Kubernets PVC1

Yml file for PV

[root@k-master yamls]# vim 4-pv-hostpath.yaml

apiVersion: v1

kind: PersistentVolume

metadata:

name: pv-hostpath

labels:

type: local

spec:

storageClassName: manual

capacity:

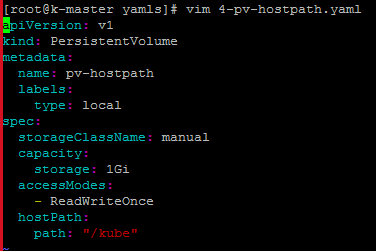
storage: 1Gi

accessModes:

- ReadWriteOnce

hostPath:

path: "/kube"



Create dir on worker node

[root@k-worker ~]# mkdir /kube

[root@k-worker ~]# chmod 777 /kube

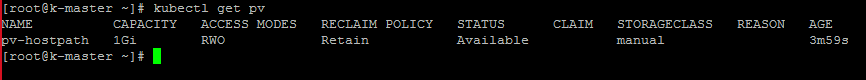
[root@k-worker ~]#

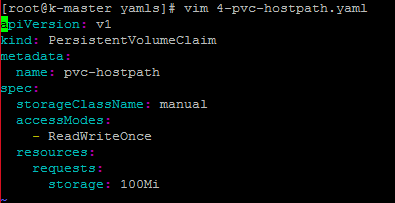
Create PV

kubectl create -f 4-pv-hostpath.yaml



kubectl get pv





[root@k-master yamls]# vim 4-pvc-hostpath.yaml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pvc-hostpath

spec:

storageClassName: manual

accessModes:

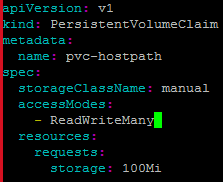
- ReadWriteOnce

resources:

requests:

storage: 100Mi

Before create pvc, we can change accessModes to ReadWriteMany,which will not work because in PV, we have only ReadWriteOnce

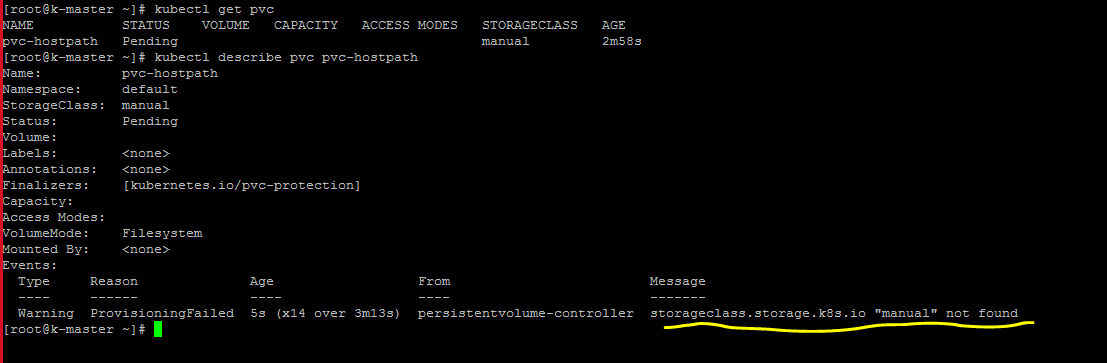


[root@k-master yamls]# kubectl create -f 4-pvc-hostpath.yaml

persistentvolumeclaim/pvc-hostpath created

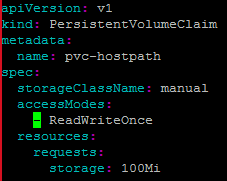
kubectl get pvc

kubectl describe pvc pvc-hostpath



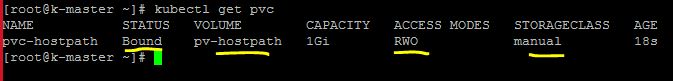
(as we don’t have access mode ReadWriteMany in PV, so in PVC, we cant claim for ReadWriteMany)

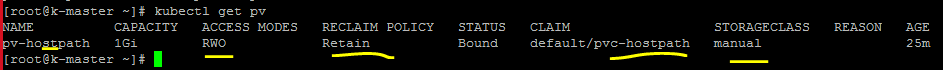
So we have to delete old pvc and change the access mode to ReadWriteOnly)



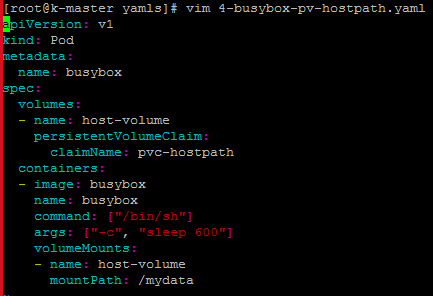
kubectl create -f 4-pvc-hostpath.yaml







Lets create pod



[root@k-master yamls]# vim 4-busybox-pv-hostpath.yaml

apiVersion: v1

kind: Pod

metadata:

name: busybox

spec:

volumes:

- name: host-volume

persistentVolumeClaim:

claimName: pvc-hostpath

containers:

