Kubernets Persistent Volume

NFS installation:-

yum install nfs\* -y

mkdir /read

mkdir /write



[root@k-master ~]# mkdir /read

[root@k-master ~]# mkdir /write

**Create file inside read dir**

[root@k-master ~]# cd /read/

[root@k-master read]# touch a{1.2.3.4.5}

[root@k-master read]# cd ..

[root@k-master /]# cd ~

**Create files inside write dir**

[root@k-master ~]# cd /write

[root@k-master write]# touch a1,a2,a3,a4,a5

**Give permission 777 to read ad write dir**

[root@k-master write]# chmod 777 /read

[root@k-master write]# ls -l

total 0

-rw-r--r--. 1 root root 0 Jun 27 01:57 a1,a2,a3,a4,a5

[root@k-master write]# chmod 777 /write

[root@k-master write]#

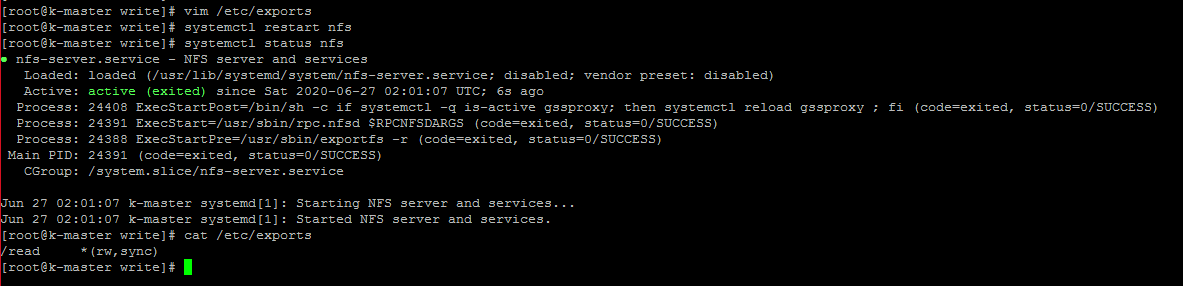
**Shared read dir**

[root@k-master write]# cat /etc/exports

/read \*(rw,sync)

