ONTAP SANドライバー

Trident

NetApp
July 01, 2026

目次

ONTAP SANドライバー

ONTAP SAN ドライバの概要

ONTAP および Cloud Volumes ONTAP SAN 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 SAN 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SAN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-san`	iSCSI SCSI over FC	ブロック	RWO、ROX、RWX、RWOP	ファイルシステムなし；rawブロックデバイス
`ontap-san`	iSCSI SCSI over FC	Filesystem	RWO、RWOP ROX と RWX は Filesystem ボリューム モードでは使用できません。	`xfs`、 `ext3`、 `ext4`
`ontap-san`	NVMe/TCP NVMe/TCP に関する追加の考慮事項を参照してください。	ブロック	RWO、ROX、RWX、RWOP	ファイルシステムなし；rawブロックデバイス
`ontap-san`	NVMe/TCP NVMe/TCP に関する追加の考慮事項を参照してください。	Filesystem	RWO、RWOP ROX と RWX は Filesystem ボリューム モードでは使用できません。	`xfs`、 `ext3`、 `ext4`
`ontap-san-economy`	iSCSI	ブロック	RWO、ROX、RWX、RWOP	ファイルシステムなし；rawブロックデバイス

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-san-economy`	iSCSI	Filesystem	RWO、RWOP ROX と RWX は Filesystem ボリューム モードでは使用できません。	`xfs`、 `ext3`、 `ext4`

- `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。

NVMe/TCPに関する追加の考慮事項

Tridentは、 `ontap-san`ドライバーを使用して不揮発性メモリエクスプレス（NVMe）プロトコルをサポートします。これには次のものが含まれます：

- IPv6を使用したチャンク アップロード署名要求がサポートされるようになりました。
- NVMe ボリュームの Snapshot とクローン
- NVMe ボリュームのサイズ変更
- Tridentの外部で作成されたNVMeボリュームをインポートして、そのライフサイクルをTridentで管理でき

るようにする

- NVMe ネイティブマルチパス
- K8sノードの正常または異常シャットダウン（24.06）

Tridentでサポートされていない機能：

- NVMeでネイティブにサポートされているDH-HMAC-CHAP
- デバイスマッパー（DM）マルチパス
- LUKS暗号化

NVMeはONTAP REST APIでのみサポートされ、ONTAPI（ZAPI）ではサポートされていません。

ONTAP SAN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 SAN ドライバーを使用した ONTAP バックエンドの設定要件と認証オプションについて理解します。

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。

"ASA r2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。ASA r2システムでは、アグリゲートの代わりにストレージの可用性ゾーンが使用されます。ASA r2システムでSVMにアグリゲートを割り当てる方法については、"事項を"ナレッジベースの記事を参照してください。

複数のドライバを同時に実行し、それぞれに対応するストレージクラスを作成できることも覚えておいてください。たとえば、 `san-dev`ドライバを使用する `ontap-san`クラスと、 `san-default`ドライバを使用する `ontap-san-economy`クラスを設定することができます。

すべての Kubernetes ワーカー ノードに適切な iSCSI 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、"ワーカーノードを準備する"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。

- 認証情報ベース：必要な権限を持つONTAPユーザーのユーザー名とパスワード。 `admin`または `vsadmin`などの事前定義されたセキュリティログインロールを使用して、ONTAPバージョンとの最大限の互換性を確保することをお勧めします。
- 証明書ベース：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-san
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: password
```

JSON

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

バックエンド定義は、クレデンシャルがプレーンテキストで保存される唯一の場所であることに留意してください。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または更新は、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の有効化

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=admin"
```

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
```

> このコマンドを実行すると、ONTAPは証明書の入力を求めます。ステップ1で生成された`k8senv.pem`ファイルの内容を貼り付け、 `END`を入力してインストールを完了します。

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 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します。<ONTAP Management LIF>と<vserver name>を管理LIF IPとSVM名に置き換えます。

```
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.json
{
"version": 1,
"storageDriverName": "ontap-san",
"backendName": "SanBackend",
"managementLIF": "1.2.3.4",
"svm": "vserver_test",
"clientCertificate": "Faaaakkkkeeee...Vaaalllluuuueeee",
"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 backend update`を実行します。

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

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

パスワード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"