- image: busybox

name: busybox

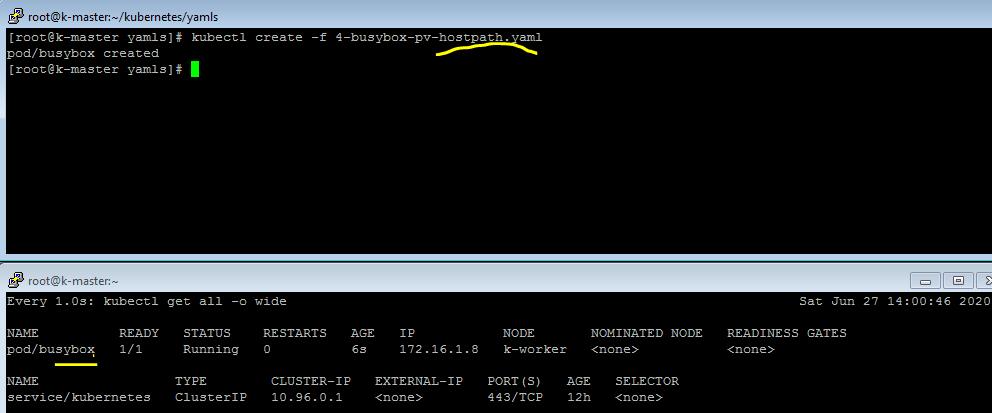
command: ["/bin/sh"]

args: ["-c", "sleep 600"]

volumeMounts:

