ONTAP NASドライバー

Trident

NetApp
July 01, 2026

目次

ONTAP NASドライバー

ONTAP NAS ドライバの概要

ONTAP および Cloud Volumes ONTAP NAS ドライバーを使用した ONTAP バックエンドの設定方法について説明します。

ONTAP NAS ドライバーの詳細

Tridentは、ONTAPクラスタと通信するために次のNASストレージドライバを提供します。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。

Driver	プロトコル	volumeMode	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`

- `ontap-san-economy`を使用するのは、永続ボリュームの使用数が"サポートされているONTAPボリューム制限"を超えることが予想される場合のみです。
- `ontap-nas-economy`を使用するのは、永続ボリュームの使用数が"サポートされているONTAPボリューム制限"を超えることが予想され、かつ `ontap-san-economy`ドライバーを使用できない場合のみです。
- データ保護、ディザスタリカバリ、モビリティの必要性が予想される場合は、使用しないでください `ontap-nas-economy`。
- NetAppでは、ontap-san以外のすべてのONTAPドライバーでFlexvolの自動拡張を使用することは推奨されません。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、ONTAPまたはSVM管理者として実行されることが想定されており、通常は `admin`クラスタユーザーまたは `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を使用します。

Amazon FSx for NetApp ONTAP環境では、TridentはONTAPまたはSVM管理者として実行されることが想定されており、クラスタ `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を使用します。 `fsxadmin`ユーザーは、クラスタ管理者ユーザーの限定的な代替です。

`limitAggregateUsage`パラメータを使用する場合は、クラスタ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`および `fsxadmin`ユーザアカウントでは機能しません。このパラメータを指定すると、設定処理は失敗します。

ONTAP 内でより制限的なロールを作成し、Trident ドライバーで使用することは可能ですが、推奨しません。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 ドライバを使用してバックエンドを設定する準備をする

ONTAP NAS ドライバーを使用した ONTAP バックエンドの設定に関する要件、認証オプション、エクスポートポリシーを理解します。25.10リリース以降、NetApp Tridentは"NetApp AFX ストレージ システム"をサポートします。NetApp AFXストレージシステムは、ストレージ レイヤの実装において、他のONTAPシステム（ASA、AFF、FAS）とは異なります。Trident バックエンド構成では、システムが AFX であることを指定する必要はありません。 `ontap-nas`を `storageDriverName`として選択すると、Trident は AFX システムを自動的に検出します。

`ontap-nas`ドライバ（NFS プロトコル）のみが AFX システムでサポートされています。SMB プロトコルはサポートされていません。

要件

- すべての ONTAP バックエンドで、Trident では少なくとも 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`ドライバーを使用するGoldクラスと、 `ontap-nas-economy`を使用するBronzeクラスを設定できます。
- 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。詳細については、"ここをクリックしてください。"を参照してください。
- Trident は、 Windows ノード上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。詳細については、SMB ボリュームのプロビジョニングの準備を参照してください。

ONTAP バックエンドを認証します

Trident では、 ONTAP バックエンドを認証する 2 つのモードが用意されています。

- 資格情報ベース：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。ONTAPバージョンとの最大限の互換性を確保するために、 `admin`や `vsadmin`などの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。
- 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をインストールする必要があります。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。

クレデンシャルベースの認証を有効にする

Trident が ONTAP バックエンドと通信するには、 SVM スコープ / クラスタスコープの管理者のクレデンシャルが必要です。 `admin`や `vsadmin`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。これにより、将来の ONTAP リリースで公開される可能性のある機能 API を将来の Trident リリースで使用できるように、上位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 で使用することもできますが、推奨されません。

サンプルのバックエンド定義は次のようになります：

YAML

```yaml
---
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

```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、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留意してください。バックエンドが作成されると、ユーザ名/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、クレデンシャル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、Common Name（CN）を認証する ONTAP ユーザーに設定します。

```
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 `cert`認証方法をサポートしていることを確認します。

```
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 `default-data-management`に設定されていることを確認する必要があります。

```
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、クレデンシャル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。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 `tridentctl update backend`を実行します。

