



# バックエンドを設定

## Astra Trident

NetApp  
April 16, 2024

# 目次

バックエンドを設定 .....	1
Azure NetApp Files バックエンドを設定します .....	1
GCP バックエンドの CVS を設定します .....	14
NetApp HCI または SolidFire バックエンドを設定します .....	25
バックエンドに ONTAP SAN ドライバを設定します .....	31
ONTAP NASバックエンドを設定します .....	51
Amazon FSX for NetApp ONTAP で Astra Trident を使用 .....	73

# 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、ストレージクラスが定義した要件に合わせて、バックエンドからストレージプールを自動的に提供します。お使いのストレージシステムのタイプに基づいたバックエンドの設定の詳細については、こちらを参照してください。

- ["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 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 NAS ボリュームと SMB ボリュームを接続できます。

- ["準備"](#)
- ["設定オプションと例"](#)

### 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
- Astra Trident は Windows ARM アーキテクチャをサポートしていません。

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

ANF 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。

Azure NetApp Files を初めて使用する場合や新しい場所で使用する場合は、いくつかの初期設定が必要です。

- Azure NetApp Files をセットアップして NFS ボリュームを作成する方法については、[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"](#)。
- Azure NetApp Files を設定して SMB ボリュームを追加するには、[以下を参照してください。 "Azure : Azure NetApp Files 用の SMB ボリュームを作成します"](#)。

### 要件

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。

- 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

- 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Tridentシークレットが少なくとも1つあり、ANFがActive Directoryに認証できるようになっている。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='pw'
```

- 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。

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。

### バックエンド構成オプション

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	
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)

パラメータ	説明	デフォルト
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 "
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
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	<p>新しいボリュームに対するエクスポートルール</p> <p>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。</p> <p>SMBボリュームでは無視されます。</p>	"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	<p>新しいボリュームのUNIX権限（8進数の4桁）。</p> <p>SMBボリュームでは無視されます。</p>	""（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）



ANFバックエンドで作成されたすべてのボリュームに対して、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、ストレージプールごとにラベルを定義し、ストレージプール内に作成されたすべてのボリュームをグループ化できます。これは、バックエンド構成で提供されるカスタマイズ可能な一連のラベルに基づいてボリュームを区別するのに便利です。

## 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：仮想ストレージプールの構成

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。

```

{
  "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"
  },
  "location": "eastus",

  "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。

## GCP バックエンドの CVS を設定します

提供されている設定例を使用して、ネットアップCloud Volumes Service (CVS) for Google Cloud Platform (GCP) をAstra Tridentインストールのバックエンドとして設定する方法を説明します。

### CVS for GCPのAstra Tridentサポートについてご確認ください

Astra Tridentは、デフォルトのCVSサービスタイプがonに設定されたボリュームをサポートしています **"GCP"**。Astra Tridentは、CVSサービスタイプで許可されている最小値に関係なく、100GiB未満のCVSボリュームをサポートしていません。したがって、Tridentは要求されたボリュームが最小サイズよりも小さい場合、自動的に100GiBのボリュームを作成します。

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。

- ネットアップ CVS 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
- CVS サービスアカウントの API キーファイル

### バックエンド構成オプション

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



パラメータ	説明	デフォルト
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	ストレージのタイプから選択します hardware（パフォーマンス最適化済み）または software（CVSサービスタイプ）	
projectNumber	Google Cloud アカウントのプロジェクト番号。この値は、Google Cloud ポータルのホームページにあります。	
apiRegion	CVS アカウント地域。バックエンドがボリュームをプロビジョニングするリージョンです。	
apiKey	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。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）。	
proxyURL	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。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。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。	
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）
serviceLevel	新しいボリュームの CVS サービスレベル。「Standard」、「Premium」、「Extreme」のいずれかです。	標準
network	CVSボリュームに使用するGCPネットワーク	デフォルト

パラメータ	説明	デフォルト
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}。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

共有VPCネットワークを使用する場合は、両方のポートを使用します `projectNumber` および `hostProjectNumber` を指定する必要があります。その場合は、 `projectNumber` は、サービスプロジェクトです `hostProjectNumber` は、ホストプロジェクトです。

。 `apiRegion` Astra TridentがCVSボリュームを作成するGCPリージョンを表します。クロスリージョンのKubernetesクラスタを作成する場合、で作成されたCVSボリューム `apiRegion` 複数のGCPリージョンのノードでスケジュールされているワークロードで使用できます。リージョン間トラフィックは追加コストがかかることに注意してください。

- 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。例：

```
- key: topology.kubernetes.io/region
  values:
    - us-east1
    - europe-west1
```

- `storageClass` は、必要なを選択するためのオプションのパラメータです "[CVS サービスタイプ](#)"。基本CVSサービスタイプから選択できます (`storageClass=software`) またはCVS -パフォーマンスサービスのタイプ (`storageClass=hardware`) を使用します。これは、デフォルトでTridentが使用します。必ずを指定してください `apiRegion` それぞれのCVSを提供します `storageClass` バックエンドの定義に含まれています。

