます。ほとんどの場合、短縮名は同じ名前の複数のリソースに一致する可能性があるため、完全修飾名を使用することを推奨します。

。resourceGroups、netappAccounts および capacityPools 値は、検出されたリソースのセットをこのストレージバックエンドで使用可能なリソースに制限するフィルタであり、任意の組み合わせで指定できます。完全修飾名の形式は次のとおりです。

を入力します	の形式で入力し
リソースグループ	< リソースグループ >
ネットアップアカウント	< リソースグループ > / < ネットアップアカウント >
容量プール	< リソースグループ > / < ネットアップアカウント > / < 容量プール >
仮想ネットワーク	< リソースグループ > / < 仮想ネットワーク >
サブネット	< resource group > / < 仮想ネットワーク > / < サブネット >



## ボリュームのプロビジョニング

構成ファイルの特別なセクションで次のオプションを指定することで、デフォルトのボリュームプロビジョニングを制御できます。を参照してください [\[構成例\]](#) を参照してください。

パラメータ	説明	デフォルト
exportRule	新しいボリュームに対するエクスポートルール  exportRule CIDR表記のIPv4アドレスまたはIPv4サブネットの任意の組み合わせをカンマで区切って指定する必要があります。  SMBボリュームでは無視されます。	"0.0.0.0/0 "
snapshotDir	.snapshot ディレクトリの表示を制御します	いいえ
size	新しいボリュームのデフォルトサイズ	" 100G "
unixPermissions	新しいボリュームのUNIX権限（8進数の4桁）。  SMBボリュームでは無視されます。	""（プレビュー機能、サブスクリプションでホワイトリスト登録が必要）

## 構成例

### 例 1：最小限の構成

これは、バックエンドの絶対的な最小構成です。この構成では、ANF に委譲されたネットアップアカウント、容量プール、サブネットがすべて検出され、それらのプールまたはサブネットの 1 つに新しいボリュームがランダムに配置されます。理由 `nasType` は省略されています `nfs` デフォルトが適用され、バックエンドがNFSボリュームにプロビジョニングされます。

この構成は、ANF の利用を開始して何を試してみるときに理想的ですが、実際には、プロビジョニングするボリュームの範囲をさらに設定することを検討しています。

```
---
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
```

## 例 2：容量プールフィルタを使用した特定のサービスレベル設定

このバックエンド構成では、Azureにボリュームが配置されます `eastus` の場所 `Ultra` 容量プール  
：Astra Trident は、ANF に委譲されたすべてのサブネットをその場所で自動的に検出し、いずれかのサブネットに新しいボリュームをランダムに配置します。

```
---
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
```

### 例 3：高度な設定

このバックエンド構成は、ボリュームの配置を単一のサブネットにまで適用する手間をさらに削減し、一部のボリュームプロビジョニングのデフォルト設定も変更します。

```
---
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'
```

#### 例4：仮想プールの構成

このバックエンド構成では、1つのファイルに複数のストレージプールを定義します。これは、異なるサービスレベルをサポートする複数の容量プールがあり、それらを表すストレージクラスを Kubernetes で作成する場合に便利です。プールを区別するために、仮想プールのラベルを使用しました 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
```

#### ストレージクラスの定義

次のようになります StorageClass 定義は、上記のストレージプールを参照してください。

を使用した定義の例 `parameter.selector` フィールド

を使用します `parameter.selector` を指定できます `StorageClass` ボリュームをホストするために使用される仮想プール。ボリュームには、選択したプールで定義された要素があります。

```
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**ボリュームの定義例

を使用します `nasType`、``node-stage-secret-name`` および ``node-stage-secret-namespace`` を使用して、SMB ボリュームを指定し、必要な Active Directory クレデンシャルを指定できます。

### 例1：デフォルトネームスペースの基本設定

```
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"
```

### 例2：ネームスペースごとに異なるシークレットを使用する

```
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}
```

### 例3：ボリュームごとに異なるシークレットを使用する

```
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ボリュームをサポートするプールでフィルタリングします。nasType: `nfs` または nasType: `null` NFSプールに対してフィルタを適用します。

バックエンドを作成します

バックエンド構成ファイルを作成したら、次のコマンドを実行します。

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

バックエンドの作成に失敗した場合は、バックエンドの設定に何か問題があります。次のコマンドを実行すると、ログを表示して原因を特定できます。

```
tridentctl logs
```

構成ファイルで問題を特定して修正したら、create コマンドを再度実行できます。

## Google Cloudバックエンド用にCloud Volumes Service を設定します

ネットアップCloud Volumes Service for Google CloudをAstra Tridentのバックエンドとして構成する方法を、提供されている構成例を使用して説明します。

**Cloud Volumes Service for Google Cloud**に対する**Astra Trident**サポートの詳細をご確認ください

TridentがCloud Volumes Service ボリュームを作成できるのは、2つのうちの1つです "[サービスタイプ](#)" :

- **\* CVS - Performance \*** : デフォルトのAstra Tridentサービスタイプ。パフォーマンスが最適化されたこのサービスタイプは、パフォーマンスを重視する本番環境のワークロードに最適です。CVS -パフォーマンスサービスタイプは、サイズが100GiB以上のボリュームをサポートするハードウェアオプションです。のいずれかを選択できます "[3つのサービスレベル](#)" :

- standard
- premium
- extreme

- **\* CVS \*** : CVSサービスタイプは、中程度のパフォーマンスレベルに制限された高レベルの可用性を提供します。CVSサービスタイプは、ストレージプールを使用して1GiB未満のボリュームをサポートするソフトウェアオプションです。ストレージプールには最大50個のボリュームを含めることができ、すべてのボリュームでプールの容量とパフォーマンスを共有できます。のいずれかを選択できます "[2つのサービスレベル](#)" :

- standardsw
- zoneredundantstandardsw

必要なもの

を設定して使用します "Cloud Volumes Service for Google Cloud" バックエンドには次のものがが必要です。

- NetApp Cloud Volumes Service で設定されたGoogle Cloudアカウント
- Google Cloud アカウントのプロジェクト番号
- を使用するGoogle Cloudサービスアカウント `netappcloudvolumes.admin` ロール
- Cloud Volumes Service アカウントのAPIキーファイル

## バックエンド構成オプション

各バックエンドは、1つの Google Cloud リージョンにボリュームをプロビジョニングします。他のリージョンにボリュームを作成する場合は、バックエンドを追加で定義します。

パラメータ	説明	デフォルト
version		常に 1
storageDriverName	ストレージドライバの名前	"GCP-cvs"
backendName	カスタム名またはストレージバックエンド	ドライバ名 + "_" + API キーの一部
storageClass	CVSサービスタイプを指定するためのオプションのパラメータ。  使用 <code>software</code> をクリックしてCVSサービスタイプを選択します。それ以外の場合は、Astra TridentがCVSパフォーマンスサービスのタイプを引き継ぎます (hardware) 。	
storagePools	CVSサービスタイプのみ。ボリューム作成用のストレージプールを指定するオプションのパラメータ。	
projectNumber	Google Cloud アカウントのプロジェクト番号。この値は、Google Cloudポータルホームページにあります。	
hostProjectNumber	共有VPCネットワークを使用する場合は必須です。このシナリオでは、 <code>projectNumber</code> は、サービスプロジェクトです <code>hostProjectNumber</code> は、ホストプロジェクトです。	



パラメータ	説明	デフォルト
apiRegion	<p>Astra TridentがCloud Volumes Service ボリュームを作成するGoogle Cloudリージョン。複数リージョンのKubernetesクラスタを作成する場合は、に作成されたボリューム apiRegion 複数のGoogle Cloudリージョンのノードでスケジュールされたワークロードで使用できます。</p> <p>リージョン間トラフィックは追加コストを発生させます。</p>	
apiKey	<p>を使用したGoogle CloudサービスアカウントのAPIキー netappcloudvolumes.admin ロール。</p> <p>このレポートには、Google Cloud サービスアカウントの秘密鍵ファイルの JSON 形式のコンテンツが含まれています（バックエンド構成ファイルにそのままコピーされます）。</p>	
proxyURL	<p>CVSアカウントへの接続にプロキシサーバが必要な場合は、プロキシURLを指定します。プロキシサーバには、HTTP プロキシまたはHTTPS プロキシを使用できます。</p> <p>HTTPS プロキシの場合、プロキシサーバで自己署名証明書を使用するために証明書の検証はスキップされます。</p> <p>認証が有効になっているプロキシサーバはサポートされていません。</p>	
nfsMountOptions	NFS マウントオプションのきめ細かな制御。	"nfsvers=3 "
limitVolumeSize	要求されたボリュームサイズがこの値を超えている場合はプロビジョニングが失敗します。	""（デフォルトでは適用されません）

パラメータ	説明	デフォルト
serviceLevel	<p>新しいボリュームのCVS -パフォーマンスレベルまたはCVSサービスレベル。</p> <p>CVS -パフォーマンスの値はです standard、 premium`または`extreme。</p> <p>CVSの値はです standardsw または zoneredundantstandardsw。</p>	<p>CVS -パフォーマンスのデフォルトは「Standard」です。</p> <p>CVSのデフォルトは"standardsw"です。</p>
network	Cloud Volumes Service ボリュームに使用するGoogle Cloudネットワーク。	デフォルト
debugTraceFlags	<p>トラブルシューティング時に使用するデバッグフラグ。例： \{"api":false, "method":true}。</p> <p>トラブルシューティングを行って詳細なログダンプが必要な場合を除き、このオプションは使用しないでください。</p>	null
allowedTopologies	<p>クロスリージョンアクセスを有効にするには、のStorageClass定義を使用します allowedTopologies すべてのリージョンを含める必要があります。</p> <p>例： - key: topology.kubernetes.io/region values: - us-east1 - europe-west1</p>	

## ボリュームのプロビジョニングオプション

では、デフォルトのボリュームプロビジョニングを制御できます defaults 構成ファイルのセクション。

パラメータ	説明	デフォルト
exportRule	新しいボリュームのエクスポートルール。CIDR 表記の IPv4 アドレスまたは IPv4 サブネットの任意の組み合わせをカンマで区切って指定する必要があります。	"0.0.0.0/0 "

パラメータ	説明	デフォルト
snapshotDir	にアクセスします .snapshot ディレクトリ	いいえ
snapshotReserve	Snapshot 用にリザーブされている ボリュームの割合	""（CVS のデフォルト値をそのまま使用）
size	新しいボリュームのサイズ。  CVS -パフォーマンス最小値 は100GiBです。  CVS最小値は1GiBです。	CVS -パフォーマンスサービスのタイプはデフォルトで「100GiB」です。  CVSサービスのタイプではデフォルトが設定されませんが、1GiB以上が必要です。

## CVS -パフォーマンスサービスの種類の例

次の例は、CVS -パフォーマンスサービスタイプの設定例を示しています。

[illegible]

```
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
```

[illegible]

```
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
```

### 例3：仮想プールの構成

この例では、を使用します `storage` 仮想プールおよびを設定します `StorageClasses` それぞれを再度参照する。を参照してください [\[ストレージクラスの定義\]](#) をクリックして、ストレージクラスの定義方法を確認します。

ここでは、すべての仮想プールに対して特定のデフォルトが設定され、すべての仮想プールに対してが設定されます snapshotReserve 5%およびである exportRule を0.0.0.0/0に設定します。仮想プールは、で定義されます storage セクション。個々の仮想プールにはそれぞれ独自の定義があります serviceLevel をクリックすると、一部のプールでデフォルト値が上書きされます。プールを区別するために、仮想プールのラベルを使用しました performance および protection。

[illegible]



```

znHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3b1/qp8B4Kws8zX5ojY9m
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
```

## ストレージクラスの定義

次のStorageClass定義は、仮想プールの構成例に適用されます。を使用します `parameters.selector` では、ボリュームのホストに使用する仮想プールをストレージクラスごとに指定できます。ボリュームには、選択したプールで定義された要素があります。

