AccessMode: It defines how volume is mounted on host

ReadWriteOnce – the volume can be mounted as read-write by a single node

ReadOnlyMany – the volume can be mounted read-only by many nodes

ReadWriteMany – the volume can be mounted as read-write by many nodes

In the CLI, the access modes are abbreviated to:

RWO - ReadWriteOnce

ROX - ReadOnlyMany

RWX - ReadWriteMany

Current recycling policies are:

Retain – manual reclamation (Only pod deleted. Pv and data will be available)

Recycle – basic scrub (“rm -rf /thevolume/\*”) (pod and data deleted. Pv available)

Delete – associated storage asset such as AWS EBS, GCE PD or OpenStack Cinder volume is deleted

HostPath

apiVersion: v1

kind: Pod

metadata:

name: nvolpod1

spec:

containers:

- name: test

image: nginx

volumeMounts:

- mountPath: "/var/www/html"

name: mypd

volumes:

- name: mypd

hostPath:

path: /data

Link: <https://itnext.io/practical-example-of-using-k8s-pv-pvc-with-pods-5471b91d2477>

Pv

apiVersion: v1

kind: PersistentVolume

metadata:

name: nfsdata

spec:

capacity:

storage: 5Gi # Size of the nfs-server volume

volumeMode: Filesystem

accessModes:

- ReadWriteOnce

storageClassName: slow

nfs:

path: /nfsdata # path on which k8-n-4 exposes the nfs mount

server: 10.150.0.6 # nfs-server ip address

Pvc

apiVersion : v1

kind: PersistentVolumeClaim

metadata:

name: nfsclaim # Claim name which will be used by the pod

spec:

accessModes:

- ReadWriteOnce

volumeMode: Filesystem

resources:

requests:

storage: 1Gi #request size from claim to the PersistentVolume

storageClassName: slow

Pod

apiVersion: v1

kind: Pod

metadata:

name: nvolpod

spec:

containers:

- name: test

image: nginx

volumeMounts:

- mountPath: "/var/www/html" # Folder in pod mounting nfsclaim

name: mypd

volumes:

- name: mypd # volume name the container can mount

persistentVolumeClaim: # Source of storage

claimName: nfsclaim # name of pvc created

apiVersion: v1

kind: Pod

metadata:

name: nginx-server

spec:

volumes:

- name: shared-logs

emptyDir: {}

containers:

- name: nginx

image: nginx

volumeMounts:

- name: shared-logs

mountPath: /var/log/nginx

- name: sidecar-container

image: busybox

command: ["sh","-c","while true; do cat /var/log/nginx/access.log /var/log/nginx/error.log; sleep 30; done"]

volumeMounts:

- name: shared-logs

mountPath: /var/log/nginx

ETCD Backup & Restore

====================

Backup

ETCDCTL\_API=3 etcdctl snapshot save /tmp/etcd-backup.db --cacert /etc/kubernetes/pki/etcd/ca.crt --cert /etc/kubernetes/pki/etcd/server.crt --key /etc/kubernetes/pki/etcd/server.key

Restore

ETCDCTL\_API=3 etcdctl snapshot restore /tmp/etcd-backup.db --data-dir /var/lib/etcd-backup --cacert /etc/kubernetes/pki/etcd/ca.crt --cert /etc/kubernetes/pki/etcd/server.crt --key /etc/kubernetes/pki/etcd/server.key