ONTAP NAS ドライバー

Trident

NetApp
January 15, 2026

目次

ONTAP NAS ドライバー

ONTAP NAS ドライバーの概要

ONTAPおよびCloud Volumes ONTAP NAS ドライバーを使用してONTAPバックエンドを構成する方法について学習します。

ONTAP NAS ドライバーの詳細

Trident は、 ONTAPクラスタと通信するための次の NAS ストレージ ドライバーを提供します。サポートされているアクセス モードは、*ReadWriteOnce* (RWO)、*ReadOnlyMany* (ROX)、*ReadWriteMany* (RWX)、*ReadWriteOncePod* (RWOP) です。

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	サポートされているファイルシステム
`ontap-nas`	NFS SMB	Filesystem	RWO、ROX、RWX、RWOP	"", `nfs` 、 `smb`
`ontap-nas-economy`	NFS SMB	Filesystem	RWO、ROX、RWX、RWOP	"", `nfs` 、 `smb`
`ontap-nas-flexgroup`	NFS SMB	Filesystem	RWO、ROX、RWX、RWOP	"", `nfs` 、 `smb`

- 使用 `ontap-san-economy`永続ボリュームの使用数が"サポートされているONTAPボリュームの制限"。
- 使用 `ontap-nas-economy`永続ボリュームの使用数が"サポートされているONTAPボリュームの制限"そして `ontap-san-economy`ドライバーは使用できません。
- 使用しないでください `ontap-nas-economy`データ保護、災害復旧、モビリティの必要性が予想される場合。
- NetApp、ontap-san を除くすべてのONTAPドライバーで Flexvol autogrow を使用することは推奨されていません。回避策として、 Trident はスナップショット リザーブの使用をサポートし、それに応じて Flexvol ボリュームを拡張します。

ユーザー権限

Tridentは、通常、 ONTAPまたはSVM管理者として実行することを想定しています。 `admin`クラスターユーザーまたは `vsadmin`SVM ユーザー、または同じロールを持つ別の名前のユーザー。

Amazon FSx for NetApp ONTAPの導入では、 Tridentはクラスタを使用してONTAPまたはSVM管理者として実行されることが想定されています。 `fsxadmin`ユーザーまたは `vsadmin`SVM ユーザー、または同じロールを持つ別の名前のユーザー。その `fsxadmin`ユーザーは、クラスター管理者ユーザーの限定的な代替です。

を使用する場合 `limitAggregateUsage`パラメータには、クラスター管理者の権限が必要です。Amazon FSx for NetApp ONTAPをTridentで使用する場合、 `limitAggregateUsage`パラメータは、 `vsadmin`そして `fsxadmin`ユーザーアカウント。このパラメータを指定すると、構成操作は失敗します。

ONTAP内でTridentドライバーが使用できる、より制限の厳しいロールを作成することは可能ですが、お勧めしません。 Tridentのほとんどの新しいリリースでは、考慮する必要がある追加の API が呼び出されるため、アップグレードが困難になり、エラーが発生しやすくなります。

ONTAP NASドライバーを使用してバックエンドを構成する準備をする

ONTAP NAS ドライバーを使用してONTAPバックエンドを構成するための要件、認証オプション、およびエクスポート ポリシーを理解します。

要件

- すべてのONTAPバックエンドでは、 Trident少なくとも 1 つのアグリゲートを SVM に割り当てる必要があります。
- 複数のドライバーを実行し、いずれかを指すストレージ クラスを作成できます。たとえば、 `ontap-nas`ドライバーとブロンズクラスは `ontap-nas-economy`1つ。
- すべての Kubernetes ワーカーノードに適切な NFS ツールがインストールされている必要があります。参照"ここをクリックしてください。"詳細についてはこちらをご覧ください。
- 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します。参照 SMBボリュームのプロビジョニングの準備 詳細については。

ONTAPバックエンドを認証する

Trident は、 ONTAPバックエンドを認証する 2 つのモードを提供します。

- 認証情報ベース: このモードでは、 ONTAPバックエンドに対する十分な権限が必要です。次のような事前定義されたセキュリティログインロールに関連付けられたアカウントを使用することをお勧めします。 `admin`または `vsadmin`ONTAPバージョンとの最大限の互換性を確保するためです。
- 証明書ベース: このモードでは、 Trident がONTAPクラスタと通信するために、バックエンドに証明書がインストールされている必要があります。ここで、バックエンド定義には、クライアント証明書、キー、および信頼された CA 証明書（使用する場合）の Base64 エンコードされた値が含まれている必要があります（推奨）。

既存のバックエンドを更新して、資格情報ベースの方法と証明書ベースの方法間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。別の認証方法に切り替えるには、バックエンド構成から既存の方法を削除する必要があります。

資格情報と証明書の両方 を提供しようとすると、構成ファイルに複数の認証方法が提供されているというエラーが発生し、バックエンドの作成が失敗します。

資格情報ベースの認証を有効にする

Trident、 ONTAPバックエンドと通信するために、SVM スコープ/クラスタ スコープの管理者の認証情報が必要です。次のような標準の事前定義されたロールを利用することをお勧めします。 `admin`または`vsadmin`。これにより、将来のTridentリリースで使用される機能 API を公開する可能性のある将来のONTAPリリースとの前方互換性が確保されます。カスタム セキュリティ ログイン ロールを作成してTridentで使用することは可能ですが、お勧めしません。