```

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
```

- 最初のストレージクラス (cvs-extreme-extra-protection) を最初の仮想プールにマッピングします。スナップショット予約が 10% の非常に高いパフォーマンスを提供する唯一のプールです。
- 最後のストレージクラス (cvs-extra-protection) スナップショット予約が10%のストレージプールを呼び出します。Tridentが、どの仮想プールを選択するかを決定し、スナップショット予約の要件が満たされていることを確認します。

## CVSサービスタイプの例

次の例は、CVSサービスタイプの設定例を示しています。

[illegible]

```
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
```

## 例2：ストレージプールの構成

このバックエンド設定の例では、を使用して storagePools ストレージプールを設定します。

```
---
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/Y9oTbZrjAHSBgJm2nHzFq2X0rqVMAHghI6ATm4DOuWx8XGWKTGIPlc0qPqJlqS
    LLaWOH4VIZQZCAyW5IUp9CAmwqHgdG0uhFNfCgMmED6PBUvVLsLvcq86X+QSWR9k
    ETqElj/sGCenPF7ti1DhGBFafd9hPnxg9PZY29ArEZwY9G/ZjZQX7WPgs0VvxiNR
    DxZRC3GXAgMBAAECggEACn5c59bG/qnVEVI1CwMAa1M5M2z09JFh1L1ljKwntNPj
    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+nobpTuvlo56ZRJVvVk7lFMsiddzMuHH8pxfgNJemwA4P
    ThDHCEjv035NNV6Kyo00tA==
    -----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
```

## 次の手順

バックエンド構成ファイルを作成したら、次のコマンドを実行します。

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

バックエンドの作成に失敗した場合は、バックエンドの設定に何か問題があります。次のコマンドを実行すると、ログを表示して原因を特定できます。

```
tridentctl logs
```

構成ファイルで問題を特定して修正したら、create コマンドを再度実行できます。

## NetApp HCI または SolidFire バックエンドを設定します

ネットアップが提供する Trident インストールで Element バックエンドを作成して使用する方法をご確認ください。

### 作業を開始する前に

Element バックエンドを作成する前に、次の情報が必要になります。

- Element ソフトウェアを実行する、サポート対象のストレージシステム。
- NetApp HCI / SolidFire クラスタ管理者またはボリュームを管理できるテナントユーザのクレデンシャル。
- すべての Kubernetes ワーカーノードに適切な iSCSI ツールをインストールする必要があります。を参照してください ["ワーカーノードの準備情報"](#)。



## ボリュームモード

。solidfire-san ストレージドライバは、ボリュームモード（fileとblock）の両方をサポートしています。  
をクリックします Filesystem volumeMode、Astra Tridentがボリュームを作成し、ファイルシステムを作成  
ファイルシステムのタイプは StorageClass で指定されます。

ドライバ	プロトコル	ボリュームモード	サポートされている アクセスモード	サポートされるファ イルシステム
solidfire-san	iSCSI	ブロック	RWO、ROX、 RWX	ファイルシステムが ありません。raw ブ ロックデバイスで す。
solidfire-san	iSCSI	ブロック	RWO、ROX、 RWX	ファイルシステムが ありません。raw ブ ロックデバイスで す。
solidfire-san	iSCSI	ファイルシステム	RWO、ROX	xfs、ext3、ext4
solidfire-san	iSCSI	ファイルシステム	RWO、ROX	xfs、ext3、ext4



Astra Trident は強化された CSI プロビジョニング担当者として機能する場合、CHAP を使用し  
ます。CSI のデフォルトである CHAP を使用している場合は、これ以上の準備は必要ありませ  
ん。を明示的に設定することを推奨します UseCHAP CSI以外のTridentでCHAPを使用するオブ  
ション。それ以外は、を参照してください ["こちらをご覧ください"](#)。



ボリュームアクセスグループは、従来の非 CSI フレームワークである Astra Trident でのみサポ  
ートされています。CSI モードで動作するように設定されている場合、Astra Trident は CHAP  
を使用します。

どちらでもない場合 AccessGroups または UseCHAP が設定され、次のいずれかのルールが適用されます。

- デフォルトの場合は trident アクセスグループが検出され、アクセスグループが使用されます。
- アクセスグループが検出されず、Kubernetes バージョンが 1.7 以降の場合は、CHAP が使用されます。

## バックエンド構成オプション

バックエンド設定オプションについては、次の表を参照してください。

パラメータ	説明	デフォルト
version		常に 1
storageDriverName	ストレージドライバの名前	常に「solidfire-san-」
backendName	カスタム名またはストレージバック エンド	「iSCSI_」 + ストレージ（iSCSI ） IP アドレス SolidFire

パラメータ	説明	デフォルト
Endpoint	テナントのクレデンシャルを使用する SolidFire クラスタの MVIP	
SVIP	ストレージ（iSCSI）の IP アドレスとポート	
labels	ボリュームに適用する任意の JSON 形式のラベルのセット。	「」
TenantName	使用するテナント名（見つからない場合に作成）	
InitiatorIFace	iSCSI トラフィックを特定のホストインターフェイスに制限します	デフォルト
UseCHAP	CHAP を使用して iSCSI を認証します	正しいです
AccessGroups	使用するアクセスグループ ID のリスト	「trident」という名前のアクセスグループの ID を検索します。
Types	QoS の仕様	
limitVolumeSize	要求されたボリュームサイズがこの値を超えている場合、プロビジョニングが失敗します	""（デフォルトでは適用されません）
debugTraceFlags	トラブルシューティング時に使用するデバッグフラグ。例：{"API" : false、"method" : true}	null



使用しないでください debugTraceFlags トラブルシューティングを実行していて、詳細なログダンプが必要な場合を除きます。

## 例1：のバックエンド構成 solidfire-san 3種類のボリュームを備えたドライバ

次の例は、CHAP 認証を使用するバックエンドファイルと、特定の QoS 保証を適用した 3 つのボリュームタイプのモデリングを示しています。その場合は、を使用して各ストレージクラスを使用するように定義します IOPS ストレージクラスのパラメータ。

```

---
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

```

## 例2：のバックエンドとストレージクラスの設定 solidfire-san 仮想プールを備えたドライバ

この例は、仮想プールとともに、それらを参照するStorageClassesとともに構成されているバックエンド定義ファイルを示しています。

Astra Tridentは、ストレージプール上にあるラベルを、プロビジョニング時にバックエンドストレージLUNにコピーします。ストレージ管理者は、仮想プールごとにラベルを定義したり、ボリュームをラベルでグループ化したりできます。

以下に示すバックエンド定義ファイルの例では、すべてのストレージプールに対して特定のデフォルトが設定されています。これにより、が設定されます type シルバー。仮想プールは、で定義されます storage セクション。この例では、一部のストレージプールが独自のタイプを設定し、一部のプールが上記のデフォルト値を上書きします。

```

---
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

```

次のStorageClass定義は、上記の仮想プールを参照しています。を使用する `parameters.selector` 各スト

レイジクラスは、ボリュームのホストに使用できる仮想プールを呼び出します。ボリュームには、選択した仮想プール内で定義された要素があります。

最初のストレージクラス (solidfire-gold-four) を選択すると、最初の仮想プールにマッピングされます。ゴールドのパフォーマンスを提供する唯一のプール Volume Type QoS 金の。最後のストレージクラス (solidfire-silver) Silverパフォーマンスを提供するストレージプールをすべて特定します。Tridentが、どの仮想プールを選択するかを判断し、ストレージ要件を確実に満たすようにします。

```

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"

```

詳細については、こちらをご覧ください

- ["ボリュームアクセスグループ"](#)

## ONTAP SANドライバ

### ONTAP SANドライバの概要

ONTAP および Cloud Volumes ONTAP SAN ドライバを使用した ONTAP バックエンドの設定について説明します。

### ONTAP SANドライバに関する重要な情報

Astra Controlは、で作成したボリュームに対して、シームレスな保護、ディザスタリカバリ、および移動（Kubernetesクラスター間でボリュームを移動）を提供します `ontap-nas`、`ontap-nas-flexgroup` および `ontap-san` ドライバ。を参照してください ["Astra Controlレプリケーションの前提条件"](#) を参照してください。

- を使用する必要があります `ontap-nas` データ保護、ディザスタリカバリ、モビリティを必要とする本番環境のワークロード向けのサービスです。
- 使用 `ontap-san-economy` 想定されるボリューム使用量がONTAP でサポートされる量よりも大幅に多い場合
- 使用 `ontap-nas-economy` 想定されるボリューム使用量が、ONTAP でサポートされるおおよびよりも大幅に多い場合にのみ該当します `ontap-san-economy` ドライバは使用できません。
- 使用しないでください `ontap-nas-economy` データ保護、ディザスタリカバリ、モビリティのニーズが予想される場合。

### ユーザ権限

Tridentは、通常はを使用して、ONTAP 管理者またはSVM管理者のどちらかとして実行される必要があります `admin` クラスターユーザまたは `vsadmin` SVMユーザ、または同じロールを持つ別の名前のユーザ。Amazon FSX for NetApp ONTAP 環境では、Astra Tridentは、クラスターを使用して、ONTAP 管理者またはSVM管理者のどちらかとして実行されるものと想定しています `fsxadmin` ユーザまたは `vsadmin` SVMユーザ、または同じロールを持つ別の名前のユーザ。。 `fsxadmin` このユーザは、クラスター管理者ユーザを限定的に置き換えるものです。



を使用する場合 `limitAggregateUsage` クラスター管理者権限が必要です。Amazon FSX for NetApp ONTAP をAstra Tridentとともに使用している場合は、を参照してください `limitAggregateUsage` パラメータはでは機能しません `vsadmin` および `fsxadmin` ユーザアカウント：このパラメータを指定すると設定処理は失敗します。

ONTAP 内では、Trident ドライバが使用できるより制限的な役割を作成することもできますが、推奨しません。Trident の新リリースでは、多くの場合、考慮すべきAPI が追加で必要になるため、アップグレードが難しく、エラーも起こりやすくなります。

### バックエンドにONTAP SANドライバを設定する準備をします

ONTAP SANドライバでONTAPバックエンドを構成するための要件と認証オプションを

理解します。

## 要件

ONTAP バックエンドすべてに対して、Astra Trident が SVM に少なくとも 1 つのアグリゲートを割り当てておく必要があります。

複数のドライバを実行し、1 つまたは複数のドライバを参照するストレージクラスを作成することもできます。たとえば、を設定できます `san-dev` を使用するクラス `ontap-san` ドライバおよび `A san-default` を使用するクラス `ontap-san-economy` 1 つ。

すべての Kubernetes ワーカーノードに適切な iSCSI ツールをインストールしておく必要があります。を参照してください ["ワーカーノードを準備します"](#) を参照してください。

## ONTAP バックエンドの認証

Astra Trident には、ONTAP バックエンドを認証する 2 つのモードがあります。

- **credential based** : 必要な権限を持つ ONTAP ユーザのユーザ名とパスワード。など、事前定義されたセキュリティログインロールを使用することを推奨します `admin` または `vsadmin` ONTAP のバージョンとの互換性を最大限に高めるため。
- **証明書ベース** : Astra Trident は、バックエンドにインストールされた証明書を使用して ONTAP クラスターと通信することもできます。この場合、バックエンド定義には、Base64 でエンコードされたクライアント証明書、キー、および信頼された CA 証明書（推奨）が含まれている必要があります。

既存のバックエンドを更新して、クレデンシャルベースの方式と証明書ベースの方式を切り替えることができます。ただし、一度にサポートされる認証方法は1つだけです。別の認証方式に切り替えるには、バックエンド設定から既存の方式を削除する必要があります。



クレデンシャルと証明書の両方を\*指定しようとすると、バックエンドの作成が失敗し、構成ファイルに複数の認証方法が指定されているというエラーが表示されます。

### クレデンシャルベースの認証を有効にします

Trident が ONTAP バックエンドと通信するには、SVM を対象とした管理者またはクラスターを対象とした管理者のクレデンシャルが必要です。などの標準の事前定義されたロールを使用することを推奨します `admin` または `vsadmin`。これにより、今後のリリースの ONTAP との互換性が今後のリリースの Astra Trident で使用される機能 API が公開される可能性があります。カスタムのセキュリティログインロールは Astra Trident で作成して使用できますが、推奨されません。

バックエンド定義の例は次のようになります。



## YAML

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

## JSON

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

バックエンド定義は、クレデンシャルがプレーンテキストで保存される唯一の場所であることに注意してください。バックエンドが作成されると、ユーザ名とパスワードが Base64 でエンコードされ、Kubernetes シークレットとして格納されます。クレデンシャルの知識が必要なのは、バックエンドの作成または更新だけです。この処理は管理者専用で、Kubernetes / ストレージ管理者が実行します。

証明書ベースの認証を有効にします

新規または既存のバックエンドは証明書を使用して ONTAP バックエンドと通信できます。バックエンド定義には 3 つのパラメータが必要です。

- `clientCertificate` : Base64 でエンコードされたクライアント証明書の値。
- `clientPrivateKey` : Base64 でエンコードされた、関連付けられた秘密鍵の値。
- `trustedCACertificate`: 信頼された CA 証明書の Base64 エンコード値。信頼された CA を使用する場合は、このパラメータを指定する必要があります。信頼された CA が使用されていない場合は無視してかまいません。

一般的なワークフローは次の手順で構成されます。

### 手順

1. クライアント証明書とキーを生成します。生成時に、ONTAP ユーザとして認証するように Common Name (CN ; 共通名) を設定します。

