



# バックエンドの管理

## Astra Trident

NetApp  
January 14, 2026

# 目次

バックエンドの管理	1
kubectl を使用してバックエンド管理を実行します	1
バックエンドを削除します	1
既存のバックエンドを表示します	1
バックエンドを更新します	1
tridentctl を使用してバックエンド管理を実行します	2
バックエンドを作成します	2
バックエンドを削除します	2
既存のバックエンドを表示します	3
バックエンドを更新します	3
バックエンドを使用するストレージクラスを特定します	3
バックエンド管理オプション間を移動します	4
バックエンドを管理するためのオプション	4
次を使用してバックエンドを TridentBackendConfig`管理`tridentctl	4
次を使用してバックエンドを tridentctl`管理`TridentBackendConfig	8

# バックエンドの管理

## kubectl を使用してバックエンド管理を実行します

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。 kubectl

### バックエンドを削除します

を削除することで TridentBackendConfig、Astra Tridentでバックエンドを（に基づいて）削除または保持するように指示し deletionPolicy`ます。バックエンドを削除するには、がdeleteに設定されていることを確認します `deletionPolicy。のみを削除するには TridentBackendConfig、がretainに設定されていることを確認します deletionPolicy。これにより、バックエンドが引き続き存在し、を使用して管理できるようになります tridentctl。

次のコマンドを実行します。

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

Astra Tridentでは、で使用されていたKubernetesシークレットは削除されません TridentBackendConfig。Kubernetes ユーザは、シークレットのクリーンアップを担当します。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してください。

### 既存のバックエンドを表示します

次のコマンドを実行します。

```
kubectl get tbc -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 `tridentctl get backend -n trident。このリストには、で作成されたバックエンドも含まれ `tridentctl`ます。

### バックエンドを更新します

バックエンドを更新する理由はいくつかあります。

- ストレージシステムのクレデンシャルが変更されている。クレデンシャルを更新するには、オブジェクトで使用されるKubernetes Secretを `TridentBackendConfig`更新する必要があります。Astra Trident が、提供された最新のクレデンシャルでバックエンドを自動的に更新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（使用する ONTAP SVM の名前など）を更新する必要があります。

- 次のコマンドを使用して、Kubernetesから直接オブジェクトを更新できます  
TridentBackendConfig。

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

- または、次のコマンドを使用して既存のCRに変更を加えることもできます  
TridentBackendConfig。

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



- バックエンドの更新に失敗した場合、バックエンドは最後の既知の設定のまま残ります。ログを表示して原因を特定するには、またはを `kubectl describe tbc <tbc-name> -n trident` 実行し `kubectl get tbc <tbc-name> -o yaml -n trident` ます。
- 構成ファイルで問題を特定して修正したら、update コマンドを再実行できます。

## tridentctl を使用してバックエンド管理を実行します

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。 tridentctl

### バックエンドを作成します

を作成したら"[バックエンド構成ファイル](#)"、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、バックエンドの設定に何か問題があります。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
```

構成ファイルの問題を特定して修正したら、コマンドをもう一度実行できます create。

### バックエンドを削除します

Astra 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
```



Astra Trident で、まだ存在しているこのバックエンドからボリュームとスナップショットをプロビジョニングしている場合、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」状態のままになり、Trident は削除されるまでそれらのボリュームとスナップショットを管理し続けます。

## 既存のバックエンドを表示します

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
```