Astra Trident は、 Google Cloud 上の基本 CVS サービスタイプと統合されている ベータ版の機能 で、本番環境のワークロード向けではありません。Trident は、 CVS パフォーマンスサービスタイプでは完全にサポートされている \*\* で、デフォルトで使用されます。

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。

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、各ボリュームのデフォルトのプロビジョニング方法を制御できます。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
exportRule	新しいボリュームのエクスポートルール	"0.0.0.0/0 "



```

Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5o
jY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nzn
HczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrt
HisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbO
guSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKe
yAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRA
Gz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4j
K3bl/qp8B4Kws8zX5ojY9m\nXsYg6gyxy4zq7OlWgWgGa==\n-----END PRIVATE
KEY-----\n",
    "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"
  }
}

```

## 例 2：基本 CVS サービスタイプの設定

この例は、基本 CVS サービスタイプを使用するバックエンド定義を示しています。このサービスタイプは、汎用ワークロード向けであり、パフォーマンスが低く、ゾーンの可用性も高くなります。

```

{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "storageClass": "software",
  "apiRegion": "us-east4",
  "apiKey": {
    "type": "service_account",
    "project_id": "my-gcp-project",
    "private_key_id": "1234567890123456789012345678901234567890",
    "private_key": "-----BEGIN PRIVATE KEY-----
\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZ
srrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisI
sAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSa
PIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZN
chRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1z
ZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl
/qp8B4Kws8zX5ojY9m\nznHczZsrrtHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kw

```

```
s8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY
9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHc
zZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHi
sIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOgu
SaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyA
ZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz
1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3
bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4
Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5o
jY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nzn
HczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtr
HisIsAbOguSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbO
guSaPIKeyAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKe
yAZNchRAGz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRA
Gz1zZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz1zZE4j
K3bl/qp8B4Kws8zX5ojY9m\nXsYg6gyxy4zq7OlwWgLwGa==\n-----END PRIVATE
KEY-----\n",
    "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"
  }
}
```

### 例 3：単一のサービスレベルの設定

この例は、Google Cloud us-west2 リージョン内のすべての Astra Trident で作成されたストレージに同じ要素を適用するバックエンドファイルを示しています。この例は、の使用状況も示しています proxyURL バックエンド構成ファイル内。

```
{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "apiRegion": "us-west2",
  "apiKey": {
    "type": "service_account",
    "project_id": "my-gcp-project",
    "private_key_id": "1234567890123456789012345678901234567890",

```

```

    "private_key": "-----BEGIN PRIVATE KEY-----
\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZ
srtrHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisI
sAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSa
PIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZN
chRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzll
ZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl
/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4K
ws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5oj
Y9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznH
czZsrtrHisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtr
HisIsAbOguSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbO
guSaPIKeyAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKe
yAZNchRAGzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRA
GzllzZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllzZE4j
K3bl/qp8B4Kws8zX5ojY9m\nXsYg6gyxy4zq70lwWgLwGa==\n-----END PRIVATE
KEY-----\n",
    "client_email": "cloudvolumes-admin-sa@my-gcp-
project.iam.gserviceaccount.com",
    "client_id": "123456789012345678901",
    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
    "token_uri": "https://oauth2.googleapis.com/token",
    "auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",
    "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/cloudvolumes-admin-
sa%40my-gcp-project.iam.gserviceaccount.com"
  },
  "proxyURL": "http://proxy-server-hostname/",
  "nfsMountOptions": "vers=3,proto=tcp,timeo=600",
  "limitVolumeSize": "10Ti",
  "serviceLevel": "premium",
  "defaults": {
    "snapshotDir": "true",
    "snapshotReserve": "5",
    "exportRule": "10.0.0.0/24,10.0.1.0/24,10.0.2.100",
    "size": "5Ti"
  }
}

```

#### 例 4：仮想ストレージプールの構成

この例は、仮想ストレージプールとともに設定されたバックエンド定義ファイルを示しています  
StorageClasses それはそれらを再度参照する。

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 `snapshotReserve 5%` およびである `exportRule` を `0.0.0.0/0` に設定します。仮想ストレージプールは、で定義されます `storage` セクション。この例では、個々のストレージプールが独自に設定されています `'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。

```
{
  "version": 1,
  "storageDriverName": "gcp-cvs",
  "projectNumber": "012345678901",
  "apiRegion": "us-west2",
  "apiKey": {
    "type": "service_account",
    "project_id": "my-gcp-project",
    "private_key_id": "1234567890123456789012345678901234567890",
    "private_key": "-----BEGIN PRIVATE KEY-----
\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZ
srtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisI
sAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSa
PIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZN
chRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzll
ZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl
/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kw
s8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY
9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHc
zZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHi
sIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOgu
SaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyA
ZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGz
llZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3
bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4
Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5o
jY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nzn
HczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtr
HisIsAbOguSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbO
guSaPIKeyAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKe
yAZNchRAGzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRA
GzllZE4jK3bl/qp8B4Kws8zX5ojY9m\nznHczZsrtrHisIsAbOguSaPIKeyAZNchRAGzllZE4j
K3bl/qp8B4Kws8zX5ojY9m\nXsYg6gyxy4zq7OlwWgLwGa==\n-----END PRIVATE
```

```

KEY-----\n",
    "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"
}
]
}