- name: host-volume

mountPath: /mydata

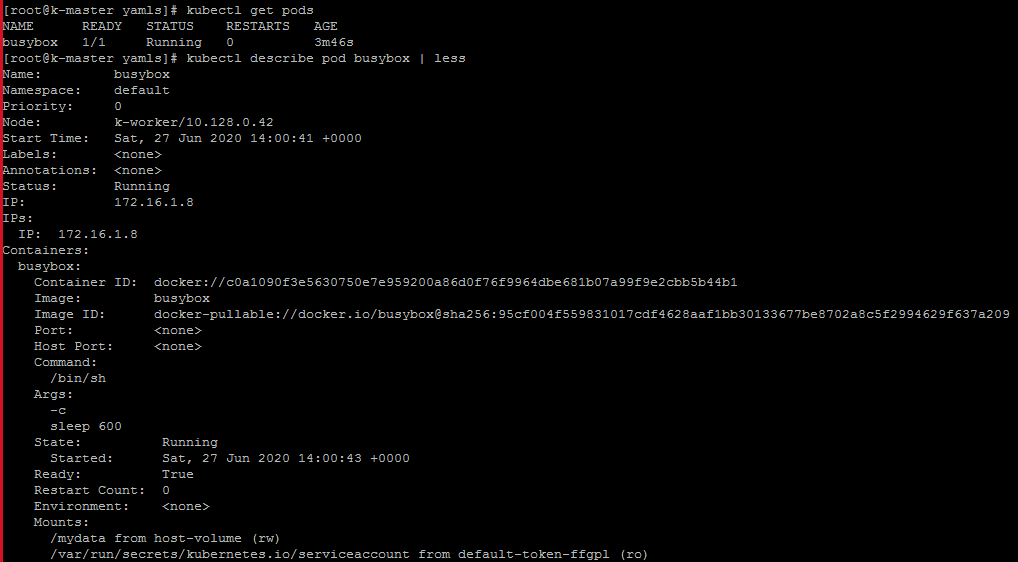


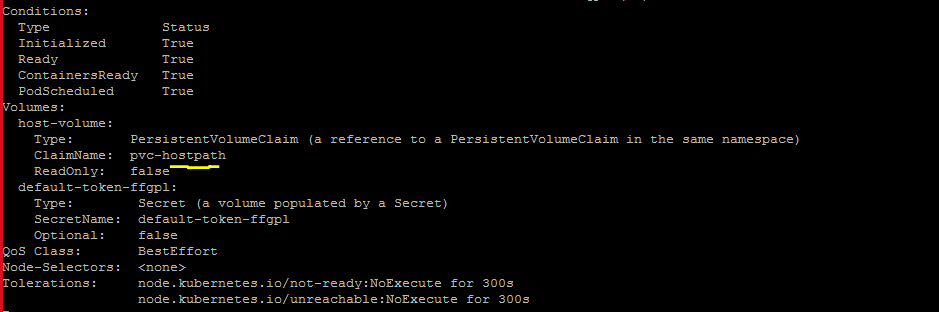
[root@k-master yamls]# kubectl get pods

NAME READY STATUS RESTARTS AGE

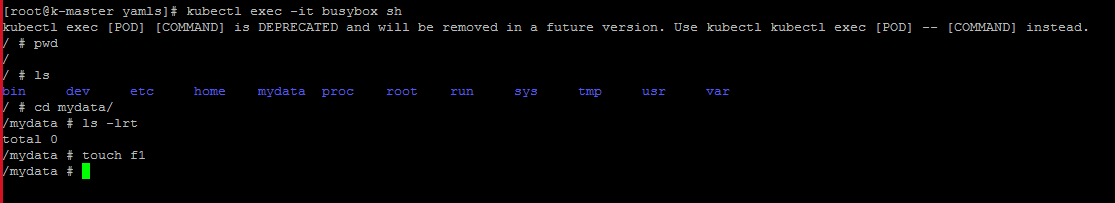
busybox 1/1 Running 0 3m46s

[root@k-master yamls]# kubectl describe pod busybox | less

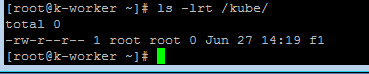




kubectl exec -it busybox sh



Lets check in /kube dir in worker node



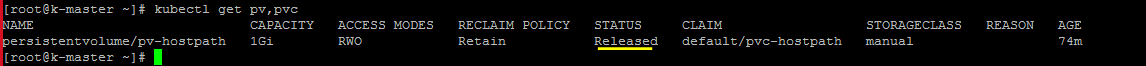
Lets delete pod

kubectl delete pod busybox

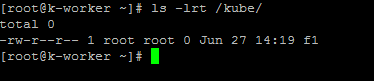
Lets remove pvc

kubectl delete pvc pvc-hostpath

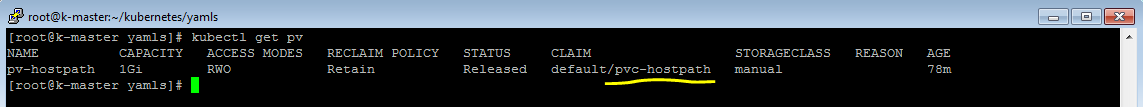
After removing pvc, you can see status has been changed to released in PV



After removing the pvc, and pod, still our data is remain exist.

