ONTAP SAN ドライバー

Trident

NetApp
January 15, 2026

目次

ONTAP SAN ドライバー

ONTAP SAN ドライバーの概要

ONTAPおよびCloud Volumes ONTAP SAN 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 SAN 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 SAN 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-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 はファイルシステム ボリューム モードでは使用できません。	`xfs`、 `ext3` 、 `ext4`
`ontap-san`	NVMe/TCP 参照NVMe/TCPに関する追加の考慮事項。	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 参照NVMe/TCPに関する追加の考慮事項。	Filesystem	RWO、RWOP ROX と RWX はファイルシステム 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

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-san-economy`	iSCSI	Filesystem	RWO、RWOP ROX と RWX はファイルシステム ボリューム モードでは使用できません。	`xfs`、 `ext3` 、 `ext4`

- 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。

NVMe/TCPに関する追加の考慮事項

Tridentは、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ています。 `ontap-san`ドライバーには以下が含まれます:

- IPv6
- NVMeボリュームのスナップショットとクローン
- 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 は以下をサポートしていません:

- 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) ではサポートされません。

ONTAP SAN 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 SAN 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と認証オプションを理解します。

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。

"ASA r2 システム"ストレージ層の実装は他のONTAPシステム (ASA、 AFF、 FAS) とは異なります。 ASA r2 システムでは、集約の代わりにストレージ可用性ゾーンが使用されます。参照"事項を"ASA r2 システムで SVM にアグリゲートを割り当てる方法に関するナレッジベースの記事。

複数のドライバーを実行し、いずれかを指すストレージ クラスを作成することもできることに注意してください。例えば、 `san-dev`を使用するクラス `ontap-san`運転手と `san-default`を使用するクラス `ontap-san-economy`1つ。

すべての Kubernetes ワーカーノードに適切な iSCSI ツールがインストールされている必要があります。参照"ワーカーノードを準備する" 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。

- 資格情報ベース: 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。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: Trident は、バックエンドにインストールされた証明書を使用してONTAPクラスターと通信することもできます。ここで、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、推奨）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。

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-san
managementLIF: 10.0.0.1
svm: svm_nfs
username: vsadmin
password: password
```

JSON

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

バックエンド定義は、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または更新は、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、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=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
```

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 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、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 backend update` 。

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

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

パスワード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"