```
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",
"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 で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。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。バックエンドのアップデートが成功したということは、TridentがONTAPバックエンドと通信でき、今後のボリューム操作を処理できることを示しています。

Trident 用のカスタム **ONTAP** ロールを作成します

最小限の権限を持つONTAPクラスタロールを作成することで、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。

Tridentカスタムロールの作成の詳細については、"Tridentカスタムロールジェネレーター"を参照してください。

ONTAPコマンドラインの使用

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. クラスタレベルでカスタムロールを作成するには、* Cluster > Settings *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 > required SVM> 設定 > ユーザーとロール*を選択します。

   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. **Roles***の下の+Add*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. **Trident**ユーザーに役割をマッピングする：+*ユーザーとロール*ページで次の手順を実行します：
   a. ユーザー*の下にある追加アイコン+*を選択します。
   b. 必要なユーザー名を選択し、*Role*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"
- "ロール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 は、エクスポート ポリシーを操作するときに 2 つのオプションを提供します。

- Tridentは、エクスポート ルール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、許容される 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 ルール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 ルールに追加されます。
- 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。Tridentは、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、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、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、事前に作成されたエクスポートポリシー（たとえばデフォルトのエクスポートポリシー）があることを必ず確認してください。常に、NetAppが推奨するベストプラクティスに従い、Trident専用のSVMを用意してください。

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`が `true`に設定されています。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各ボリュームの `svm1` SVM用のエクスポート ポリシーを作成し、 `autoexportCIDRs` アドレス ブロックを使用してルールの追加と削除を処理することを示しています。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に、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が使用するエクスポート ポリシーにも追加されます。
  - 次に例を示します。
    - 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 `trident-403b5326-8482-40db96d0-d83fb3f4daec`のエクスポート ポリシー、qtree という名前の `trident_pvc_a79bcf5f_7b6d_4a40_9876_e2551f159c1c`のエクスポート ポリシー、および SVM 上の `trident_empty`という名前の空のエクスポート ポリシーを作成します。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すべての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`にはアドレス 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 ノードで検出され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、このアドレス範囲内にあるKubernetesノードのIPが、公開されている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が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、 `autoExportCIDRs`で提供されたアドレスブロックと照合します。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。

> `autoExportPolicy`と `autoExportCIDRs`は、バックエンドを作成した後に更新できます。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。バックエンドの `autoExportPolicy`を無効にして、手動で作成したエクスポート ポリシーにフォールバックすることもできます。これには、バックエンド構成で `exportPolicy`パラメータを設定する必要があります。

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

```
./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て、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 を削除することで、Trident は、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、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、 `tridentctl update backend`でバックエンドを更新することで、Tridentがエクスポート ポリシーを自動的に管理するようになります。これにより、バックエンドのUUIDとqtree名にちなんで名付けられた2つの新しいエクスポート ポリシーが必要に応じて作成されます。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。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、 `ontap-nas`ドライバーを使用してSMBボリュームをプロビジョニングできます。

ONTAP オンプレミスクラスタ用の `ontap-nas-economy` SMB ボリュームを作成するには、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。これらのプロトコルのいずれかを設定しないと、SMB ボリュームの作成が失敗します。

`autoExportPolicy` は 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 ノード上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 のクレデンシャルを含む少なくとも 1 つの Trident シークレット。シークレットを生成するには `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。 `csi-proxy`を設定するには、Windows 上で実行されている Kubernetes ノード用の"GitHub：CSI Proxy"または"GitHub：Windows用CSIプロキシ"を参照してください。

手順

1. オンプレミス ONTAP の場合、必要に応じて SMB 共有を作成するか、 Trident に作成させることができます。

   SMB 共有は Amazon FSx for ONTAP に必要です。

   SMB 管理共有は、"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 のいずれかを使用して、2 つの方法のいずれかで作成できます。ONTAP CLI を使用して SMB 共有を作成するには：

   a. 必要に応じて、共有のディレクトリ パス構造を作成します。

      `vserver cifs share create`コマンドは、共有の作成中に `-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
```

詳細については、"SMB共有を作成する"を参照してください。

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。すべての FSx for ONTAP バックエンドの設定オプションについては、"FSx for ONTAP 設定オプションと例"を参照してください。