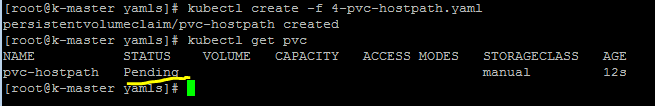


Here, you can see here PVC still attach with PV

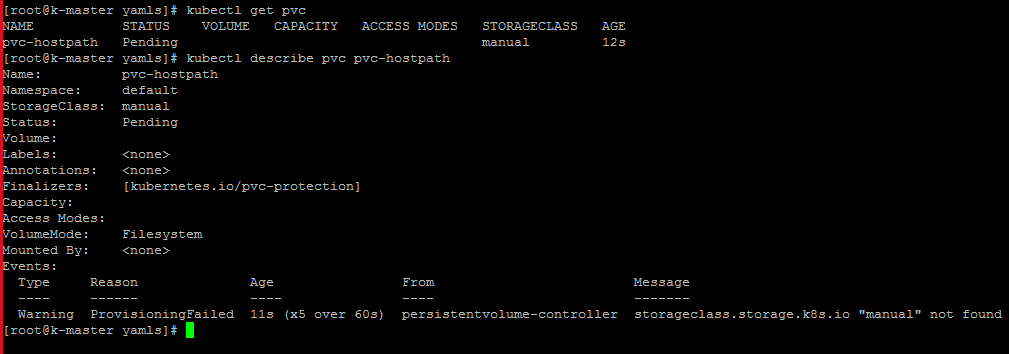


So other PVC cant request to attach with same PV

kubectl create -f 4-pvc-hostpath.yaml



kubectl describe pvc pvc-hostpath



Lets delete pv and pvc here

Then create pv again

kubectl delete pv pv-hostpath

kubectl delete pvc pvc-hostpath





<http://centosquestions.com/kubernetes-delete-label/>

[root@k-master yamls]# kubectl label node k-worker1 demoserver=true

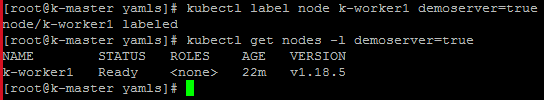
node/k-worker1 labeled

[root@k-master yamls]# kubectl get nodes -l demoserver=true

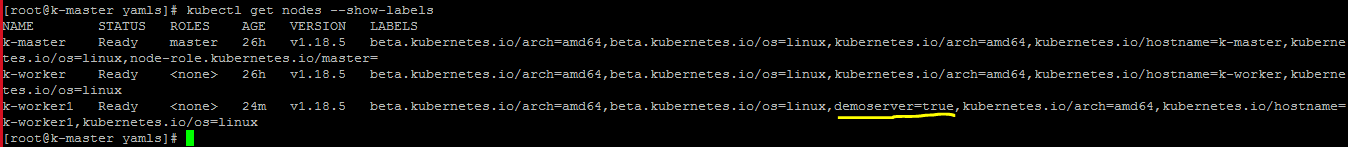
NAME STATUS ROLES AGE VERSION

k-worker1 Ready <none> 22m v1.18.5

[root@k-master yamls]#



kubectl get nodes --show-labels



Lets create pv, then pvc

[root@k-master yamls]# kubectl create -f 4-pv-hostpath.yaml

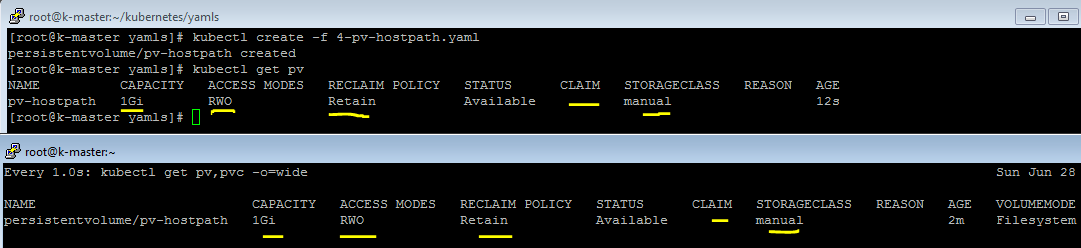
persistentvolume/pv-hostpath created

[root@k-master yamls]# kubectl get pv

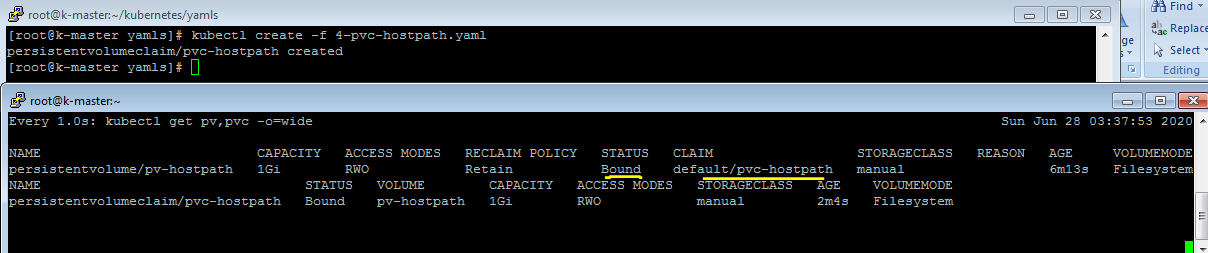
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE

pv-hostpath 1Gi RWO Retain Available manual 12s

[root@k-master yamls]#

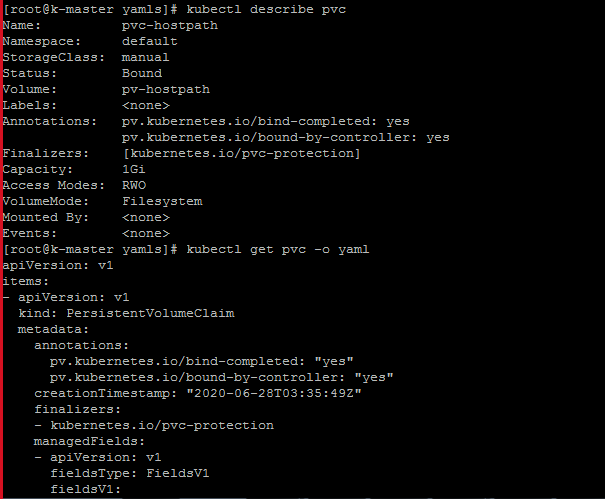
Now lets create pvc

After creating pvc, available status got changed to “Bound” and claim status also got changed as well.



kubectl describe pvc

kubectl get pvc -o yaml



Going to create pod in which I have add nodeselector to match label

vim 4-busybox-pv-hostpath.yaml

[root@k-master yamls]# vim 4-busybox-pv-hostpath.yaml

apiVersion: v1

kind: Pod

metadata:

name: busybox

spec:

volumes:

- name: host-volume

persistentVolumeClaim:

claimName: pvc-hostpath

containers:

- image: busybox

name: busybox

command: ["/bin/sh"]

args: ["-c", "sleep 600"]

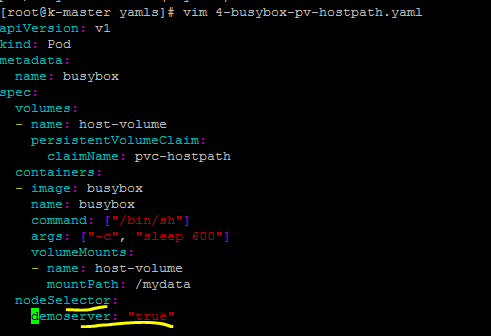
volumeMounts:

- name: host-volume

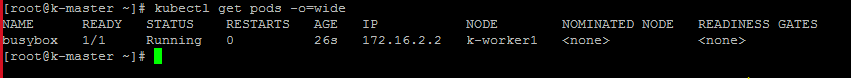
mountPath: /mydata

nodeSelector:

demoserver: "true"



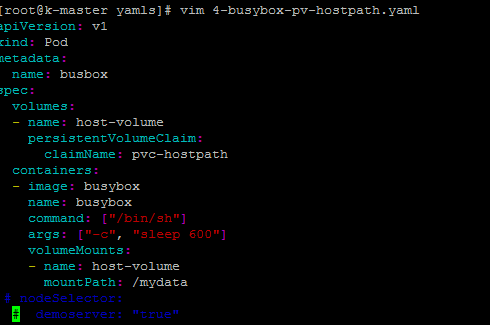
kubectl create -f 4-busybox-pv-hostpath.yaml