```

次の StorageClass 定義は、上記のストレージプールを参照してください。を使用します  
`parameters.selector` フィールドでは、ボリュームのホストに使用され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。

最初のストレージクラス (`cvs-extreme-extra-protection`) を最初の仮想ストレージ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、どの仮想ストレージプールを選択するかを決定し、Snapshot リザーブの要件を確実に満たします。

```

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

```

## 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方法をご確認ください。

### 必要なもの

- 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"](#)。

### 知っておくべきこと

。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を使用するオプション。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
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）

パラメータ	説明	デフォルト
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。



Astra Trident は、ボリュームを作成すると、ストレージプール上のすべてのラベルを、プロビジョニング時にバックアップストレージ LUN にコピーします。ストレージ管理者は、ストレージプールごとにラベルを定義し、ストレージプール内に作成されたすべてのボリュームをグループ化できます。これにより、バックエンド構成で提供されるカスタマイズ可能な一連のラベルに基づいてボリュームを簡単に区別できます。

## 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 仮想ストレージプール用のドライバ

この例は、仮想ストレージプールで設定されたバックエンド定義ファイルと、それらを参照する StorageClasses を示しています。

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。仮想ストレージプールは、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で独自のタイプが設定されており、一部のプールでは上記で設定したデフォルト値が上書きされます。

```

{
  "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、上記の仮想ストレージプールを参照してください。を使用する `parameters.selector` 各ストレージクラスは、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、最初の仮想ストレージ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、どの仮想ストレージ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"](#)

## バックエンドに **ONTAP SAN** ドライバを設定します

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。

- ["準備"](#)
- ["設定と例"](#)

### 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 クラスタユーザまたはです 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 ユーザまたはです 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 ドライバを使用して ONTAP バックエンドを設定するための準備方法について説明します。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 ["こちらをご覧ください"](#) 詳細：

### 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。

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

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=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",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "clientCertificate": "Faaaakkkkeeee...Vaaalllluuueeeee",
  "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",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "username": "vsadmin",
  "password": "secret",
  "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。

## igroup を指定します

Astra Trident は、igroup を使用して、プロビジョニングするボリューム（LUN）へのアクセスを制御します。管理者はバックエンドに igroup を指定する方法として、次の 2 つを選択できます。

- Astra Trident では、バックエンドごとに igroup を自動的に作成、管理できます。状況 igroupName はバックエンドの定義に含まれていないため、Astra Trident がという名前の igroup を作成します `trident-  
<backend-UUID>` 指定します。これにより、各バックエンドに専用の igroup が割り当てられ、Kubernetes ノードの IQN の自動追加や削除が処理されます。

- また、事前に作成された igroup もバックエンドの定義で提供できます。これは、を使用して実行できます。igroupName パラメータを設定します。Astra Trident が、Kubernetes ノードの IQN を既存の igroup に追加または削除します。

を含むバックエンドの場合 igroupName 定義されている igroupName を使用して削除できます。tridentctl backend update Astra Tridentでigroupを自動処理すでにワークロードに接続されているボリュームへのアクセスが中断されることはありません。今後作成される igroup Astra Trident を使用して接続を処理します。



Astra Trident の一意のインスタンスごとに igroup を専用にすることを推奨します。これは、Kubernetes 管理者とストレージ管理者にとって有益です。CSI Trident は、クラスターノード IQN の igroup への追加と削除を自動化し、管理を大幅に簡易化します。Kubernetes 環境（および Astra Trident インストール）全体で同じ SVM を使用する場合、専用の igroup を使用することで、ある Kubernetes クラスターに対する変更が、別の Kubernetes クラスターに関連付けられた igroup に影響しないようにできます。また、Kubernetes クラスター内の各ノードに一意の IQN を設定することも重要です。前述のように、Astra Trident は IQN の追加と削除を自動的に処理します。ホスト間で IQN を再使用すると、ホスト間で誤って認識されて LUN にアクセスできないような、望ましくないシナリオが発生する可能性があります。

Astra Trident が CSI Provisioner として機能するように設定されている場合、Kubernetes ノード IQN は自動的に igroup に追加 / 削除されます。ノードがKubernetesクラスターに追加されると、trident-csi DemonSetによってポッドが展開されます (trident-csi-xxxxx) を追加し、ボリュームを接続できる新しいノードを登録します。ノード IQN もバックエンドの igroup に追加されます。ノードが遮断され、削除され、Kubernetes から削除された場合も、同様の手順で IQN の削除が処理されます。

Astra Trident が CSI Provisioner として実行されない場合は、Kubernetes クラスター内のすべてのワーカーノードからの iSCSI IQN を含むように、igroup を手動で更新する必要があります。Kubernetes クラスターに参加するノードの IQN を igroup に追加する必要があります。同様に、Kubernetes クラスターから削除されたノードの IQN を igroup から削除する必要があります。

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": "FaKePaSsWoRd",
  "igroupName": "trident",
  "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）。
- イニシエータへの追加の管理 igroupName バックエンドで提供されます。指定しない場合、デフォルトはです trident。

バックエンドが作成されると、対応するが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": "FaKePaSsWoRd",
  "igroupName": "trident",
  "chapInitiatorSecret": "cl9qxUpDaTeD",
  "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
}
```

```
./tridentctl update backend ontap_san_chap -f backend-san.json -n trident
+-----+-----+-----+-----+
+-----+-----+
|  NAME          | STORAGE DRIVER |                               UUID                               |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| ontap_san_chap | ontap-san      | aa458f3b-ad2d-4378-8a33-1a472ffbe5c |
