

## 36Kubernetes 系列（三十三）跨 StorageClass 迁移存储

### Kubernetes 系列（三十三）跨 StorageClass 迁移存储

本文着重介绍如何在两个 StorageClass 类型之前迁移 PV PVC，为什么？我就碰到我们 nfs 的存储需要切换环境，要换 nfs client 的情况，当然不同类型的存储可能存在便捷的迁移方式，这边讲述的主要还是针对所有存储都通用的迁移方式。

#### 迁移环境信息

Storage Class : 旧 nfs-client / 新 nfs-client-new

PVC: test/pvc-busybox

Bound PV: pvc-a9f0dcf6-5fba-486e-9d2a-7c62c6d64e36

PVC 挂载应用: deployment - busybox

需要将如上的 PV 数据从 nfs-client 迁移到 nfs-client-new 中去

#### 实施步骤

##### 备份

备份 pv 以及 pvc 的配置

```
kubectl get pv pvc-a9f0dcf6-5fba-486e-9d2a-7c62c6d64e36 -o yaml > pv_bak.yaml
kubectl get pvc pvc-busybox -n test -o yaml > pvc_bak.yaml
```

##### 缩减应用

缩减 PVC 挂载的所有应用控制器副本数为 0，这样一来，就可以对PVC 进行后续的删除以及更新操作，例如：

```
kubectl scale deployment busybox -n test --replicas=0
```

ps：如果是其他类型的控制器，更换相应的类型即可

##### 创建新的 PVC

按照旧的 PVC 一模一样的配置创建 PVC,指定新的 storage class:

```
kubectl apply -f pvc_replace.yaml
```



```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc-busybox-new
  namespace: test
spec:
  storageClassName: nfs-client-new
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
```

## 拷贝存储数据

现在作为替换的新的 pvc 已创建，现在需要运行一个 deployment 挂载 新旧 两个 pvc，以实现数据拷贝：

```
kubectl apply -f deployment_cp.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: test
  labels:
    app: datacopy
  name: datacopy
spec:
  replicas: 1
  selector:
    matchLabels:
      app: datacopy
  template:
    metadata:
      labels:
        app: datacopy
    spec:
      containers:
        - name: datacopy
          image: busybox
          command:
            - 'sleep'
          args:
            - infinity
          volumeMounts:
            - name: old-pvc
              readOnly: false
              mountPath: /mnt/old
            - name: new-pvc
              readOnly: false
              mountPath: /mnt/new
      volumes:
        - name: old-pvc
          persistentVolumeClaim:
            claimName: pvc-busybox
        - name: new-pvc
          persistentVolumeClaim:
            claimName: pvc-busybox-new
```

注意：将以上对应的 命名空间、pvc 信息替换为实际的，然后执行部署

部署成功后，进入容器命令行，检查 /mnt/old /mnt/new 目录，旧的目录是否有数据，新的目录是否是空的；

之后执行命令：`(cd /mnt/old; tar -cf - .) | (cd /mnt/new; tar -xpf -)` 确保所有数据的所有权和权限被继承；

执行完成后，验证新的 PV 的挂载点是否包含旧的PV 的数据，以及所有权限是否被正确继承。

至此，复制数据就完成了，现在我们需要将 datacopy deployment 的副本缩减为0，这样两个pvc 就和它失去关联，可以执行后续操作：

```
kubectl scale deployment datacopy -n test --replicas=0
```

ps: 如果该deployment 用不到了, 可以直接删除

## 修改新旧 PV 的删除策略

默认情况下, PV 的回收策略是 Delete,也就是一旦删除 PVC, 与之绑定的 PV和 PV 里面的数据都不会被删除, 因为后面需要删除 PVC,所以为了防止数据丢失, 需要确认 删除策略:

```
[root@k8s-master yy-work]# kubectl describe pv pvc-a9f0dcf6-5fba-486e-9d2a-7c62c6d64e36
Name:          pvc-a9f0dcf6-5fba-486e-9d2a-7c62c6d64e36
Labels:        <none>
Annotations:   pv.kubernetes.io/provisioned-by: cluster.local/nfs-client-provisioner
Finalizers:    [kubernetes.io/pv-protection]
StorageClass:  nfs-client
Status:        Bound
Claim:         test/pvc-busybox
Reclaim Policy: Delete
Access Modes:  RWX
VolumeMode:    Filesystem
Capacity:      10Gi
Node Affinity: <none>
Message:
Source:
  Type:        NFS (an NFS mount that lasts the lifetime of a pod)
  Server:      172.28.104.42
  Path:        /data/nfsdata/test-pvc-busybox-pvc-a9f0dcf6-5fba-486e-9d2a-7c62c6d64e36
  ReadOnly:    false
Events:        <none>
```

可以看到, Reclaim Policy 的值为 Delete, 需要修改:

```
kubectl patch pv pvc-a9f0dcf6-5fba-486e-9d2a-7c62c6d64e36 -p '{"spec":{"persistentVolumeReclaimPolicy":"ReclaimPolicy":"Retain"}}
```

注意: 新旧两个 PV 都要检查

## 删除新旧 PVC

```
kubectl delete pvc pvc-busybox -n test
kubectl delete pvc pvc-busybox-new -n test
```

## 解绑新的 PV PVC 关系

```
kubectl edit pv pvc-dcceb327-5f94-4d10-a56d-d854964690fd
```

将 claimRef 的内容全部删除，然后保存

## 创建旧的 PVC 同名的 PVC，并保持相关参数一致，同时绑定新的 PV

kubectl apply -f pvc\_new.yaml

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  # 保持与旧的 PVC 同名
  name: pvc-busybox
  namespace: test
spec:
  # 使用新的 sc 类型
  storageClassName: nfs-client-new
  # 配置保持与旧的 PVC 一致
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  volumeMode: Filesystem
  # 新的 PV
  volumeName: pvc-dcceb327-5f94-4d10-a56d-d854964690fd
```

## 恢复原先应用副本数

```
kubectl scale deployment busybox -n test --replicas=1
```

## 恢复新的 PV 的删除策略(需要)

```
kubectl patch pv pvc-dcceb327-5f94-4d10-a56d-d854964690fd -p '{"spec":{"persistentVolumeReclaimPolicy":"De
```

## 删除旧PV

确认无误后，删除旧的 PV，至此完成迁移

欢迎关注我的公众号“云原生拓展”，原创技术文章第一时间推送。