As we have specified to create pod which node label matches

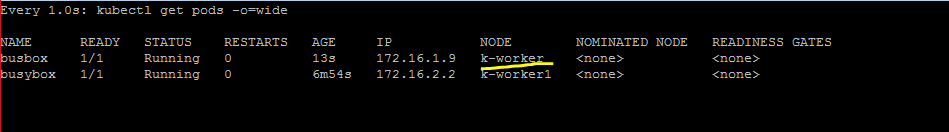
So it created under k-worker1

Lets comment nodeselector to create pod



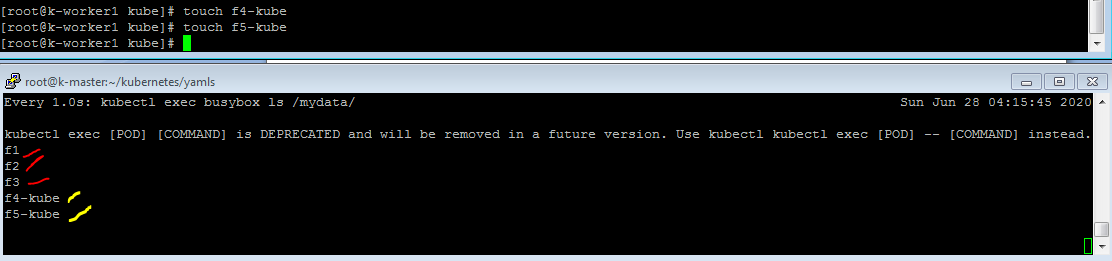
Lets create pod with different name which is busbox, let see on which it is creating

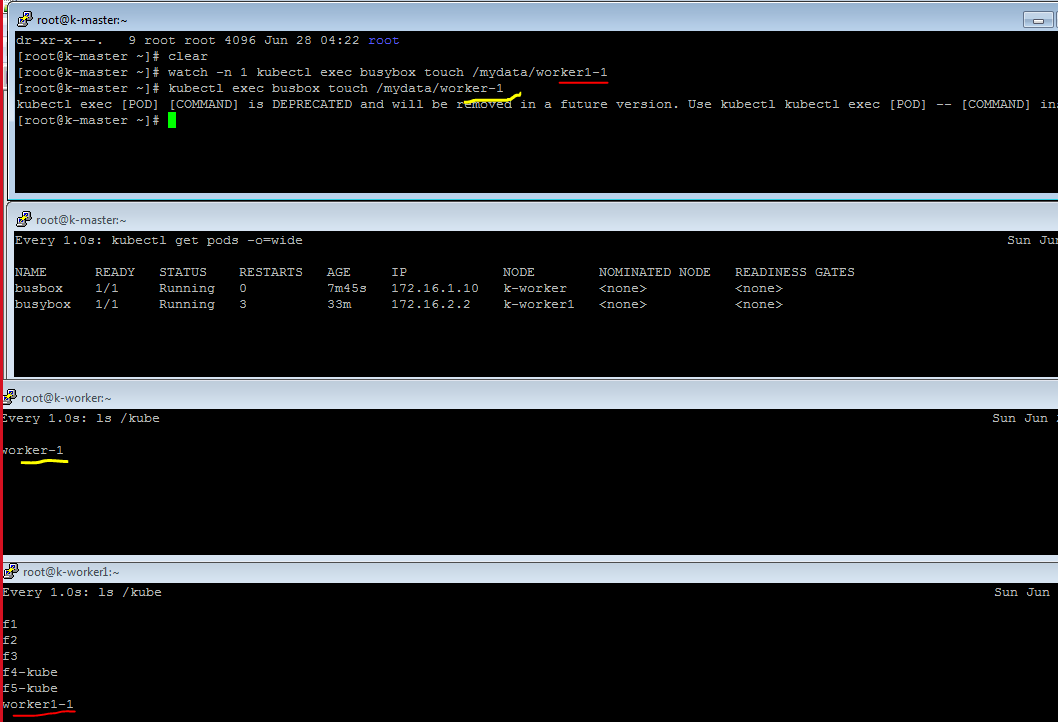
As we have removed node selector,it can create on any node like it create on k-worker node