```
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. 信頼された CA 証明書を ONTAP クラスタに追加します。この処理は、ストレージ管理者がすでに行っている可能性があります。信頼できる CA が使用されていない場合は無視します。

```
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 クラスタにクライアント証明書とキーをインストールします（手順 1）。

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

4. ONTAP セキュリティログインロールでサポートされていることを確認する cert 認証方式。

```
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. 生成された証明書を使用して認証をテスト ONTAP 管理 LIF > と <vserver name> は、管理 LIF の IP アドレスおよび SVM 名に置き換えてください。

```
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. Base64 で証明書、キー、および信頼された CA 証明書をエンコードする。

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

7. 前の手順で得た値を使用してバックエンドを作成します。

```
cat cert-backend.json
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "backendName": "SanBackend",
  "managementLIF": "1.2.3.4",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaallllluuuueeee",
  "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 |
+-----+-----+-----+-----+
+-----+-----+
```

認証方法を更新するか、クレデンシャルをローテーションして

既存のバックエンドを更新して、別の認証方法を使用したり、クレデンシャルをローテーションしたりできます。これはどちらの方法でも機能します。ユーザ名とパスワードを使用するバックエンドは証明書を使用するように更新できますが、証明書を使用するバックエンドはユーザ名とパスワードに基づいて更新できます。これを行うには、既存の認証方法を削除して、新しい認証方法を追加する必要があります。次に、更新されたbackend.jsonファイルに必要なパラメータが含まれたものを使用して実行します `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 |
+-----+-----+-----+
+-----+-----+
```



パスワードのローテーションを実行する際には、ストレージ管理者が最初に ONTAP でユーザのパスワードを更新する必要があります。この後にバックエンドアップデートが続きます。証明書のローテーションを実行する際に、複数の証明書をユーザに追加することができます。その後、バックエンドが更新されて新しい証明書が使用されるようになります。この証明書に続く古い証明書は、ONTAP クラスタから削除できます。

バックエンドを更新しても、すでに作成されているボリュームへのアクセスは中断されず、その後のボリューム接続にも影響しません。バックエンドの更新が成功した場合、Astra Trident が ONTAP バックエンドと通信し、以降のボリューム処理を処理できることを示しています。

## 双方向 **CHAP** を使用して接続を認証します

Astra Tridentは、に対して双方向CHAPを使用してiSCSIセッションを認証できます `ontap-san` および `ontap-san-economy` ドライバ。これには、を有効にする必要があります `useCHAP` バックエンド定義のオプション。に設定すると `true`、Astra Tridentは、SVMのデフォルトのイニシエータセキュリティを双方向CHAPに設定し、バックエンドファイルからのユーザ名とシークレットを設定します。接続の認証には双方向 CHAPを使用することを推奨します。次の設定例を参照してください。

```

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

```



。 useCHAP パラメータは、1回だけ設定できるブール値のオプションです。デフォルトでは false に設定されています。true に設定したあとで、 false に設定することはできません。

に加えて useCHAP=true、chapInitiatorSecret、chapTargetInitiatorSecret、chapTargetUsername および chapUsername フィールドはバックエンド定義に含める必要があります。を実行すると、バックエンドが作成されたあとでシークレットを変更できます `tridentctl update`。

#### 動作の仕組み

を設定します useCHAP trueに設定すると、ストレージ管理者は、ストレージバックエンドでCHAPを設定するようにAstra Tridentに指示します。これには次のものが含まれます。

- SVM で CHAP をセットアップします。
  - SVMのデフォルトのイニシエータセキュリティタイプがnone（デフォルトで設定）\*で、ボリュームに既存のLUNがない場合、Astra Tridentはデフォルトのセキュリティタイプをに設定します CHAP CHAPイニシエータとターゲットのユーザ名およびシークレットの設定に進みます。
  - SVM に LUN が含まれている場合、Trident は SVM で CHAP を有効にしません。これにより、SVM にすでに存在する LUN へのアクセスが制限されることはありません。
- CHAP イニシエータとターゲットのユーザ名とシークレットを設定します。これらのオプションは、バックエンド構成で指定する必要があります（上記を参照）。

バックエンドが作成されると、対応するAstra Tridentによって作成されます `tridentbackend` CRDを実行し、CHAPシークレットとユーザ名をKubernetesシークレットとして保存します。このバックエンドのAstra Trident によって作成されたすべての PVS がマウントされ、CHAP 経由で接続されます。

#### クレデンシャルをローテーションし、バックエンドを更新

CHAPクレデンシャルを更新するには、でCHAPパラメータを更新します `backend.json` ファイル。CHAPシークレットを更新し、を使用する必要があります `tridentctl update` 変更を反映するためのコマンドです。



バックエンドのCHAPシークレットを更新する場合は、を使用する必要があります  
tridentctl バックエンドを更新します。Astra Trident では変更を取得できないため、CLI /  
ONTAP UI からストレージクラスタのクレデンシャルを更新しないでください。

```
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",
  "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-4378-8a33-1a472ffbe5c |
online |       7 |
+-----+-----+-----+-----+
+-----+-----+
```

既存の接続は影響を受けません。SVM の Astra Trident でクレデンシャルが更新されても、引き続きアクティブです。新しい接続では更新されたクレデンシャルが使用され、既存の接続は引き続きアクティブです。古い PVS を切断して再接続すると、更新されたクレデンシャルが使用されます。

## ONTAP のSAN構成オプションと例

ONTAP SAN ドライバを作成して Astra Trident インストールで使用方法をご確認ください。このセクションでは、バックエンド構成の例と、バックエンドをストレージクラスにマッピングする方法を詳しく説明します。

### バックエンド構成オプション

バックエンド設定オプションについては、次の表を参照してください。

パラメータ	説明	デフォルト
version		常に 1
storageDriverName	ストレージドライバの名前	ontap-nas、ontap-nas-economy、ontap-nas-flexgroup、ontap-san、ontap-san-economy
backendName	カスタム名またはストレージバックエンド	ドライバ名 + "_" + データ LIF
managementLIF	<p>クラスタ管理 LIF または SVM 管理 LIF の IP アドレス</p> <p>MetroClusterのシームレスなスイッチオーバーを実現するには、SVM 管理LIFを指定する必要があります。</p> <p>Fully Qualified Domain Name (FQDN；完全修飾ドメイン名) を指定できます。</p> <p>を使用してAstra Tridentをインストールした場合、IPv6アドレスを使用するようにを設定できます --use-ipv6 フラグ。IPv6アドレスは、[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]などの角かっこで定義する必要があります。</p>	「 10.0.0.1 」、 「 [2001:1234:abcd::fefe] 」
dataLIF	<p>プロトコル LIF の IP アドレス。</p> <p>* iSCSIには指定しないでください。* Astra Tridentが使用します <a href="#">"ONTAP の選択的LUNマップ"</a> iSCSI LIFを検出するには、マルチパスセッションを確立する必要があります。の場合は警告が生成されます dataLIF は明示的に定義されます。</p>	SVMの派生物です
useCHAP	<p>CHAPを使用してONTAP SANドライバのiSCSIを認証します（ブーリアン）。</p> <p>をに設定します true Astra Tridentでは、バックエンドで指定されたSVMのデフォルト認証として双方向CHAPを設定して使用します。 を参照してください <a href="#">"バックエンドにONTAP SANドライバを設定する準備をします"</a> を参照してください。</p>	false

パラメータ	説明	デフォルト
chapInitiatorSecret	CHAP イニシエータシークレット。の場合は必須です useCHAP=true	「」
labels	ボリュームに適用する任意の JSON 形式のラベルのセット	「」
chapTargetInitiatorSecret	CHAP ターゲットイニシエータシークレット。の場合は必須です useCHAP=true	「」
chapUsername	インバウンドユーザ名。の場合は必須です useCHAP=true	「」
chapTargetUsername	ターゲットユーザ名。の場合は必須です useCHAP=true	「」
clientCertificate	クライアント証明書の Base64 エンコード値。証明書ベースの認証に使用されます	「」
clientPrivateKey	クライアント秘密鍵の Base64 エンコード値。証明書ベースの認証に使用されます	「」
trustedCACertificate	信頼された CA 証明書の Base64 エンコード値。任意。証明書ベースの認証に使用されます。	「」
username	ONTAP クラスタとの通信に必要なユーザ名。クレデンシャルベースの認証に使用されます。	「」
password	ONTAP クラスタとの通信にパスワードが必要です。クレデンシャルベースの認証に使用されます。	「」
svm	使用する Storage Virtual Machine	SVMの場合に生成されます managementLIF を指定します
storagePrefix	SVM で新しいボリュームをプロビジョニングする際に使用するプレフィックスを指定します。  あとから変更することはできません。このパラメータを更新するには、新しいバックエンドを作成する必要があります。	trident



パラメータ	説明	デフォルト
limitAggregateUsage	<p>使用率がこの割合を超えている場合は、プロビジョニングが失敗します。</p> <p>NetApp ONTAP バックエンドにAmazon FSXを使用している場合は、指定しないでください limitAggregateUsage。提供された fsxadmin および vsadmin アグリゲートの使用状況を取得し、Astra Tridentを使用して制限するために必要な権限が含まれていない。</p>	""（デフォルトでは適用されません）
limitVolumeSize	<p>要求されたボリュームサイズがこの値を超えている場合、プロビジョニングが失敗します。</p> <p>また、qtreeおよびLUNに対して管理するボリュームの最大サイズを制限します。</p>	""（デフォルトでは適用されません）
lunsPerFlexvol	FlexVol あたりの最大 LUN 数。有効な範囲は 50 、 200 です	100
debugTraceFlags	<p>トラブルシューティング時に使用するデバッグフラグ。例： {"API" : false 、 "method" : true}</p> <p>トラブルシューティングを行い、詳細なログダンプが必要な場合を除き、は使用しないでください。</p>	null

パラメータ	説明	デフォルト
useREST	<p>ONTAP REST API を使用するためのブーリアンパラメータ。* テクニカルレビュー *</p> <p>useREST は、テクニカルレビューとして提供されています。テスト環境では、本番環境のワークロードでは推奨されません。に設定すると true`Astra Trident は、ONTAP REST APIを使用してバックエンドと通信します。この機能にはONTAP 9.11.1以降が必要です。また、使用するONTAP ログインロールにはへのアクセス権が必要です `ontap アプリケーション：これは事前定義されたによって満たされます vsadmin および cluster-admin ロール。</p> <p>useREST は、MetroCluster ではサポートされていません。</p>	false

#### ボリュームのプロビジョニング用のバックエンド構成オプション

これらのオプションを使用して、のデフォルトプロビジョニングを制御できます defaults 設定のセクション。例については、以下の設定例を参照してください。

パラメータ	説明	デフォルト
spaceAllocation	space-allocation for LUN のコマンドを指定します	正しいです
spaceReserve	スペースリザーベーションモード：「none」（シン）または「volume」（シック）	なし
snapshotPolicy	使用する Snapshot ポリシー	なし