サンプルのバックエンド定義は次のようになります。

ヤムル

```
---
version: 1
backendName: ExampleBackend
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
dataLIF: 10.0.0.2
svm: svm_nfs
credentials:
  name: secret-backend-creds
```

JSON

```
{
  "version": 1,
  "backendName": "ExampleBackend",
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.0.0.1",
  "dataLIF": "10.0.0.2",
  "svm": "svm_nfs",
  "credentials": {
        "name": "secret-backend-creds"
    }
}
```

バックエンド定義は、資格情報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留意してください。バックエンドが作成されると、ユーザー名とパスワードは Base64 でエンコードされ、Kubernetes シークレットとして保存されます。バックエンドの作成/更新は、資格情報に関する知識が必要となる唯一のステップです。したがって、これは Kubernetes/ストレージ管理者によって実行される管理者専用の操作です。

証明書ベースの認証を有効にする

新規および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: クライアント証明書の Base64 エンコードされた値。
- clientPrivateKey: 関連付けられた秘密キーの Base64 エンコードされた値。
- trustedCACertificate: 信頼された CA 証明書の Base64 エンコードされた値。信頼できる 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。信頼できる CA が使用されていない場合は、これを無視できます。

一般的なワークフローには次の手順が含まれます。

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、認証するONTAPユーザーに共通名 (CN) 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. 信頼できる CA 証明書をONTAPクラスタに追加します。これはストレージ管理者によってすでに処理され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. クライアント証明書とキー（手順 1 から）をONTAPクラスタにインストールします。

```
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がサポートしていることを確認する `cert`認証方法。

```
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します。 < ONTAP Management LIF> と <vserver name> を管理 LIF IP と SVM 名に置き換えます。 LIFのサービスポリシーが次のよう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. 証明書、キー、および信頼された CA 証明書を Base64 でエンコードします。

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

7. 前の手順で取得し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...Vaaalllluuuueeee",
"clientPrivateKey": "LS0tFaKE...0VaLuES0tLS0K",
"storagePrefix": "myPrefix_"
}

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

認証方法を更新するか、資格情報をローテーションする

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、資格情報をローテーションしたりすることができます。これは両方向に機能します。ユーザー名/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。また、証明書を使用するバックエンドは、ユーザー名/パスワード ベースに更新できます。これを行うには、既存の認証方法を削除し、新しい認証方法を追加する必要があります。次に、必要なパラメータを含む更新されたbackend.jsonファイルを使用して実行します。 `tridentctl update backend` 。

```
cat cert-backend-updated.json
```

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

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

> パスワードをローテーションする場合、ストレージ管理者はまずONTAP上のユーザーのパスワードを更新する必要があります。続いてバックエンドの更新が行われます。証明書をローテーションする場合、ユーザーに複数の証明書を追加できます。その後、バックエンドは新しい証明書を使用するように更新され、その後、古い証明書を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、その後に行われたボリューム接続にも影響はありません。バックエンドの更新が成功すると、 Trident がONTAPバックエンドと

通信し、将来のボリューム操作を処理できることを示します。

Trident用のカスタムONTAPロールを作成する

最小限の権限を持つONTAPクラスタ ロールを作成すると、 Tridentで操作を実行するためにONTAP管理者ロールを使用する必要がなくなります。 Tridentバックエンド構成にユーザー名を含めると、 Trident は作成したONTAPクラスタ ロールを使用して操作を実行します。

参照"Tridentカスタムロールジェネレーター"Tridentカスタム ロールの作成の詳細については、こちらをご覧ください。

ONTAP CLIの使用

1. 次のコマンドを使用して新しいロールを作成します。

```
security login role create <role_name\> -cmddirname "command" -access all
–vserver <svm_name\>
```

2. Tridentユーザーのユーザー名を作成します。

```
security login create -username <user_name\> -application ontapi
-authmethod <password\> -role <name_of_role_in_step_1\> –vserver
<svm_name\> -comment "user_description"
```

3. ロールをユーザーにマップします。

```
security login modify username <user_name\> –vserver <svm_name\> -role
<role_name\> -application ontapi -application console -authmethod
<password\>
```

System Managerを使用

ONTAP System Manager で次の手順を実行します。

1. カスタムロールを作成する:
   a. クラスター レベルでカスタム ロールを作成するには、クラスター **>** 設定 を選択します。

      （または）SVMレベルでカスタムロールを作成するには、ストレージ **>** ストレージ**VM >** を選択します。 **`required SVM`** **>** 設定 **>** ユーザーとロール。

   b. ユーザーとロール*の横にある矢印アイコン (→*) を選択します。
   c. 役割*の下の+追加*を選択します。
   d. ロールのルールを定義し、「保存」をクリックします。
2. 役割を**Trident**ユーザーにマップします: + ユーザーと役割 ページで次の手順を実行します。
   a. ユーザー*の下の追加アイコン+*を選択します。
   b. 必要なユーザー名を選択し、*役割*のドロップダウン メニューで役割を選択します。
   c. *保存*をクリックします。

詳細については、次のページを参照してください。

- "ONTAPの管理用のカスタム ロール"または"カスタム ロールの定義"
- "役割とユーザーを操作する"

NFSエクスポートポリシーを管理する

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。

Trident、エクスポート ポリシーを操作するときに 2 つのオプションが提供されます。

- Trident はエクスポート ポリシー自体を動的に管理できます。この動作モードでは、ストレージ管理者は許容 IP アドレスを表す CIDR ブロックのリストを指定します。 Trident は、公開時にこれらの範囲内にある該当するノード IP をエクスポート ポリシーに自動的に追加します。あるいは、CIDR が指定されていない場合は、ボリュームが公開されるノードで見つかったすべてのグローバル 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、ルールを手動で追加できます。構成で別のエクスポート ポリシー名が指定されていない限り、 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する