パラメータ	概要	例
`smbShare`	次のいずれかを指定できます：Microsoft Management Console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、またはパラメータを空白のままにしてボリュームへの共通共有アクセスを防止できます。このパラメータは、オンプレミスONTAPではオプションです。このパラメータは、Amazon FSx for ONTAPバックエンドでは必須であり、空白にすることはできません。	`smb-share`
`nasType`	* `smb`に設定する必要があります。*nullの場合、デフォルトは `nfs`です。	`smb`
`securityStyle`	新しいボリュームのセキュリティ スタイル。**SMB** ボリュームの場合は、 **`ntfs`**または **`mixed`**に設定する必要があります。	`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。	""

セキュア**SMB**を有効にする

25.06リリース以降、NetApp Tridentは、 `ontap-nas`および `ontap-nas-economy`バックエンドを使用して作成された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。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

- `ontap-nas-economy`ボリュームのインポートはサポートされていません。
- `ontap-nas-economy`ボリュームでは、読み取り専用クローンのみがサポートされています。
- セキュア SMB が有効になっている場合、Trident はバックエンドで指定された 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 のアノテーションで指定された SMB 共有 ACL は、ソース PVC の ACL よりも優先されます。
- セキュアな SMB を有効にする際には、有効な AD ユーザーを指定してください。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 `ontap-nas`管理対象ボリュームのインポートでサポートされており、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. 次の例のように、TridentBackendConfig で adAdminUser を指定してください：

```
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。

   セキュアな SMB を確実に有効にするには、 `trident.netapp.io/smbShareAdUser`アノテーションをストレージ クラスに追加します。アノテーション `trident.netapp.io/smbShareAdUser`に指定されたユーザー値は、 `smbcreds`シークレットで指定されたユーザー名と同じである必要があります。 `smbShareAdUserPermission`には、 `full_control`、 `change`、または `read`のいずれかを選択できます。デフォルトの権限は `full_control`です。

```
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 構成オプションと例

Tridentインストールで使用するONTAP NASドライバーの作成方法と使用方法について説明します。このセクションでは、バックエンドの設定例と、バックエンドを StorageClasses にマッピングするための詳細について説明します。25.10リリース以

降、NetApp Tridentは"NetApp AFX ストレージ システム"をサポートします。NetApp AFXストレージシステムは、ストレージ レイヤの実装において、他のONTAPベースのシステム（ASA、AFF、FAS）とは異なります。

> `ontap-nas`ドライバー（NFSプロトコル付き）のみがNetApp AFXシステムでサポートされています。SMBプロトコルはサポートされていません。

バックエンド構成オプション

Tridentバックエンド構成では、システムがNetApp AFXストレージシステムであることを指定する必要はありません。 `ontap-nas`を `storageDriverName`として選択すると、TridentはAFXストレージシステムを自動的に検出します。一部のバックエンド構成パラメータは、AFXストレージシステムには適用できません。

以下の表は、バックエンドの設定オプションを示しています：

パラメータ	概要	デフォルト
`version`		常に1
`storageDriverName`	ストレージドライバーの名前 NetApp AFXシステムでは、 `ontap-nas`のみがサポートされます。	`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）も指定可能）Trident が IPv6 フラグを使用してインストールされている場合、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 の例を参照してください。	"10.0.0.1"、"[2001:1234:abcd::fefe]"
`dataLIF`	プロトコル LIF の IP アドレス。NetApp では `dataLIF`を指定することを推奨します。指定しない場合、`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`）を指定することで、ラウンドロビン `DNS` を作成し、複数の `dataLIF` 間で負荷分散を行うことができます。初期設定後でも変更可能です。を参照してください。`Trident` が `IPv6` フラグを使用してインストールされている場合、`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。MetroCluster の例を参照してください。	指定されたアドレス、または指定されていない場合は SVM から導出されます（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。MetroCluster の例を参照してください。	SVM `managementLIF`が指定されている場合に導出されます

