Persistent Volumes

Table of Content

[Lifecycle of a volume and claim](#)
- [Provisioning](#)
- [Binding](#)
- [Using](#)
- [Storage Object in Use Protection](#)
- [Reclaiming](#)

[Type of Volumes](#)

[How to attach PV to Pod](#)

[References](#)

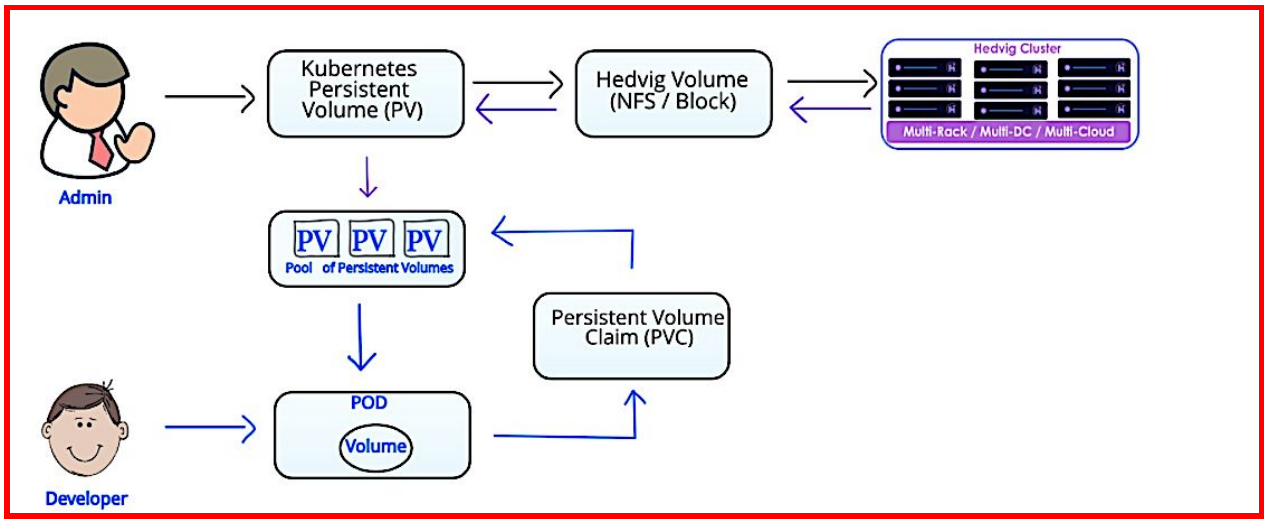
Terms

pv= persistent volume
pvc= persistent volume claim

A **PersistentVolume (PV)** is a piece of storage in the cluster that has been provisioned by an administrator. It is a resource in the cluster just like a node is a cluster resource. PVs are volume plugins like Volumes, but have a life cycle independent of any individual pod that uses the PV. This API object captures the details of the implementation of the storage, be that NFS, iSCSI, or a cloud-provider-specific storage system.

A **PersistentVolumeClaim (PVC)** is a request for storage by a user. It is similar to a pod. Pods consume node resources and PVCs consume PV resources. Pods can request specific levels of resources (CPU and Memory). Claims can request specific size and access modes (e.g., can be mounted once read/write or many times read-only).

Basic Idea for Kubernetes Volume



Lifecycle of a volume and claim

1. **Provisioning**

   Creating PV in this step

2. **Binding**

   Create PVC in this step

3. **Using**

   Bound PV using PVC

4. **Storage Object in Use Protection**

   Give protection of bounded PCV to avoid uncertainty like data loss

5. **Reclaiming**

   Delete current PVC and Set data availability policy for PV. like after reclaiming delete data or retain previously written data on volume

Type of Volumes

Read this to get more types
https://kubernetes.io/docs/concepts/storage/volumes/

How to attach PV to Pod

1. Write specification for PV

   pv-test.yaml

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: task-pv-volume
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/work/volume_test"
```

   Create volume

```
kubectl create -f pv-test.yaml
```

2. Create PCV for volume
   pv-claim.yaml

```
apiVersion: v1
metadata:
  name: task-pv-claim
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

Create PVC

```
kubectl create -f pv-claim.yaml
```

3. Attach PV to **pod** using PVC

deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ui-service
  labels:
    name: ui-service
spec:
  replicas: 2
  selector:
    matchLabels:
      name: ui-service
  template:
    metadata:
      labels:
        name: ui-service
    spec:
      volumes:                          #1st in specification section
      - name: task-pv-storage
        persistentVolumeClaim:
         claimName: task-pv-claim
      containers:
      - name: ui
        image: docker-server.com/nodeapptwo:v1.5
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 8080
        resources:
          requests:
```

```
        memory: "100Mi"
        cpu: "500m"
      limits:
        memory: "120Mi"
        cpu: "550m"
    volumeMounts:                    #2nd in container section
    - mountPath: "/usr/src/app/abc"
      name: task-pv-storage
  imagePullSecrets:
  - name: private-secret
```

Create pods using deployment.yaml

```
kubectl create -f  deployment.yaml
```

4. Exec into pod and add some files at mounted directory

References

1. https://kubernetes.io/docs/concepts/storage/persistent-volumes/
2. **TutorialPoints**

Why we can't share volume among more than 1 pvc and which volume type shares among multiple pvc
https://groups.google.com/forum/#!topic/kubernetes-users/6A1ZwSYkG8I

Learn more about volume
https://kubernetes.io/docs/tasks/configure-pod-container/configure-persistent-volume-storage/#access-control