Trident 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する機能を提供します。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 IP に許可されるアドレス空間を指定できるようになります。これにより、エクスポート ポリシーの管理が大幅に簡素化され、エクスポート ポリシーを変更する際にストレージ クラスターで手動で介入する必要がなくなります。さらに、これにより、ボリュームをマウントしており、指定された範囲内の IP を持つワーカー ノードのみにストレージ クラスターへのアクセスが制限され、きめ細かな自動管理がサポートされます。

> ⓘ 動的エクスポート ポリシーを使用する場合は、ネットワーク アドレス変換 (NAT) を使用しないでください。 NAT では、ストレージ コントローラは実際の IP ホスト アドレスではなくフロントエンド NAT アドレスを認識するため、エクスポート ルールに一致するものが見つからない場合はアクセスが拒否されます。

例

使用する必要がある構成オプションが 2 つあります。バックエンドの定義の例を次に示します。

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

この機能を使用する場合は、SVM のルート ジャンクションに、ノード CIDR ブロックを許可するエクスポート ルールを含むエクスポート ポリシーが以前に作成されていることを確認する必要があります (デフォルトのエクスポート ポリシーなど)。 SVM をTrident専用にする場合は、常にNetApp が推奨するベスト プラクティスに従ってください。

上記の例を使用して、この機能がどのように機能するかを説明します。

- `autoExportPolicy`設定されている `true`。これは、 Tridentがこのバックエンドでプロビジョニングされたボリュームごとにエクスポートポリシーを作成することを示します。 `svm1` SVMを使用してルールの追加と削除を処理します `autoexportCIDRs`アドレス ブロック。ボリュームがノードに接続されるまで、そのボリュームへの不要なアクセスを防ぐためのルールのない空のエクスポート ポリシーがボリュームで使用されます。ボリュームがノードに公開されると、 Trident は指定された CIDR ブロック内のノード IP を含む基礎となる qtree と同じ名前のエクスポート ポリシーを作成します。これらのIPは、親FlexVol volumeで使用されるエクスポートポリシーにも追加されます。
  - 例えば：
    - バックエンドUUID 403b5326-8482-40db-96d0-d83fb3f4daec
    - `autoExportPolicy`に設定 `true`
    - ストレージプレフィックス `trident`
    - PVC UUID a79bcf5f-7b6d-4a40-9876-e2551f159c1c
    - trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1cという名前のqtreeは、 FlexVolという名前のエクスポートポリシーを作成します。 `trident-403b5326-8482-40db96d0-d83fb3f4daec` 、qtreeのエクスポートポリシー `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c`、および空のエクスポートポリシー `trident_empty`SVM 上。 FlexVolエクスポート ポリシーのルールは、qtree エクスポート ポリシーに含まれるルールのスーパーセットになります。空のエクスポート ポリシーは、接続されていないボリュームによって再利用されます。
- `autoExportCIDRs`アドレス ブロックのリストが含まれます。このフィールドはオプションであり、デフォルトは ["0.0.0.0/0", "::/0"] になります。定義されていない場合、 Trident はパブリケーションを持つワーカー ノードで見つかったすべてのグローバル スコープのユニキャスト アドレスを追加します。

この例では、 `192.168.0.0/24`アドレス空間が提供されます。これは、公開されているこのアドレス範囲内にある Kubernetes ノード IP が、 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。Tridentは、そのノードを登録する際に、そのノードのIPアドレスを取得し、それを以下のアドレスブロックと照合します。 `autoExportCIDRs`公開時に、IP をフィルタリングした後、 Trident は公開先のノードのク

ライアント IP のエクスポート ポリシー ルールを作成します。

更新できます `autoExportPolicy`そして `autoExportCIDRs`バックエンドを作成した後。自動的に管理されるバックエンドに新しい CIDR を追加したり、既存の CIDR を削除したりできます。 CIDR を削除するときは、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`バックエンドにエクスポート ポリシーを手動で作成し、フォールバックします。これには、 `exportPolicy`バックエンド構成のパラメータ。