双方向**CHAP**による接続の認証

Tridentは、 `ontap-san`および `ontap-san-economy`ドライバで双方向CHAPを使用してiSCSIセッションを認証できます。これには、バックエンド定義で `useCHAP`オプションを有効にする必要があります。 `true`に設定すると、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

ファイルからユーザー名とシークレットを設定します。NetAppは、接続の認証に双方向CHAPを使用することを推奨します。次のサンプル構成を参照してください：

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap_san_chap
managementLIF: 192.168.0.135
svm: ontap_iscsi_svm
useCHAP: true
username: vsadmin
password: password
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
```

`useCHAP`パラメータは、一度だけ設定できるブーリアン パラメータです。デフォルトではfalseに設定されています。trueに設定した後は、falseに設定することはできません。

`useCHAP=true`に加えて、 `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。バックエンドを作成した後、 `tridentctl update`を実行することでシークレットを変更できます。

仕組み

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

- SVM で CHAP を設定する：
  - SVMのデフォルトのイニシエータセキュリティタイプがなし（デフォルトで設定）であり、かつボリューム内に既存のLUNが存在しない場合は、Tridentはデフォルトのセキュリティタイプを `CHAP`に設定し、CHAPイニシエータとターゲットのユーザー名とシークレットの構成に進みます。
  - 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上にすでに存在するLUNへのアクセスが制限されなくなります。
- CHAP イニシエーターとターゲットのユーザー名およびシークレットを構成します。これらのオプションは、バックエンド構成で指定する必要があります（上記を参照）。

バックエンドが作成されると、Tridentは対応する `tridentbackend`CRDを作成し、CHAPシークレットとユーザー名をKubernetesシークレットとして保存します。このバックエンドでTridentによって作成されたすべてのPVは、CHAP経由でマウントおよび接続されます。

資格情報をローテーションしてバックエンドを更新する

`backend.json`ファイルでCHAPパラメータを更新することで、CHAP認証情報を更新できます。これには、CHAPシークレットを更新し、 `tridentctl update`コマンドを使用してこれらの変更を反映する必要があります。

バックエンドのCHAPシークレットを更新する場合は、 `tridentctl`を使用してバックエンドを更新する必要があります。ONTAP CLIまたはONTAP System Managerを使用してストレージクラスタの資格情報を更新しないでください。Tridentはこれらの変更を反映できません。

```
cat backend-san.json
{
    "version": 1,
    "storageDriverName": "ontap-san",
    "backendName": "ontap_san_chap",
    "managementLIF": "192.168.0.135",
    "svm": "ontap_iscsi_svm",
    "useCHAP": true,
    "username": "vsadmin",
    "password": "password",
    "chapInitiatorSecret": "cl9qxUpDaTeD",
    "chapTargetInitiatorSecret": "rqxigXgkeUpDaTeD",
    "chapTargetUsername": "iJF4heBRT0TCwxyz",
    "chapUsername": "uh2aNCLSd6cNwxyz",
}

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

既存の接続は影響を受けません。Trident が SVM で資格情報を更新しても、引き続きアクティブなままになります。新しい接続では更新された資格情報が使用され、既存の接続は引き続きアクティブなままになります。古い PV を切断して再接続すると、更新された資格情報が使用されるようになります。

ONTAP SAN 構成オプションと例

Trident インストールで ONTAP SAN ドライバを作成して使用する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。"ASA r2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。これらの違いは、記載されている特定のパラメータの使用に影響します。"ASA r2システムとその他のONTAPシステムの違いについて詳しくは、こちらをご覧ください"。Trident バックエンド構成では、システムが ASA r2 であることを指定する必要はありません。`ontap-san`を `storageDriverName`として選択すると、Trident は ASA r2 またはその他の ONTAP システムを自動的に検出します。以下の表に記載されているように、一部のバックエンド構成パラメータは ASA r2 システムには適用されません。

`ontap-san`ドライバー（iSCSI、NVMe/TCP、FCプロトコル）のみがASA r2システムでサポートされています。

バックエンド構成オプション

バックエンド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	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	`ontap-san` または `ontap-san-economy`
`backendName`	カスタム名またはストレージバックエンド	ドライバー名 + "_" + 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、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。 「vsadmin」の資格情報を使用している場合は、 `managementLIF` SVMのものでなければなりません。「admin」の資格情報を使用する場合は、 `managementLIF` クラスターのものである必要があります。	"10.0.0.1"、"[2001:1234:abcd::fefe]"

