PV,PVC and Storage Class

ow

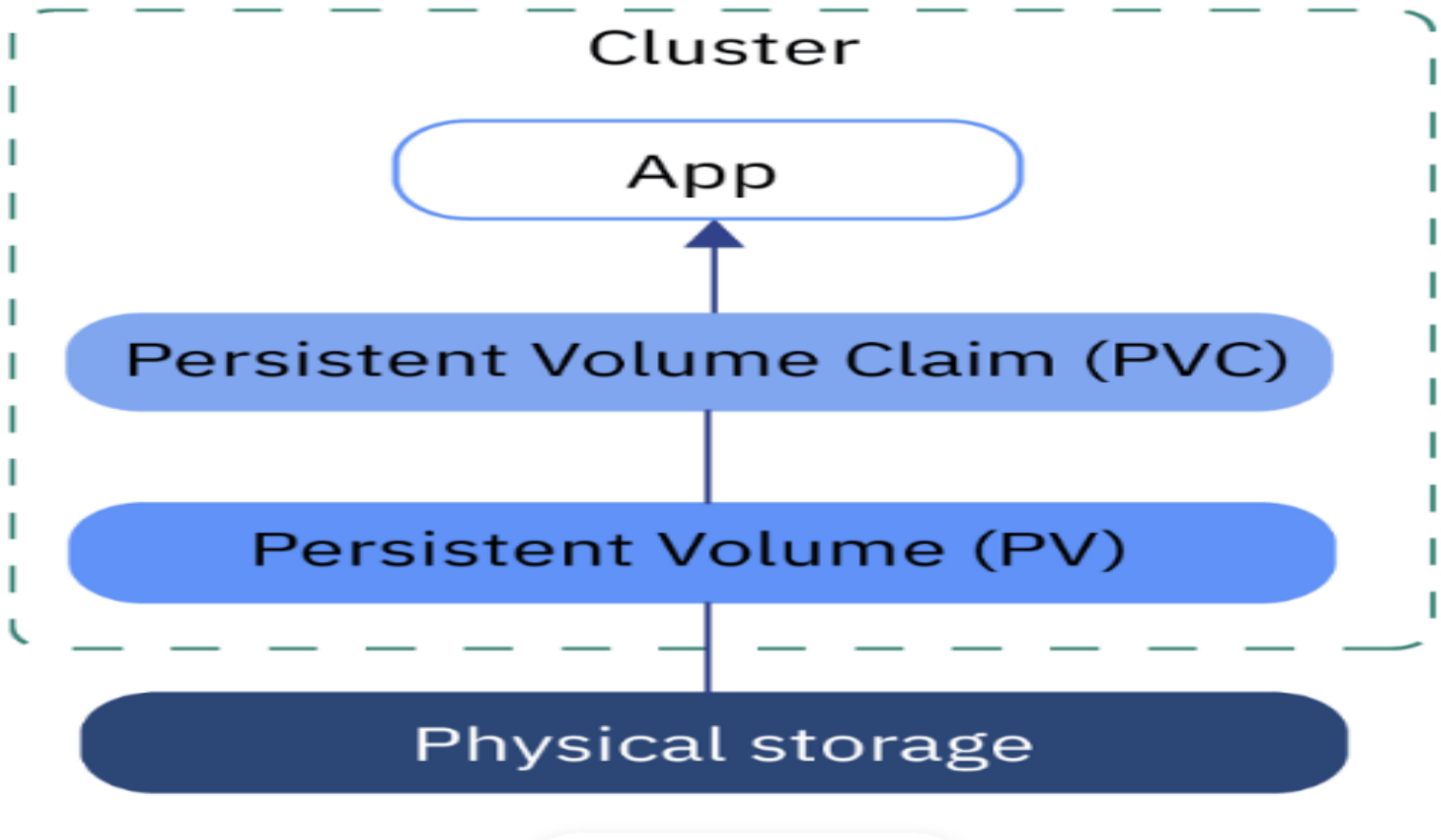
What is PV and PVC?