パラメータ	概要	デフォルト
`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]。 `autoExportPolicy`および `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`	`autoExportPolicy`が有効になっている場合に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`および `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	""
`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。任意。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。クレデンシャル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 Trident を認証"を参照してください。	
`password`	クラスター/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。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 Trident を認証"を参照してください。	
`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 プレフィックスが含まれますが、qtree にはストレージ プレフィックスが埋め込まれません。	「trident」

パラメータ	概要	デフォルト
`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）。 `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、利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 Controller を再起動することなく、SVM をポーリングすることで Trident で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 で特定のアグリゲートを設定している場合、その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 アグリゲートのポーリング中にバックエンドが Trident で障害状態に移行します。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。 **AFX**ストレージシステムには指定しないでください。	""
`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。**AFX**ストレージシステムには指定しないでください。	""（デフォルトでは強制されません）

パラメータ	概要	デフォルト
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でアグリゲートリストが更新されると、Trident ControllerをTridentせずにSVMをポーリングすることで、リストはTridentで自動的に更新されます。Tridentでボリュームをプロビジョニングするために特定のアグリゲートリストを設定している場合、アグリゲート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アグリゲートのポーリング中にバックエンドがTridentで障害状態に移行します。バックエンドをオンラインに戻すには、アグリゲートリストをSVM上に存在するリストに変更するか、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。	""（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} `debugTraceFlags`を使用しないでください。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。	null
`nasType`	NFS または SMB ボリュームの作成を設定します。オプションは `nfs`、 `smb`、またはnullです。null に設定すると、デフォルトで NFS ボリュームになります。指定する場合は、**AFX**ストレージシステムでは常に`**nfs**`に設定してください。	`nfs`
`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 オプションを使用するようになります。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 は関連付けられている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、範囲 [50, 300] 内である必要があります	"200"

パラメータ	概要	デフォルト
smbShare	次のいずれかを指定できます：Microsoft Management Console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、またはパラメータを空白のままにしてボリュームへの共通共有アクセスを防止できます。このパラメータは、オンプレミスONTAPではオプションです。このパラメータは、Amazon FSx for ONTAPバックエンドでは必須であり、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リアン パラメータ。 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`に設定すると、TridentはONTAPI（ZAPI）呼び出しを使用してバックエンドと通信します。この機能にはONTAP 9.11.1以降が必要です。さらに、使用するONTAPログインロールには、`ontapi`アプリケーションへのアクセス権が必要です。これは、事前定義された `vsadmin`および `cluster-admin`ロールで満たされます。Trident 24.06リリースおよびONTAP 9.15.1以降では、 `useREST`はデフォルトで `true`に設定されます。ONTAPI（ZAPI）呼び出しを使用するには、 `useREST`を `false`に変更します。指定する場合は、**AFX**ストレージシステムでは常に `**true**`に設定してください。	ONTAP 9.15.1以降の場合は`true`、それ以外の場合は false。
limitVolumePoolSize	ontap-nas-economy バックエンドで Qtree を使用する場合の最大リクエスト可能 FlexVol サイズ。	""（デフォルトでは強制されません）
denyNewVolumePools	バックエンドが Qtree を格納する新しい FlexVol ボリュームを作成できないように制限 `ontap-nas-economy`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 共有へのフルコントロールの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用のバックエンド設定オプション