**Start NFS service**

systemctl restart nfs



systemctl enable nfs



Monitoring files

exportfs –v



Now install nfs-utils on worker

yum install nfs-util

showmount -e k-master



mkdir /readonly

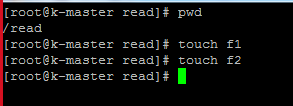
mount k-master:/read /readonly

mount

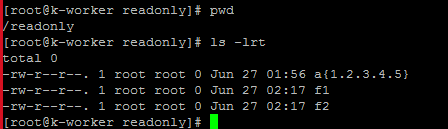
df –HTP

cd /readonly

Go to master and create files in /readdir



Lets see in worker to confirm

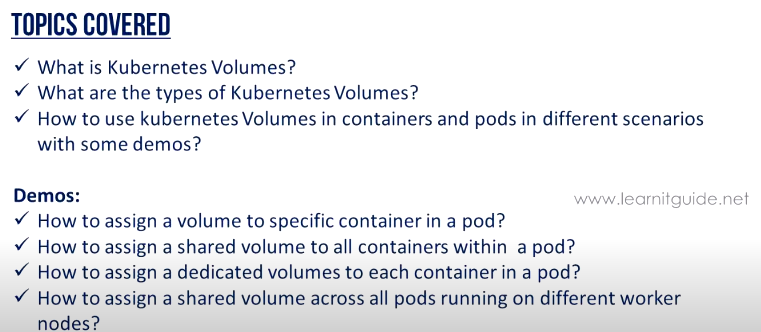


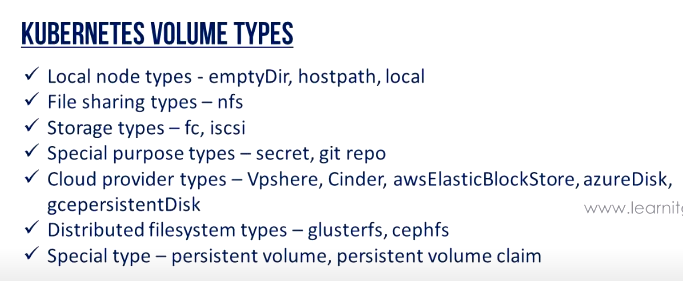
Reference <URL:->

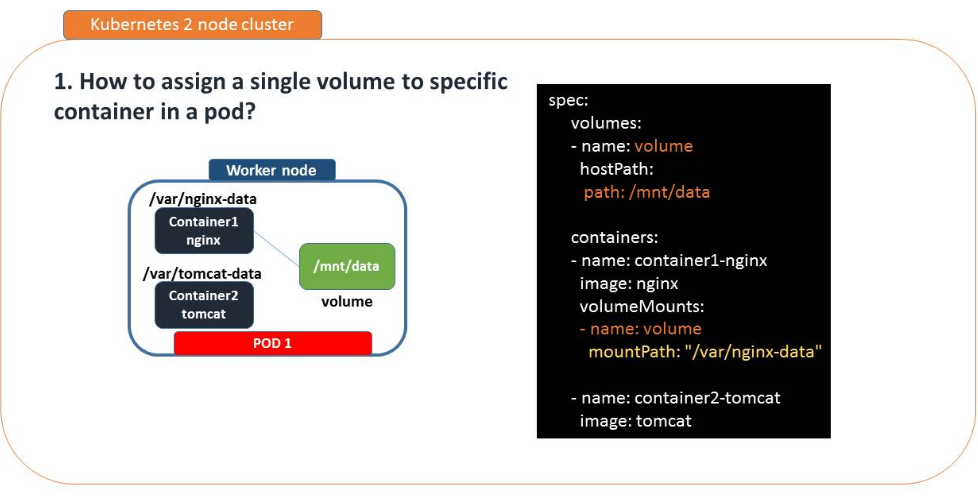
<https://www.youtube.com/watch?v=AXi2oENUJHo&list=PLYliwzTugkt3_8H2JPTUfzQ6NpsEzF41K&index=7>

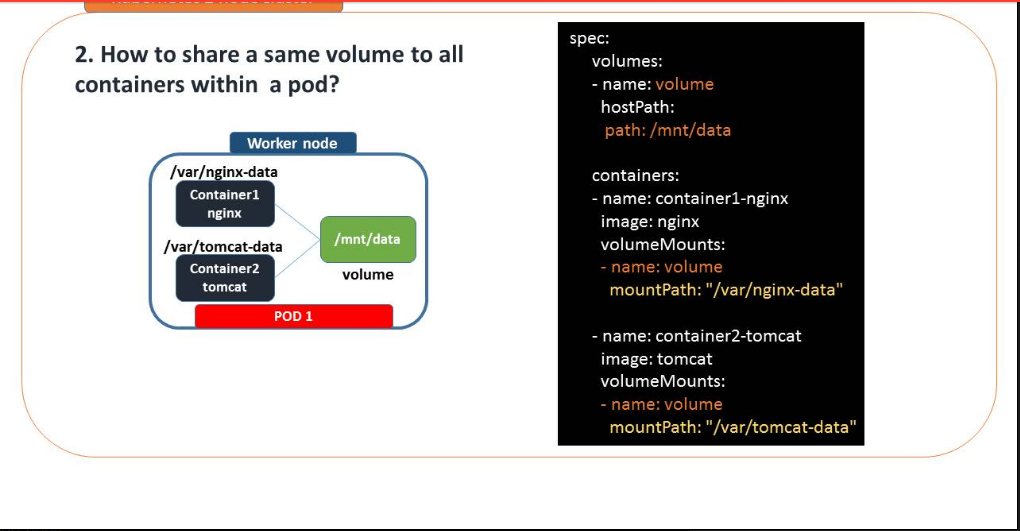
<https://www.learnitguide.net/2020/03/kubernetes-volumes-explained.html>