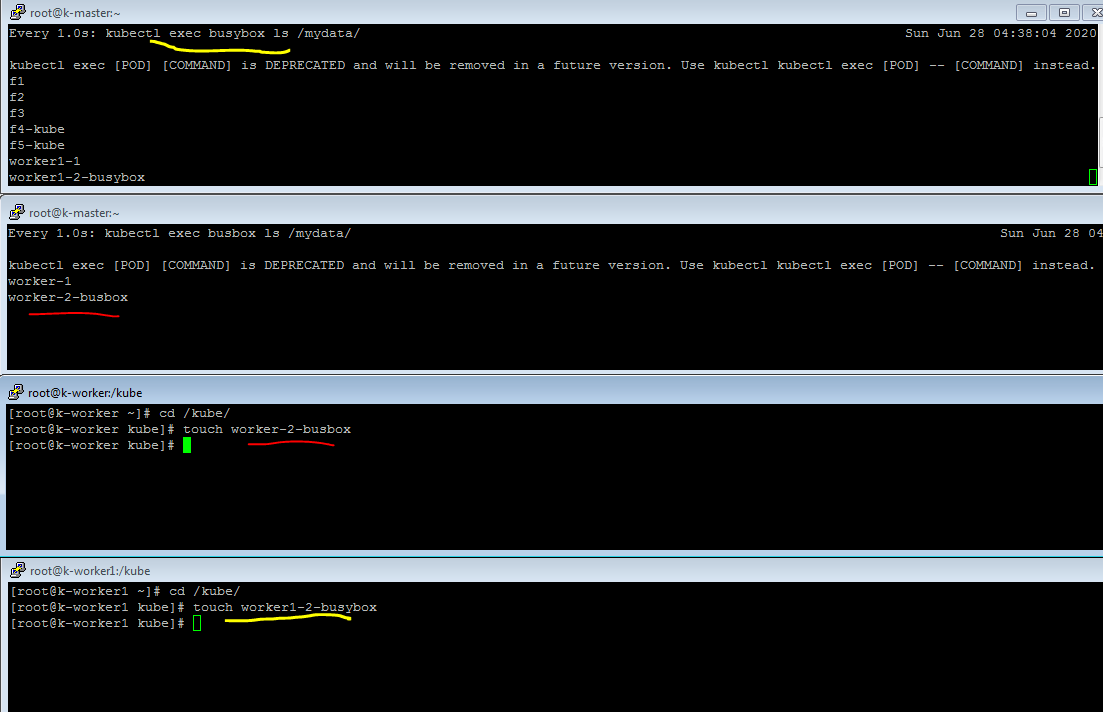
Here, we create f1,f2,f3 in container which are able to see in /kube in worker1 node

Similarly we create f4-kube,f5-kube in worker1 node, which we are able to see in container



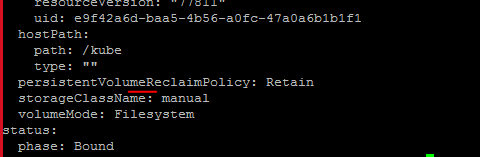


Now we are creating files under /kube which is local dir, same file would show in container