Tridentがバックエンドを作成または更新した後、以下のコマンドで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>
      fileSystemType: ext4
```

ノードが削除されると、 Trident はすべてのエクスポート ポリシーをチェックし、そのノードに対応するアクセス ルールを削除します。管理対象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 を削除することにより、この IP がクラスター内の新しいノードによって再利用されない限り、 Trident は不正なマウントを防止します。

既存のバックエンドの場合は、バックエンドを次のように更新します。 `tridentctl update backend` Trident がエクスポート ポリシーを自動的に管理することを保証します。これにより、必要に応じて、バックエンドの UUID と qtree 名に基づいて名前が付けられた 2 つの新しいエクスポート ポリシーが作成されます。バックエンドに存在するボリュームは、アンマウントされて再度マウントされた後、新しく作成されたエクスポート ポリシーを使用します。

> 自動管理エクスポート ポリシーを持つ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 ポリシーも削除されます。バックエンドが再作成されると、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Tridentポッドを再起動する必要があります。そ

の後、 Trident は、この IP 変更を反映するために、管理するバックエンドのエクスポート ポリシーを更新します。

SMBボリュームのプロビジョニングの準備

少しの追加の準備をすれば、SMBボリュームをプロビジョニングできます。 `ontap-nas`ドライバー。

> SVMでNFSとSMB/CIFSプロトコルの両方を設定する必要があります。 `ontap-nas-economy` ONTAPオンプレミス クラスターの SMB ボリューム。これらのプロトコルのいずれかを構成しないと、SMB ボリュームの作成が失敗します。

> `autoExportPolicy`SMB ボリュームではサポートされません。

開始する前に

SMB ボリュームをプロビジョニングする前に、次のものが必要です。

- Linux コントローラー ノードと、Windows Server 2022 を実行する少なくとも 1 つの Windows ワーカー ノードを備えた Kubernetes クラスター。 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します。
- Active Directory 資格情報を含む少なくとも 1 つのTridentシークレット。秘密を生成する `smbcreds`:

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

- Windows サービスとして構成された CSI プロキシ。設定するには `csi-proxy`、参照"GitHub: CSIプロキシ"または"GitHub: Windows 用 CSI プロキシ"Windows 上で実行されている Kubernetes ノード用。

手順

1. オンプレミスのONTAPの場合、オプションで SMB 共有を作成するか、 Trident で作成することもできます。

   > Amazon FSx for ONTAPには SMB 共有が必要です。

   SMB管理共有は、次の2つの方法のいずれかで作成できます。"Microsoft管理コンソール"共有フォルダ スナップインまたはONTAP CLI を使用します。 ONTAP CLI を使用して SMB 共有を作成するには、次の手順を実行します。

   a. 必要に応じて、共有のディレクトリ パス構造を作成します。

      その `vserver cifs share create`コマンドは、共有の作成中に -path オプションで指定されたパスをチェックします。指定されたパスが存在しない場合、コマンドは失敗します。

   b. 指定された SVM に関連付けられた SMB 共有を作成します。

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

c. 共有が作成されたことを確認します。

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

参照"SMB共有を作成する"詳細についてはこちらをご覧ください。

2. バックエンドを作成するときは、SMB ボリュームを指定するために以下を構成する必要があります。FSx for ONTAPのバックエンド構成オプションについては、以下を参照してください。"FSx for ONTAP の構成オプションと例" 。

パラメータ	説明	例
`smbShare`	次のいずれかを指定できます: Microsoft 管理コンソールまたはONTAP CLI を使用して作成された SMB 共有の名前、 Trident がSMB 共有を作成できるようにする名前、またはボリュームへの共通共有アクセスを防止するためにパラメータを空白のままにしておくことができます。このパラメータは、オンプレミスのONTAPではオプションです。このパラメータはAmazon FSx for ONTAPバックエンドに必須であり、空白にすることはできません。	`smb-share`
`nasType`	設定する必要があります **`smb`**. nullの場合、デフォルトは `nfs`。	`smb`
`securityStyle`	新しいボリュームのセキュリティ スタイル。 設定する必要があります **`ntfs`**または **`mixed`SMB** ボリュームの場合。	`ntfs`または `mixed`SMB ボリューム用
`unixPermissions`	新しいボリュームのモード。 **SMB** ボリュームの場合は空のままにする必要があります。	""

安全な**SMB**を有効にする

25.06リリース以降、 NetApp Tridentは、以下の方法で作成されたSMBボリュームの安全なプロビジョニングをサポートします。 `ontap-nas`そして `ontap-nas-economy`バックエンド。セキュア SMB を有効にすると、アクセス制御リスト (ACL) を使用して、Active Directory (AD) ユーザーおよびユーザー グループに SMB 共有への制御されたアクセスを提供できます。

覚えておくべきポイント

- インポート `ontap-nas-economy`ボリュームはサポートされていません。
- 読み取り専用クローンのみがサポートされています `ontap-nas-economy`ボリューム。
- Secure SMB が有効になっている場合、 Trident はバックエンドに記載されている SMB 共有を無視します。

- PVC アノテーション、ストレージ クラス アノテーション、およびバックエンド フィールドを更新しても、SMB 共有 ACL は更新されません。
- クローン PVC の注釈で指定された SMB 共有 ACL は、ソース PVC の ACL よりも優先されます。
- 安全な SMB を有効にする際には、有効な AD ユーザーを提供するようにしてください。無効なユーザーは ACL に追加されません。
- バックエンド、ストレージ クラス、PVC で同じ AD ユーザーに異なる権限を指定した場合、権限の優先順位は PVC、ストレージ クラス、バックエンドの順になります。
- セキュアSMBは以下でサポートされています `ontap-nas`管理対象ボリュームのインポートには適用され、管理対象外ボリュームのインポートには適用されません。

手順

1. 次の例に示すように、TridentBackendConfig で adAdminUser を指定します。

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.193.176.x
  svm: svm0
  useREST: true
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

2. ストレージ クラスに注釈を追加します。

   追加する `trident.netapp.io/smbShareAdUser`ストレージ クラスにアノテーションを追加して、安全な SMB を確実に有効にします。注釈に指定されたユーザー値 `trident.netapp.io/smbShareAdUser`指定されたユーザー名と同じである必要があります `smbcreds`秘密。次のいずれかを選択できます `smbShareAdUserPermission`: `full_control` 、 `change` 、 または `read`。デフォルトの権限は `full_control`。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

1. PVCを作成します。

   次の例では、PVC を作成します。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/snapshotDirectory: "true"