パラメータ	概要	デフォルト
`dataLIF`	プロトコル LIF の IP アドレス。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]`。**iSCSI** の場合は指定しないでください。***Trident** は**"ONTAP セレクティブLUNマップ"**を使用して、マルチパス セッションを確立するために必要な **iSCSI LIF** を検出します。警告が発生するのは、 **`dataLIF`** が明示的に定義されている場合です。***MetroCluster** の場合は省略します。MetroCluster の例を参照してください。	SVMによって導出された
`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
`useCHAP`	ONTAP SAN ドライバーの iSCSI 認証に CHAP を使用します [ブール値]。 `true`に設定すると、バックエンドで指定された SVM のデフォルト認証として、Trident が双方向 CHAP を設定して使用します。詳細については、"ONTAP SAN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"を参照してください。**FCP** または **NVMe/TCP** ではサポートされません。	`false`
`chapInitiatorSecret`	CHAP イニシエータシークレット。次の場合は必須 `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。次の場合は必須 `useCHAP=true`	""
`chapUsername`	受信ユーザー名。次の場合は必須 `useCHAP=true`	""
`chapTargetUsername`	ターゲットユーザー名。次の場合は必須 `useCHAP=true`	""
`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されます。	""
`username`	ONTAP クラスタとの通信に必要なユーザー名。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。	""

パラメータ	概要	デフォルト
`password`	ONTAP クラスタとの通信に必要なパスワード。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。	""
`svm`	使用するStorage Virtual Machine	SVM `managementLIF`が指定されている場合に導出されます
`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。後で変更することはできません。このパラメータを更新するには、新しいバックエンドを作成する必要があります。	`trident`
`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。 **ASA r2**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 NetApp ONTAP バックエンドを使用している場合は、`limitAggregateUsage`を指定しないでください。提供された `fsxadmin`と `vsadmin`には、Trident を使用してアグリゲートの使用状況を取得して制限するために必要な権限が含まれていません。**ASA r2**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）
`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、LUNに対して管理するボリュームの最大サイズも制限します。	""（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
`lunsPerFlexvol`	FlexVolあたりの最大LUN数は[50, 200]の範囲でなければなりません	`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}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、使用しないでください。	`null`
`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。 `useREST`は、NVMe/TCP に完全対応しています。 NVMeはONTAP REST APIでのみサポートされ、ONTAPI（ZAPI）ではサポートされていません。 指定されている場合は、常に **ASA r2** システムの**`true`**に設定します。	ONTAP 9.15.1以降の場合は`true`、それ以外の場合は `false`。
`sanType`	iSCSIの場合は `iscsi`、NVMe/TCPの場合は `nvme`、SCSI over Fibre Channel（FC）の場合は `fcp`を選択します。	`iscsi` 空白の場合

パラメータ	概要	デフォルト
formatOptions	`formatOptions`を使用して、`mkfs`コマンドのコマンドライン引数を指定します。これは、ボリュームがフォーマットされるたびに適用されます。これにより、設定に応じてボリュームをフォーマットできます。デバイス パスを除き、mkfsコマンドのオプションと同様にformatOptionsを指定してください。例："-E nodiscard" • `ontap-san`および`ontap-san-economy`ドライバでiSCSIプロトコルを使用する場合にサポートされます。*さらに、**iSCSI**および**NVMe/TCP**プロトコルを使用する場合、**ASA r2**システムでサポートされます。	
limitVolumePoolSize	ontap-san-economy バックエンドで LUN 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 LUN 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-san-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。	

formatOptionsの使用に関する推奨事項

Tridentは、フォーマット処理を高速化するために次のオプションを推奨します：

- **-E nodiscard (ext3、ext4):** mkfs 時にブロックを破棄しません（最初にブロックを破棄することは、ソリッド ステート デバイスやスパース / シンプロビジョニング ストレージでは有効です）。これは非推奨のオプション「-K」に代わるもので、ext3、ext4 ファイル システムに適用できます。
- **-K (xfs):** mkfs 時にブロックを破棄しません。このオプションは xfs ファイル システムに適用できます。

