Here is the configuration file for the hostPath PersistentVolume:

vim pv.yaml

kind: PersistentVolume

apiVersion: v1

metadata:

name: my-pv

labels:

type: local

spec:

storageClassName: manual

capacity:

storage: 5Gi

accessModes:

- ReadWriteOnce

hostPath:

path: "/tmp/data"

kubectl create -f pv.yaml

kubectl get pv my-pv



Create a PersistentVolumeClaim

The next step is to create a PersistentVolumeClaim. Pods use PersistentVolumeClaims to request physical storage. In this exercise, you create a PersistentVolumeClaim that requests a volume of at least three gibibytes that can provide read-write access for at least one Node.

Here is the configuration file for the PersistentVolumeClaim

vim pvc.yaml

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

name: my-pv-claim

spec:

storageClassName: manual

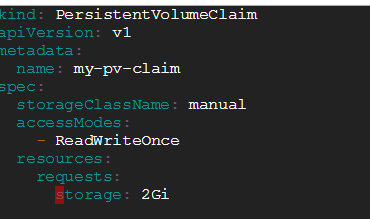
accessModes:

- ReadWriteOnce

resources:

requests:

storage: 2Gi



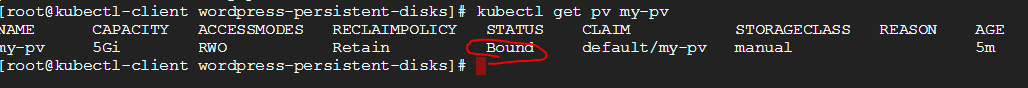
kubectl create -f pvc.yaml



kubectl get pvc my-pv-claim



kubectl get pv my-pv



vim podvolume.yaml

kind: Pod

apiVersion: v1

metadata:

name: ngix-pod-with-volume1

spec:

volumes:

- name: my-pv

persistentVolumeClaim:

claimName: my-pv-claim

containers:

- name: task-pv-container

image: nginx

ports:

- containerPort: 80

name: "http-server"

volumeMounts:

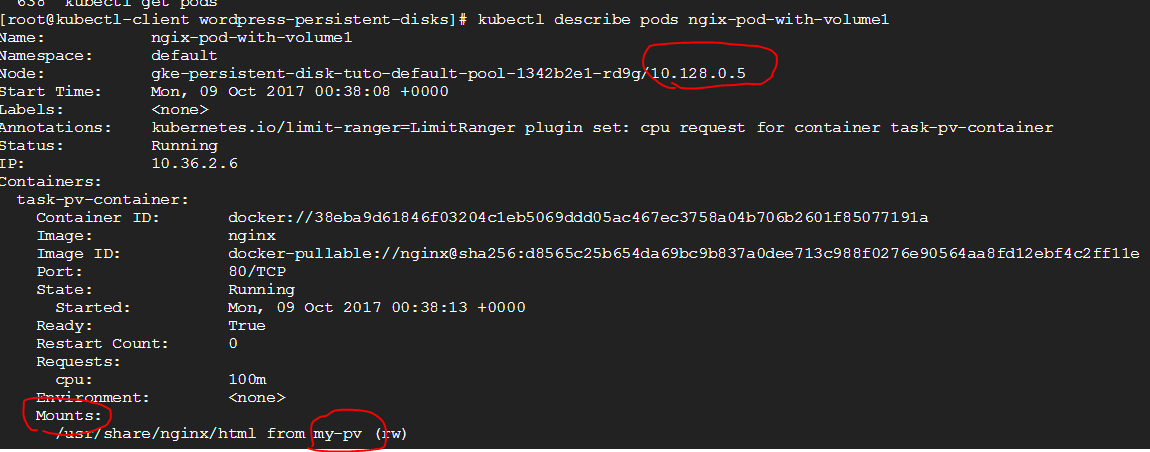
- mountPath: "/usr/share/nginx/html"