    trident.netapp.io/smbShareAccessControl: |
      read:
        - tridentADtest
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

ONTAP NAS の構成オプションと例

TridentインストールでONTAP NAS ドライバーを作成して使用する方法を学習します。このセクションでは、バックエンドを StorageClasses にマッピングするためのバックエンド構成の例と詳細について説明します。

バックエンド構成オプション

バックエンドの構成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 ドライバーの名前	`ontap-nas`、 `ontap-nas-economy` 、 または `ontap-nas-flexgroup`
`backendName`	カスタム名またはストレージバックエンド	ドライバー名 + "_" + dataLIF
`managementLIF`	クラスタまたは SVM 管理 LIF の IP アドレス。完全修飾ドメイン名 (FQDN) を指定できます。 TridentがIPv6 フラグを使用してインストールされている場合は、IPv6 アドレスを使用するように設定できます。 IPv6アドレスは角括弧で囲んで定義する必要があります。例: `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]` 。シームレスなMetroClusterスイッチオーバーについては、MetroClusterの例 。	"10.0.0.1"、"[2001:1234:abcd::fefe]"
`dataLIF`	プロトコル LIF の IP アドレス。 NetAppは以下を指定することを推奨しています `dataLIF`。指定されない場合、 Trident はSVM から dataLIF を取得します。NFS マウント操作に使用する完全修飾ドメイン名 (FQDN) を指定できるため、複数のデータ LIF 間で負荷分散を行うラウンドロビン DNS を作成できます。初期設定後も変更可能です。参照。 Trident がIPv6 フラグを使用してインストールされている場合は、IPv6 アドレスを使用するように設定できます。 IPv6アドレスは角括弧で囲んで定義する必要があります。例: `[28e8:d9fb:a825:b7bf:69a8:d02f:9e7b:3555]` 。 *Metrocluster の場合は省略します。*参照MetroClusterの例。	指定されたアドレス、または指定されていない場合は SVM から派生したアドレス（非推奨）
`svm`	使用するストレージ仮想マシン *Metrocluster の場合は省略。*参照MetroClusterの例。	SVMの場合導出される `managementLIF`指定されている
`autoExportPolicy`	自動エクスポート ポリシーの作成と更新を有効にします [ブール値]。使用して `autoExportPolicy`そして `autoExportCIDRs`オプションを使用すると、 Tridentはエクスポート ポリシーを自動的に管理できます。	false
`autoExportCIDRs`	KubernetesのノードIPをフィルタリングするためのCIDRのリスト `autoExportPolicy`が有効になります。使用して `autoExportPolicy`そして `autoExportCIDRs`オプションを使用すると、 Tridentはエクスポート ポリシーを自動的に管理できます。	["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 エンコードされた値。証明書ベースの認証に使用	""

パラメータ	説明	デフォルト
`clientPrivateKey`	クライアント秘密キーの Base64 エンコードされた値。証明書ベースの認証に使用	""
`trustedCACertificate`	信頼された CA 証明書の Base64 エンコードされた値。オプション。証明書ベースの認証に使用	""
`username`	クラスタ/SVM に接続するためのユーザー名。資格情報ベースの認証に使用されます。Active Directory認証については、 "Active Directory の認証情報を使用して、バックエンド SVM に対してTrident を認証する"。	
`password`	クラスター/SVM に接続するためのパスワード。資格情報ベースの認証に使用されます。Active Directory認証については、 "Active Directory の認証情報を使用して、バックエンド SVM に対してTrident を認証する"。	
`storagePrefix`	SVM で新しいボリュームをプロビジョニングするときに使用されるプレフィックス。設定後は更新できません ontap-nas-economy と 24 文字以上の storagePrefix を使用する場合、ボリューム名にはストレージ プレフィックスが埋め込まれますが、qtree には埋め込まれません。	"トライデント"
`aggregate`	プロビジョニング用のアグリゲート (オプション。設定する場合は、SVM に割り当てる必要があります)。のために `ontap-nas-flexgroup`ドライバーの場合、このオプションは無視されます。割り当てられていない場合は、使用可能なアグリゲートのいずれかを使用してFlexGroupボリュームをプロビジョニングできます。 SVM でアグリゲートが更新されると、Tridentコントローラーを再起動せずに、SVM をポーリングすることによってTridentでも自動的に更新されます。Tridentで特定のアグリゲートをボリュームのプロビジョニング用に構成した場合、アグリゲートの名前が変更されたり、SVM から移動されたりすると、SVM アグリゲートをポーリングしているときに、 Tridentでバックエンドが障害状態になります。バックエンドをオンラインに戻すには、アグリゲートを SVM 上に存在するものに変更するか、完全に削除する必要があります。	""

パラメータ	説明	デフォルト
`limitAggregateUsage`	使用率がこのパーセンテージを超える場合、プロビジョニングは失敗します。 * Amazon FSx for ONTAPには適用されません*。	"" (デフォルトでは強制されません)
flexgroupAggregateList	プロビジョニング用のアグリゲートのリスト (オプション。設定する場合は、SVM に割り当てる必要があります)。 SVM に割り当てられたすべてのアグリゲートは、 FlexGroupボリュームのプロビジョニングに使用されます。 **ontap-nas-flexgroup** ストレージ ドライバーでサポートされています。 SVM で集計リストが更新されると、Tridentコントローラーを再起動せずに、SVM をポーリングすることによってTridentのリストが自動的に更新されます。ボリュームをプロビジョニングするためにTridentで特定の集約リストを設定した場合、集約リストの名前が変更されたり、SVM から移動されたりすると、SVM 集約をポーリングしているときに、 Tridentでバックエンドが障害状態になります。バックエンドをオンラインに戻すには、集約リストをSVM 上に存在するリストに変更するか、集約リストを完全に削除する必要があります。	""
`limitVolumeSize`	要求されたボリューム サイズがこの値を超える場合、プロビジョニングは失敗します。また、qtreeの管理対象となるボリュームの最大サイズを制限し、`qtreesPerFlexvol`オプションにより、 FlexVol volumeあたりの qtree の最大数をカスタマイズできます。	"" (デフォルトでは強制されません)
`debugTraceFlags`	トラブルシューティング時に使用するデバッグ フラグ。例: {"api":false, "method":true} 使用しないでください `debugTraceFlags`ただし、トラブルシューティングを行っており、詳細なログ ダンプが必要な場合を除きます。	ヌル
`nasType`	NFS または SMB ボリュームの作成を構成します。オプションは `nfs`、 `smb`または null。 null に設定すると、デフォルトで NFS ボリュームになります。	`nfs`

パラメータ	説明	デフォルト
`nfsMountOptions`	NFS マウント オプションのコンマ区切りリスト。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、 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たマウント 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、 Trident は関連付けられた永続ボリュームにマウント オプションを設定しません。	""
`qtreesPerFlexvol`	FlexVolあたりの最大Qtree数は、[50, 300]の範囲でなければなりません	「200」