パラメータ	説明	デフォルト
qosPolicy	<p>作成したボリュームに割り当てる QoS ポリシーグループ。ストレージプール / バックエンドごとに QOSPolicy または adaptiveQosPolicy のいずれかを選択します。</p> <p>Trident が Astra で QoS ポリシーグループを使用するには、ONTAP 9.8 以降が必要です。非共有の QoS ポリシーグループを使用して、各コンスチチュエントに個別にポリシーグループを適用することを推奨します。共有 QoS ポリシーグループにより、すべてのワークロードの合計スループットに対して上限が適用されます。</p>	「」
adaptiveQosPolicy	<p>アダプティブ QoS ポリシーグループ：作成したボリュームに割り当てます。ストレージプール / バックエンドごとに QOSPolicy または adaptiveQosPolicy のいずれかを選択します</p>	「」
snapshotReserve	Snapshot 「0」用にリザーブされているボリュームの割合	状況 snapshotPolicy は「none」、それ以外は「」です。
splitOnClone	作成時にクローンを親からスプリットします	いいえ
encryption	<p>新しいボリュームで NetApp Volume Encryption (NVE) を有効にします。デフォルトは false。このオプションを使用するには、クラスターで NVE のライセンスが設定され、有効になっている必要があります。</p> <p>NAE がバックエンドで有効になっている場合は、Astra Trident でプロビジョニングされたすべてのボリュームが NAE に有効になります。</p> <p>詳細については、以下を参照してください。"<a href="#">Astra Trident と NVE および NAE の相互運用性</a>"。</p>	いいえ
luksEncryption	LUKS 暗号化を有効にします。を参照してください " <a href="#">Linux Unified Key Setup (LUKS; 統合キーセットアップ)</a> を使用"。	""
securityStyle	新しいボリュームのセキュリティ形式	unix

パラメータ	説明	デフォルト
tieringPolicy	「none」を使用する階層化ポリシー	ONTAP 9.5 よりも前の SVM-DR 構成の「スナップショットのみ」

#### ボリュームプロビジョニングの例

デフォルトが定義されている例を次に示します。

```

---
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
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'

```



を使用して作成したすべてのボリューム ontap-san ドライバである Astra Trident が、FlexVol のメタデータに対応するために、さらに 10% の容量を追加 LUN は、ユーザが PVC で要求したサイズとまったく同じサイズでプロビジョニングされます。Astra Trident が FlexVol に 10% を追加（ONTAP で利用可能なサイズとして表示）ユーザには、要求した使用可能容量が割り当てられます。また、利用可能なスペースがフルに活用されていないかぎり、LUN が読み取り専用になることもありません。これは、ONTAP と SAN の経済性には該当しません。

を定義するバックエンドの場合 `snapshotReserve` Trident は、次のようにボリュームサイズを計算します。

```

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

```

1.1 は、Astra Trident の 10% の追加料金で、FlexVol のメタデータに対応します。の場合 snapshotReserve = 5%、PVC 要求 = 5GiB、ボリュームの合計サイズは 5.79GiB、使用可能なサイズは 5.5GiB です。。 volume show 次の例のような結果が表示されます。

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.

現在、既存のボリュームに対して新しい計算を行うには、サイズ変更だけを使用します。

### 最小限の設定例

次の例は、ほとんどのパラメータをデフォルトのままにする基本的な設定を示しています。これは、バックエンドを定義する最も簡単な方法です。



Amazon FSx on NetApp ONTAPとAstra Tridentを使用している場合は、IPアドレスではなく、LIFのDNS名を指定することを推奨します。

### ONTAP SANの最小構成の例

これは、ontap-san ドライバ。

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
username: vsadmin
password: <password>
```

### ONTAP SANエコノミーの最小構成の例

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
username: vsadmin
password: <password>
```

この基本的な設定例では、`clientCertificate`、`clientPrivateKey` および `trustedCACertificate`（信頼されたCAを使用している場合はオプション）が入力されます `backend.json` およびは、クライアント証明書、秘密鍵、信頼されたCA証明書のbase64エンコード値をそれぞれ取得します。

```
---
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
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

## 双方向CHAPの例

次の例では、useCHAP をに設定します true。

### ONTAP SAN CHAPの例

```
---
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
username: vsadmin
password: <password>
```

### ONTAP SANエコノミーCHAPの例

```
---
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
username: vsadmin
password: <password>
```

## 仮想プールを使用するバックエンドの例

これらのサンプルバックエンド定義ファイルでは、次のような特定のデフォルトがすべてのストレージプールに設定されています。spaceReserve 「なし」の場合は、spaceAllocation との誤り encryption 実行されます。仮想プールは、ストレージセクションで定義します。

Astra Tridentは、[Comments]フィールドにプロビジョニングラベルを設定します。FlexVol にコメントが設定されます。Astra Tridentは、プロビジョニング時に仮想プール上にあるすべてのラベルをストレージボリュームにコピーします。ストレージ管理者は、仮想プールごとにラベルを定義したり、ボリュームをラベルでグル

ープ化したりできます。

これらの例では、一部のストレージプールが独自の `spaceReserve`、`spaceAllocation` および `encryption` 値、および一部のプールはデフォルト値よりも優先されます。





```

---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSD6cNwxyz
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'

```

```
---
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
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'
- labels:
  department: legal
  creditpoints: '5000'
```

```
zone: us_east_1c
defaults:
  spaceAllocation: 'true'
  encryption: 'false'
```

バックエンドを **StorageClasses** にマッピングします

次のStorageClass定義は、[\[仮想プールを使用するバックエンドの例\]](#)。を使用する `parameters.selector` フィールドでは、各StorageClassがボリュームのホストに使用できる仮想プールを呼び出します。ボリュームには、選択した仮想プール内で定義された要素があります。

- 。 `protection-gold` StorageClassは、 `ontap-san` バックエンド：ゴールドレベルの保護を提供する唯一のプールです。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- 。 `protection-not-gold` StorageClassは、内の2番目と3番目の仮想プールにマッピングされます。 `ontap-san` バックエンド：これらは、ゴールド以外の保護レベルを提供する唯一のプールです。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- 。 `app-mysqldb` StorageClassは内の3番目の仮想プールにマッピングされます `ontap-san-economy` バックエンド：これは、`mysqldb`タイプアプリケーション用のストレージプール構成を提供する唯一のプールです。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- 。 protection-silver-creditpoints-20k StorageClassは内の2番目の仮想プールにマッピングされます ontap-san バックエンド：シルバーレベルの保護と20000クレジットポイントを提供する唯一のプールです。

```
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"
```

- 。 creditpoints-5k StorageClassは内の3番目の仮想プールにマッピングされます ontap-san バックエンドと内の4番目の仮想プール ontap-san-economy バックエンド：これらは、5000クレジットポイントを持つ唯一のプールオフリングです。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

Tridentが、どの仮想プールを選択するかを判断し、ストレージ要件を確実に満たすようにします。

## ONTAP NAS ドライバ

### ONTAP NASドライバの概要

ONTAP および Cloud Volumes ONTAP の NAS ドライバを使用した ONTAP バックエン

ドの設定について説明します。

## ONTAP NASドライバに関する重要な情報

Astra Controlは、で作成したボリュームに対して、シームレスな保護、ディザスタリカバリ、および移動（Kubernetesクラスター間でボリュームを移動）を提供します `ontap-nas`、`ontap-nas-flexgroup` および `ontap-san` ドライバ。を参照してください "[Astra Controlレプリケーションの前提条件](#)" を参照してください。

- を使用する必要があります `ontap-nas` データ保護、ディザスタリカバリ、モビリティを必要とする本番環境のワークロード向けのサービスです。
- 使用 `ontap-san-economy` 想定されるボリューム使用量がONTAP でサポートされる量よりも大幅に多い場合
- 使用 `ontap-nas-economy` 想定されるボリューム使用量が、ONTAP でサポートされるおおよよりも大幅に多い場合にのみ該当します `ontap-san-economy` ドライバは使用できません。
- 使用しないでください `ontap-nas-economy` データ保護、ディザスタリカバリ、モビリティのニーズが予想される場合。

## ユーザ権限

Tridentは、通常はを使用して、ONTAP 管理者またはSVM管理者のどちらかとして実行される必要があります `admin` クラスターユーザまたは `vsadmin` SVMユーザ、または同じロールを持つ別の名前のユーザ。

Amazon FSX for NetApp ONTAP 環境では、Astra Tridentは、クラスターを使用して、ONTAP 管理者またはSVM管理者のどちらかとして実行されるものと想定しています `fsxadmin` ユーザまたは `vsadmin` SVMユーザ、または同じロールを持つ別の名前のユーザ。。 `fsxadmin` このユーザは、クラスター管理者ユーザを限定的に置き換えるものです。



を使用する場合 `limitAggregateUsage` クラスター管理者権限が必要です。Amazon FSX for NetApp ONTAP をAstra Tridentとともに使用している場合は、を参照してください `limitAggregateUsage` パラメータはでは機能しません `vsadmin` および `fsxadmin` ユーザアカウント：このパラメータを指定すると設定処理は失敗します。

ONTAP内でTridentドライバが使用できる、より制限の厳しいロールを作成することは可能ですが、推奨しません。Trident の新リリースでは、多くの場合、考慮すべきAPI が追加で必要になるため、アップグレードが難しく、エラーも起こりやすくなります。

## ONTAP NASドライバを使用してバックエンドを設定する準備をします

ONTAP NASドライバでONTAPバックエンドを設定するための要件、認証オプション、およびエクスポートポリシーを理解します。

### 要件

- ONTAP バックエンドすべてに対して、Astra Trident が SVM に少なくとも 1 つのアグリゲートを割り当てておく必要があります。
- 複数のドライバを実行し、どちらか一方を参照するストレージクラスを作成できます。たとえば、を使用するGoldクラスを設定できます `ontap-nas` ドライバとを使用するBronzeクラス `ontap-nas-economy` 1つ。

- すべてのKubernetesワーカーノードに適切なNFSツールをインストールしておく必要があります。を参照してください ["こちらをご覧ください"](#) 詳細：
- Astra Tridentは、Windowsノードで実行されているポッドにマウントされたSMBボリュームのみをサポートを参照してください [SMBボリュームをプロビジョニングする準備をします](#) を参照してください。

## ONTAPバックエンドの認証

Astra Trident には、ONTAP バックエンドを認証する 2 つのモードがあります。

- credential based : 必要な権限を持つ ONTAP ユーザのユーザ名とパスワード。など、事前定義されたセキュリティログインロールを使用することを推奨します admin または vsadmin ONTAP のバージョンとの互換性を最大限に高めるため。
- 証明書ベース : Astra Trident は、バックエンドにインストールされた証明書を使用して ONTAP クラスタと通信することもできます。この場合、バックエンド定義には、Base64 でエンコードされたクライアント証明書、キー、および信頼された CA 証明書（推奨）が含まれている必要があります。

既存のバックエンドを更新して、クレデンシャルベースの方式と証明書ベースの方式を切り替えることができます。ただし、一度にサポートされる認証方法は1つだけです。別の認証方式に切り替えるには、バックエンド設定から既存の方式を削除する必要があります。



クレデンシャルと証明書の両方を\*指定しようとする、バックエンドの作成が失敗し、構成ファイルに複数の認証方法が指定されているというエラーが表示されます。

クレデンシャルベースの認証を有効にします

Trident が ONTAP バックエンドと通信するには、SVM を対象とした管理者またはクラスタを対象とした管理者のクレデンシャルが必要です。などの標準の事前定義されたロールを使用することを推奨します admin または vsadmin。これにより、今後のリリースの ONTAP との互換性が今後のリリースの Astra Trident で使用される機能 API が公開される可能性があります。カスタムのセキュリティログインロールは Astra Trident で作成して使用できますが、推奨されません。

バックエンド定義の例は次のようになります。

## 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
```

## JSON

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

バックエンド定義は、クレデンシャルがプレーンテキストで保存される唯一の場所であることに注意してください。バックエンドが作成されると、ユーザ名とパスワードが Base64 でエンコードされ、Kubernetes シークレットとして格納されます。クレデンシャルの知識が必要なのは、バックエンドの作成と更新だけです。この処理は管理者専用で、Kubernetes / ストレージ管理者が実行します。