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-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	クラスタ管理 LIF または SVM 管理 LIF の IP アドレス  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。	「10.0.0.1」、「[2001:1234:abcd::fefe]」
dataLIF	プロトコル LIF の IP アドレス。IPv6 には角かっこを使用します。設定後に更新することはできません	特に指定がないかぎり、SVM が派生します
useCHAP	CHAP を使用して ONTAP SAN ドライバ用の iSCSI を認証する [ブーリアン]	いいえ
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	「」

パラメータ	説明	デフォルト
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
igroupName	SAN ボリュームで使用する igroup の名前	"trident-<backend-UUID> "
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）
lunsPerFlexvol	FlexVol あたりの最大 LUN 数。有効な範囲は 50、200 です	100
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
useREST	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 *  MetroClusterではサポートされません。	いいえ

#### <code>useREST</code> 考慮事項



- 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。この機能を使用するには、ONTAP 9.10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。
- useREST は、MetroCluster ではサポートされていません。

ONTAP クラスタと通信するには、認証パラメータを指定する必要があります。これは、セキュリティログインまたはインストールされている証明書のユーザ名 / パスワードです。



ネットアップONTAP バックエンドにAmazon FSXを使用している場合は、を指定しないでください `limitAggregateUsage` パラメータ。 `fsxadmin` および `vsadmin` Amazon FSX for NetApp ONTAP のロールには、アグリゲートの使用状況を取得し、Astra Tridentを通じて制限するために必要なアクセス権限が含まれていません。



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。

をクリックします `ontap-san` ドライバのデフォルトでは、SVMのすべてのデータLIF IPが使用され、iSCSI マルチパスが使用されます。のデータLIFのIPアドレスを指定します `ontap-san` ドライバは、マルチパスを無効にして、指定されたアドレスだけを使用します。



バックエンドを作成するときは、この点に注意してください `dataLIF` および `storagePrefix` 作成後に変更することはできません。これら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。

`igroupName` ONTAP クラスタですでに作成されているigroupに設定できます。指定しない場合、Trident は `trident-<backend-UUID>` という名前の igroup を自動的に作成します。事前に定義された `igroupName` を指定する場合は、各 Kubernetes クラスタで igroup を使用することを推奨します。ただし、SVM が環境間で共有される場合です。これは、Astra Trident が IQN の追加や削除を自動的に維持するために必要です。

バックエンドは、作成後に igroup を更新することもできます。

- `igroupName` は、Astra Trident の外部の SVM で作成および管理される新しい igroup を指すように更新できます。
- `igroupName` は省略できます。この場合、Astra Trident は Trident によって `trident-<backend-UUID>` igroup が自動的に作成および管理されます。

どちらの場合も、ボリュームの添付ファイルには引き続きアクセスできます。以降のボリューム接続では、更新された igroup が使用されます。この更新によって、バックエンドにあるボリュームへのアクセスが中断されることはありません。

には完全修飾ドメイン名 (FQDN) を指定できます `managementLIF` オプション

``managementLIF`` すべてのONTAP ドライバをIPv6  
アドレスに設定することもできます。Tridentをに必ずインストールしてください ``---use-ipv6`` フラグ。定義には注意が必要です ``managementLIF`` 角かっこ内のIPv6アドレス。



IPv6アドレスを使用する場合は、を確認してください `managementLIF` および `dataLIF` (バックエンド定義に含まれている場合) は、`[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]`などの角括弧内に定義されます。状況 `dataLIF` が指定されていない場合、Astra TridentがSVMからIPv6 データLIFを取得します。

SANドライバでCHAPを使用できるようにするには、を設定します `useCHAP` パラメータの値 `true` バックエンドの定義に含まれています。その後、Astra Trident が、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 ["こちらをご覧ください"](#) その仕組みについては、を参照してください。

をクリックします `ontap-san-economy` ドライバ、`limitVolumeSize` オプションを使用すると、`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されます。



Trident から、を使用して作成したすべてのボリュームの「Comments」フィールドにプロビジョニングラベルが設定されます `ontap-san` ドライバ。作成された各ボリュームについて、FlexVol の [Comments] フィールドに、配置先のストレージプールにあるすべてのラベルが入力されます。ストレージ管理者は、ストレージプールごとにラベルを定義し、ストレージプール内に作成されたすべてのボリュームをグループ化できます。これにより、バックエンド構成で提供されるカスタマイズ可能な一連のラベルに基づいてボリュームを簡単に区別できます。

#### 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、構成の特別なセクションで各ボリュームをデフォルトでプロビジョニングする方法を制御できます。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
<code>spaceAllocation</code>	<code>space-allocation</code> for LUN のコマンドを指定します	正しいです
<code>spaceReserve</code>	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
<code>snapshotPolicy</code>	使用する Snapshot ポリシー	なし
<code>qosPolicy</code>	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 <code>QOSPolicy</code> または <code>adaptiveQosPolicy</code> のいずれかを選択します	「」
<code>adaptiveQosPolicy</code>	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 <code>QOSPolicy</code> または <code>adaptiveQosPolicy</code> のいずれかを選択します	「」
<code>snapshotReserve</code>	Snapshot 「0」 用にリザーブされているボリュームの割合	状況 <code>snapshotPolicy</code> は「none」、それ以外は「」です。
<code>splitOnClone</code>	作成時にクローンを親からスプリットします	いいえ
<code>splitOnClone</code>	作成時にクローンを親からスプリットします	いいえ

パラメータ	説明	デフォルト
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 <code>false</code>。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) を使用"</a> 。	""
securityStyle	新しいボリュームのセキュリティ形式	「UNIX」
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」



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、各コンスチテ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。

次に、デフォルトが定義されている例を示します。

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "trident_svm",
  "username": "admin",
  "password": "password",
  "labels": {"k8scluster": "dev2", "backend": "dev2-sanbackend"},
  "storagePrefix": "alternate-trident",
  "igroupName": "custom",
  "debugTraceFlags": {"api": false, "method": true},
  "defaults": {
    "spaceReserve": "volume",
    "qosPolicy": "standard",
    "spaceAllocation": "false",
    "snapshotPolicy": "default",
    "snapshotReserve": "10"
  }
}
```