- The PersistentVolume subsystem provides an API for users and administrators that abstracts details of how storage is provided from how it is consumed.
- To do this, we introduce two new API resources: PersistentVolume and PersistentVolumeClaim.
- A *PersistentVolume* (PV) is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned . It is a resource in the cluster just like a node is a cluster resource.
- 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., they can be mounted ReadWriteOnce, ReadOnlyMany or ReadWriteMany

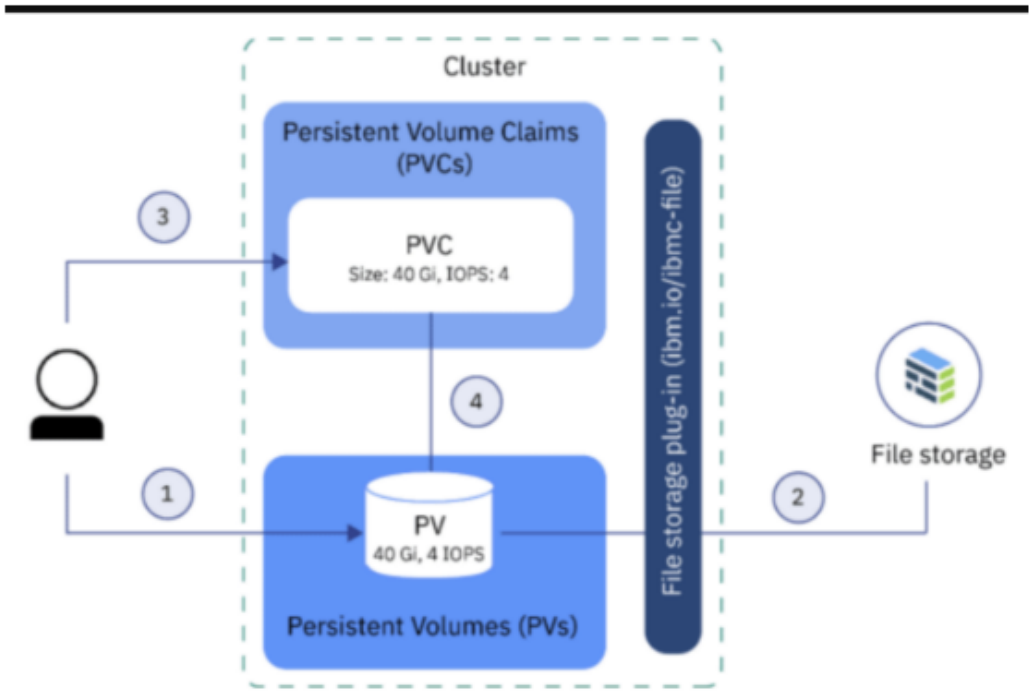
Provisioning and Reclaim

- Two ways PVs can provisioned:
  - Static
  - Dynamic
- Reclaim types
  - Retain
  - Delete