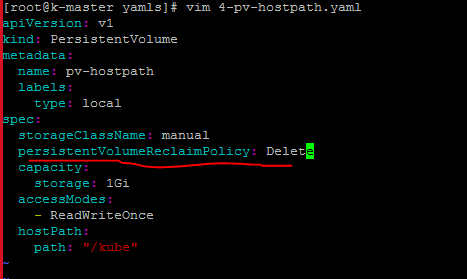


Lets use persistentVolumeReclaimPolicy: Delete

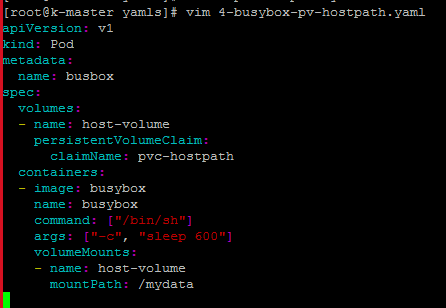
kubectl get pv pv-hostpath -o yaml



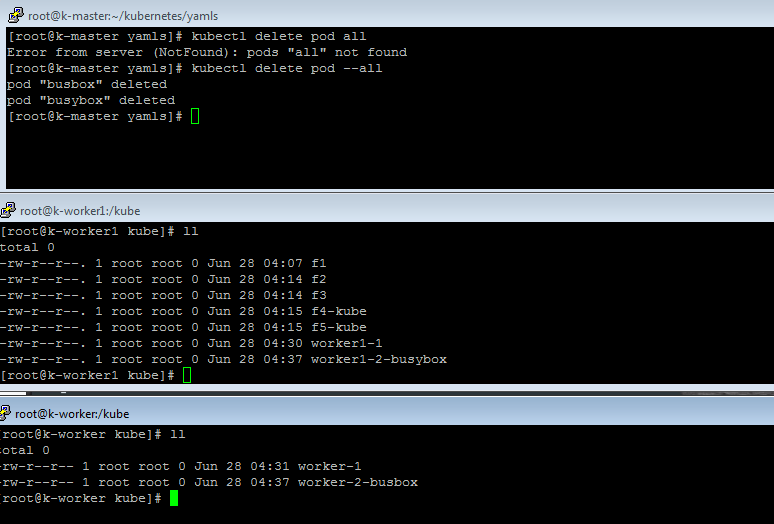
Changed in PV



Lets remove node selecto from pod yaml

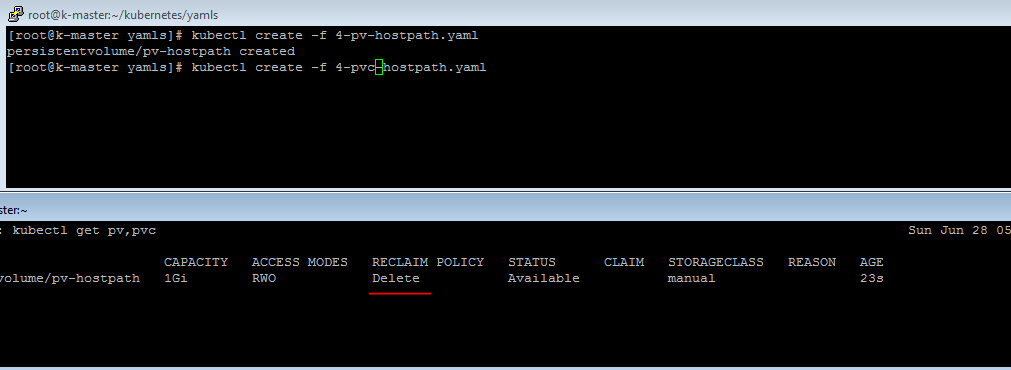


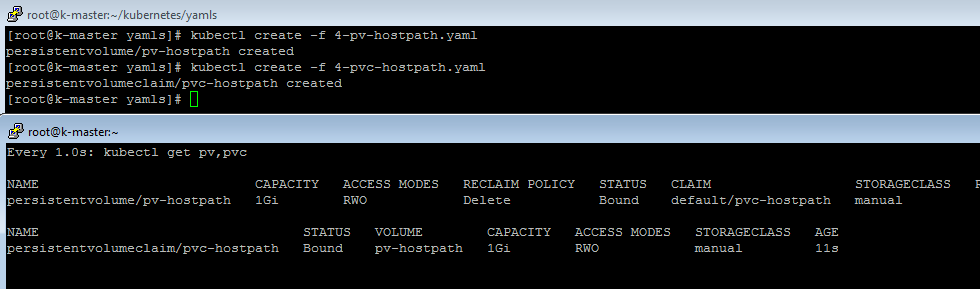
We removed existing pod, still data is available because reclampolicy was retain

