**KUBERNETES VOLUMES**

* On-disk files in a container are ephemeral, problem is the loss of files when a container crashes. A second problem occurs when sharing files between containers running together in a Pod.
* A volume can be thought of as a directory that is accessible to the containers in a pod. As soon as the life of a pod ended, the volume was also lost.
* A key advantage of Kubernetes volume is, it supports different kinds of storage wherein the pod can use multiple of them at the same time.

### Examples of the Volume Types:

### emptyDir: An emptyDir volume is first created when a Pod is assigned to a Node and exists as long as that Pod is running on that node. As the name says, it is initially empty. When a Pod is removed from a node for any reason, the data in the emptyDir is deleted forever.

### ****hostPath:**** A hostPath volume mounts a file or directory from the host node's filesystem into your Pod.

* **awsElasticBlockStore Cloud volumes:** An awsElasticBlockStore volume mounts an Amazon Web Services (AWS) EBS Volume into your Pod.
* **nfs:** An nfs volume allows an existing NFS (Network File System) share to be mounted into a Pod. Unlike emptyDir, which is erased when a Pod is removed, the contents of an nfs volume are preserved and the volume is merely unmounted. This means that an NFS volume can be pre-populated with data, and that data can be shared between pods. NFS can be mounted by multiple writers simultaneously.
* **persistentVolumeClaim** A persistentVolumeClaim volume is used to mount a PersistentVolume into a Pod.
* A PersistentVolume (PV) is a piece of storage in the cluster just like a node is a cluster resource.
* A PersistentVolumeClaim (PVC) is a request for storage by a user.
* A PersistentVolume can be mounted on a host in any way supported by the resource provider.
* A PersistentVolume can be mounted on a host in any way supported by the resource provider.
* A PersistentVolume can be mounted on a host in any way supported by the resource provider.

**ReadWriteOnce:** Read-write by a single node

**ReadOnlyMany:** Read-only by many nodes

* **ReadWriteMany:** Read-write by many nodes
* **ReadWriteOncePod:** Read-write only one pod in the cluster