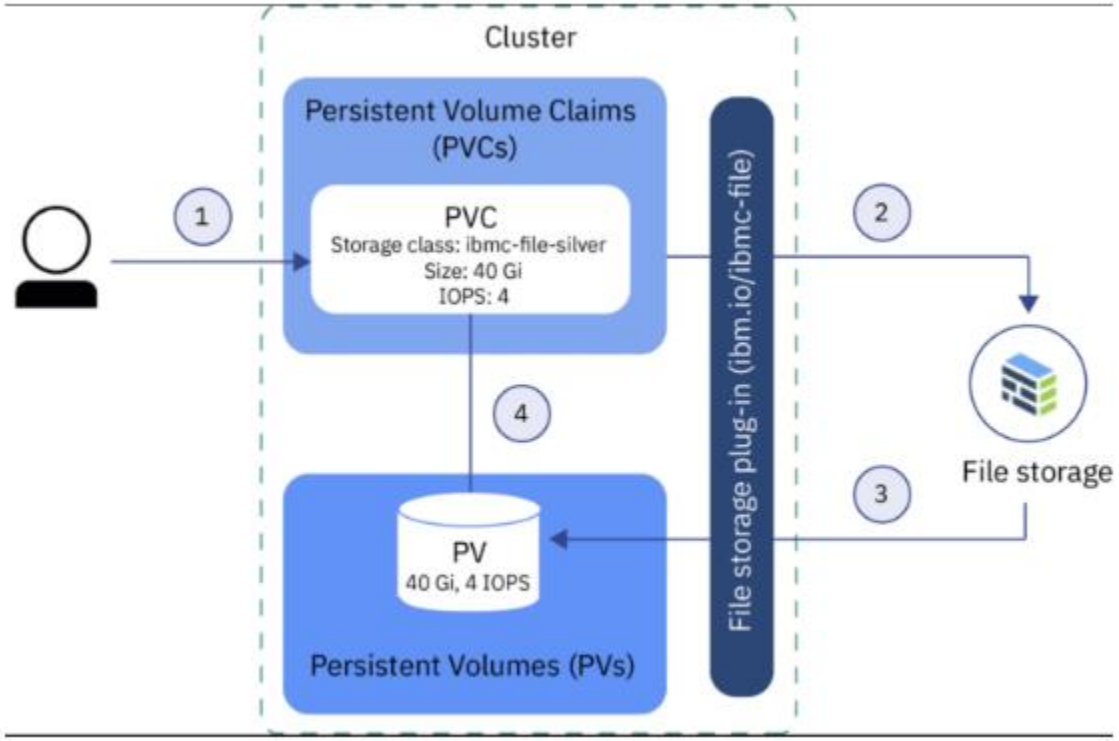
..


Cluster
App
Persistent Volume Claim (PVC)
Persistent Volume (PV)
Physical storage

- Static provisioning



Dynamic Provisioning



Pv yaml example

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

Yaml explain

- Capacity: Generally, a PV will have a specific storage capacity. This is set using the PV's capacity attribute.
- VolumeMode:Kubernetes supports two volumeModes of PersistentVolumes: Filesystem and Block.
- volumeMode is an optional API parameter. Filesystem is the default mode used when volumeMode parameter is omitted.

Access Mode

- ReadWriteOnce
  - the volume can be mounted as read-write by a single node. ReadWriteOnce access mode still can allow multiple pods to access the volume when the pods are running on the same node.
- ReadOnlyMany
  - the volume can be mounted as read-only by many nodes.
- ReadWriteMany
  - the volume can be mounted as read-write by many nodes.
- ReadWriteOncePod
  - the volume can be mounted as read-write by a single Pod.

PVC

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

Map to pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: task-pv-pod
spec:
  volumes:
    - name: task-pv-storage
      persistentVolumeClaim:
        claimName: task-pv-claim
  containers:
    - name: task-pv-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: task-pv-storage
```

StorageClass

- A StorageClass provides a way for administrators to describe the "classes" of storage they offer. Different classes might map to quality-of-service levels, or to backup policies, or to arbitrary policies determined by the cluster administrators.
- Each StorageClass contains the fields provisioner, parameters, and reclaimPolicy, which are used when a PersistentVolume belonging to the class needs to be dynamically provisioned.

Dynamic Provisioning

- With dynamic provisioning, these two steps are automated, eliminating the need for cluster administrators to pre-provision storage.
- Instead, the storage resources can be dynamically provisioned using the provisioner specified by the StorageClass object (see user-guide).
- StorageClasses are essentially blueprints that abstract away the underlying storage provider, as well as other parameters, like disk-type (e.g.; solid-state vs standard disks)

• •

Cloud Provider	Default StorageClass Name	Default Provisioner
Amazon Web Services	gp2	aws-ebs
Microsoft Azure	standard	azure-disk
Google Cloud Platform	standard	gce-pd
OpenStack	standard	cinder
VMware vSphere	thin	vsphere-volume

yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard
provisioner: kubernetes.io/aws-ebs
parameters:
  type: gp2
reclaimPolicy: Retain
allowVolumeExpansion: true
mountOptions:
  - debug
volumeBindingMode: Immediate
```