Active Directory の認証情報を使用してバックエンド **SVM** に **Trident** を認証

Tridentを設定して、Active Directory（AD）認証情報を使用してバックエンドSVMに認証できます。ADアカウントがSVMにアクセスする前に、クラスターまたはSVMへのADドメイン コントローラアクセスを設定する必要があります。ADアカウントを使用してクラスターを管理するには、ドメイン トンネルを作成する必要があります。詳細については、"ONTAPでActive Directoryドメイン コントローラ アクセスを設定する"を参照してください。

手順

1. バックエンド SVM のドメイン ネーム システム（DNS）設定を構成します：

```
vserver services dns create -vserver <svm_name> -dns-servers
<dns_server_ip1>,<dns_server_ip2>
```

2. 次のコマンドを実行して、Active Directory に SVM のコンピュータ アカウントを作成します：

```
vserver active-directory create -vserver DataSVM -account-name ADSERVER1
-domain demo.netapp.com
```

3. このコマンドを使用して、クラスタまたはSVMを管理するためのADユーザまたはグループを作成します

```
security login create -vserver <svm_name> -user-or-group-name
<ad_user_or_group> -application <application> -authentication-method domain
-role vsadmin
```

4. Tridentバックエンド設定ファイルで、 `username`および `password`パラメータをそれぞれADユーザー名またはグループ名とパスワードに設定します。

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`	LUNのスペース割り当て	"true" 指定されている場合は、**ASA r2** システム用に `**true**`に設定します。
`spaceReserve`	スペース予約モード。「none」（シン）または「volume」（シック）。**ASA r2** システムの場合は `none`に設定します。	「なし」
`snapshotPolicy`	使用するSnapshotポリシー。**ASA r2**システムの場合は `**none**`に設定。	「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てください。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。	""
`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	""
`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。**ASA r2**システムには指定しないでください。	`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"

パラメータ	概要	デフォルト
`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" 指定されている場合は、**ASA r2** システム **`true`**に設定します。
`luksEncryption`	LUKS暗号化を有効にします。"Linux Unified Key Setup（LUKS）を使用する"を参照してください。	"" **ASA r2** システムの場合は **`false`** に設定。
`tieringPolicy`	階層化ポリシーは「なし」を使用する **ASA r2** システムには指定しないでください。	
`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。	""

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-san
managementLIF: 10.0.0.1
svm: trident_svm
username: admin
password: <password>
labels:
  k8scluster: dev2
  backend: dev2-sanbackend
storagePrefix: alternate-trident
debugTraceFlags:
  api: false
  method: true
defaults:
  spaceReserve: volume
  qosPolicy: standard
  spaceAllocation: 'false'
  snapshotPolicy: default
  snapshotReserve: '10'
```

`ontap-san`ドライバを使用して作成されたすべてのボリュームについて、TridentはLUNメタデータに対応するためにFlexVolに10%の容量を追加します。LUNは、ユーザーがPVCで要求した正確なサイズでプロビジョニングされます。TridentはFlexVolに10%を追加します（ONTAPでは使用可能なサイズとして表示されます）。ユーザーは要求した使用可能な容量を取得できるようになりました。この変更により、使用可能なスペースが完全に使用されない限り、LUNが読み取り専用になることも防止されます。これはontap-san-economyには適用されません。

`snapshotReserve`を定義するバックエンドの場合、Tridentはボリュームのサイズを次のように計算します：

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

1.1は、TridentがLUNメタデータに対応するためにFlexVolに追加する10パーセントの追加分です。snapshotReserve = 5%、PVC要求 = 5 GiBの場合、ボリュームの合計サイズは5.79 GiB、使用可能なサイズは5.5 GiBになります。 `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：

現在、既存のボリュームに対して新しい計算を使用する唯一の方法は、サイズ変更です。

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、NetApp では、IP アドレスではなく LIF の DNS 名を指定することを推奨しています。

ONTAP SANの例

これは、 `ontap-san`ドライバを使用した基本的な設定です。

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

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、 `svm`パラメータは省略します。例：

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

ONTAP SANエコノミーの例

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

証明書ベースの認証の例

この基本構成例では、 `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）が `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
storageDriverName: ontap-san
backendName: DefaultSANBackend
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
clientCertificate: ZXR0ZXJwYXB...ICMgJ3BhcGVyc2
clientPrivateKey: vciwKIyAgZG...0cnksIGRlc2NyaX
trustedCACertificate: zcyBbaG...b3Igb3duIGNsYXNz
```