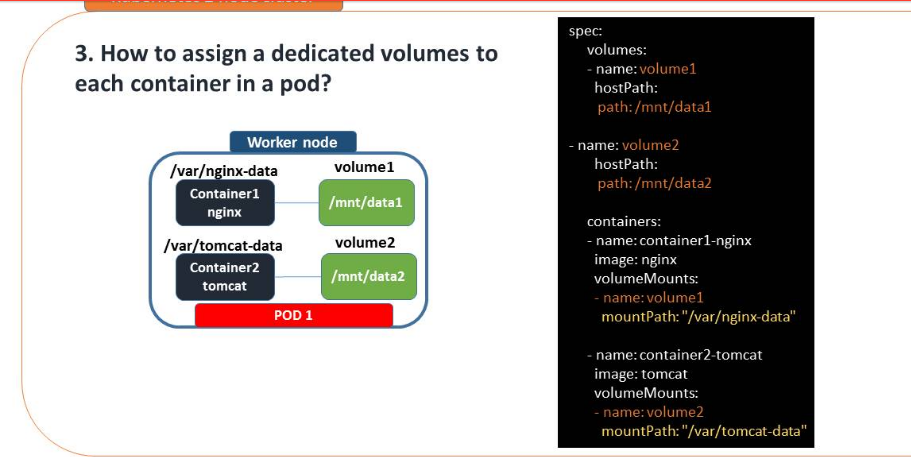


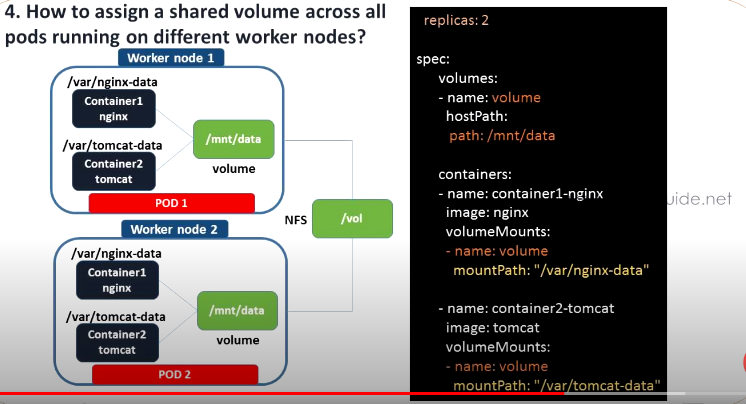












In order to achieve we takes /mnt/data dir to mount with nfs in both worker and master node

**Perform on master node**

mkdir /vol

chmod -R 777 /vol/

[root@k-master ~]# vim /etc/exports

/read \*(rw,sync)

/vol \*(rw,sync)



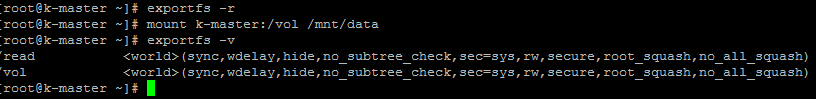
Then refresh nfs

exportfs -r

mkdir /mnt/data

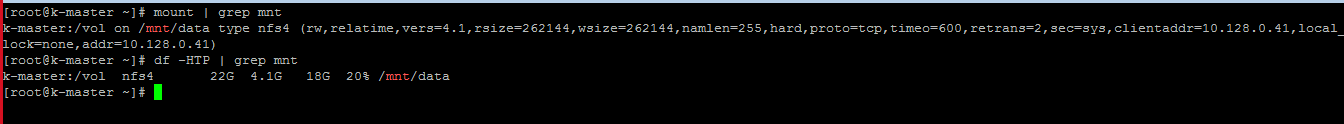
mount k-master:/vol /mnt/data

exportfs –v



mount | grep mnt

df -HTP | grep mnt



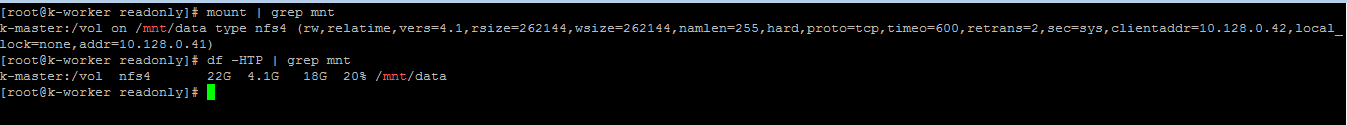
**Perform on worker node**

mkdir /mnt/data

mount k-master:/vol /mnt/data

mount | grep mnt

df -HTP | grep mnt



**Yml file to share nfs volume**

[root@k-master ~]# vim http-nfs-basic-deployment-use4.yaml

kind: Deployment

apiVersion: apps/v1

metadata:

