

Volumes



- A volume is just a directory, possibly with some data in it, which is accessible to the containers in a pod.
- The content is determined by the particular volume type used.

Volumes - types



- **Host-based**
 - EmptyDir
 - HostPath
- **Block Storage**
 - Amazon EBS
 - GCE Persistent Disk
 - Azure Disk
 - vSphere Volume
 - ...
- **Distributed File System**
 - NFS
 - Ceph
 - Gluster
 - Amazon EFS
 - Azure File System
 - ...
- **Other**
 - Flocker
 - iScsi
 - Git Repo
 - Quobyte
 - ...

Volumes



- Volumes bring persistence to Pods
- Kubernetes volumes are similar to Docker volumes, but managed differently
- All containers in a Pod can access the volume
- Volumes are associated with the lifecycle of Pod
- Directories in the host are exposed as volumes
- Volumes may be based on a variety of storage backends

Persistent volume



PersistentVolume (PV)

- Networked storage in the cluster pre-provisioned by an administrator.

PersistentVolumeClaim (PVC)

- Storage resource requested by a user.



Dynamic Volume Provisioning

Dynamic volume provisioning allows storage volumes to be created on-demand.

Cluster administrators have to manually make calls to their cloud or storage provider to create new storage volumes, and then create `PersistentVolume` objects to represent them in Kubernetes.

Dynamic Volume Provisioning



The implementation of dynamic volume provisioning is based on the API object **StorageClass**

A cluster administrator can define as many StorageClass objects as needed, each specifying a volume plugin.

Dynamic Volume Provisioning

To enable dynamic provisioning, a cluster administrator needs to pre-create one or more `StorageClass` objects for users.

Users request dynamically provisioned storage by including a storage class in their `PersistentVolumeClaim`.

Dynamic Volume Provisioning



```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: local-fast
provisioner: kubernetes.io/host-path
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nginx-claim
spec:
  storageClassName: local-fast
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

```
apiVersion: v1
kind: Pod
metadata:
  name: mysamplepod
spec:
  containers:
    - name: frontend
      image: nginx
      volumeMounts:
        - mountPath: "/var/www/html"
          name: myvolume
  volumes:
    - name: myvolume
      persistentVolumeClaim:
        claimName: nginx-claim
```