



バックエンドを管理する Trident

NetApp
October 29, 2026

目次

バックエンドを管理する	1
kubectlを使用したTridentバックエンドの管理	1
バックエンドを削除する	1
既存のバックエンドを表示する	1
バックエンドを更新する	1
tridentctl を使用した Trident バックエンドの管理	2
バックエンドを作成する	2
バックエンドを削除する	3
既存のバックエンドを表示する	3
バックエンドを更新する	3
バックエンドを使用するストレージクラスを特定する	4
Tridentバックエンド管理方法を選択する	4
バックエンドを管理するオプション	4
tridentctl`を使用してバックエンドを管理 `TridentBackendConfig`	4
TridentBackendConfig`を使用してバックエンドを管理 `tridentctl`	9

バックエンドを管理する

kubectlを使用したTridentバックエンドの管理

`kubectl`を使用してバックエンド管理操作を実行する方法について説明します。

バックエンドを削除する

`TridentBackendConfig`を削除すると、Tridentに対してバックエンドを削除/保持するように指示します（`deletionPolicy`に基づく）。バックエンドを削除するには、`deletionPolicy`がdeleteに設定されていることを確認してください。`TridentBackendConfig`だけを削除するには、`deletionPolicy`がretainに設定されていることを確認してください。これにより、バックエンドが引き続き存在し、`tridentctl`を使用して管理できることが保証されます。

次のコマンドを実行します。

```
kubectl delete tbc <tbc-name> -n trident
```

Tridentは`TridentBackendConfig`で使用されていたKubernetes Secretを削除しません。Kubernetesユーザーはシークレットをクリーンアップする責任があります。シークレットを削除するときは注意が必要です。シークレットは、バックエンドで使用されていない場合にのみ削除する必要があります。

既存のバックエンドを表示する

次のコマンドを実行します。

```
kubectl get tbc -n trident
```

```
`tridentctl get backend -n trident`または `tridentctl get backend -o yaml  
-n  
trident`を実行して、存在するすべてのバックエンドのリストを取得することもできます。このリ  
ストには、`tridentctl`で作成されたバックエンドも含まれます。
```

バックエンドを更新する

バックエンドを更新する理由は複数考えられます：

- ストレージシステムへのクレデンシャルが変更されました。クレデンシャルを更新するには、`TridentBackendConfig`オブジェクトで使用されているKubernetes Secretを更新する必要があります。

す。Tridentは、提供された最新のクレデンシャルを使用してバックエンドを自動的に更新します。次のコマンドを実行してKubernetes Secretを更新します：

```
kubectl apply -f <updated-secret-file.yaml> -n trident
```

- パラメータ（使用されている ONTAP SVM の名前など）を更新する必要があります。
 - `TridentBackendConfig` オブジェクトは、次のコマンドを使用してKubernetes経由で直接更新できます：

```
kubectl apply -f <updated-backend-file.yaml>
```

- あるいは、次のコマンドを使用して既存の TridentBackendConfig CR に変更を加えることができます：

```
kubectl edit tbc <tbc-name> -n trident
```



- バックエンドの更新が失敗した場合、バックエンドは最後の既知の構成のままになります。ログを表示して原因を特定するには、`kubectl get tbc <tbc-name> -o yaml -n trident` または `kubectl describe tbc <tbc-name> -n trident` を実行します。
- 構成ファイルの問題を特定して修正したら、更新コマンドを再実行できます。

tridentctl を使用した Trident バックエンドの管理

`tridentctl` を使用してバックエンド管理操作を実行する方法について説明します。

バックエンドを作成する

"バックエンド設定ファイル"を作成したら、次のコマンドを実行します：

```
tridentctl create backend -f <backend-file> -n trident
```

バックエンドの作成に失敗した場合は、バックエンドの構成に問題があります。次のコマンドを実行すると、ログを表示して原因を特定できます：

```
tridentctl logs -n trident
```