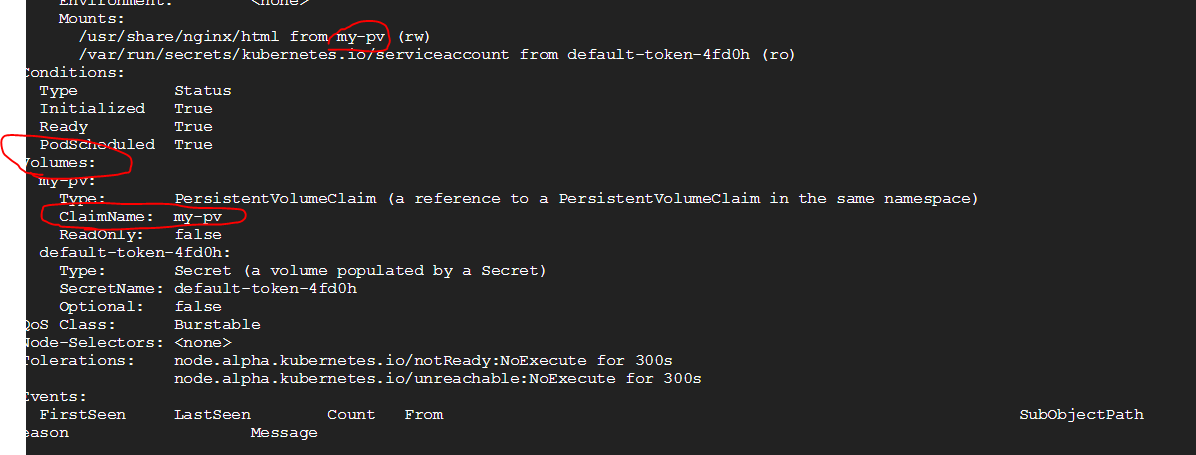
name: my-pv

kubectl create -f podvolume.yaml

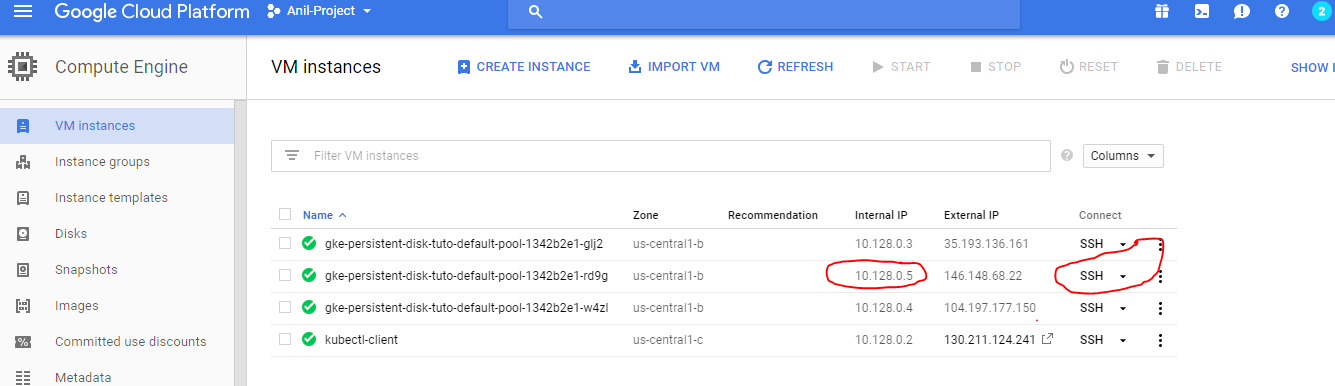
kubectl get pods

kubectl describe pods ngix-pod-with-volume1





Ssh to the node where pod is deployed



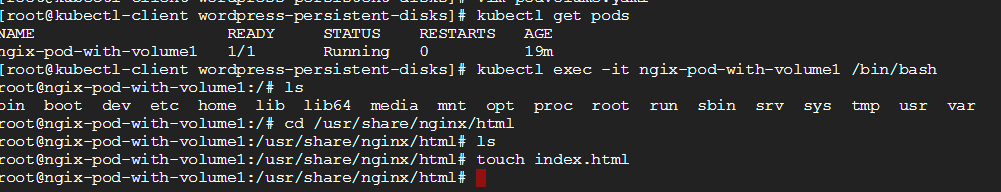
Verify if you directory exist

ls -l /tmp/data/

go back to your master node login and create index.html file

kubectl exec -it ngix-pod-with-volume1 /bin/bash

touch /usr/share/nginx/html/index.html



Login to you minnion node and verify if file exit on host

ls /tmp/data

