

PersistentVolume (PV) and PersistentVolumeClaim (PVC)

Persistent Volumes (PVs) and PersistentVolumeClaims (PVCs) are fundamental concepts in Kubernetes for managing persistent storage. They provide a way to decouple storage from Pods, making applications more resilient and portable.

PersistentVolume (PV)

A PersistentVolume represents a piece of storage in the cluster. It's a cluster-level object that abstracts the underlying storage infrastructure. Think of it as a provisioned storage resource.

It represents: A PV represents a physical storage resource, such as:

- A disk on a Node.
- A volume in a cloud provider (e.g., AWS EBS, Google Persistent Disk, Azure Disk).
- A network file system (e.g., NFS, GlusterFS).

Key Characteristics:

Storage Capacity: The amount of storage available on the PV.

Access Modes: How the PV can be accessed (e.g., ReadWriteOnce, ReadOnlyMany, ReadWriteMany). These modes define how many Pods can access the volume simultaneously and whether they can read and write to it.

Storage Class: The storage class associated with the PV (if any). Storage classes define different tiers of storage with varying performance and cost characteristics.

Capacity: The amount of storage available.

Reclaim Policy: What happens to the PV when it's no longer bound to a PVC (e.g., Delete, Recycle, Retain).

PVs can be created in two ways:

Static Provisioning: Manually creating PVs with specific configurations. This is suitable for environments where storage is pre-provisioned.

Dynamic Provisioning: Using a StorageClass to automatically provision PVs when a PVC is created. This is the preferred method in most Kubernetes deployments.

PersistentVolumeClaim (PVC)

A PersistentVolumeClaim is a request for storage by a user. It's a namespace-scoped object that defines the requirements for a volume. Think of it as a request for storage from a Pod.

It represents: A PVC represents a request for a specific amount of storage with certain access modes. It doesn't guarantee that storage will be immediately available, but it expresses the application's storage needs.

Key Characteristics:

Requested Storage Size: The amount of storage requested by the PVC.

Access Modes: The access modes required by the application (e.g., ReadWriteOnce, ReadOnlyMany, ReadWriteMany).

Storage Class: The storage class the PVC prefers to use.

Selector: A selector that can be used to match PVs with specific labels.

PVCs are created by users or applications. When a PVC is created, Kubernetes attempts to bind it to a suitable PV.

Binding: Kubernetes automatically binds a PVC to a PV that meets the PVC's requirements. If no suitable PV is available, the PVC will remain in a `Pending` state.