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。



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

ontap-san 証明書ベースの認証を使用するドライバ

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,
  "storageDriverName": "ontap-san",
  "backendName": "DefaultSANBackend",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "clientCertificate": "ZXR0ZXJwYXB...ICMgJ3BhcGVyc2",
  "clientPrivateKey": "vciwKIyAgZG...0cnksIGRlc2NyaX",
  "trustedCACertificate": "zcyBbaG...b3Igb3duIGNsYXNz"
}
```

ontap-san 双方向**CHAP**を備えたドライバ

これは、バックエンドの最小限の設定例です。この基本設定では、が作成されます ontap-san バックエンドの指定 useCHAP をに設定します true。

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "labels": {"k8scluster": "test-cluster-1", "backend": "testcluster1-
sanbackend"},
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSD6cNwxyz",
  "igroupName": "trident",
  "username": "vsadmin",
  "password": "secret"
}
```

ontap-san-economy ドライバ

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

### 仮想ストレージプールを使用するバックエンドの例

次のバックエンド定義ファイルの例では、などの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。spaceReserve「なし」の場合は、spaceAllocation との誤り encryption 実行されます。仮想ストレージプールは、ストレージセクションで定義します。

この例では、一部のストレージプールが独自に設定されています。spaceReserve、spaceAllocation、および encryption 値を指定すると、一部のプールでは、上記のデフォルト値が上書きされます。



```

{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.3",
  "svm": "svm_iscsi",
  "useCHAP": true,
  "chapInitiatorSecret": "cl9qxIm36DKyawxy",
  "chapTargetInitiatorSecret": "rqxigXgkesIpwxyz",
  "chapTargetUsername": "iJF4heBRT0TCwxyz",
  "chapUsername": "uh2aNCLSd6cNwxyz",
  "igroupName": "trident",
  "username": "vsadmin",
  "password": "secret",

  "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"
    }
}
]
}

```

のiSCSIの例を次に示します ontap-san-economy ドライバ:

```

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

  "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"
      }
    }
  ]
}

```

バックエンドを **StorageClasses** にマッピングします

次の StorageClass 定義は、上記の仮想ストレージプールを参照してください。を使用する `parameters.selector` 各ストレージクラスは、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 (`protection-gold`) を指定すると、内の1番目と2番目の仮想ストレージプールにマッピングされます `ontap-nas-flexgroup` 内の最初の仮想ストレージプール `ontap-san` バックエンド：ゴールドレベルの保護を提供している唯一のプールです。
- 2つ目のStorageClass (`protection-not-gold`) は、の3番目、4番目の仮想ストレージプールにマッピングされます `ontap-nas-flexgroup` のバックエンドと2番目の3番目の仮想ストレージプール `ontap-san` 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。
- 第3のストレージクラス (`app-mysqldb`) をクリックすると、で4番目の仮想ストレージプールにマッピングされます `ontap-nas` のバックエンドと3つ目の仮想ストレージプール `ontap-san-economy` バックエンド：mysqldb タイプのアプリケーション用のストレージプール設定を提供しているプールは、これだけです。
- 第4のストレージクラス (`protection-silver-creditpoints-20k`) は、の3番目の仮想ストレージプールにマッピングされます `ontap-nas-flexgroup` のバックエンドと2つ目の仮想ストレージプール `ontap-san` バックエンド：ゴールドレベルの保護を提供している唯一のプールは、20000 の利用可能なクレジットポイントです。
- 第5のストレージクラス (`creditpoints-5k`) をクリックすると、で2つ目の仮想ストレージプールにマッピングされます `ontap-nas-economy` のバックエンドと3つ目の仮想ストレージプール `ontap-san` バックエンド：5000 ポイントの利用可能な唯一のプールは以下のとおりです。