証明書ベースの認証を有効にします

新規または既存のバックエンドは証明書を使用して ONTAP バックエンドと通信できます。バックエンド定義には 3 つのパラメータが必要です。

- `clientCertificate` : Base64 でエンコードされたクライアント証明書の値。
- `clientPrivateKey` : Base64 でエンコードされた、関連付けられた秘密鍵の値。
- `trustedCACertificate`: 信頼された CA 証明書の Base64 エンコード値。信頼された CA を使用する場合は、このパラメータを指定する必要があります。信頼された CA が使用されていない場合は無視してかまいません。

一般的なワークフローは次の手順で構成されます。

### 手順

1. クライアント証明書とキーを生成します。生成時に、ONTAP ユーザとして認証するように Common



Name (CN ; 共通名) を設定します。

```
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. 信頼された CA 証明書を ONTAP クラスタに追加します。この処理は、ストレージ管理者がすでに行っている可能性があります。信頼できる CA が使用されていない場合は無視します。

```
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 クラスタにクライアント証明書とキーをインストールします (手順 1)。

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

4. ONTAP セキュリティログインロールでサポートされていることを確認する cert 認証方式。

```
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. 生成された証明書を使用して認証をテスト ONTAP 管理 LIF > と <vserver name> は、管理 LIF の IP アドレスおよび SVM 名に置き換えてください。LIF のサービスポリシーがに設定されていることを確認する必要があります 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. Base64 で証明書、キー、および信頼された CA 証明書をエンコードする。

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

## 7. 前の手順で得た値を使用してバックエンドを作成します。

```
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...Vaaalllluuueeeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "storagePrefix": "myPrefix_"
}

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

NAME	STORAGE DRIVER	UUID
NasBackend	ontap-nas	98e19b74-aec7-4a3d-8dcf-128e5033b214

認証方法を更新するか、クレデンシャルをローテーションして

既存のバックエンドを更新して、別の認証方法を使用したり、クレデンシャルをローテーションしたりできます。これはどちらの方法でも機能します。ユーザ名とパスワードを使用するバックエンドは証明書を使用するように更新できますが、証明書を使用するバックエンドはユーザ名とパスワードに基づいて更新できます。これを行うには、既存の認証方法を削除して、新しい認証方法を追加する必要があります。次に、更新されたbackend.jsonファイルに必要なパラメータが含まれたものを使用して実行します 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 |
+-----+-----+-----+
+-----+-----+

```



パスワードのローテーションを実行する際には、ストレージ管理者が最初に ONTAP でユーザのパスワードを更新する必要があります。この後にバックエンドアップデートが続きます。証明書のローテーションを実行する際に、複数の証明書をユーザに追加することができます。その後、バックエンドが更新されて新しい証明書が使用されるようになります。この証明書に続く古い証明書は、ONTAP クラスタから削除できます。

バックエンドを更新しても、すでに作成されているボリュームへのアクセスは中断されず、その後のボリューム接続にも影響しません。バックエンドの更新が成功した場合、Astra Trident が ONTAP バックエンドと通信し、以降のボリューム処理を処理できることを示しています。

## NFS エクスポートポリシーを管理します

Astra Trident は、NFS エクスポートポリシーを使用して、プロビジョニングするボリュームへのアクセスを制御します。

Astra Trident には、エクスポートポリシーを使用する際に次の 2 つのオプションがあります。

- Astra Trident は、エクスポートポリシー自体を動的に管理できます。このモードでは、許容可能な IP アドレスを表す CIDR ブロックのリストをストレージ管理者が指定します。Astra Trident は、この範囲に含まれるノード IP をエクスポートポリシーに自動的に追加します。または、CIDRs が指定されていない場

合は、ノード上で検出されたグローバルスコープのユニキャスト IP がエクスポートポリシーに追加されます。

- ストレージ管理者は、エクスポートポリシーを作成したり、ルールを手動で追加したりできます。構成に別のエクスポートポリシー名を指定しないと、Astra Trident はデフォルトのエクスポートポリシーを使用します。

#### エクスポートポリシーを動的に管理

CSI Trident の 20.04 リリースでは、ONTAP バックエンドのエクスポートポリシーを動的に管理できます。これにより、ストレージ管理者は、明示的なルールを手動で定義するのではなく、ワーカーノードの IP で許容されるアドレススペースを指定できます。エクスポートポリシーの管理が大幅に簡易化され、エクスポートポリシーを変更しても、ストレージクラスタに対する手動の操作は不要になります。さらに、この方法を使用すると、ストレージクラスタへのアクセスを指定した範囲内の IP を持つワーカーノードだけに制限できるため、きめ細かい管理が可能になります。



エクスポートポリシーの動的管理は CSI Trident でのみ使用できます。ワーカーノードが NAT 処理されていないことを確認することが重要です。

#### 例

2 つの設定オプションを使用する必要があります。バックエンド定義の例を次に示します。

```
---
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
```



この機能を使用する場合は、SVMのルートジャンクションに、ノードのCIDRブロックを許可するエクスポートルール（デフォルトのエクスポートポリシーなど）を含む事前に作成したエクスポートポリシーがあることを確認する必要があります。ネットアップが推奨する、Astra Trident 専用のベストプラクティスを常に守ってください。

ここでは、上記の例を使用してこの機能がどのように動作するかについて説明します。

- `autoExportPolicy` がに設定されます `true`。これは、Astra Tridentがのエクスポートポリシーを作成することを示します `svm1` SVMで、を使用してルールの追加と削除を処理します `autoExportCIDRs` アドレスブロック。たとえば、UUID 403b5326-842-40dB-96d0-d83fb3f4daecのバックエンドです `autoExportPolicy` をに設定します `true` という名前のエクスポートポリシーを作成します `trident-403b5326-8482-40db-96d0-d83fb3f4daec` 指定します。
- `autoExportCIDRs` アドレスブロックのリストが含まれます。このフィールドは省略可能で、デフォルト値は `["0.0.0/0", ":::/0"]` です。定義されていない場合は、Astra Trident が、ワーカーノードで検出された

すべてのグローバルにスコープ指定されたユニキャストアドレスを追加します。

この例では、を使用しています 192.168.0.0/24 アドレススペースが指定されています。このアドレス範囲に含まれる Kubernetes ノードの IP が、Astra Trident が作成するエクスポートポリシーに追加されることを示します。Astra Tridentは、実行されているノードを登録すると、ノードのIPアドレスを取得し、で指定されたアドレスブロックと照合してチェックします autoExportCIDRs。IP をフィルタリングすると、Trident が検出したクライアント IP のエクスポートポリシールールを作成し、特定したノードごとに 1 つのルールが設定されます。

更新できます autoExportPolicy および autoExportCIDRs バックエンドを作成したあとのバックエンドの場合自動的に管理されるバックエンドに新しい CIDRs を追加したり、既存の CIDRs を削除したりできます。CIDRs を削除する際は、既存の接続が切断されないように注意してください。無効にすることもできます autoExportPolicy をバックエンドに追加し、手動で作成したエクスポートポリシーに戻します。これにはを設定する必要があります exportPolicy バックエンド構成のパラメータ。

Astra Tridentがバックエンドを作成または更新したら、を使用してバックエンドを確認できます tridentctl または対応する 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
```

Kubernetes クラスタにノードを追加して Astra Trident コントローラに登録すると、既存のバックエンドのエクスポートポリシーが更新されます（に指定されたアドレス範囲に含まれる場合） autoExportCIDRs バックエンドの場合）をクリックします。

ノードを削除すると、Astra Trident はオンラインのすべてのバックエンドをチェックして、そのノードのアクセスルールを削除します。管理対象のバックエンドのエクスポートポリシーからこのノード IP を削除することで、Astra Trident は、この IP がクラスタ内の新しいノードによって再利用されないかぎり、不正なマウントを防止します。

以前のバックエンドの場合は、を使用してバックエンドを更新します tridentctl update backend で

は、Astra Tridentがエクスポートポリシーを自動的に管理します。これにより、バックエンドの UUID のあとにという名前の新しいエクスポートポリシーが作成され、バックエンドに存在するボリュームは、新しく作成したエクスポートポリシーを使用して、再びマウントします。



自動管理されたエクスポートポリシーを使用してバックエンドを削除すると、動的に作成されたエクスポートポリシーが削除されます。バックエンドが再作成されると、そのバックエンドは新しいバックエンドとして扱われ、新しいエクスポートポリシーが作成されます。

ライブノードの IP アドレスが更新された場合は、ノード上の Astra Trident ポッドを再起動する必要があります。Trident が管理するバックエンドのエクスポートポリシーを更新して、この IP の変更を反映させます。

## SMBボリュームをプロビジョニングする準備をします

多少の準備が必要な場合は、次のツールを使用してSMBボリュームをプロビジョニングできます。ontap-nas ドライバ。



を作成するには、SVMでNFSプロトコルとSMB / CIFSプロトコルの両方を設定する必要があります。ontap-nas-economy オンプレミスのONTAP 用のSMBボリューム。これらのプロトコルのいずれかを設定しないと、原因 SMBボリュームの作成が失敗します。

作業を開始する前に

SMBボリュームをプロビジョニングする前に、以下を準備しておく必要があります。

- Linuxコントローラノードと少なくとも1つのWindowsワーカーノードでWindows Server 2019を実行しているKubernetesクラスター。Astra Tridentは、Windowsノードで実行されているポッドにマウントされたSMBボリュームのみをサポート
- Active Directoryのクレデンシャルを含むAstra Tridentのシークレットが少なくとも1つ必要です。シークレットを生成します smbcreds :

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

- Windowsサービスとして設定されたCSIプロキシ。を設定します `csi-proxy` を参照してください "[GitHub: CSIプロキシ](#)" または "[GitHub: Windows向けCSIプロキシ](#)" Windowsで実行されているKubernetesノードの場合。

手順

1. オンプレミスのONTAPの場合は、必要に応じてSMB共有を作成するか、Astra TridentでSMB共有を作成できます。



Amazon FSx for ONTAPにはSMB共有が必要です。

SMB管理共有は、のいずれかの方法で作成できます "[Microsoft管理コンソール](#)" 共有フォルダスナップインまたはONTAP CLIを使用します。ONTAP CLIを使用してSMB共有を作成するには、次の手順を実行します

- a. 必要に応じて、共有のディレクトリパス構造を作成します。

。 `vserver cifs share create` コマンドは、共有の作成時に `-path` オプションで指定されているパスを確認します。指定したパスが存在しない場合、コマンドは失敗します。

- b. 指定したSVMに関連付けられているSMB共有を作成します。

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

- c. 共有が作成されたことを確認します。

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



を参照してください ["SMB 共有を作成"](#) 詳細については、

2. バックエンドを作成する際に、SMBボリュームを指定するように次の項目を設定する必要があります。ONTAP バックエンド構成オプションのすべてのFSXについては、を参照してください ["FSX \(ONTAP の構成オプションと例\)"](#)。

パラメータ	説明	例
<code>smbShare</code> Microsoft管理コンソールまたはONTAP CLIを使用して作成されたSMB共有の名前、Astra TridentでSMB共有を作成できる名前、ボリュームへの共有アクセスを禁止する場合はパラメータを空白のままにすることができます。  オンプレミスのONTAPでは、このパラメータはオプションです。  このパラメータはAmazon FSx for ONTAPバックエンドで必須であり、空にすることはできません。	<code>smb-share</code>	<code>nasType</code>
をに設定する必要があります <b>smb</b> . <code>null</code> の場合、デフォルトはです <code>nfs</code> 。	<code>smb</code>	<code>securityStyle</code>
新しいボリュームのセキュリティ形式。  をに設定する必要があります <b>ntfs</b> または <b>mixed SMB</b> ボリューム	<code>ntfs</code> または <code>mixed SMB</code> ボリュームの場合	<code>unixPermissions</code>

## ONTAP NASの設定オプションと例

Astra Tridentのインストール環境でONTAP NASドライバを作成して使用方法をご確認ください。このセクションでは、バックエンド構成の例と、バックエンドをストレージクラスにマッピングする方法を詳しく説明します。

