persistent volume (PV) is the "physical" volume on the host machine that stores your persistent data.

A persistent volume claim (PVC) is a request for the platform to create a PV for you, and you attach PVs to your pods via a PVC.

For openshift convert this to json or retain yaml and use

1 create PV

2 create PVC

3 call PVC in the pods

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

#PV

apiVersion: v1

kind: PersistentVolume

metadata:

name: pv-volume

labels:

type: local

spec:

storageClassName: manual

capacity:

storage: 10Gi

accessModes:

- ReadWriteOnce

hostPath:

path: "/mnt/data"

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

#PVC

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pv-claim

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 2Gi

volumeName: pv0001

=######create deploy,service

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: jenkins

spec:

replicas: 1

selector:

matchLabels:

app: jenkins

template:

metadata:

labels:

app: jenkins

spec:

containers:

- name: jenkins

image: jenkins:2.60.3

ports:

- containerPort: 8080

volumeMounts:

- name: pv0001

mountPath: "/var/jenkins\_home"

volumes:

- name: pv0001

persistentVolumeClaim:

claimName: pv-claim

---

###create a service and then route it

apiVersion: v1

kind: Service

metadata:

name: jenkins

spec:

selector:

app: jenkins

ports:

- protocol: TCP

port: 80

targetPort: 8080

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

First when pv is created 10GB is allocated

when pvc.yaml is created then 2GB out of 10 get allocated and a separate pv,pvc gets created

[root@ip-10-100-50-122 pv]# kubectl get pv,pvc

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE

persistentvolume/pv-volume 10Gi RWO Retain Available manual 6m46s

persistentvolume/pvc-1d50d08e-2fe1-4c00-acb1-e24fcc357c3a 2Gi RWO Delete Bound default/pv-claim standard 4m4s

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

persistentvolumeclaim/pv-claim Bound pvc-1d50d08e-2fe1-4c00-acb1-e24fcc357c3a 2Gi RWO standard 4m4s

kubectl describe persistentvolumeclaim/pv-claim

Name: pv-claim

Namespace: default

StorageClass: standard

Status: Bound

Volume: pvc-1d50d08e-2fe1-4c00-acb1-e24fcc357c3a

Labels: <none>

Annotations: control-plane.alpha.kubernetes.io/leader:

{"holderIdentity":"085dbe07-7326-11ea-8449-022020e33f23","leaseDurationSeconds":15,"acquireTime":"2020-03-31T10:06:57Z","renewTime":"2020-...

pv.kubernetes.io/bind-completed: yes

pv.kubernetes.io/bound-by-controller: yes

volume.beta.kubernetes.io/storage-provisioner: k8s.io/minikube-hostpath

Finalizers: [kubernetes.io/pvc-protection]

Capacity: 2Gi

Access Modes: RWO

VolumeMode: Filesystem

Mounted By: jenkins-84d6755c9b-tw96h

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal ExternalProvisioning 5m10s (x3 over 5m10s) persistentvolume-controller waiting for a volume to be created, either by external provisioner "k8s.io/minikube-hostpath" or manually created by system administrator

Normal Provisioning 5m10s k8s.io/minikube-hostpath 085dbe07-7326-11ea-8449-022020e33f23 External provisioner is provisioning volume for claim "default/pv-claim"

Normal ProvisioningSucceeded 5m10s k8s.io/minikube-hostpath 085dbe07-7326-11ea-8449-022020e33f23 Successfully provisioned volume pvc-1d50d08e-2fe1-4c00-acb1-e24fcc357c3a

kubectl describe pv/pvc-1d50d08e-2fe1-4c00-acb1-e24fcc357c3a

Name: pvc-1d50d08e-2fe1-4c00-acb1-e24fcc357c3a

Labels: <none>

Annotations: hostPathProvisionerIdentity: 085dbd90-7326-11ea-8449-022020e33f23

pv.kubernetes.io/provisioned-by: k8s.io/minikube-hostpath

Finalizers: [kubernetes.io/pv-protection]

StorageClass: standard

Status: Bound

Claim: default/pv-claim

Reclaim Policy: Delete

Access Modes: RWO

VolumeMode: Filesystem

Capacity: 2Gi

Node Affinity: <none>

Message:

Source:

Type: HostPath (bare host directory volume)

Path: /tmp/hostpath-provisioner/pvc-1d50d08e-2fe1-4c00-acb1-e24fcc357c3a ####this is where Jenkins is mounted

HostPathType:

Events: <none>