name: ebay-app

spec:

selector:

matchLabels:

environment: dev

app: ebay

replicas: 1

template:

metadata:

labels:

environment: dev

app: ebay

spec:

volumes:

- name: volume

hostPath:

path: /mnt/data

containers:

- name: container1-nginx

image: nginx

volumeMounts:

- name: volume

mountPath: "/var/nginx-data"

- name: container2-tomcat

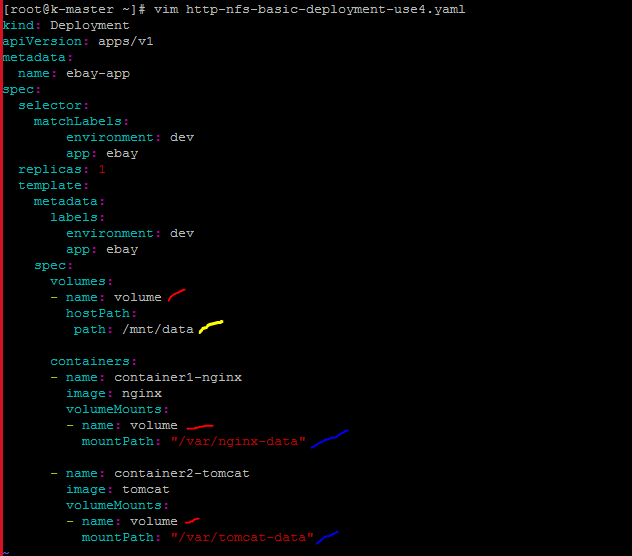
image: tomcat

volumeMounts:

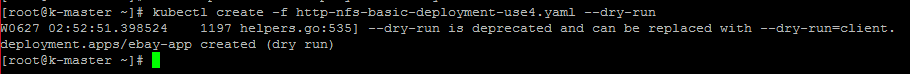
- name: volume

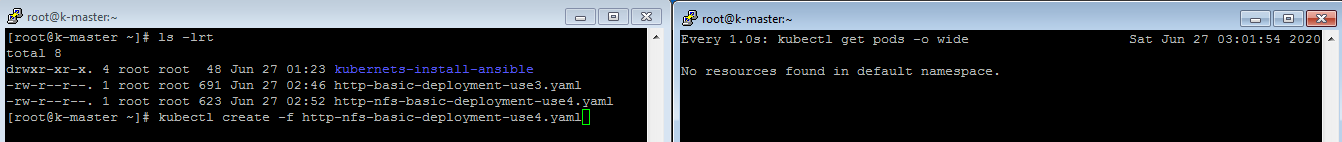
mountPath: "/var/tomcat-data"