### バックエンド構成オプション

バックエンド設定オプションについては、次の表を参照してください。

パラメータ	説明	デフォルト
version		常に 1
storageDriverName	ストレージドライバの名前	「ONTAP-NAS」、「ONTAP-NAS-エコノミー」、「ONTAP-NAS-flexgroup」、「ONTAP-SAN」、「ONTAP-SAN-エコノミー」
backendName	カスタム名またはストレージバックエンド	ドライバ名 + "_" + データ LIF
managementLIF	<p>クラスタ管理 LIF または SVM 管理 LIF の IP アドレス</p> <p>MetroClusterのシームレスなスイッチオーバーを実現するには、SVM 管理LIFを指定する必要があります。</p> <p>Fully Qualified Domain Name (FQDN；完全修飾ドメイン名) を指定できます。</p> <p>を使用してAstra Tridentをインストールした場合、IPv6アドレスを使用するようにを設定できます --use-ipv6 フラグ。IPv6アドレスは、[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]などの角かっこで定義する必要があります。</p>	「10.0.0.1」、「[2001:1234:abcd::fefe]」



パラメータ	説明	デフォルト
dataLIF	<p>プロトコル LIF の IP アドレス。</p> <p>を指定することを推奨します dataLIF。指定しない場合は、Astra TridentがSVMからデータLIFを取得します。NFSマウント処理に使用するFully Qualified Domain Name（FQDN；完全修飾ドメイン名）を指定して、ラウンドロビンDNSを作成して複数のデータLIF間で負荷を分散することができます。</p> <p>初期設定後に変更できます。を参照してください。</p> <p>を使用してAstra Tridentをインストールした場合、IPv6アドレスを使用するようにを設定できます --use-ipv6 フラグ。IPv6アドレスは、[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]などの角かっこで定義する必要があります。</p>	指定されたアドレス、または指定されていない場合はSVMから取得されるアドレス（非推奨）
autoExportPolicy	<p>エクスポートポリシーの自動作成と更新を有効にします[ブーリアン]。</p> <p>を使用する autoExportPolicy および autoExportCIDRs ネットアップのAstra Tridentなら、エクスポートポリシーを自動的に管理できます。</p>	いいえ
autoExportCIDRs	<p>KubernetesのノードIPをいつからフィルタリングするかを示すCIDRsのリスト autoExportPolicy が有効になります。</p> <p>を使用する autoExportPolicy および autoExportCIDRs ネットアップのAstra Tridentなら、エクスポートポリシーを自動的に管理できます。</p>	[0.0.0.0/0]、[::/0]
labels	ボリュームに適用する任意のJSON形式のラベルのセット	「」
clientCertificate	クライアント証明書の Base64 エンコード値。証明書ベースの認証に使用されます	「」

パラメータ	説明	デフォルト
clientPrivateKey	クライアント秘密鍵の Base64 エンコード値。証明書ベースの認証に使用されます	「」
trustedCACertificate	信頼された CA 証明書の Base64 エンコード値。任意。証明書ベースの認証に使用されます	「」
username	クラスタ / SVM に接続するためのユーザ名。クレデンシャルベースの認証に使用されます	
password	クラスタ / SVM に接続するためのパスワード。クレデンシャルベースの認証に使用されます	
svm	使用する Storage Virtual Machine	SVMの場合に生成されます managementLIF を指定します
storagePrefix	SVM で新しいボリュームをプロビジョニングする際に使用するプレフィックスを指定します。設定後に更新することはできません	Trident
limitAggregateUsage	使用率がこの割合を超えている場合は、プロビジョニングが失敗します。  * Amazon FSX for ONTAP * には適用されません	""（デフォルトでは適用されません）
limitVolumeSize	要求されたボリュームサイズがこの値を超えている場合、プロビジョニングが失敗します。	""（デフォルトでは適用されません）
limitVolumeSize	要求されたボリュームサイズがこの値を超えている場合、プロビジョニングが失敗します。  また、qtreeおよびLUN用に管理するボリュームの最大サイズも制限します qtreesPerFlexvol オプションを使用すると、FlexVol あたりの最大qtree数をカスタマイズできます。	""（デフォルトでは適用されません）
lunsPerFlexvol	FlexVol あたりの最大 LUN 数。有効な範囲は 50、200 です	100

パラメータ	説明	デフォルト
debugTraceFlags	<p>トラブルシューティング時に使用するデバッグフラグ。例：{"API" : false、"method" : true}</p> <p>使用しないでください debugTraceFlags トラブルシューティングを実行していて、詳細なログダンプが必要な場合を除きます。</p>	null
nasType	<p>NFSボリュームまたはSMBボリュームの作成を設定</p> <p>オプションはです nfs、 smb または null。nullに設定すると、デフォルトでNFSボリュームが使用されます。</p>	nfs
nfsMountOptions	<p>NFSマウントオプションをカンマで区切ったリスト。</p> <p>Kubernetes永続ボリュームのマウントオプションは通常はストレージクラスで指定されますが、ストレージクラスでマウントオプションが指定されていない場合、Astra Tridentはストレージバックエンドの構成ファイルで指定されているマウントオプションを使用します。</p> <p>ストレージクラスや構成ファイルにマウントオプションが指定されていない場合、Astra Tridentは関連付けられた永続的ボリュームにマウントオプションを設定しません。</p>	「」
qtreesPerFlexvol	FlexVol あたりの最大 qtree 数。有効な範囲は [50、 300] です。	「200」

パラメータ	説明	デフォルト
smbShare	<p>Microsoft管理コンソールまたはONTAP CLIを使用して作成されたSMB共有の名前、Astra TridentでSMB共有を作成できる名前、ボリュームへの共有アクセスを禁止する場合はパラメータを空白のままにすることができます。</p> <p>オンプレミスのONTAPでは、このパラメータはオプションです。</p> <p>このパラメータはAmazon FSx for ONTAPバックエンドで必須であり、空にすることはできません。</p>	smb-share
useREST	<p>ONTAP REST API を使用するためのブーリアンパラメータ。* テクニカルプレビュー *</p> <p>useREST は、テクニカルプレビューとして提供されています。テスト環境では、本番環境のワークロードでは推奨されません。に設定すると true`Astra Trident は、ONTAP REST APIを使用してバックエンドと通信します。この機能にはONTAP 9.11.1以降が必要です。また、使用するONTAP ログインロールにはへのアクセス権が必要です `ontap アプリケーション：これは事前定義されたによって満たされます vsadmin および cluster-admin ロール。</p> <p>useREST は、MetroCluster ではサポートされていません。</p>	いいえ

#### ボリュームのプロビジョニング用のバックエンド構成オプション

これらのオプションを使用して、のデフォルトプロビジョニングを制御できます defaults 設定のセクション。例については、以下の設定例を参照してください。

パラメータ	説明	デフォルト
spaceAllocation	space-allocation for LUN のコマンドを指定します	正しいです
spaceReserve	スペースリザーベーションモード：「none」（シン）または「volume」（シック）	なし
snapshotPolicy	使用する Snapshot ポリシー	なし

パラメータ	説明	デフォルト
qosPolicy	作成したボリュームに割り当てる QoS ポリシーグループ。ストレージプール / バックエンドごとに QOSPolicy または adaptiveQosPolicy のいずれかを選択します	「」
adaptiveQosPolicy	アダプティブ QoS ポリシーグループ：作成したボリュームに割り当てます。ストレージプール / バックエンドごとに QOSPolicy または adaptiveQosPolicy のいずれかを選択します。  経済性に影響する ONTAP - NAS ではサポートされません。	「」
snapshotReserve	Snapshot 「0」用にリザーブされているボリュームの割合	状況 snapshotPolicy は「none」、それ以外は「」です。
splitOnClone	作成時にクローンを親からスプリットします	いいえ
encryption	新しいボリュームでNetApp Volume Encryption (NVE) を有効にします。デフォルトは false。このオプションを使用するには、クラスタで NVE のライセンスが設定され、有効になっている必要があります。  NAEがバックエンドで有効になっている場合は、Astra TridentでプロビジョニングされたすべてのボリュームがNAEに有効になります。  詳細については、以下を参照してください。 <a href="#">"Astra TridentとNVEおよびNAEの相互運用性"</a> 。	いいえ
tieringPolicy	「none」を使用する階層化ポリシー	ONTAP 9.5 よりも前の SVM-DR 構成の「スナップショットのみ」
unixPermissions	新しいボリュームのモード	NFSボリュームの場合は「777」、SMBボリュームの場合は空（該当なし）
snapshotDir	の表示/非表示を制御します .snapshot ディレクトリ	いいえ
exportPolicy	使用するエクスポートポリシー	デフォルト

パラメータ	説明	デフォルト
securityStyle	<p>新しいボリュームのセキュリティ形式。</p> <p>NFSのサポート mixed および unix セキュリティ形式</p> <p>SMBはをサポートします mixed および ntfs セキュリティ形式</p>	<p>NFSのデフォルトはです unix。</p> <p>SMBのデフォルト： ntfs。</p>



Trident が Astra で QoS ポリシーグループを使用するには、ONTAP 9.8 以降が必要です。共有されない QoS ポリシーグループを使用して、各コンスチチュエントに個別にポリシーグループを適用することを推奨します。共有 QoS ポリシーグループにより、すべてのワークロードの合計スループットに対して上限が適用されます。

## ボリュームプロビジョニングの例

デフォルトが定義されている例を次に示します。

```
---
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'
```

の場合 ontap-nas および ontap-nas-flexgroups`Tridentが新たに計算を使用して、FlexVol のサイズがsnapshotReserveの割合とPVCで正しく設定されていることを確認するようになりました。ユーザが

PVC を要求すると、Astra Trident は、新しい計算を使用して、より多くのスペースを持つ元の FlexVol を作成します。この計算により、ユーザは要求された PVC 内の書き込み可能なスペースを受信し、要求されたスペースよりも少ないスペースを確保できます。v21.07 より前のバージョンでは、ユーザが PVC を要求すると（5GiB など）、snapshotReserve が 50% に設定されている場合、書き込み可能なスペースは 2.5GiB のみになります。これは、ユーザが要求したボリューム全体とがであるためです。snapshotReserve には、その割合を指定します。Trident 21.07では、ユーザが要求したものが書き込み可能なスペースであり、Astra Tridentが定義します snapshotReserve ボリューム全体に対する割合として示されます。には適用されません ontap-nas-economy。この機能の仕組みについては、次の例を参照してください。

計算は次のとおりです。

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

snapshotReserve = 50%、PVC 要求 = 5GiB の場合、ボリュームの合計サイズは  $2/0.5 = 10\text{GiB}$  であり、使用可能なサイズは 5GiB であり、これが PVC 要求で要求されたサイズです。。 volume show 次の例のような結果が表示されます。

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.

