创建和管理存储类

Trident

NetApp

July 01, 2026

目录

创建和管理存储类

创建存储类

配置 Kubernetes StorageClass 对象并创建存储类，以指导 Trident 如何配置卷。

配置 **Kubernetes StorageClass** 对象

`https://kubernetes.io/docs/concepts/storage/storage-classes/["Kubernetes StorageClass 对象"^] 将 Trident 识别为用于该类的预配程序，并指示 Trident 如何预配卷。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
```

请参阅"Kubernetes 和 Trident 对象"，了解存储类如何与 `PersistentVolumeClaim`交互，以及用于控制 Trident 配置卷的参数详情。

创建存储类

创建 StorageClass 对象后，可以创建存储类。[存储类示例] 提供了一些可以使用或修改的基本示例。

步骤

1. 这是一个 Kubernetes 对象，因此请使用 `kubectl` 在 Kubernetes 中创建它。

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

2. 现在，您应该在 Kubernetes 和 Trident 中都看到 **basic-csi** 存储类，并且 Trident 应该已经发现了后端的池。

```
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 可提供 "用于特定后端的简单存储类定义"。

或者，您可以编辑安装程序附带的 `sample-input/storage-class-csi.yaml.templ` 文件，并将 *`BACKEND_TYPE`* 替换为存储驱动程序名称。

```
./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、标识存储类后端以及删除存储类。

查看现有存储类

- 要查看现有的 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 Persistent Volume Claim (PVC) 中指定存储类，则此存储类将用于配置 Persistent Volume。

- 通过在存储类定义中将注释 `storageclass.kubernetes.io/is-default-class`设置为 true 来定义默认存储类。根据规范，注释的任何其他值或缺失将被解释为 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"}}}'
```

- 类似地，您可以使用以下命令删除默认存储类注释：

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

Trident 安装程序包中还有包含此注释的示例。

群集中一次只能有一个默认存储类。从技术上讲，Kubernetes 不会阻止您拥有多个存储类，但它会表现得好像根本没有默认存储类一样。

标识存储类的后端

这是一个示例，说明您可以使用 `tridentctl` 为 Trident 后端对象输出的 JSON 来回答的问题类型。这使用了 `jq` 实用程序，您可能需要先安装它。

```
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>
```

`<storage-class>` 应替换为您的存储类。

通过此存储类创建的任何持久卷将保持不变，Trident 将继续管理它们。

Trident 为其创建的卷强制使用空白 `fsType`。对于 iSCSI 后端，建议在 StorageClass 中强制执行 `parameters.fsType`。您应该删除现有的 StorageClasses 并使用指定的 `parameters.fsType`重新创建它们。