



バックエンドを管理する Trident

NetApp
January 15, 2026

目次

バックエンドを管理する	1
kubectl を使用してバックエンド管理を実行する	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 でバックエンド管理を実行する	2
バックエンドを作成する	2
バックエンドを削除する	2
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識別する	3
バックエンド管理オプション間を移動する	4
バックエンドを管理するためのオプション	4
管理 tridentctl`バックエンドを使用する`TridentBackendConfig	4
管理 TridentBackendConfig`バックエンドを使用する`tridentctl	9

バックエンドを管理する

kubectl を使用してバックエンド管理を実行する

バックエンド管理操作を実行する方法について学習します。 kubectl。

バックエンドを削除する

削除することで TridentBackendConfig、 Tridentにバックエンドを削除/保持するように指示します（`deletionPolicy`）。バックエンドを削除するには、 `deletionPolicy` 削除するように設定されています。削除するには `TridentBackendConfig、必ず deletionPolicy` 保持するように設定されています。これにより、バックエンドがまだ存在し、以下を使用して管理できることが保証されます。
`tridentctl。`

次のコマンドを実行します。

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

Tridentは、使用されていたKubernetes Secretsを削除しません。 TridentBackendConfig。 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
```

実行することもできます `tridentctl get backend -n trident` または `tridentctl get backend -o yaml -n trident` 存在するすべてのバックエンドのリストを取得します。このリストには、 `tridentctl。`

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資格情報が変更されました。資格情報を更新するには、 `TridentBackendConfig` オブジェクトを更新する必要があります。 Trident は、 提供された最新の資格情報を使用してバックエンドを自動的に更新します。 Kubernetes シークレットを更新するには、次のコマンドを実行します。

```
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。
 - 更新できます `TridentBackendConfig` 次のコマンドを使用して、 Kubernetes 経由でオブジェクトを直

接実行します。

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

- あるいは、既存の `TridentBackendConfig` 次のコマンドを使用して CR を実行します。

```
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、次のコマンドを実行します。 `kubectl get tbc <tbc-name> -o yaml -n trident` または `kubectl describe tbc <tbc-name> -n trident`。
- 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 でバックエンド管理を実行する

バックエンド管理操作を実行する方法について学習します。 tridentctl。

バックエンドを作成する

作成後 "バックエンド設定ファイル"、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`再度コマンドを実行します。

バックエンドを削除する

Tridentからバックエンドを削除するには、次の手順を実行します。

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、新しいボリュームがプロビジョニングされなくなります。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、次の手順を実行します。

- 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`再度コマンドを実行します。

バックエンドを使用するストレージクラスを識別する

これはJSONで答えられる質問の例です。`tridentctl` バックエンド オブジェクトの出力。これは、`jq`インストールする必要があるユーティリティ。

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

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

バックエンド管理オプション間を移動する

Tridentでバックエンドを管理するさまざまな方法について学びます。

バックエンドを管理するためのオプション

の導入により `TridentBackendConfig` 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。

- ・ バックエンドは以下を使用して作成できます `tridentctl`、管理される `TridentBackendConfig`?
- ・ バックエンドは以下を使用して作成できます `TridentBackendConfig`、管理するには `tridentctl`?

管理 `tridentctl`、バックエンドを使用する `TridentBackendConfig`

このセクションでは、以下の手順で作成されたバックエンドを管理するために必要な手順について説明します。`tridentctl` Kubernetes インターフェースから直接作成することで `TridentBackendConfig` オブジェクト。

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`、なぜなら、それらは `tridentctl`。
- ・ 作成された新しいバックエンド `tridentctl`、他の `TridentBackendConfig` オブジェクトが存在します。

どちらのシナリオでも、バックエンドは引き続き存在し、Tridentがボリュームをスケジュールして操作します。管理者はここで 2 つの選択肢のいずれかを選択できます。

- ・ 引き続き使用 `tridentctl`、これを使用して作成されたバックエンドを管理します。
- ・ 作成したバックエンドをバインドする `tridentctl`、新しい `TridentBackendConfig` 物体。そうすることで、バックエンドは次のように管理されることになります。 `kubectl`、そしてそうではない `tridentctl`。

既存のバックエンドを管理するには `kubectl` を作成する必要があります `TridentBackendConfig` 既存のバックエンドにバインドします。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。シークレットには、Trident がストレージ クラスター/サービスと通信するために必要な資格情報が含まれています。
2. 作成する `TridentBackendConfig` 物体。これには、ストレージ クラスター/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（例えば `spec.backendName`、`spec.storagePrefix`、`spec.storageDriverName`、等々）。

`spec.backendName` 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

を作成するには `TridentBackendConfig` 既存のバックエンドにバインドする場合は、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: 作成する `TridentBackendConfig` CR

次のステップは、`TridentBackendConfig` 既存の CR に自動的にバインドされる `ontap-nas-backend` (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定義されている `spec.backendName`。
- 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。
- 資格情報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` 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: mysqldb
        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: ステータスを確認する `TridentBackendConfig` CR

その後 TridentBackendConfig が作成された場合、そのフェーズは `Bound`。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
PHASE    STATUS
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
| STATE  | VOLUMES | 
+-----+-----+
+-----+-----+-----+
| ontap-nas-backend | ontap-nas      | 52f2eb10-e4c6-4160-99fc-
96b3be5ab5d7 | online |      25 |
+-----+-----+
+-----+-----+-----+

```

バックエンドは、 tbc-ontap-nas-backend `TridentBackendConfig` 物体。

管理 TridentBackendConfig、バックエンドを使用する `tridentctl`

`tridentctl` 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
`TridentBackendConfig`。さらに、管理者は、次のようなバックエンドを完全に管理すること
もできます。 `tridentctl` 削除することで `TridentBackendConfig`、そして確認する
`spec.deletionPolicy` 設定されている `retain`。

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次のように作成されたとします。 TridentBackendConfig :

```

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

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 |
+-----+-----+
+-----+-----+

```

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ます [バックエンドのUUIDを確認してください]。

ステップ1: 確認 deletionPolicy 設定されている `retain`

の値を見てみましょう deletionPolicy。これを設定する必要があります retain。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。

```

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

```



以下の場合を除き、次のステップに進まないでください。`deletionPolicy`が`retain`で設定されている場合。

ステップ2: 削除する `TridentBackendConfig` CR

最後のステップは、`TridentBackendConfig` CR。確認後、`deletionPolicy`が`retain`で設定されている場合、削除することができます。

```
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 |
+-----+-----+
+-----+-----+
```

削除すると、`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、単にオブジェクトを削除します。

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隨時、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 5225.227-19（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権を有し、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権については、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。