双方向**CHAP**による接続の認証

Tridentは、双方向CHAPを使用してiSCSIセッションを認証できます。 `ontap-san`そして `ontap-san-

economy`ドライバー。これには、 `useCHAP`バックエンド定義のオプション。に設定すると `true`Trident は、SVM のデフォルトのイニシエータ セキュリティを双方向 CHAP に設定し、バックエンド ファイルからユーザー名とシークレットを設定します。 NetApp、接続の認証に双方向 CHAP を使用することを推奨しています。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
```

その `useCHAP`パラメータは一度だけ設定できるブールオプションです。デフォルトでは false に設定されています。 true に設定した後は、false に設定することはできません。

に加えて `useCHAP=true`、 `chapInitiatorSecret` 、 `chapTargetInitiatorSecret` 、 `chapTargetUsername` 、 そして `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、次のコマンドを実行することでシークレットを変更できます。 `tridentctl update` 。

仕組み

設定により `useCHAP`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 を構成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none（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、 Tridentはデフォルトのセキュリティタイプを次のように設定します。 `CHAP` CHAP イニシエーター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 上で 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と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 は、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によって作成されるすべての PV は、CHAP 経由でマウントおよび接続されます。

認証情報をローテーションしてバックエンドを更新する

CHAP認証情報を更新するには、CHAPパラメータを更新します。 `backend.json`ファイル。これにはCHAPシークレットを更新し、 `tridentctl update`これらの変更を反映するコマンド。

> バックエンドのCHAPシークレットを更新する場合は、 `tridentctl`バックエンドを更新します。Trident はこれらの変更を取得できないため、 ONTAP CLI またはONTAP System Manager を使用してストレージ クラスターの資格情報を更新しないでください。

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

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

既存の接続は影響を受けません。SVM 上のTridentによって資格情報が更新されると、既存の接続は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。

"ASA r2 システム"ストレージ層の実装は他のONTAPシステム (ASA、 AFF、 FAS) とは異なります。これらのバリエーションは、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。"。

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

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。選択すると `ontap-san`として `storageDriverName`Trident は、 ASA r2 または従来のONTAPシステムを自動的に検出します。以下の表に示すように、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。

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-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 。 「vsadmin」の資格情報を使用している場合は、 `managementLIF` SVMの認証情報である必要があります。「admin」認証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	説明	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]` 。 *iSCSI の場合は指定しないでください。*Tridentは"ONTAP選択的 LUN マップ"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`明示的に定義されています。 *Metrocluster の場合は省略します。*参照MetroClusterの例。	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。*参照MetroClusterの例。	SVMの場合導出される `managementLIF`指定されている
`useCHAP`	CHAP を使用してONTAP SAN ドライバーの iSCSI を認証します [ブール値]。設定 `true`Trident がバックエンドで指定された SVM のデフォルト認証として双方向 CHAP を設定して使用できるようにします。参照 "ONTAP SAN 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" 詳細については。 **FCP** または **NVMe/TCP** ではサポートされません。	`false`
`chapInitiatorSecret`	CHAP イニシエーター シークレット。必須の場合 `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 イニシエーター シークレット。必須の場合 `useCHAP=true`	""
`chapUsername`	受信ユーザー名。必須の場合 `useCHAP=true`	""
`chapTargetUsername`	ターゲットユーザー名。必須の場合 `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。オプション。証明書ベースの認証に使用されます。	""
`username`	ONTAPクラスタと通信するために必要なユーザー名。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"。	""

パラメータ	説明	デフォルト
`password`	ONTAPクラスタと通信するために必要なパスワード。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"。	""
`svm`	使用するストレージ仮想マシン	SVMの場合導出される `managementLIF`指定されている
`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。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)。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。 • ASA r2 システムで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、指定しないでください。 `limitAggregateUsage` 。提供された `fsxadmin`そして `vsadmin`集計使用量を取得し、 Trident を使用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 システムで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	説明	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。 `useREST`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) ではサポートされません。 指定されている場合、常に **`true`ASA r2** システムの場合。	`true`ONTAP 9.15.1以降の場合、それ以外の場合 `false`。
`sanType`	選択するには使用 `iscsi`iSCSIの場合、 `nvme` NVMe/TCPの場合または `fcp`SCSI over Fibre Channel (FC) 用。	`iscsi`空白の場合

パラメータ	説明	デフォルト
`formatOptions`	使用 `formatOptions`コマンドライン引数を指定するには `mkfs`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、好みに応じてボリュームをフォーマットできます。デバイス パスを除いて、mkfs コマンド オプションと同様の formatOptions を指定してください。例: "-E nodiscard" 対応機種 **`ontap-san`**そして **`ontap-san-economy`iSCSI** プロトコルを使用したドライバー。さらに、**iSCSI** および **NVMe/TCP** プロトコルを使用する場合、 **ASA r2** システムでもサポートされます。	
`limitVolumePoolSize`	ontap-san-economy バックエンドで LUN を使用する場合の、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-san-economy`バックエンドが LUN 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。	

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 破棄なし:

- 保持し、mkfs 時にブロックを破棄しないでください (最初にブロックを破棄することは、ソリッド ステート デバイスおよびスパース / シン プロビジョニング ストレージで役立ちます)。これは非推奨のオプション「-K」に代わるもので、すべてのファイルシステム (xfs、ext3、ext4) 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** に対して**Trident** を認証する

Active Directory (AD) 認証情報を使用してバックエンド SVM に対して認証するようにTrident を設定できます。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。参照 "ONTAPでActive Directoryドメインコントローラのアクセスを構成する" 詳細については。

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ユーザー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` そして `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渡します。

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`	LUNのスペース割り当て	"true" 指定されている場合は、**`true` ASA r2** システムの場合。
`spaceReserve`	スペース予約モード。「なし」(薄い) または「ボリューム」(厚い)。 設定 **`none`ASA r2** システムの場合。	"なし"
`snapshotPolicy`	使用するスナップショット ポリシー。 設定 **`none`ASA r2** システムの場合。	"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。 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ます	""
`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 システムでは指定しないでください*。	「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" 。	"false" 指定されている場合は、**`true` ASA r2** システムの場合。

パラメータ	説明	デフォルト
`luksEncryption`	LUKS 暗号化を有効にします。参照"Linux Unified Key Setup (LUKS) を使用する"。	"" 設定 `**false**`ASA r2 システムの場合。
`tieringPolicy`	階層化ポリシーは「なし」を使用します。* ASA r2 システムで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'
```

作成されたすべてのボリュームについて `ontap-san`ドライバーにより、 Trident はLUN メタデータに対応するためにFlexVolに 10 パーセント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 パーセントを追加します (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す。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。

定義するバックエンドの場合 `snapshotReserve`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
```

にTridentがFlexVolに追加する10%の容量です。のために `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。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。

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。

> Tridentを搭載したNetApp ONTAPでAmazon FSx を使用している場合、 NetApp、 LIF に IP アドレスではなく 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。

```
---
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

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 `svm`パラメータ。例えば：

```
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` 、 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
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`に設定 `true`。

ONTAP SAN CHAPの例

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

ONTAP SANエコノミーCHAPの例

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

NVMe/TCPの例

ONTAPバックエンドに NVMe が設定された 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バックエンドに FC が設定された 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}}"
```

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

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

仮想プールを備え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。 `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。コメントはFlexVol volumeに設定され、 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 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

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-san`バックエンド。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"
```

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

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

- その `app-mysqldb`StorageClassは3番目の仮想プールにマッピングされます `ontap-san-economy`バックエンド。これは、mysqldb タイプ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"
```

- その `protection-silver-creditpoints-20k`ストレージクラスは2番目の仮想プールにマッピングされます `ontap-san`バックエンド。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`StorageClassは3番目の仮想プールにマッピングされます `ontap-san`バックエンドと4番目の仮想プール `ontap-san-economy`バックエンド。これらは 5000 クレジットポイント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"
```

- その `my-test-app-sc`StorageClassは `testAPP`仮想プール `ontap-san`ドライバー付き `sanType: nvme`。これは唯一のプールの提供です `testApp`。

```
---
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