以前のインストールからの既存のバックエンドは、Astra Trident のアップグレード時に前述のようにボリュームをプロビジョニングします。アップグレード前に作成したボリュームについては、変更が反映されるようにボリュームのサイズを変更する必要があります。たとえば、が搭載されている2GiB PVCなどです snapshotReserve=50 以前は、書き込み可能なスペースが1GiBのボリュームが作成されていました。たとえば、ボリュームのサイズを 3GiB に変更すると、アプリケーションの書き込み可能なスペースが 6GiB のボリュームで 3GiB になります。

### 最小限の設定例

次の例は、ほとんどのパラメータをデフォルトのままにする基本的な設定を示しています。これは、バックエンドを定義する最も簡単な方法です。



ネットアップ ONTAP で Trident を使用している場合は、IP アドレスではなく LIF の DNS 名を指定することを推奨します。

#### 最小限の構成 `<code>ontap-nas-economy</code>`

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

#### 最小限の構成 `<code>ontap-nas-flexgroup</code>`

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

#### SMBボリュームの最小限の設定

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
nasType: smb
securityStyle: ntfs
unixPermissions: ""
dataLIF: 10.0.0.2
svm: svm_nfs
username: vsadmin
password: password
```



## 証明書ベースの認証

これは、バックエンドの最小限の設定例です。clientCertificate、clientPrivateKey`および`trustedCACertificate（信頼されたCAを使用している場合はオプション）が入力されます backend.json およびは、クライアント証明書、秘密鍵、信頼されたCA証明書のbase64エンコード値をそれぞれ取得します。

```
---
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_
```

## 自動エクスポートポリシー

この例は、動的なエクスポートポリシーを使用してエクスポートポリシーを自動的に作成および管理するように Astra Trident に指示する方法を示しています。これは、でも同様に機能します ontap-nas-economy および ontap-nas-flexgroup ドライバ。

```
---
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
```

## IPv6アドレスを使用している

この例は、を示しています managementLIF IPv6アドレスを使用している。

```
---
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
```

## SMBボリュームを使用したAmazon FSx for ONTAP

。 smbShare SMBボリュームを使用するFSx for ONTAPの場合、パラメータは必須です。

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

## 仮想プールを使用するバックエンドの例

以下に示すサンプルのバックエンド定義ファイルでは、次のような特定のデフォルトがすべてのストレージプールに設定されています。 spaceReserve 「なし」 の場合は、 spaceAllocation との誤り encryption 実行されます。仮想プールは、ストレージセクションで定義します。

Astra Tridentは、[Comments]フィールドにプロビジョニングラベルを設定します。コメントは次のFlexVolに設定されています： ontap-nas またはFlexGroup for ontap-nas-flexgroup。 Astra Tridentは、プロビジョニング時に仮想プール上にあるすべてのラベルをストレージボリュームにコピーします。ストレージ管理者は、仮想プールごとにラベルを定義したり、ボリュームをラベルでグループ化したりできます。

これらの例では、一部のストレージプールが独自の `spaceReserve`、`spaceAllocation` および `encryption` 値、および一部のプールはデフォルト値よりも優先されます。

```

---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
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:
  department: legal
  creditpoints: '5000'
  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
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
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:
  protection: bronze
  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'
```

バックエンドを **StorageClasses** にマッピングします

次のStorageClass定義は、を参照してください。[[仮想プールを使用するバックエンドの例](#)]。を使用する `parameters.selector` フィールドでは、各StorageClassがボリュームのホストに使用できる仮想プールを呼び出します。ボリュームには、選択した仮想プール内で定義された要素があります。

- 。 `protection-gold` StorageClassは、 `ontap-nas-flexgroup` バックエンド：ゴールドレベルの保護を提供する唯一のプールです。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
```

- 。 `protection-not-gold` StorageClassは、内の3番目と4番目の仮想プールにマッピングされます。 `ontap-nas-flexgroup` バックエンド：金色以外の保護レベルを提供する唯一のプールです。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
```

- 。 `app-mysqldb` StorageClassは内の4番目の仮想プールにマッピングされます。 `ontap-nas` バックエンド：これは、`mysqldb`タイプアプリ用のストレージプール構成を提供する唯一のプールです。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
```

- [t] protection-silver-creditpoints-20k StorageClassは、ontap-nas-flexgroup バックエンド：シルバーレベルの保護と20000クレジットポイントを提供する唯一のプールです。

```
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"
```

- 。 creditpoints-5k StorageClassは、ontap-nas バックエンドと内の2番目の仮想プール ontap-nas-economy バックエンド：これらは、5000クレジットポイントを持つ唯一のプールオフファリングです。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"
```

Tridentが、どの仮想プールを選択するかを判断し、ストレージ要件を確実に満たすようにします。

#### 更新 dataLIF 初期設定後

初期設定後にデータLIFを変更するには、次のコマンドを実行して、更新されたデータLIFを新しいバックエンドJSONファイルに指定します。

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



PVCが1つ以上のポッドに接続されている場合は、対応するすべてのポッドを停止してから、新しいデータLIFを有効にするために稼働状態に戻す必要があります。

## NetApp ONTAP 対応の Amazon FSX

### Amazon FSX for NetApp ONTAP で Astra Trident を使用

"NetApp ONTAP 対応の Amazon FSX" は、NetApp ONTAP ストレージオペレーティングシステムを基盤とするファイルシステムの起動や実行を可能にする、フルマネージドのAWSサービスです。FSX for ONTAP を使用すると、使い慣れたネットアップの機能、パフォーマンス、管理機能を活用しながら、AWSにデータを格納するためのシンプルさ、即応性、セキュリティ、拡張性を活用できます。FSX for ONTAP は、ONTAP ファイルシステムの機能と管理APIをサポートしています。

#### 概要

ファイルシステムは、オンプレミスの ONTAP クラスタに似た、Amazon FSX のプライマリリソースです。各 SVM 内には、ファイルとフォルダをファイルシステムに格納するデータコンテナである 1 つ以上のボリュームを作成できます。Amazon FSX for NetApp ONTAP を使用すると、Data ONTAP はクラウド内の管理対象ファイルシステムとして提供されます。新しいファイルシステムのタイプは \* NetApp ONTAP \* です。

Amazon Elastic Kubernetes Service (EKS) で実行されている Astra Trident と Amazon FSX for NetApp ONTAP を使用すると、ONTAP がサポートするブロックボリュームとファイル永続ボリュームを確実にプロビジョニングできます。

NetApp ONTAP 用の Amazon FSX では、を使用します **"FabricPool"** ストレージ階層を管理します。データへのアクセス頻度に基づいて階層にデータを格納することができます。

#### 考慮事項

- SMBボリューム：
  - SMBボリュームは、を使用してサポートされます `ontap-nas` ドライバーのみ。
  - Astra Tridentは、Windowsノードで実行されているポッドにマウントされたSMBボリュームのみをサポート
- 自動バックアップが有効になっているAmazon FSXファイルシステムで作成されたボリュームはTridentで削除できません。PVC を削除するには、PV と ONTAP ボリュームの FSX を手動で削除する必要があります。この問題を回避するには、次の手順
  - ONTAP ファイル・システム用の FSX を作成する場合は **'Quick create'** を使用しないでくださいクイック作成ワークフローでは、自動バックアップが有効になり、オプトアウトオプションはありません。
  - **Standard create** を使用する場合は、自動バックアップを無効にしてください。自動バックアップを無効にすると、Trident は手動操作なしでボリュームを正常に削除できます。

## ▼ Backup and maintenance - *optional*

### Daily automatic backup [Info](#)

Amazon FSx can protect your data through daily backups

☐ Enabled

☒ Disabled

## ドライバ

次のドライバを使用して、Astra TridentをAmazon FSX for NetApp ONTAP と統合できます。

- `ontap-san`：プロビジョニングされる各PVは、NetApp ONTAP ボリューム用に独自のAmazon FSX内にあるLUNです。
- `ontap-san-economy`：プロビジョニングされる各PVは、Amazon FSXあたり、NetApp ONTAP ボリューム用に構成可能なLUN数を持つLUNです。
- `ontap-nas`：プロビジョニングされた各PVは、NetApp ONTAP ボリュームのAmazon FSX全体です。
- `ontap-nas-economy`：プロビジョニングされる各PVはqtreeで、NetApp ONTAP ボリュームのAmazon FSXごとに設定可能な数のqtreeがあります。
- `ontap-nas-flexgroup`：プロビジョニングされた各PVは、NetApp ONTAP FlexGroup ボリュームのAmazon FSX全体です。

ドライバの詳細については、を参照してください ["ONTAP ドライバ"](#)。

## 認証

Astra Tridentは、2種類の認証モードを提供します。

- 証明書ベース：Astra Trident は、SVM にインストールされている証明書を使用して、FSX ファイルシステムの SVM と通信します。
- クレデンシャルベース：を使用できます `fsxadmin` ユーザが自身のファイルシステムまたはに割り当てられます `vsadmin` ユーザがSVM用に設定します。



Astra Tridentは `vsadmin` SVMユーザまたは同じロールを持つ別の名前のユーザ。NetApp ONTAP 対応のAmazon FSXには、が搭載されています `fsxadmin` ONTAP を限定的に交換するユーザ `admin` クラスタユーザ：を使用することを強く推奨します `vsadmin` ネットアップが実現します。

証明書ベースの方法と証明書ベースの方法を切り替えるために、バックエンドを更新できます。ただし、\*クレデンシャルと\*証明書を入力しようとすると、バックエンドの作成に失敗します。別の認証方式に切り替えるには、バックエンド設定から既存の方式を削除する必要があります。

認証を有効にする方法の詳細については、使用しているドライバタイプの認証を参照してください。

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

詳細については、こちらをご覧ください

- ["Amazon FSX for NetApp ONTAP のドキュメント"](#)
- ["Amazon FSX for NetApp ONTAP に関するブログ記事です"](#)

## NetApp ONTAP 向けAmazon FSXを統合します

Amazon Elastic Kubernetes Service (EKS) で実行されているKubernetesクラスターが、ONTAP によってサポートされるブロックおよびファイルの永続ボリュームをプロビジョニングできるように、Amazon ONTAP ファイルシステム用のAmazon FSXをAstra Tridentに統合することができます。

### 要件

に加えて ["Astra Trident の要件"](#)FSX for ONTAP とAstra Tridentを統合するには、次のものがが必要です。

- 既存のAmazon EKSクラスターまたはを使用する自己管理型Kubernetesクラスター `kubectl` インストール済み。
- クラスターのワーカーノードから到達可能な既存のAmazon FSx for NetApp ONTAPファイルシステムおよびStorage Virtual Machine (SVM) 。
- 準備されているワーカーノード ["NFSまたはiSCSI"](#)。



Amazon LinuxおよびUbuntuで必要なノードの準備手順を実行します ["Amazon Machine Images の略"](#) (AMIS) EKS の AMI タイプに応じて異なります。

- Astra Tridentは、Windowsノードで実行されているポッドにマウントされたSMBボリュームのみをサポートを参照してください [SMBボリュームをプロビジョニングする準備をします](#) を参照してください。

## ONTAP SANとNASドライバの統合



SMBボリュームについて設定する場合は、を参照してください [SMBボリュームをプロビジョニングする準備をします](#) バックエンドを作成する前に。

### 手順

1. のいずれかを使用してAstra Tridentを導入 ["導入方法"](#)。
2. SVM管理LIFのDNS名を収集します。たとえば、AWS CLIを使用してを検索します `DNSName` の下のエントリ `Endpoints` → `Management` 次のコマンドを実行した後：

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

3. 用の証明書を作成してインストールします ["NASバックエンド認証"](#) または ["SANバックエンド認証"](#)。



ファイルシステムにアクセスできる任意の場所から SSH を使用して、ファイルシステムにログイン（証明書をインストールする場合など）できます。を使用します `fsxadmin user`、ファイルシステムの作成時に設定したパスワード、およびの管理DNS名 `aws fsx describe-file-systems`。

4. 次の例に示すように、証明書と管理 LIF の DNS 名を使用してバックエンドファイルを作成します。

#### 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
```

#### JSON

```
{
  "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"
}
```

バックエンドの作成については、次のリンクを参照してください。

- ["ONTAP NASドライバを使用したバックエンドの設定"](#)
- ["バックエンドに ONTAP SAN ドライバを設定します"](#)

#### 結果

導入後、を作成できます ["ストレージクラスを定義してボリュームをプロビジョニングし、ポッドでボリュームをマウント"](#)。

## SMBボリュームをプロビジョニングする準備をします

を使用してSMBボリュームをプロビジョニングできます `ontap-nas` ドライバ。をクリックしてください [ONTAP SANとNASドライバの統合](#) 次の手順を実行します。

作業を開始する前に

SMBボリュームをプロビジョニングする前に `ontap-nas` ドライバー、あなたは以下を持っている必要があります。

- Linuxコントローラノードと少なくとも1つのWindowsワーカーノードでWindows Server 2019を実行しているKubernetesクラスター。Astra Tridentは、Windowsノードで実行されているポッドにマウントされたSMBボリュームのみをサポート
- Active Directoryのクレデンシャルを含むAstra Tridentのシークレットが少なくとも1つ必要です。シークレットを生成します `smbcreds` :

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

- Windowsサービスとして設定されたCSIプロキシ。を設定します `'csi-proxy'` を参照してください ["GitHub: CSIプロキシ"](#) または ["GitHub: Windows向けCSIプロキシ"](#) Windowsで実行されているKubernetesノードの場合。