双方向CHAPの例

これらの例では、 `useCHAP`を `true`に設定してバックエンドを作成します。

ONTAP SAN CHAPの例

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
labels:
  k8scluster: test-cluster-1
  backend: testcluster1-sanbackend
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

ONTAP SAN economy CHAPの例

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
```

NVMe/TCPの例

ONTAP バックエンドに NVMe で構成された SVM が必要です。これは、NVMe/TCP の基本的なバックエンド構成です。

```
---
version: 1
backendName: NVMeBackend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_nvme
username: vsadmin
password: password
sanType: nvme
useREST: true
```

SCSI over FC（FCP）の例

ONTAP バックエンドで FC を使用して構成された SVM が必要です。これは FC の基本的なバックエンド構成です。

```
---
version: 1
backendName: fcp-backend
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_fc
username: vsadmin
password: password
sanType: fcp
useREST: true
```

nameTemplateを使用したバックエンド構成の例

```
---
version: 1
storageDriverName: ontap-san
backendName: ontap-san-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}}"
```

formatOptions ontap-san-economy ドライバーの例

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: ""
svm: svm1
username: ""
password: "!"
storagePrefix: whelk_
debugTraceFlags:
  method: true
  api: true
defaults:
  formatOptions: -E nodiscard
```

仮想プールを使用したバックエンドの例

これらのサンプルバックエンド定義ファイルでは、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 volume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 SANの例

```
---
version: 1
storageDriverName: ontap-san
managementLIF: 10.0.0.1
svm: svm_iscsi
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
  qosPolicy: standard
labels:
  store: san_store
  kubernetes-cluster: prod-cluster-1
region: us_east_1
storage:
  - labels:
      protection: gold
      creditpoints: "40000"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
      adaptiveQosPolicy: adaptive-extreme
  - labels:
      protection: silver
      creditpoints: "20000"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
      qosPolicy: premium
  - labels:
      protection: bronze
      creditpoints: "5000"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
```

ONTAP SANエコノミーの例

```
---
version: 1
storageDriverName: ontap-san-economy
managementLIF: 10.0.0.1
svm: svm_iscsi_eco
useCHAP: true
chapInitiatorSecret: cl9qxIm36DKyawxy
chapTargetInitiatorSecret: rqxigXgkesIpwxyz
chapTargetUsername: iJF4heBRT0TCwxyz
chapUsername: uh2aNCLSd6cNwxyz
username: vsadmin
password: <password>
defaults:
  spaceAllocation: "false"
  encryption: "false"
labels:
  store: san_economy_store
region: us_east_1
storage:
  - labels:
      app: oracledb
      cost: "30"
    zone: us_east_1a
    defaults:
      spaceAllocation: "true"
      encryption: "true"
  - labels:
      app: postgresdb
      cost: "20"
    zone: us_east_1b
    defaults:
      spaceAllocation: "false"
      encryption: "true"
  - labels:
      app: mysqldb
      cost: "10"
    zone: us_east_1c
    defaults:
      spaceAllocation: "true"
      encryption: "false"
  - labels:
      department: legal
      creditpoints: "5000"
```

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

NVMe/TCPの例

```
---
version: 1
storageDriverName: ontap-san
sanType: nvme
managementLIF: 10.0.0.1
svm: nvme_svm
username: vsadmin
password: <password>
useREST: true
defaults:
  spaceAllocation: "false"
  encryption: "true"
storage:
  - labels:
      app: testApp
      cost: "20"
    defaults:
      spaceAllocation: "false"
      encryption: "false"
```

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。

- `protection-gold` StorageClassは、 `ontap-san`バックエンドの最初の仮想プールにマッピングされます。これはゴールド 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-san`バックエンドの2番目と3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"
```

- この `app-mysqldb` StorageClass は `ontap-san-economy`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、mysqldb タイプのアプリにストレージ 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"
```

- この `protection-silver-creditpoints-20k` StorageClassは `ontap-san`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"
```

- この `creditpoints-5k` StorageClassは、 `ontap-san`バックエンドの3番目の仮想プールと `ontap-san-economy`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"
```

- この `my-test-app-sc` StorageClassは、 `testAPP`仮想プールに `ontap-san`ドライバで `sanType: nvme`マッピングされます。これは `testApp`を提供する唯一のプールです。

```
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: my-test-app-sc
provisioner: csi.trident.netapp.io
parameters:
  selector: "app=testApp"
  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。

著作権に関する情報

商標に関する情報