~



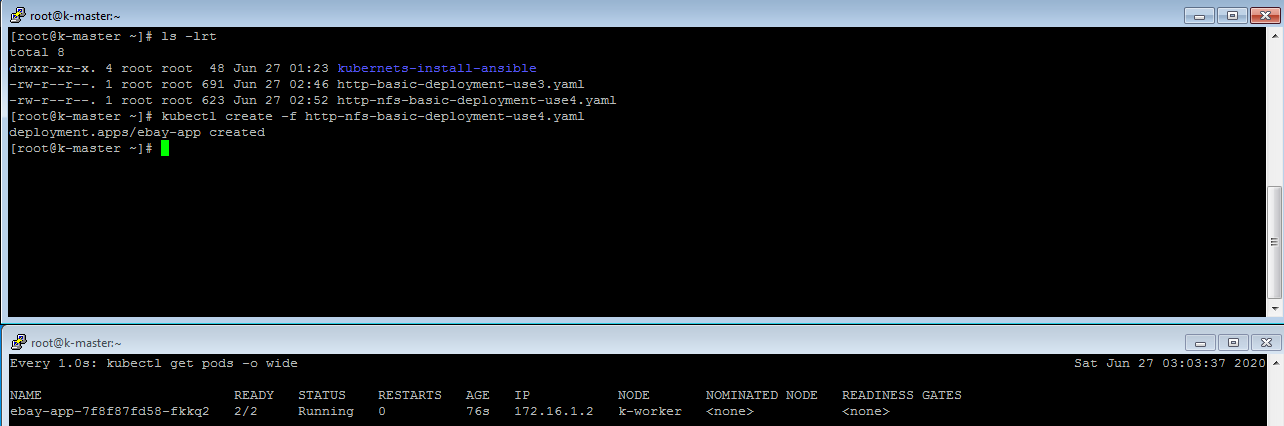
kubectl create -f http-nfs-basic-deployment-use4.yaml --dry-run





kubectl create -f http-nfs-basic-deployment-use4.yaml

watch –n 1 kubectl create -f http-nfs-basic-deployment-use4.yaml



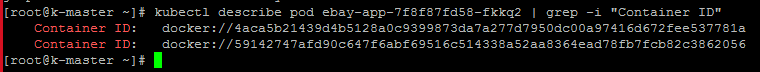
Here, we are running describe command to get container id

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

NAME READY STATUS RESTARTS AGE

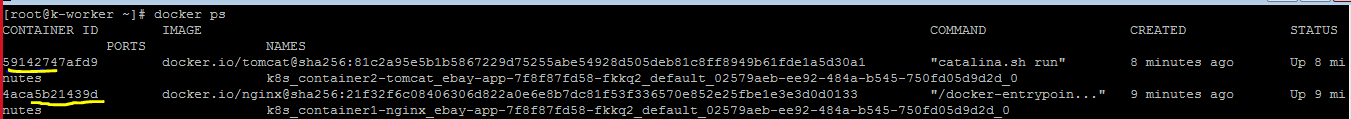
ebay-app-7f8f87fd58-fkkq2 2/2 Running 0 4m39s

kubectl describe pod ebay-app-7f8f87fd58-fkkq2 | grep -i "Container ID"



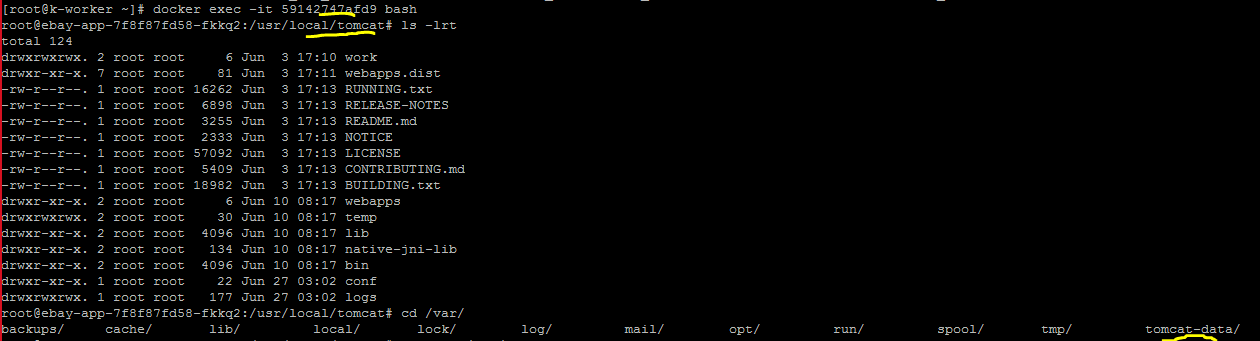
Move to worker node to match container id

docker ps

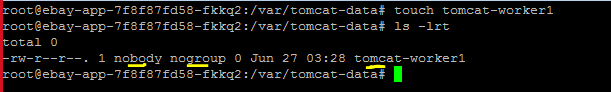


Now we need to move to inside the container to verify nfs mount point(mnt data)