`smbShare`	次のいずれかを指定できます: Microsoft 管理コンソールまたはONTAP CLI を使用して作成された SMB 共有の名前、 Trident がSMB 共有を作成できるようにする名前、またはボリュームへの共通共有アクセスを防止するためにパラメータを空白のままにしておくことができます。このパラメータは、オンプレミスのONTAPではオプションです。このパラメータはAmazon FSx for ONTAPバックエンドに必須であり、空白にすることはできません。	`smb-share`
`useREST`	ONTAP REST API を使用するためのブール パラメーター。 `useREST`に設定すると `true`、 TridentはONTAP REST APIを使用してバックエンドと通信します。 `false` Trident は、バックエンドとの通信にONTAPI (ZAPI) 呼び出しを使用します。この機能にはONTAP 9.11.1 以降が必要です。さらに、使用するONTAPログインロールには、 `ontapi`応用。これは、事前に定義された `vsadmin`そして `cluster-admin`役割。 Trident 24.06リリースおよびONTAP 9.15.1以降では、 `useREST`設定されている `true`デフォルト; 変更 `useREST`に `false`ONTAPI (ZAPI) 呼び出しを使用します。	`true`ONTAP 9.15.1以降の場合、それ以外の場合 `false`。
`limitVolumePoolSize`	ontap-nas-economy バックエンドで Qtree を使用する場合の、要求可能なFlexVol の最大サイズ。	"" (デフォルトでは強制されません)
`denyNewVolumePools`	制限 `ontap-nas-economy`バックエンドが Qtree を格納するための新しいFlexVolボリュームを作成できないようにします。新しい PV のプロビジョニングには、既存の Flexvol のみが使用されます。	
`adAdminUser`	SMB 共有へのフルアクセス権を持つ Active Directory 管理者ユーザーまたはユーザー グループ。このパラメータを使用して、SMB 共有への完全な制御権限を持つ管理者権限を付与します。	

ボリュームのプロビジョニングのためのバックエンド構成オプション

デフォルトのプロビジョニングは、以下のオプションを使用して制御できます。 `defaults`構成のセクション。例については、以下の構成例を参照してください。

パラメータ	説明	デフォルト
`spaceAllocation`	Qtreeのスペース割り当て	"真実"
`spaceReserve`	スペース予約モード。「なし」（薄い）または「ボリューム」（厚い）	"なし"
`snapshotPolicy`	使用するスナップショットポリシー	"なし"
`qosPolicy`	作成され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`	作成されたボリュームに割り当てるアダプティブ 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。 ontap-nas-economy ではサポートされていません。	""
`snapshotReserve`	スナップショット用に予約されているボリュームの割合	「0」の場合 `snapshotPolicy`は「なし」、それ以外の場合は「」
`splitOnClone`	クローン作成時に親からクローンを分割する	"間違い"
`encryption`	新しいボリュームでNetAppボリューム暗号化（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、NVE のライセンスを取得し、クラスターで有効にする必要があります。バックエンドで NAE が有効になっている場合、 Tridentでプロビジョニングされたすべてのボリュームで NAE が有効になります。詳細については、以下を参照してください。"Trident がNVE および NAE と連携する仕組み" 。	"間違い"
`tieringPolicy`	「なし」を使用する階層化ポリシー	
`unixPermissions`	新しいボリュームのモード	NFSボリュームの場合は「777」、SMBボリュームの場合は空（該当なし）
`snapshotDir`	アクセスを制御します `.snapshot`ディレクトリ	NFSv4の場合は「true」、NFSv3の場合は「false」
`exportPolicy`	使用するエクスポートポリシー	"デフォルト"
`securityStyle`	新しいボリュームのセキュリティ スタイル。 NFSサポート `mixed`そして `unix`セキュリティ スタイル。 SMBサポート `mixed`そして `ntfs`セキュリティ スタイル。	NFSのデフォルトは `unix`。 SMBのデフォルトは `ntfs`。
`nameTemplate`	カスタムボリューム名を作成するためのテンプレート。	""

Trident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。共有されていない QoS ポリシー グループを使用し、ポリシー グループが各構成要素に個別に適用されるようにする必要があります。共有 QoS ポリシー グループは、すべてのワークロードの合計スループットの上限を適用します。

ボリュームプロビジョニングの例

デフォルトを定義した例を次に示します。

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

のために ontap-nas`そして `ontap-nas-flexgroups`Trident、新しい計算を使用して、SnapshotReserve のパーセンテージと PVC に合わせてFlexVol のサイズが適切に設定されるようになりました。ユーザーがPVCを要求すると、 Tridentは新しい計算方法を用いて、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、ユーザーはPVCで要求した書き込み可能なスペースを確実に受け取り、要求したスペースよりも少ないスペースを受け取ることはありません。v21.07より前のバージョンでは、ユーザーがSnapshotReserveを50%に設定してPVC（例えば5GiB）を要求した場合、書き込み可能なスペースは2.5GiBしか得られませんでした。これは、ユーザーが要求したのは全巻であり、`snapshotReserve`それはそのパーセンテージです。 Trident 21.07では、ユーザーが要求するのは書き込み可能なスペースであり、 Tridentはそれを定義します。 `snapshotReserve`全体の量の割合として数値を表示します。これは適用されません `ontap-nas-economy。これがどのように機能するかを確認するには、次の例を参照してください

計算は次のようになります。

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

snapshotReserve = 50%、PVCリクエスト = 5 GiBの場合、ボリュームの合計サイズは5/0.5 = 10 GiBとなり、使用可能なサイズはユーザーがPVCリクエストで要求した5 GiBになりますその `volume show`コマンドを実行すると、次の例のような結果が表示されます。

以前のインストールからの既存のバックエンドは、 Tridentをアップグレードする際に、上記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、変更を反映させるためにボリュームのサイズを変更する必要があります。例えば、2GiBのPVCで `snapshotReserve=50`以前は、1 GiB の書き込み可能なスペースを提供するボリュームが作成されました。例えば、ボリュームを3GiBにサイズ変更すると、6GiBのボリュームで3GiBの書き込み可能領域がアプリケーションに提供されます。

最小限の構成例

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構成を示しています。これはバックエンドを定義する最も簡単な方法です。

> Tridentを搭載したNetApp ONTAPでAmazon FSx を使用している場合は、LIF に IP アドレスではなく DNS 名を指定することをお勧めします。

ONTAP NASエコノミーの例

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

ONTAP NAS Flexgroupの例

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

MetroClusterの例

バックエンドを設定することで、スイッチオーバーとスイッチバック後にバックエンド定義を手動で更新する必要がなくなります。"SVMのレプリケーションとリカバリ"。

シームレスなスイッチオーバーとスイッチバックを行うには、SVMを次のように指定します。`managementLIF`そして省略する `dataLIF`そして `svm`パラメータ。例えば：

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 192.168.1.66
username: vsadmin
password: password
```