Trident が、どの仮想ストレージ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: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: netapp.io/trident
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```

# ONTAP NASバックエンドを設定します

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。

- "準備"
- "設定と例"



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ワークロード向けのドライバ。Astra Tridentは、で作成され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、モビリティを提供します `ontap-nas` ドライバ。。 `ontap-nas-economy` ドライバは、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はるかに高く、予想されるデータ保護、ディザスタリカバリ、モビリティ（Kubernetesクラスター間でのボリュームの移動）の要件がないと予想される限られたユースケースでのみ使用してください。

## 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。

## 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ドライバを使用してONTAPバックエンドを構成するための準備方法について説明します。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。

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：

## 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。

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

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...Vaaalllluuuuueeee",
  "clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
  "storagePrefix": "myPrefix_"
}

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

NAME	STORAGE DRIVER	UUID
NasBackend	ontap-nas	98e19b74-aec7-4a3d-8dcf-128e5033b214
online	9	



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-nas",
  "backendName": "NasBackend",
  "managementLIF": "1.2.3.4",
  "dataLIF": "1.2.3.8",
  "svm": "vserver_test",
  "username": "vsadmin",
  "password": "secret",
  "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 で許可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、ストレージクラスタへのアクセスを、指定した範囲の IP を持つワーカーノードだけに制限することで、きめ細かな管理と自動化をサポートします。



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": "FaKePaSsWoRd",
  "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）を含む事前に作成され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", ":::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。

### 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について説明します。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	クラスタ管理 LIF または SVM 管理 LIF の IP アドレス  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。	「 10.0.0.1 」、「 [2001:1234:abcd::fefe] 」

パラメータ	説明	デフォルト
dataLIF	プロトコル LIF の IP アドレス。IPv6 には角かっこを使用します。設定後に更新することはできません	特に指定がないかぎり、SVM が派生します
autoExportPolicy	エクスポートポリシーの自動作成と更新を有効にする [ブーリアン]	いいえ
autoExportCIDRs	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	[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 / 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
igroupName	SAN ボリュームで使用する igroup の名前	"trident-<backend-UUID> "
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）

パラメータ	説明	デフォルト
lunsPerFlexvol	FlexVol あたりの最大 LUN 数。有効な範囲は 50 、 200 です	100
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
nfsMountOptions	NFS マウントオプションをカンマで区切ったリスト	「」
qtreesPerFlexvol	FlexVol あたりの最大 qtree 数。有効な範囲は [50 、 300] です。	「200」
useREST	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 *  MetroClusterではサポートされません。	いいえ

#### <code>useREST</code> 考慮事項



- 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。この機能を使用するには、ONTAP 9.10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。
- useREST は、MetroCluster ではサポートされていません。

ONTAP クラスタと通信するには、認証パラメータを指定する必要があります。これは、セキュリティログインまたはインストールされている証明書のユーザ名 / パスワードです。



ネットアップONTAP バックエンドにAmazon FSXを使用している場合は、を指定しないでください limitAggregateUsage パラメータ。 fsxadmin および vsadmin Amazon FSX for NetApp ONTAP のロールには、アグリゲートの使用状況を取得し、Astra Tridentを通じて制限するために必要なアクセス権限が含まれていません。



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。



バックエンドを作成するときは、を忘れないでください dataLIF および storagePrefix 作成後に変更することはできません。これら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。

には完全修飾ドメイン名 (FQDN) を指定できます managementLIF オプションにFQDNを指定することもできます dataLIF オプション。その場合は、NFSマウント処理にFQDNが使用されます。こうすることで、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。

`managementLIF` すべてのONTAP ドライバをIPv6アドレスに設定することもできます。Astra Tridentは、必ずを使用してインストールしてください `--use-ipv6` フラグ。を定義する際は注意が必要です `managementLIF` 角かっこ内のIPv6アドレス。



IPv6アドレスを使用する場合は、を確認してください managementLIF および dataLIF（バックエンド定義に含まれている場合）は、[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]などの角括弧内に定義されます。状況 dataLIF が指定されていない場合、Astra TridentがSVMからIPv6 データLIFを取得します。

を使用する autoExportPolicy および autoExportCIDRs CSI Tridentでは、エクスポートポリシーを自動的に管理できます。これはすべての ONTAP-NAS-\* ドライバでサポートされています。

をクリックします ontap-nas-economy ドライバ、 limitVolumeSize オプションを使用すると、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されます 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。

。 nfsMountOptions パラメータを使用すると、マウントオプションを指定できます。Kubernetes 永続ボリュームのマウントオプションは通常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。ストレージクラスまたは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。