## 手順

1. SMB共有を作成SMB管理共有は、のいずれかの方法で作成できます ["Microsoft管理コンソール"](#) 共有フォルダスナップインまたはONTAP CLIを使用します。ONTAP CLIを使用してSMB共有を作成するには、次の手順を実行します

- a. 必要に応じて、共有のディレクトリパス構造を作成します。

。 `vserver cifs share create` コマンドは、共有の作成時に `-path` オプションで指定されているパスを確認します。指定したパスが存在しない場合、コマンドは失敗します。

- b. 指定したSVMに関連付けられているSMB共有を作成します。

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

- c. 共有が作成されたことを確認します。

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



を参照してください ["SMB 共有を作成"](#) 詳細については、

2. バックエンドを作成する際に、SMBボリュームを指定するように次の項目を設定する必要があります。ONTAP バックエンド構成オプションのすべてのFSXについては、を参照してください ["FSX"](#)

(ONTAP の構成オプションと例) "。

パラメータ	説明	例
smbShare	次のいずれかを指定できます。Microsoft管理コンソールまたはONTAP CLIを使用して作成されたSMB共有の名前、またはAstra TridentでSMB共有を作成できるようにする名前。  このパラメータは、Amazon FSx for ONTAPバックエンドに必要です。	smb-share
nasType	をに設定する必要があります <b>smb</b> . nullの場合、デフォルトはです <b>nfs</b> 。	smb
securityStyle	新しいボリュームのセキュリティ形式。  をに設定する必要があります <b>ntfs</b> または <b>mixed SMB</b> ボリューム	ntfs または mixed SMB ボリュームの場合
unixPermissions	新しいボリュームのモード。* SMBボリュームは空にしておく必要があります。*	""

## FSX (ONTAP の構成オプションと例)

Amazon FSX for ONTAP のバックエンド構成オプションについて説明します。ここでは、バックエンドの設定例を示します。

### バックエンド構成オプション

バックエンド設定オプションについては、次の表を参照してください。

パラメータ	説明	例
version		常に 1
storageDriverName	ストレージドライバの名前	ontap-nas、ontap-nas-economy、ontap-nas-flexgroup、ontap-san、ontap-san-economy
backendName	カスタム名またはストレージバックエンド	ドライバ名 + "_" + データ LIF



パラメータ	説明	例
managementLIF	<p>クラスタ管理 LIF または SVM 管理 LIF の IP アドレス</p> <p>MetroClusterのシームレスなスイッチオーバーを実現するには、SVM 管理LIFを指定する必要があります。</p> <p>Fully Qualified Domain Name (FQDN；完全修飾ドメイン名) を指定できます。</p> <p>を使用してAstra Tridentをインストールした場合、IPv6アドレスを使用するようにを設定できます  --use-ipv6 フラグ。IPv6アドレスは、[28e8：d9fb：a825：b7bf：69a8：d02f：9e7b：3555]などの角かっこで定義する必要があります。</p>	<p>「 10.0.0.1 」、「 [2001:1234:abcd::fefe] 」</p>
dataLIF	<p>プロトコル LIF の IP アドレス。</p> <p>* ONTAP NASドライバ*：データLIFを指定することを推奨します。指定しない場合は、Astra TridentがSVMからデータLIFを取得します。NFSマウント処理に使用するFully Qualified Domain Name (FQDN；完全修飾ドメイン名) を指定して、ラウンドロビンDNSを作成して複数のデータLIF間で負荷を分散することができます。初期設定後に変更できます。を参照してください。</p> <p>* ONTAP SANドライバ*：iSCSIには指定しないでくださいTrident がONTAP の選択的LUNマップを使用して、マルチパスセッションの確立に必要なiSCSI LIFを検出します。データLIFが明示的に定義されている場合は警告が生成されません。</p> <p>を使用してAstra Tridentをインストールした場合、IPv6アドレスを使用するようにを設定できます  --use-ipv6 フラグ。IPv6アドレスは、[28e8：d9fb：a825：b7bf：69a8：d02f：9e7b：3555]などの角かっこで定義する必要があります。</p>	

パラメータ	説明	例
autoExportPolicy	<p>エクスポートポリシーの自動作成と更新を有効にします[ブーリアン]。</p> <p>を使用する autoExportPolicy および autoExportCIDRs ネットアップのAstra Tridentなら、エクスポートポリシーを自動的に管理できます。</p>	false
autoExportCIDRs	<p>KubernetesのノードIPをいつからフィルタリングするかを示すCIDRsのリスト</p> <p>autoExportPolicy が有効になります。</p> <p>を使用する autoExportPolicy および autoExportCIDRs ネットアップのAstra Tridentなら、エクスポートポリシーを自動的に管理できます。</p>	「[0.0.0.0/0]、 「::/0」 」
labels	ボリュームに適用する任意のJSON形式のラベルのセット	""
clientCertificate	クライアント証明書の Base64 エンコード値。証明書ベースの認証に使用されます	""
clientPrivateKey	クライアント秘密鍵の Base64 エンコード値。証明書ベースの認証に使用されます	""
trustedCACertificate	信頼された CA 証明書の Base64 エンコード値。任意。証明書ベースの認証に使用されます。	""
username	クラスタまたはSVMに接続するためのユーザ名。クレデンシャルベースの認証に使用されます。たとえば、vsadminのように指定します。	
password	クラスタまたはSVMに接続するためのパスワード。クレデンシャルベースの認証に使用されます。	
svm	使用する Storage Virtual Machine	SVM管理LIFが指定されている場合に生成されます。

パラメータ	説明	例
storagePrefix	<p>SVM で新しいボリュームをプロビジョニングする際に使用するプレフィックスを指定します。</p> <p>作成後に変更することはできません。このパラメータを更新するには、新しいバックエンドを作成する必要があります。</p>	trident
limitAggregateUsage	<p>* Amazon FSx for NetApp ONTAP には指定しないでください。*</p> <p>提供された fsxadmin および vsadmin アグリゲートの使用状況を取得し、Astra Tridentを使用して制限するために必要な権限が含まれていない。</p>	使用しないでください。
limitVolumeSize	<p>要求されたボリュームサイズがこの値を超えている場合、プロビジョニングが失敗します。</p> <p>また、qtreeおよびLUN用に管理するボリュームの最大サイズも制限します qtreesPerFlexvol オプションを使用すると、FlexVol あたりの最大qtree数をカスタマイズできます。</p>	""（デフォルトでは適用されません）
lunsPerFlexvol	<p>FlexVol あたりの最大LUN数。有効な範囲は50、200です。</p> <p>SANのみ。</p>	100
debugTraceFlags	<p>トラブルシューティング時に使用するデバッグフラグ。例：{"API" : false、"method" : true}</p> <p>使用しないでください debugTraceFlags トラブルシューティングを実行していて、詳細なログダンプが必要な場合を除きます。</p>	null

パラメータ	説明	例
nfsMountOptions	<p>NFSマウントオプションをカンマで区切ったリスト。</p> <p>Kubernetes永続ボリュームのマウントオプションは通常はストレージクラスで指定されますが、ストレージクラスでマウントオプションが指定されていない場合、Astra Tridentはストレージバックエンドの構成ファイルで指定されているマウントオプションを使用します。</p> <p>ストレージクラスや構成ファイルにマウントオプションが指定されていない場合、Astra Tridentは関連付けられた永続的ボリュームにマウントオプションを設定しません。</p>	""
nasType	<p>NFSボリュームまたはSMBボリュームの作成を設定</p> <p>オプションはです <code>nfs</code>、<code>smb</code>、または<code>null</code>。</p> <p>*をに設定する必要があります <code>smb</code> SMBボリューム。*を<code>null</code>に設定すると、デフォルトでNFSボリュームが使用されます。</p>	<code>nfs</code>
qtreesPerFlexvol	FlexVol あたりの最大 qtree 数。有効な範囲は [50 、 300] です。	200
smbShare	<p>次のいずれかを指定できます。Microsoft管理コンソールまたはONTAP CLIを使用して作成されたSMB共有の名前、またはAstra TridentでSMB共有を作成できるようにする名前。</p> <p>このパラメータは、Amazon FSx for ONTAPバックエンドに必要です。</p>	<code>smb-share</code>

パラメータ	説明	例
useREST	<p>ONTAP REST API を使用するためのブーリアンパラメータ。* テクニカルレビュー *</p> <p>useREST は、テクニカルレビューとして提供されています。テスト環境では、本番環境のワークロードでは推奨されません。に設定すると true`Astra Trident は、ONTAP REST APIを使用してバックエンドと通信します。</p> <p>この機能にはONTAP 9.11.1以降が必要です。また、使用するONTAP ログインロールにはへのアクセス権が必要です `ontap アプリケーション：これは事前定義されたによって満たされます vsadmin および cluster-admin ロール。</p>	false

#### 更新 dataLIF 初期設定後

初期設定後にデータLIFを変更するには、次のコマンドを実行して、更新されたデータLIFを新しいバックエンドJSONファイルに指定します。

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



PVCが1つ以上のポッドに接続されている場合は、対応するすべてのポッドを停止してから、新しいデータLIFを有効にするために稼働状態に戻す必要があります。

#### ボリュームのプロビジョニング用のバックエンド構成オプション

これらのオプションを使用して、のデフォルトプロビジョニングを制御できます defaults 設定のセクション。例については、以下の設定例を参照してください。

パラメータ	説明	デフォルト
spaceAllocation	space-allocation for LUN のコマンドを指定します	true
spaceReserve	スペースリザーベーションモード：「none」（シン）または「volume」（シック）	none
snapshotPolicy	使用する Snapshot ポリシー	none

パラメータ	説明	デフォルト
qosPolicy	<p>作成したボリュームに割り当てる QoS ポリシーグループ。ストレージプールまたはバックエンドごとに、QOSPolicyまたはadaptiveQosPolicyのいずれかを選択します。</p> <p>Trident が Astra で QoS ポリシーグループを使用するには、ONTAP 9.8 以降が必要です。</p> <p>非共有のQoSポリシーグループを使用して、各コンスティチュエントに個別にポリシーグループを適用することを推奨します。共有 QoS ポリシーグループにより、すべてのワークロードの合計スループットに対して上限が適用されます。</p>	「」
adaptiveQosPolicy	<p>アダプティブ QoS ポリシーグループ：作成したボリュームに割り当てます。ストレージプールまたはバックエンドごとに、QOSPolicyまたはadaptiveQosPolicyのいずれかを選択します。</p> <p>経済性に影響する ONTAP - NAS ではサポートされません。</p>	「」
snapshotReserve	Snapshot 「0」 用にリザーブされているボリュームの割合	状況 snapshotPolicy はです none、else 「」
splitOnClone	作成時にクローンを親からスプリットします	false
encryption	<p>新しいボリュームでNetApp Volume Encryption (NVE) を有効にします。デフォルトはです false。このオプションを使用するには、クラスタで NVE のライセンスが設定され、有効になっている必要があります。</p> <p>NAEがバックエンドで有効になっている場合は、Astra TridentでプロビジョニングされたすべてのボリュームがNAEに有効になります。</p> <p>詳細については、以下を参照してください。 <a href="#">"Astra TridentとNVEおよびNAEの相互運用性"</a>。</p>	false

パラメータ	説明	デフォルト
luksEncryption	LUKS暗号化を有効にします。を参照してください " <a href="#">Linux Unified Key Setup (LUKS; 統合キーセットアップ)</a> を使用"。  SANのみ。	""
tieringPolicy	使用する階層化ポリシー none	snapshot-only ONTAP 9.5より前のSVM-DR構成の場合
unixPermissions	新しいボリュームのモード。  * SMBボリュームは空にしておきます。*	「」
securityStyle	新しいボリュームのセキュリティ形式。  NFSのサポート mixed および unix セキュリティ形式  SMBはをサポートします mixed および ntfs セキュリティ形式	NFSのデフォルトはです unix。  SMBのデフォルト： ntfs。

## 例

を使用します nasType、node-stage-secret-name`および `node-stage-secret-namespace`を使用して、SMBボリュームを指定し、必要なActive Directoryクレデンシャルを指定できます。SMBボリュームは、を使用してサポートされます `ontap-nas` ドライバーのみ。

```
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"
```

## 著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。