SMBボリュームの例

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

証明書ベースの認証の例

これは最小限のバックエンド構成の例です。 `clientCertificate` 、 `clientPrivateKey` 、 そして `trustedCACertificate`（信頼できるCAを使用する場合はオプション）が入力されます `backend.json`クライアント証明書、秘密キー、信頼できる CA 証明書の base64 エンコードされた値をそれぞれ取得します。

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

自動エクスポートポリシーの例

この例では、動的エクスポート 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Tridentに指示する方法を示します。これは、 `ontap-nas-economy`そして `ontap-nas-flexgroup` ドライバー。

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

IPv6アドレスの例

この例は `managementLIF`IPv6 アドレスを使用します。

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

SMB ボリュームを使用した**Amazon FSx for ONTAP**の例

その `smbShare`SMB ボリュームを使用する FSx for ONTAPにはこのパラメータが必要です。

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

nameTemplate を使用したバックエンド構成の例

```
---
version: 1
storageDriverName: ontap-nas
backendName: ontap-nas-backend
managementLIF: <ip address>
svm: svm0
username: <admin>
password: <password>
defaults:
  nameTemplate:
"{{.volume.Name}}_{{.labels.cluster}}_{{.volume.Namespace}}_{{.vo\
    lume.RequestName}}"
labels:
  cluster: ClusterA
  PVC: "{{.volume.Namespace}}_{{.volume.RequestName}}"
```

仮想プールを備えたバックエンドの例

以下に示すサンプルのバックエンド定義ファイルでは、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`どれも、 `spaceAllocation`偽で、そして `encryption`偽です。仮想プールはストレージ セクションで定義されます。

Trident は、 「コメント」フィールドにプロビジョニング ラベルを設定します。 FlexVolにコメントが設定されている `ontap-nas`または`FlexGroup `ontap-nas-flexgroup`。 Trident は、プロビジョニング時に仮想プールに存在するすべてのラベルをストレージ ボリュームにコピーします。便宜上、ストレージ管理者は仮想プールごとにラベルを定義し、ラベルごとにボリュームをグループ化できます。

これらの例では、一部のストレージプールは独自の `spaceReserve`、 `spaceAllocation`、 そして `encryption`値があり、一部のプールはデフォルト値を上書きします。

ONTAP NASの例

```
---
version: 1
storageDriverName: ontap-nas
managementLIF: 10.0.0.1
svm: svm_nfs
username: admin
password: <password>
nfsMountOptions: nfsvers=4
defaults:
  spaceReserve: none
  encryption: "false"
  qosPolicy: standard
labels:
  store: nas_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      app: msoffice
      cost: "100"
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: "true"
      unixPermissions: "0755"
      adaptiveQosPolicy: adaptive-premium
  - labels:
      app: slack
      cost: "75"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      department: legal
      creditpoints: "5000"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
```

```
      app: wordpress
      cost: "50"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
  - labels:
      app: mysqldb
      cost: "25"
    zone: us_east_1d
    defaults:
      spaceReserve: volume
      encryption: "false"
      unixPermissions: "0775"
```