Tridentから、を使用して作成したすべてのボリュームの「Comments」フィールドにプロビジョニングラベルが設定されます(ontap-nas および(ontap-nas-flexgroup。使用するドライバに基づいて、FlexVol にコメントが設定されます (ontap-nas) またはFlexGroup のいずれかです (ontap-nas-flexgroup) )。Trident が、ストレージプール上にあるすべてのラベルを、プロビジョニング時に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、ストレージプールごとにラベルを定義し、ストレージプール内に作成されたすべてのボリュームをグループ化できます。これにより、バックエンド構成で提供されるカスタマイズ可能な一連のラベルに基づいてボリュームを簡単に区別できます。

#### 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、構成の特別なセクションで各ボリュームをデフォルトでプロビジョニングする方法を制御できます。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
securityStyle	新しいボリュームのセキュリティ形式	「UNIX」
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	「777」
Snapshot ディレクトリ	の表示/非表示を制御します .snapshot ディレクトリ	いいえ
エクスポートポリシー	使用するエクスポートポリシー	デフォルト
securityStyle の追加	新しいボリュームのセキュリティ形式	「UNIX」





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、各コンスチテ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。

次に、デフォルトが定義されている例を示します。

```
{
  "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。

ontap-nas 証明書ベースの認証を使用するドライバ

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_"
}
```

ontap-nas ドライバと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": "secret",
  "nfsMountOptions": "nfsvers=4",
}
```

`ontap-nas-flexgroup` ドライバ

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

`ontap-nas` **IPv6**対応ドライバ

```
{
  "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": "netapp123"
}
```

ontap-nas-economy ドライバ

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

### 仮想ストレージプールを使用するバックエンドの例

次のバックエンド定義ファイルの例では、などの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。spaceReserve 「なし」の場合は、spaceAllocation との誤り encryption 実行されません。仮想ストレージプールは、ストレージセクションで定義します。

この例では、一部のストレージプールが独自に設定されています。spaceReserve、spaceAllocation、および encryption 値を指定すると、一部のプールでは、上記のデフォルト値が上書きされます。

ontap-nas ドライバ

```
{
  {
    "version": 1,
    "storageDriverName": "ontap-nas",
    "managementLIF": "10.0.0.1",
    "dataLIF": "10.0.0.2",
    "svm": "svm_nfs",
    "username": "admin",
    "password": "secret",
```

```

"nfsMountOptions": "nfsvers=4",

"defaults": {
    "spaceReserve": "none",
    "encryption": "false",
    "qosPolicy": "standard"
},
"labels":{"store":"nas_store", "k8scluster": "prod-cluster-1"},
"region": "us_east_1",
"storage": [
    {
        "labels":{"app":"msoffice", "cost":"100"},
        "zone":"us_east_1a",
        "defaults": {
            "spaceReserve": "volume",
            "encryption": "true",
            "unixPermissions": "0755",
            "adaptiveQosPolicy": "adaptive-premium"
        }
    },
    {
        "labels":{"app":"slack", "cost":"75"},
        "zone":"us_east_1b",
        "defaults": {
            "spaceReserve": "none",
            "encryption": "true",
            "unixPermissions": "0755"
        }
    },
    {
        "labels":{"app":"wordpress", "cost":"50"},
        "zone":"us_east_1c",
        "defaults": {
            "spaceReserve": "none",
            "encryption": "true",
            "unixPermissions": "0775"
        }
    },
    {
        "labels":{"app":"mysqldb", "cost":"25"},
        "zone":"us_east_1d",
        "defaults": {
            "spaceReserve": "volume",
            "encryption": "false",
            "unixPermissions": "0775"
        }
    }
]

```

```
}  
]  
}
```

ontap-nas-flexgroup ドライバ

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

```

ontap-nas-economy ドライバ

```

{
    "version": 1,
    "storageDriverName": "ontap-nas-economy",
    "managementLIF": "10.0.0.1",
    "dataLIF": "10.0.0.2",
    "svm": "svm_nfs",
    "username": "vsadmin",
    "password": "secret",

    "defaults": {
        "spaceReserve": "none",
        "encryption": "false"
    },
    "labels":{"store":"nas_economy_store"},
    "region": "us_east_1",
    "storage": [
        {
            "labels":{"department":"finance", "creditpoints":"6000"},
            "zone":"us_east_1a",
            "defaults": {
                "spaceReserve": "volume",
                "encryption": "true",
                "unixPermissions": "0755"
            }
        },
        {
            "labels":{"department":"legal", "creditpoints":"5000"},

```

```

        "zone": "us_east_1b",
        "defaults": {
            "spaceReserve": "none",
            "encryption": "true",
            "unixPermissions": "0755"
        }
    },
    {
        "labels": {"department": "engineering", "creditpoints": "3000"},
        "zone": "us_east_1c",
        "defaults": {
            "spaceReserve": "none",
            "encryption": "true",
            "unixPermissions": "0775"
        }
    },
    {
        "labels": {"department": "humanresource",
"creditpoints": "2000"},
        "zone": "us_east_1d",
        "defaults": {
            "spaceReserve": "volume",
            "encryption": "false",
            "unixPermissions": "0775"
        }
    }
]
}

