



ストレージクラスの作成と管理

Trident

NetApp
March 05, 2026

目次

ストレージクラスの作成と管理	1
ストレージクラスを作成する	1
Kubernetes StorageClassオブジェクトを構成する	1
ストレージクラスを作成する	1
ストレージクラスの管理	3
既存のストレージクラスを表示する	3
デフォルトのストレージクラスを設定する	4
ストレージクラスのバックエンドを識別する	4
ストレージクラスを削除する	4

ストレージクラスの作成と管理

ストレージクラスを作成する

Kubernetes StorageClass オブジェクトを構成し、ボリュームのプロビジョニング方法をTrident に指示するストレージクラスを作成します。

Kubernetes StorageClassオブジェクトを構成する

その "Kubernetes StorageClassオブジェクト"そのクラスに使用されるプロビジョナーとしてTrident を識別し、ボリュームをプロビジョニングする方法をTrident に指示します。例えば：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ontap-gold
provisioner: csi.trident.netapp.io
mountOptions:
  - nfsvers=3
  - nolock
parameters:
  backendType: "ontap-nas"
  media: "ssd"
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

参照"[KubernetesとTridentオブジェクト](#)"ストレージクラスがどのように相互作用するかの詳細については、PersistentVolumeClaim Trident がボリュームをプロビジョニングする方法を制御するためのパラメータ。

ストレージクラスを作成する

StorageClass オブジェクトを作成したら、ストレージクラスを作成できます。[\[ストレージクラスのサンプル\]](#)使用したり変更したりできるいくつかの基本的なサンプルを提供します。

手順

1. これはKubernetesオブジェクトなので、`kubectl` Kubernetes で作成します。

```
kubectl create -f sample-input/storage-class-basic-csi.yaml
```

2. これで、Kubernetes とTridentの両方に **basic-csi** ストレージクラスが表示され、Trident はバックエンドでプールを検出しているはずです。

```
kubectl get sc basic-csi
```

NAME	PROVISIONER	AGE
basic-csi	csi.trident.netapp.io	15h

```
./tridentctl -n trident get storageclass basic-csi -o json
```

```
{
  "items": [
    {
      "Config": {
        "version": "1",
        "name": "basic-csi",
        "attributes": {
          "backendType": "ontap-nas"
        },
        "storagePools": null,
        "additionalStoragePools": null
      },
      "storage": {
        "ontapnas_10.0.0.1": [
          "aggr1",
          "aggr2",
          "aggr3",
          "aggr4"
        ]
      }
    }
  ]
}
```

ストレージクラスのサンプル

Tridentは **"特定のバックエンド向けのシンプルなストレージクラス定義"**。

あるいは、編集することもできます `sample-input/storage-class-csi.yaml.templ` インストーラーに付属のファイルと置き換えます `BACKEND_TYPE` ストレージ ドライバー名を使用します。

```

./tridentctl -n trident get backend
+-----+-----+-----+-----+
+-----+-----+
|   NAME   | STORAGE DRIVER |           UUID           |
STATE | VOLUMES |
+-----+-----+-----+-----+
+-----+-----+
| nas-backend | ontap-nas      | 98e19b74-aec7-4a3d-8dcf-128e5033b214 |
online |         0 |
+-----+-----+-----+-----+
+-----+-----+

cp sample-input/storage-class-csi.yaml.templ sample-input/storage-class-
basic-csi.yaml

# Modify __BACKEND_TYPE__ with the storage driver field above (e.g.,
ontap-nas)
vi sample-input/storage-class-basic-csi.yaml

```

ストレージクラスの管理

既存のストレージ クラスを表示したり、デフォルトのストレージ クラスを設定したり、ストレージ クラスのバックエンドを識別したり、ストレージ クラスを削除したりできます。

既存のストレージクラスを表示する

- 既存の Kubernetes ストレージ クラスを表示するには、次のコマンドを実行します。

```
kubectl get storageclass
```

- Kubernetes ストレージ クラスの詳細を表示するには、次のコマンドを実行します。

```
kubectl get storageclass <storage-class> -o json
```

- Trident の同期されたストレージ クラスを表示するには、次のコマンドを実行します。

```
tridentctl get storageclass
```

- Trident の同期ストレージ クラスの詳細を表示するには、次のコマンドを実行します。

```
tridentctl get storageclass <storage-class> -o json
```

デフォルトのストレージクラスを設定する

Kubernetes 1.6 では、デフォルトのストレージ クラスを設定する機能が追加されました。これは、ユーザーが永続ボリューム要求 (PVC) でストレージ クラスを指定しない場合に、永続ボリュームをプロビジョニングするために使用されるストレージ クラスです。

- アノテーションを設定してデフォルトのストレージクラスを定義します `storageclass.kubernetes.io/is-default-class` ストレージ クラス定義で true に設定します。仕様によれば、その他の値または注釈の欠如は false として解釈されます。
- 次のコマンドを使用して、既存のストレージ クラスをデフォルトのストレージ クラスとして構成できます。

```
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
```

- 同様に、次のコマンドを使用して、デフォルトのストレージ クラス注釈を削除できます。

```
kubectl patch storageclass <storage-class-name> -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"false"}}}'
```

Tridentインストーラー バンドルにも、この注釈を含む例があります。



クラスター内に一度に存在できるデフォルトのストレージ クラスは 1 つだけです。Kubernetes では、技術的には複数のストレージ クラスを持つことを禁止していませんが、デフォルトのストレージ クラスがまったく存在しないかのように動作します。

ストレージクラスのバックエンドを識別する

これはJSONで答えられる質問の例です。tridentctl Tridentバックエンド オブジェクトの出力。これは、jq ユーティリティを最初にインストールする必要がある場合があります。

```
tridentctl get storageclass -o json | jq '[.items[] | {storageClass: .Config.name, backends: [.storage]|unique}]'
```

ストレージクラスを削除する

Kubernetes からストレージ クラスを削除するには、次のコマンドを実行します。

```
kubectl delete storageclass <storage-class>
```

`<storage-class>`ストレージ クラスに置き換える必要があります。

このストレージ クラスを通じて作成された永続ボリュームはそのまま残り、Tridentによって引き続き管理されます。



Tridentは空白を強制する `fsType` 作成されるボリュームに対して、iSCSIバックエンドの場合、強制することが推奨されます `parameters.fsType` StorageClass 内。既存のStorageClassesを削除して、`parameters.fsType` 指定された。

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。