创建和管理存储类
Trident

NetApp
February 02, 2026

This PDF was generated from https://docs.netapp.com/zh-cn/trident/trident-use/create-stor-class.html on February 02, 2026. Always check docs.netapp.com for the latest.

目录

创建和管理存储类 . . . 1
- 创建存储类。 . . . 1
  - 配置Kubnetes StorageClass对象 . . . 1
  - 创建存储类。 . . . 1
- 管理存储类 . . . 3
  - 查看现有存储类 . . . 3
  - 设置默认存储类 . . . 4
  - 确定存储类的后端 . . . 4
  - 删除存储类 . . . 4

创建和管理存储类

创建存储类。

配置Kubnetes StorageClass对象并创建存储类、以指示Trident如何配置卷。

配置**Kubnetes StorageClass**对象

```
https://kubernetes.io/docs/concepts/storage/storage-classes/["Kubnetes StorageClass对象"^]将Trident标识为用于该类的配置程序、并指示Trident如何配置卷。例如：
```

```
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
```

有关存储类如何与和参数交互以控制Trident如何配置卷的详细信息 `PersistentVolumeClaim`、请参见"Kubernetes 和 Trident 对象"。

创建存储类。

创建StorageClass对象后、您可以创建存储类。 [存储类示例] 提供了一些可供您使用或修改的基本示例。

步骤

1. 这是一个Kubbernetes对象、因此请使用 `kubectl` 以在Kubernetes中创建。

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

2. 现在，Kubernetes和Trident中都应显示一个*BASIC－Csi*存储类，并且Trident应已发现后端的池。

```
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"
        ]
      }
    }
  ]
}
```

存储类示例

Trident提供 "适用于特定后端的简单存储类定义"。

或者、您也可以进行编辑 `sample-input/storage-class-csi.yaml.templ` 安装程序随附的文件、然后进行替换 *`BACKEND_TYPE`* 和存储驱动程序名称。

```
./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
```

管理存储类

您可以查看现有存储类、设置默认存储类、确定存储类后端以及删除存储类。

查看现有存储类

- 要查看现有 Kubernetes 存储类，请运行以下命令：

```
kubectl get storageclass
```

- 要查看 Kubernetes 存储类详细信息，请运行以下命令：

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

- 要查看Trident的同步存储类、请运行以下命令：

```
tridentctl get storageclass
```

- 要查看Trident的已同步存储类详细信息、请运行以下命令：

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

设置默认存储类

Kubernetes 1.6 增加了设置默认存储类的功能。如果用户未在永久性卷声明（ PVC ）中指定永久性卷，则此存储类将用于配置永久性卷。

- 通过在存储类定义中将标注 `storageclass.Kubernetes 。 io/is-default-class` 设置为 true 来定义默认存储类。根据规范，任何其他值或标注不存在均视为 false 。
- 您可以使用以下命令将现有存储类配置为默认存储类：

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

- 同样，您也可以使用以下命令删除默认存储类标注：

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

Trident 安装程序包中也有包含此标注的示例。

集群中一次只能有一个默认存储类。Kubernetes 在技术上不会阻止您拥有多个存储类，但其行为就像根本没有默认存储类一样。

确定存储类的后端

以下是您可以使用为Trident后端对象输出的JSON回答的问题示例 `tridentctl`。这将使用 `jq`实用程序、您可能需要先安装该实用程序。

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

删除存储类

要从 Kubernetes 中删除存储类，请运行以下命令：

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

` < 存储类 >` 应替换为您的存储类。

通过此存储类创建的任何永久性卷将保持不变、Trident将继续对其进行管理。

Trident会对其创建的卷强制使用空白 `fsType`。对于iSCSI后端、建议在StorageClass中强制实施 `parameters.fsType`。您应删除现有StorageClasses、然后使用指定的重新创建它们 `parameters.fsType`。

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。 美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 http://www.netapp.com/TM 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。