```

バックエンドを **StorageClasses** にマッピングします

次の StorageClass 定義は、上記の仮想ストレージプールを参照してください。を使用する `parameters.selector` 各ストレージクラスは、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 (protection-gold) を指定すると、内の1番目と2番目の仮想ストレージプールにマッピングされます `ontap-nas-flexgroup` 内の最初の仮想ストレージプール `ontap-san` バックエンド：ゴールドレベルの保護を提供している唯一のプールです。
- 2つ目のStorageClass (protection-not-gold) は、の3番目、4番目の仮想ストレージプールにマッピングされます `ontap-nas-flexgroup` のバックエンドと2番目の3番目の仮想ストレージプール `ontap-san` 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。
- 第3のストレージクラス (app-mysqldb) をクリックすると、で4番目の仮想ストレージプールにマッピングされます `ontap-nas` のバックエンドと3つ目の仮想ストレージプール `ontap-san-economy` バックエンド：mysqldb タイプのアプリケーション用のストレージプール設定を提供しているプールは、これだけです。
- 第4のストレージクラス (protection-silver-creditpoints-20k) は、の3番目の仮想ストレージプ



ールにマッピングされます `ontap-nas-flexgroup` のバックエンドと2つ目の仮想ストレージプール `ontap-san` バックエンド：ゴールドレベルの保護を提供している唯一のプールは、20000 の利用可能なクレジットポイントです。

- 第5のストレージクラス (`creditpoints-5k`) をクリックすると、で2つ目の仮想ストレージプールにマッピングされます `ontap-nas-economy` のバックエンドと3つ目の仮想ストレージプール `ontap-san` バックエンド：5000 ポイントの利用可能な唯一のプールは以下のとおりです。

Trident が、どの仮想ストレージ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: protection-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-not-gold
provisioner: netapp.io/trident
parameters:
  selector: "protection!=gold"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: app-mysqldb
provisioner: netapp.io/trident
parameters:
  selector: "app=mysqldb"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: netapp.io/trident
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: creditpoints-5k
provisioner: netapp.io/trident
parameters:
  selector: "creditpoints=5000"
  fsType: "ext4"

```

# 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・システムを搭載したファイル・システムの起動と実行を可能にする、フルマネージドの AWS サービスです。Amazon FSX for NetApp 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際の簡易性、即応性、セキュリティ、拡張性を活用できます。FSX は、ONTAP のファイルシステム機能と管理 API の多くをサポートしています。

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。

## ONTAP ファイルシステム用に Amazon FSX を作成します

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

## Astra Trident の詳細をご確認ください

Astra Trident を初めて使用する場合は、以下のリンクを使用して確認してください。

- ["よくある質問です"](#)
- ["Astra Trident を使用するための要件"](#)
- ["Astra Trident を導入"](#)

- "ネットアップ ONTAP 用に ONTAP、Cloud Volumes ONTAP、Amazon FSX を設定する際のベストプラクティス"
- "Astra Trident を統合"
- "ONTAP SAN バックエンド構成"
- "ONTAP NASバックエンド構成"

ドライバ機能の詳細をご覧ください ["こちらをご覧ください"](#)。

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。

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 クラスターユーザ：を使用することは推奨されません fsxadmin Tridentを使用したユーザ vsadmin SVMユーザは、より多くのAstra Trident機能にアクセスできます。

## ドライバ

Astra Trident と Amazon FSX for NetApp ONTAP を統合するには、次のドライバを使用します。

- `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。

## 認証

Astra Trident には、次の 2 つの認証モードがあります。

- 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。



を使用することを強く推奨します vsadmin ユーザがではなく fsxadmin バックエンドを設定します。Astra Trident は、このユーザ名とパスワードを使用して FSX ファイルシステムと通信します。

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。

認証の詳細については、次のリンクを参照してください。

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

## Amazon FSX for NetApp ONTAP を使用して、EKS に Astra Trident を導入して設定する

### 必要なもの

- 既存のAmazon EKSクラスタまたはを使用する自己管理型Kubernetesクラスタ `kubectl` インストール済み。
- クラスタのワーカーノードからアクセスできる、NetApp ONTAP ファイルシステムと Storage Virtual Machine (SVM) 用の既存の Amazon FSX。
- 準備されているワーカーノード ["NFS か iSCSI か"](#)。



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 のその他の要件については、を参照してください ["こちらをご覧ください"](#)。

### 手順

1. のいずれかを使用してAstra Tridentを導入 ["導入方法"](#)。
2. Trident を設定する手順は次のとおりです。
  - a. SVM の管理 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. 認証用の証明書を作成してインストールします。を使用する場合 `ontap-san` バックエンド。を参照してください ["こちらをご覧ください"](#)。を使用する場合 `ontap-nas` バックエンド。を参照してください ["こちらをご覧ください"](#)。



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。

```
{
  "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"](#)



指定しないでください dataLIF をクリックします ontap-san および ontap-san-economy Astra Tridentがマルチパスを使用できるようにするためのドライバ。



。 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。

導入後、次の手順を実行してを作成します ["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"](#)。

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"](#)

## 著作権に関する情報

Copyright © 2024 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。