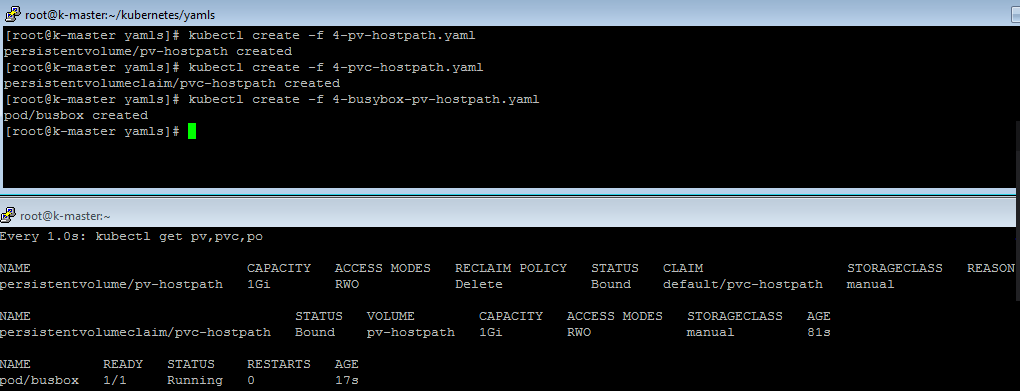


Lets remove existing pv and pvc

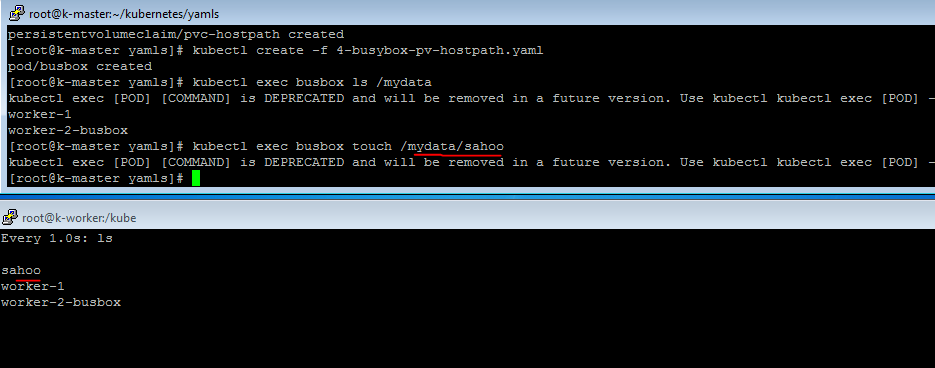
Here, we create pv,pvc again





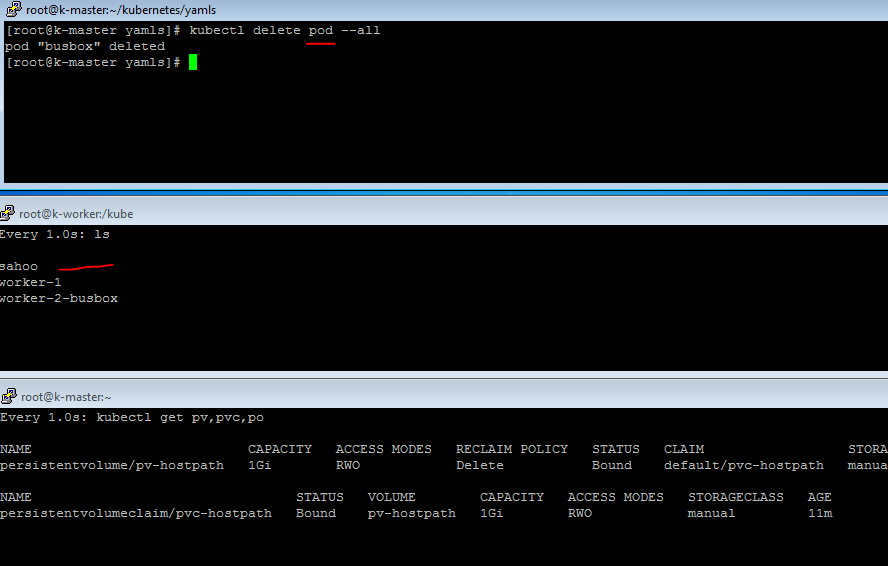


Create new file in container



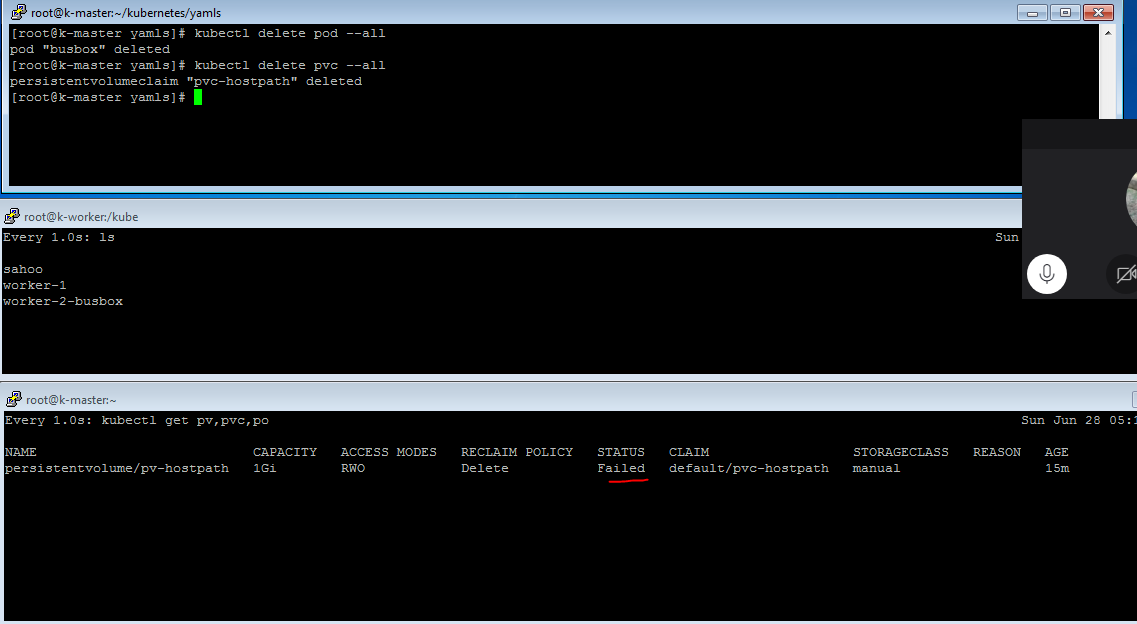
Use case 1 :-

Delete pod, still data is there

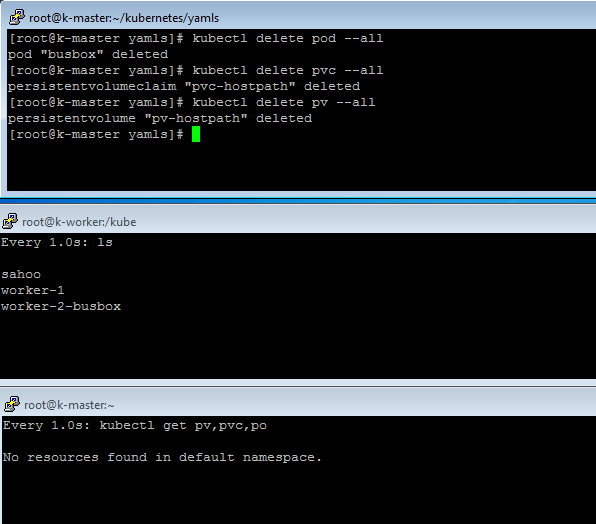


Use case 2

Removed PVC, stil; data is available



Use case 4



1. NFS pending

2. Read Write Many

3. Claimpolicy to delete, data is not getting delete even after removing pvc,pv and pod