設定ファイルの問題を特定して修正したら、`create` コマンドを再度実行します。

バックエンドを削除する

バックエンドをTridentから削除するには、次の操作を行います：

1. バックエンド名を取得します：

```
tridentctl get backend -n trident
```

2. バックエンドを削除します：

```
tridentctl delete backend <backend-name> -n trident
```



Tridentがこのバックエンドからプロビジョニングしたボリュームとスナップショットがまだ存在する場合、バックエンドを削除すると、新しいボリュームをプロビジョニングできなくなります。バックエンドは「削除中」の状態で存在し続けます。

既存のバックエンドを表示する

Tridentが認識しているバックエンドを表示するには、次の操作を行います。

- 概要を取得するには、次のコマンドを実行します：

```
tridentctl get backend -n trident
```

- すべての詳細を取得するには、次のコマンドを実行します：

```
tridentctl get backend -o json -n trident
```

バックエンドを更新する

新しいバックエンド構成ファイルを作成したら、次のコマンドを実行します：

```
tridentctl update backend <backend-name> -f <backend-file> -n trident
```

バックエンドの更新が失敗した場合は、バックエンドの構成に問題があるか、無効な更新を試行しました。次のコマンドを実行すると、ログを表示して原因を特定できます：

```
tridentctl logs -n trident
```

設定ファイルの問題を特定して修正したら、`update`コマンドを再度実行します。

バックエンドを使用するストレージクラスを特定する

これは、`tridentctl`がバックエンドオブジェクトに対して出力するJSONで回答できる質問の種類の例です。これは`jq`ユーティリティを使用しており、インストールする必要があります。

```
tridentctl get backend -o json | jq '[.items[] | {backend: .name, storageClasses: [.storage[].storageClasses] | unique}]'
```

これは、`TridentBackendConfig`を使用して作成されたバックエンドにも適用されます。

Tridentバックエンド管理方法を選択する

Trident でバックエンドを管理するさまざまな方法について説明します。

バックエンドを管理するオプション

`TridentBackendConfig`の導入により、管理者は、バックエンドを管理する 2 つの独自の方法を利用できるようになりました。これにより、次の疑問が生じます：

- `tridentctl`を使用して作成されたバックエンドは、`TridentBackendConfig`で管理できますか？
- `TridentBackendConfig`を使用して作成されたバックエンドは、`tridentctl`を使用して管理できますか？

tridentctl`を使用してバックエンドを管理 `TridentBackendConfig

このセクションでは、Kubernetesインターフェイスを通じて`tridentctl`を直接作成し、`TridentBackendConfig`オブジェクトを作成することで作成されたバックエンドの管理手順について説明します。

これは次のシナリオに適用されます：

- 既存のバックエンドには`TridentBackendConfig`がありません。これは`tridentctl`で作成されたためです。
- `tridentctl`で作成された新しいバックエンド（他の`TridentBackendConfig`オブジェクトが存在する場合）。

どちらのシナリオでも、バックエンドは存在し続け、Tridentがボリュームのスケジューリング設定とボリュームに対する操作を行います。管理者には次の2つの選択肢があります（：）

- それを使用して作成されたバックエンドの管理には`tridentctl`を引き続きご利用ください。
- `tridentctl`を使用して作成されたバックエンドを新しい`TridentBackendConfig`オブジェクトにバインドします。これにより、バックエンドは`kubectl`で管理され、`tridentctl`では管理されなくなります。

`kubectl`を使用して既存のバックエンドを管理するには、既存のバックエンドにバインドする

`TridentBackendConfig`を作成する必要があります。その仕組みの概要は次のとおりです：

1. Kubernetes シークレットを作成します。このシークレットには、Trident がストレージクラスタ/サービスと通信するために必要なクレデンシャルが含まれています。
2. `TridentBackendConfig` オブジェクトを作成します。これには、ストレージクラスタ/サービスに関する詳細が含まれており、前の手順で作成されたシークレットが参照されます。同一の設定パラメータ（`spec.backendName`、`spec.storagePrefix`、`spec.storageDriverName`など）を指定するように注意する必要があります。`spec.backendName`は既存のバックエンドの名前に設定する必要があります。