ONTAP NAS FlexGroupの例

```
---
version: 1
storageDriverName: ontap-nas-flexgroup
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: <password>
defaults:
  spaceReserve: none
  encryption: "false"
labels:
  store: flexgroup_store
  k8scluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      protection: gold
      creditpoints: "50000"
    zone: us_east_1a
    defaults:
      spaceReserve: volume
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      protection: gold
      creditpoints: "30000"
    zone: us_east_1b
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0755"
  - labels:
      protection: silver
      creditpoints: "20000"
    zone: us_east_1c
    defaults:
      spaceReserve: none
      encryption: "true"
      unixPermissions: "0775"
  - labels:
      protection: bronze
      creditpoints: "10000"
    zone: us_east_1d
```

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

ONTAP NASエコノミーの例

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

```
      spaceReserve: volume
      encryption: "false"
      unixPermissions: "0775"
```

バックエンドを**StorageClasses**にマッピングする

以下のStorageClass定義は、[仮想プールを備えたバックエンドの例] 。使用して `parameters.selector`フィールドでは、各 StorageClass はボリュームをホストするため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で定義された側面が設定されます。

- その `protection-gold`ストレージクラスは、 `ontap-nas-flexgroup`バックエンド。これらはゴールド レベルの保護を提供する唯一のプールです。

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

- その `protection-not-gold`ストレージクラスは、 `ontap-nas-flexgroup`バックエンド。これらは、ゴールド以外の保護レベルを提供する唯一のプールです。

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

- その `app-mysqldb`ストレージクラスは、 `ontap-nas`バックエンド。これは、mysqldb タイプのアプリ用のストレージ プール構成を提供する唯一のプールです。

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

- TThe `protection-silver-creditpoints-20k`ストレージクラスは、 `ontap-nas-flexgroup`バックエンド。これは、シルバー レベルの保護と 20,000 クレジット ポイントを提供する唯一のプールです。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: protection-silver-creditpoints-20k
provisioner: csi.trident.netapp.io
parameters:
  selector: "protection=silver; creditpoints=20000"
  fsType: "ext4"
```

- その `creditpoints-5k`ストレージクラスは、 `ontap-nas`バックエンドと2番目の仮想プール `ontap-nas-economy`バックエンド。これらは 5000 クレジットポイントで提供される唯一のプールです。

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

Trident はどの仮想プールが選択されるか決定し、ストレージ要件が満たされていることを確認します。

アップデート `dataLIF`初期設定後

次のコマンドを実行して、更新された dataLIF を含む新しいバックエンド JSON ファイルを提供することにより、初期構成後に dataLIF を変更できます。

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

PVC が 1 つまたは複数のポッドに接続されている場合、新しい dataLIF を有効にするには、対応するすべてのポッドを停止してから再度起動する必要があります。

セキュアな中小企業の例

ontap-nas ドライバーを使用したバックエンド構成

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

ontap-nas-economy ドライバーを使用したバックエンド構成

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas-economy
  managementLIF: 10.0.0.1
  svm: svm2
  nasType: smb
  defaults:
    adAdminUser: tridentADtest
  credentials:
    name: backend-tbc-ontap-invest-secret
```

ストレージプールを使用したバックエンド構成

```
apiVersion: trident.netapp.io/v1
kind: TridentBackendConfig
metadata:
  name: backend-tbc-ontap-nas
  namespace: trident
spec:
  version: 1
  storageDriverName: ontap-nas
  managementLIF: 10.0.0.1
  svm: svm0
  useREST: false
  storage:
  - labels:
      app: msoffice
    defaults:
      adAdminUser: tridentADuser
  nasType: smb
  credentials:
    name: backend-tbc-ontap-invest-secret
```

ontap-nas ドライバーを使用したストレージクラスの例

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADtest
parameters:
  backendType: ontap-nas
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

必ず追加してください `annotations`安全な SMB を有効にします。バックエンドまたは PVC で設定された構成に関係なく、アノテーションがないとセキュア SMB は機能しません。

ontap-nas-economy ドライバーを使用したストレージクラスの例

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-smb-sc
  annotations:
    trident.netapp.io/smbShareAdUserPermission: change
    trident.netapp.io/smbShareAdUser: tridentADuser3
parameters:
  backendType: ontap-nas-economy
  csi.storage.k8s.io/node-stage-secret-name: smbcreds
  csi.storage.k8s.io/node-stage-secret-namespace: trident
  trident.netapp.io/nasType: smb
provisioner: csi.trident.netapp.io
reclaimPolicy: Delete
volumeBindingMode: Immediate
```

単一の **AD** ユーザーによる **PVC** の例

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc4
  namespace: trident
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      change:
        - tridentADtest
      read:
        - tridentADuser
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: ontap-smb-sc
```

複数の **AD** ユーザーによる **PVC** の例

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-test-pvc
  annotations:
    trident.netapp.io/smbShareAccessControl: |
      full_control:
        - tridentTestuser
        - tridentuser
        - tridentTestuser1
        - tridentuser1
      change:
        - tridentADuser
        - tridentADuser1
        - tridentADuser4
        - tridentTestuser2
      read:
        - tridentTestuser2
        - tridentTestuser3
        - tridentADuser2
        - tridentADuser3
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

著作権に関する情報

商標に関する情報