デフォルトのプロビジョニングは、設定の `defaults`セクションにあるこれらのオプションを使用して制御できます。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	概要	デフォルト
spaceAllocation	Qtreeのスペース割り当て	"true"
spaceReserve	スペース予約モード：「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	概要	デフォルト
`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てください	""
`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てください。ontap-nas-economy ではサポートされていません。	""
`snapshotReserve`	Snapshot 用に予約されているボリュームの割合	`snapshotPolicy`が「none」の場合は「0」、それ以外の場合は「」
`splitOnClone`	作成時にクローンを親から分離する	"false"
`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`です。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ボリュームはすべてNAEが有効になります。詳細については、次を参照してください：Tridentと NVE および NAE の連携。	"false"
`tieringPolicy`	階層化ポリシーで「none」を使用	
`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`	`.snapshot`ディレクトリへのアクセスを制御します	`true`、 `false` （明示的に設定）。
`exportPolicy`	使用するエクスポートポリシー	"default"
`securityStyle`	新しいボリュームのセキュリティ スタイル。NFSは `mixed`および `unix`セキュリティ スタイルをサポートします。SMBは `mixed`および `ntfs`セキュリティ スタイルをサポートします。	NFSのデフォルトは `unix`です。SMBのデフォルトは `ntfs`です。
`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"
```

`ontap-nas`および `ontap-nas-flexgroups`の場合、Tridentは新しい計算方法を使用して、FlexVolがsnapshotReserveのパーセンテージとPVCで正しくサイズ設定されるようにします。ユーザーがPVCを要求すると、Tridentは新しい計算方法を使用して、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ることができ、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より前では、ユーザーがPVC（たとえば5GiB）を要求し、snapshotReserveを50パーセントにすると、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ボリューム全体であり、`snapshotReserve`はその割合であるためです。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`snapshotReserve`の数値をボリューム全体の割合として定義します。これは `ontap-nas-economy`には適用されません。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：

計算は次のとおりです：

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

snapshotReserve = 50%、PVC要求 = 5 GiBの場合、ボリュームの合計サイズは5/.5 = 10 GiBとなり、使用可能なサイズは5 GiBになります。これは、ユーザーがPVC要求で要求したサイズです。 `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 のアップグレード時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の場合は、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。たとえば、 `snapshotReserve=50`を使用した 2 GiB の PVC では、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。たとえば、ボリュームのサイズを 3 GiB に変更すると、6 GiB のボリューム上で 3 GiB の書き込み可能な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。

Amazon FSx for NetApp ONTAP を Trident とともに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推奨します。

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

"SVMのレプリケーションとリカバリ"中の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いように、バックエンドを設定できます。

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、 `managementLIF`を使用してSVMを指定し、 `dataLIF`および `svm`パラメータは省略します。例：

```
---
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`、および `trustedCACertificate`（信頼できるCAを使用する場合はオプション）は `backend.json`に入力され、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、Tridentに動的エクスポート 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指示する方法を示します。これは `ontap-nas-economy`ドライバと `ontap-nas-flexgroup`ドライバで同じように機能します。

```
---
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` を使用した IPv6 アドレスを示しています。

```
---
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
```

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

`smbShare`パラメータは、`SMB` ボリュームを使用する `FSx for ONTAP` に必要です。

```
---
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}}"
```

仮想プールを使用し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。たとえば、 `spaceReserve`は none、 `spaceAllocation`は false、 `encryption`は false 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。コメントは FlexVol の場合は `ontap-nas`、FlexGroup の場合は `ontap-nas-flexgroup`に設定されます。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`、および `encryption`値を設定し、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

次のStorageClass定義は[仮想プールを使用したバックエンドの例]を参照しています。 `parameters.selector` フィールドを使用して、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。

- この `protection-gold` StorageClass は、 `ontap-nas-flexgroup` バックエンドの最初と 2 番目の仮想プールにマッピングされます。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"
```

- `protection-not-gold` StorageClassは、 `ontap-nas-flexgroup`バックエンドの3番目と4番目の仮想プールにマッピングされます。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"
```

- `app-mysqldb`StorageClassは、 `ontap-nas`バックエンドの4番目の仮想プールにマッピングされます。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"
```

- `protection-silver-creditpoints-20k` StorageClassは `ontap-nas-flexgroup`バックエンドの3番目の仮想プールにマッピングされます。これは、シルバーレベルの保護と20000クレジットポイントを提供する唯一のプールです。

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

- `creditpoints-5k` StorageClassは、 `ontap-nas`バックエンドの3番目の仮想プールと `ontap-nas-economy`バックエンドの2番目の仮想プールにマッピングされます。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を決定し、ストレージ要件が満たされていることを確認します。

初期設定後にアップデート `dataLIF`

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 SMB の例

ontap-nas ドライバ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
```

`annotations`を追加して、セキュアな SMB を有効にしてください。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な SMB は機能しません。

ontap-nas-economy ドライバ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** ユーザを使用した **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** ユーザを使用した **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
```