- すべての詳細を確認するには、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します

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、バックエンドの設定に問題があるか、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
```

構成ファイルの問題を特定して修正したら、コマンドをもう一度実行できます `update`。

## バックエンドを使用するストレージクラスを特定します

これは、バックエンドオブジェクト用に出力するJSONで回答できる質問の例 `tridentctl` です。これは、インストールする必要があるユーティリティを使用し `jq` ます。

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

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

## バックエンド管理オプション間を移動します

Astra Trident でバックエンドを管理するさまざまな方法をご確認ください。

### 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2つの独自の方法で管理できるようになりました。これには、次のような質問があります。

- を使用して作成したバックエンドはで管理 `TridentBackendConfig` で `tridentctl` ますか。
- を使用して作成したバックエンドはを使用して管理 `tridentctl` で `TridentBackendConfig` ますか。

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

このセクションでは、オブジェクトを作成してKubernetesインターフェイスから直接 `TridentBackendConfig` 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 `tridentctl` ます。

これは、次のシナリオに該当します。

- を使用して作成された `tridentctl`既存のバックエンドにはが`ありません。  
``TridentBackendConfig``
- 他のオブジェクトが存在するときに、 `TridentBackendConfig`で`作成された新しいバックエンド  
``tridentctl`。`

どちらの場合も、Trident でボリュームをスケジューリングし、処理を行っているバックエンドは引き続き存在します。管理者には次の2つの選択肢があります。

- を使用して作成されたバックエンドの管理に引き続き使用し `tridentctl` ます。
- を使用して作成したバックエンドを新しいオブジェクトに `TridentBackendConfig`バインドし``  
``tridentctl`` ます。これは、バックエンドがではなくを使用して管理されることを意味します  
``kubectl tridentctl`。`

を使用して既存のバックエンドを管理するには `kubectl`、既存のバックエンドにバインドするを作成する必要があります `TridentBackendConfig`。その仕組みの概要を以下に示します。

1. Kubernetes Secret を作成します。シークレットには、ストレージクラスタ / サービスと通信するために Trident から必要なクレデンシャルが含まれています。
2. オブジェクトを作成し `TridentBackendConfig`` ます。ストレージクラスタ / サービスの詳細を指定し、前の手順で作成したシークレットを参照します。同一の設定パラメータ (、  
``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`。`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。`

## 手順 0 : バックエンドを特定します

既存のバックエンドにバインドするを作成するには TridentBackendConfig、バックエンド設定を取得する必要があります。この例では、バックエンドが次の JSON 定義を使用して作成されているとします。

```
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"
    }
  }
]
}
```

### 手順 1 : Kubernetes Secret を作成します

次の例に示すように、バックエンドのクレデンシャルを含むシークレットを作成します。

```
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
```

### 手順2 : CRを作成する TridentBackendConfig

次の手順では、（この例のように）既存のに自動的にバインドするCRを `ontap-nas-backend` 作成し `TridentBackendConfig` ます。次の要件が満たされていることを確認します。

- には、同じバックエンド名が定義されていて `spec.backendName` ます。
- 設定パラメータは元のバックエンドと同じです。
- 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である必要があります。
- 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、は `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
```

### 手順3：CRのステータスを確認する TridentBackendConfig

が作成されたら TridentBackendConfig、そのフェーズはにする必要があります Bound。また、既存のバックエンドと同じバックエンド名と UUID が反映されている必要があります。

```

kubect1 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 |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

```

これで、バックエンドはオブジェクトを使用して完全に管理され `tbc-ontap-nas-backend` `TridentBackendConfig` ます。

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

```

`tridentctl` を使用して作成されたバックエンドの一覧表示に使用でき
`TridentBackendConfig` ます。さらに、管理者は、を削除して、がに設定されている
`retain` ことを確認する `spec.deletionPolicy` ことで、
`TridentBackendConfig` こ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`。

```

手順 0 : バックエンドを特定します

たとえば、次のバックエンドがを使用して作成されたとし `TridentBackendConfig` 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
```

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

出力からは、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 `TridentBackendConfig` ます ([Observe the backend's UUID]) 。

手順1: **Confirm**がに設定されている `retain`` ことを確認 `deletionPolicy`

の値を見てみましょう `deletionPolicy`。これはに設定する必要があり `retain` ます。これにより、CRが削除されてもバックエンド定義が存在し、で管理できるように `TridentBackendConfig` なり `tridentctl` 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
```



がに設定され `retain` していない場合は、次の手順に進まないで `deletionPolicy` ください。

## 手順2: CRを削除する TridentBackendConfig

最後のステップはCRを削除することです TridentBackendConfig。がに設定されている `retain` ことを確認したら `deletionPolicy`、削除を続行できます。

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

オブジェクトが削除されると TridentBackendConfig、Astra Tridentはバックエンド自体を削除せず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。