ステップ0：バックエンドを特定する

既存のバックエンドにバインドする `TridentBackendConfig`を作成するには、バックエンドの構成を取得する必要があります。この例では、次の JSON 定義を使用してバックエンドが作成されたと仮定します：

```
tridentctl get backend ontap-nas-backend -n trident
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |          UUID          |
| STATE  | VOLUMES |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontap-nas-backend      | ontap-nas      | 52f2eb10-e4c6-4160-99fc-96b3be5ab5d7 |
| online |      25 |          |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

```
cat ontap-nas-backend.json
```

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "managementLIF": "10.10.10.1",
  "dataLIF": "10.10.10.2",
  "backendName": "ontap-nas-backend",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "admin-password",
  "defaults": {
    "spaceReserve": "none",
    "encryption": "false"
  },
  "labels": {
    "store": "nas_store"
  }
}
```

```

},
"region": "us_east_1",
"storage": [
  {
    "labels": {
      "app": "msoffice",
      "cost": "100"
    },
    "zone": "us_east_1a",
    "defaults": {
      "spaceReserve": "volume",
      "encryption": "true",
      "unixPermissions": "0755"
    }
  },
  {
    "labels": {
      "app": "mysqldb",
      "cost": "25"
    },
    "zone": "us_east_1d",
    "defaults": {
      "spaceReserve": "volume",
      "encryption": "false",
      "unixPermissions": "0775"
    }
  }
]
}

```

ステップ1：Kubernetesシークレットを作成する

次の例に示すように、バックエンドの資格情報を含む Secret を作成します：

```
cat tbc-ontap-nas-backend-secret.yaml
```

```

apiVersion: v1
kind: Secret
metadata:
  name: ontap-nas-backend-secret
type: Opaque
stringData:
  username: cluster-admin

```

```
password: admin-password
```

```
kubectl create -f tbc-ontap-nas-backend-secret.yaml -n trident  
secret/backend-tbc-ontap-san-secret created
```

ステップ2: TridentBackendConfig CRを作成する

次のステップは、既存の `ontap-nas-backend` に自動的にバインドされる `TridentBackendConfig` CRを作成することです（この例のように）。次の要件が満たされていることを確認してください：

- 同じバックエンド名が `spec.backendName` で定義されています。
- 構成パラメータは元のバックエンドと同一です。
- 仮想プール（存在する場合）は、元のバックエンドと同じ順序を維持する必要があります。
- クレデンシャルはプレーンテキストではなく、Kubernetes Secret を通じて提供されます。

この場合、`TridentBackendConfig` は次のようになります：

```
cat backend-tbc-ontap-nas.yaml
```

```
apiVersion: trident.netapp.io/v1  
kind: TridentBackendConfig  
metadata:  
  name: tbc-ontap-nas-backend  
spec:  
  version: 1  
  storageDriverName: ontap-nas  
  managementLIF: 10.10.10.1  
  dataLIF: 10.10.10.2  
  backendName: ontap-nas-backend  
  svm: trident_svm  
  credentials:  
    name: mysecret  
  defaults:  
    spaceReserve: none  
    encryption: 'false'  
  labels:  
    store: nas_store  
    region: us_east_1  
  storage:  
  - labels:  
    app: msoffice  
    cost: '100'
```

```

zone: us_east_1a
defaults:
  spaceReserve: volume
  encryption: 'true'
  unixPermissions: '0755'
- labels:
  app: mysqlldb
  cost: '25'
zone: us_east_1d
defaults:
  spaceReserve: volume
  encryption: 'false'
  unixPermissions: '0775'

```

```

kubectl create -f backend-tbc-ontap-nas.yaml -n trident
tridentbackendconfig.trident.netapp.io/tbc-ontap-nas-backend created

```

ステップ3: **TridentBackendConfig CR**のステータスを確認する

`TridentBackendConfig`が作成された後、そのフェーズは`Bound`である必要があります。また、既存のバックエンドと同じバックエンド名とUUIDを反映する必要があります。

```

kubectl get tbc tbc-ontap-nas-backend -n trident

```

NAME	BACKEND NAME	BACKEND UUID
tbc-ontap-nas-backend	ontap-nas-backend	52f2eb10-e4c6-4160-99fc-96b3be5ab5d7
Bound	Success	