As it is container 2, that’s why we see tomcat-data dir



Lets create file inside tomcat container

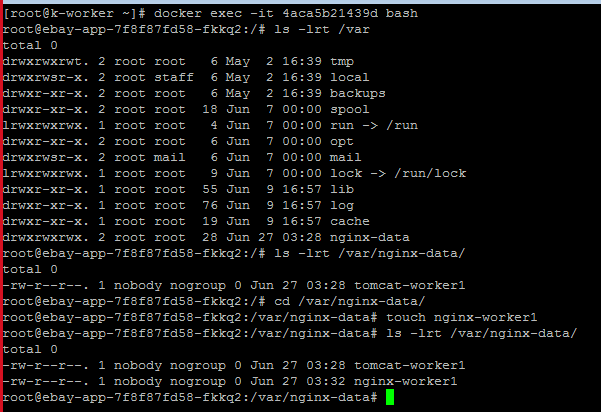


Lets login to nginx container

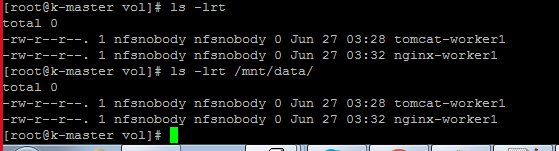
docker exec -it 4aca5b21439d bash

cd /var/nginx-data/

touch nginx-worker1



Let see data in master node

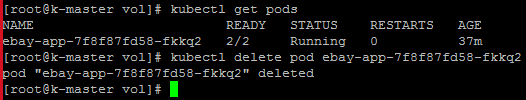


# Scenario 1

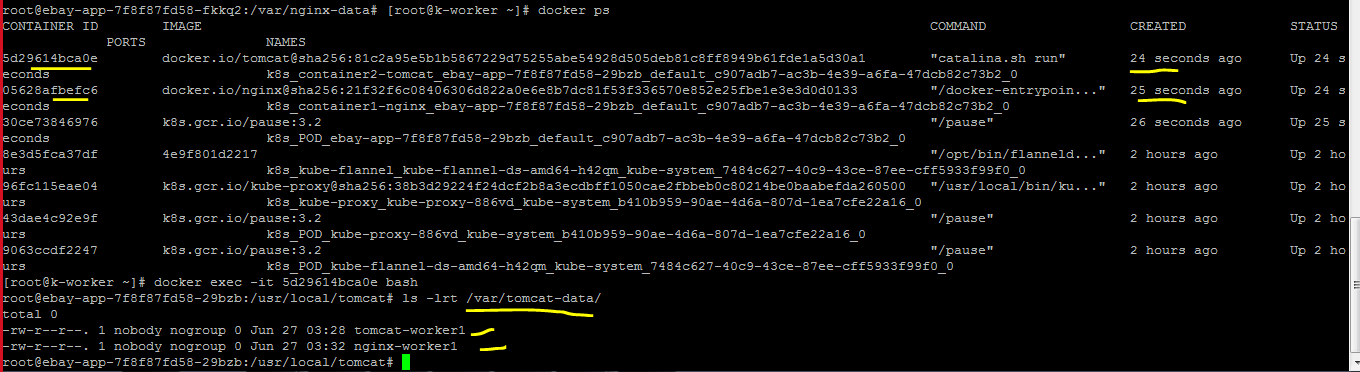
(here, we are trying to remove pod to check persistent volume data)

kubectl get pods

kubectl delete pod ebay-app-7f8f87fd58-fkkq

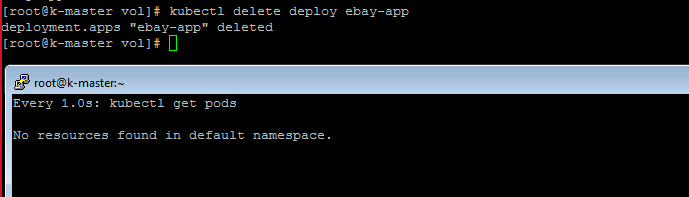


As we have replica set enable, so container got create automatic



# Scenario 2

Lets remove deploy



Let check data inside /mnt/data which is mount with nfs