#confirm that no new backends were created (i.e., TridentBackendConfig did not end up creating a new backend)

```

tridentctl get backend -n trident

```

NAME	STORAGE DRIVER	UUID
ontap-nas-backend	ontap-nas	52f2eb10-e4c6-4160-99fc-96b3be5ab5d7
online	25	

バックエンドは、`tbcb-backend-tbcb-ontap-nas-backend TridentBackendConfig` オブジェクトを使用して完全に管理されるようになります。

`TridentBackendConfig` を使用してバックエンドを管理 `tridentctl`

`tridentctl` は、`TridentBackendConfig` を使用して作成されたバックエンドを一覧表示するために使用できます。さらに、管理者は `tridentctl` を通じてそのようなバックエンドを完全に管理することもでき、`TridentBackendConfig` を削除し、`spec.deletionPolicy` が `retain` に設定されていることを確認できます。

ステップ0: バックエンドを特定する

例えば、次のバックエンドが `TridentBackendConfig` を使用して作成されたとします：

```
kubectl get tbc backend-tbcb-ontap-nas -n trident -o wide
NAME                                BACKEND NAME                                BACKEND UUID
PHASE    STATUS    STORAGE DRIVER    DELETION POLICY
backend-tbcb-ontap-nas    ontap-nas-backend    81abcb27-ea63-49bb-b606-0a5315ac5f82    Bound    Success    ontap-nas    delete

tridentctl get backend ontap-nas-backend -n trident
+-----+-----+
+-----+-----+-----+-----+-----+-----+
|      NAME      | STORAGE DRIVER |                                UUID                                |
| STATE  | VOLUMES  |                                |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| ontap-nas-backend | ontap-nas      | 81abcb27-ea63-49bb-b606-0a5315ac5f82 |
| online  | 33        |                                |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
```

出力から、`TridentBackendConfig` が正常に作成され、バックエンドにバインドされていることがわかります [バックエンドの UUID を確認してください] 。

ステップ1: `deletionPolicy` が `retain` に設定されていることを確認します

`deletionPolicy` の値を見てみましょう。これを `retain` に設定する必要があります。これにより、`TridentBackendConfig` CR が削除されても、バックエンドの定義はそのまま残り、`tridentctl` で管理できます。

```
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME            BACKEND UUID
PHASE    STATUS    STORAGE DRIVER    DELETION POLICY
backend-tbc-ontap-san    ontap-san-backend    81abcb27-ea63-49bb-b606-
0a5315ac5f82    Bound    Success    ontap-san    delete

# Patch value of deletionPolicy to retain
kubectl patch tbc backend-tbc-ontap-san --type=merge -p
'{"spec":{"deletionPolicy":"retain"}}' -n trident
tridentbackendconfig.trident.netapp.io/backend-tbc-ontap-san patched

#Confirm the value of deletionPolicy
kubectl get tbc backend-tbc-ontap-san -n trident -o wide
NAME                                BACKEND NAME            BACKEND UUID
PHASE    STATUS    STORAGE DRIVER    DELETION POLICY
backend-tbc-ontap-san    ontap-san-backend    81abcb27-ea63-49bb-b606-
0a5315ac5f82    Bound    Success    ontap-san    retain
```



‘deletionPolicy’が‘retain’に設定されていない限り、次のステップに進まないでください。

ステップ2：TridentBackendConfig CRを削除する

最後のステップは、TridentBackendConfig CRを削除することです。‘deletionPolicy’が‘retain’に設定されていることを確認した後、削除を進めることができます：

```
kubectl delete tbc backend-tbc-ontap-san -n trident
tridentbackendconfig.trident.netapp.io "backend-tbc-ontap-san" deleted

tridentctl get backend ontap-san-backend -n trident
+-----+-----+
+-----+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |                               UUID
| STATE  | VOLUMES |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| ontap-san-backend | ontap-san      | 81abcb27-ea63-49bb-b606-
0a5315ac5f82 | online |          33 |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
```

‘TridentBackendConfig’オブジェクトを削除すると、Tridentはバックエンド自体を実際に削除せずに、単に削除だけです。

著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。