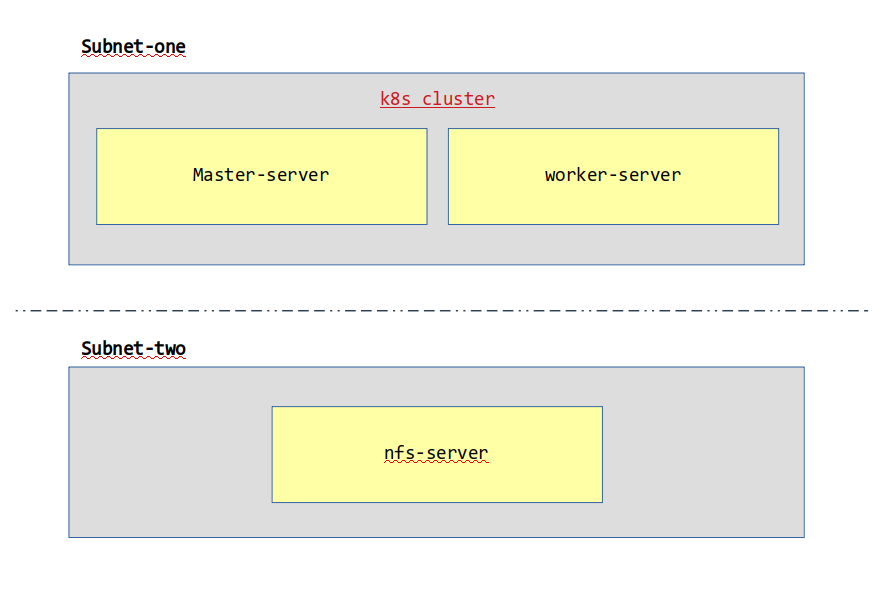
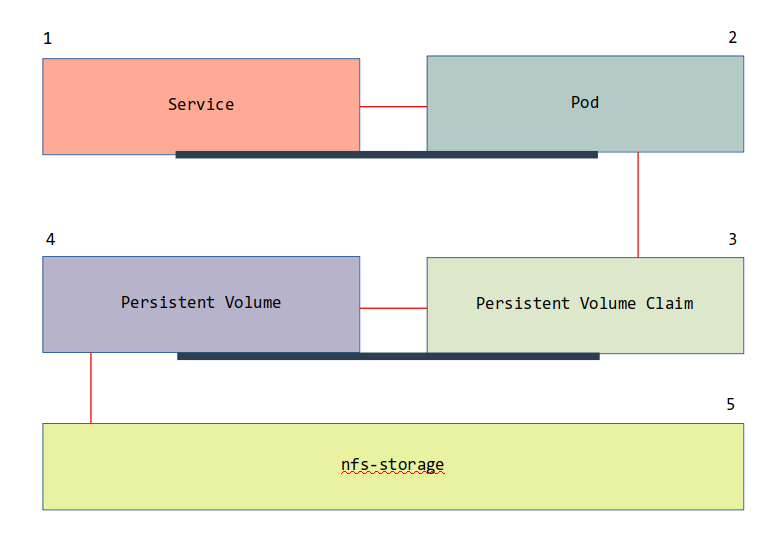
**Nfs-server**

Sharing data between containers is often a necessary component of container-based services and applications. You usually have various pods that need access to the same information on an external persistent volume. While creating an NFS Server on an VM is another form of persistent shared storage.



**Volume attached process**

****

**Create nfs-server**

Requirements:

1. ubuntu server

nfs-server.sh

#!/bin/bash

# This script should be executed on Linux Ubuntu Virtual Machine

EXPORT\_DIRECTORY=${1:-/export/data}

DATA\_DIRECTORY=${2:-/data}

SUBNET=${3:-\*}

apt-get -y update

apt-get -y install nfs-kernel-server

mkdir -p ${DATA\_DIRECTORY}

mkdir -p ${EXPORT\_DIRECTORY}

mount --bind ${DATA\_DIRECTORY} ${EXPORT\_DIRECTORY}

chmod 777 ${EXPORT\_DIRECTORY}

parentdir="$(dirname "$EXPORT\_DIRECTORY")"

chmod 777 $parentdir

echo "${DATA\_DIRECTORY} ${EXPORT\_DIRECTORY} none bind 0 0" >> /etc/fstab

echo "/export ${SUBNET}(rw,async,insecure,fsid=0,crossmnt,no\_subtree\_check)" >> /etc/exports

echo "/export localhost(rw,async,insecure,fsid=0,crossmnt,no\_subtree\_check)" >> /etc/exports

Create kubernetes cluster

Requirements:

1. two ubuntu linux server

assign hostname:

master server name – master

worker server name - node

this link provide ansible tutomation script for kubernetes

packages details:

1. apt install ansible -y

2. apt install sshpass -y

ansible server setup (my local machine act as ansible-server)

vim hosts

master ansible\_user=node ansible\_host=13.71.68.213 ansible\_ssh\_pass=Password@1234567

node ansible\_user=node ansible\_host=13.71.68.221 ansible\_ssh\_pass=Password@1234567

vim ansible.cfg

[defaults]

inventory=hosts

host\_key\_checking = False

stdout\_callback = unixy

refer:

1. https://raw.githubusercontent.com/FourTimes/aws-kubernetes/master/k8s-manual.yml

2. <https://raw.githubusercontent.com/FourTimes/aws-kubernetes/master/token.j2>

3. https://raw.githubusercontent.com/FourTimes/aws-kubernetes/master/daemon.json.j2

# after install login into master node execute

# kubectl get nodes

# kubectl get all

# kubectl get ns

Create persistent volume (pv)

apiVersion: v1

kind: PersistentVolume

metadata:

name: nfs-pv # → pv callback name

labels:

type: nfs

spec:

capacity:

storage: 1Gi

accessModes:

- ReadWriteMany

nfs:

server: 10.0.1.6 # → this is nfs server IP

path: /data # → this is nfs-server path

Create persistent volume claim from persistent volume

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: nfs-pvc # → pvc calback name

spec:

accessModes:

- ReadWriteMany

storageClassName: "" # → "storageClassName" needs to remain an empty string or the claim won't work.

resources:

requests:

storage: 1Gi

selector:

matchLabels:

type: nfs # → this label from nfs pv

create pod with node port services

apiVersion: v1

kind: Pod

metadata:

name: nginx-nfs-pod

labels:

name: nginx-nfs-pod

spec:

containers:

- name: nginx-nfs-pod

image: nginx

ports:

- name: web

containerPort: 80

volumeMounts:

- name: nfsvol

mountPath: /usr/share/nginx/html

volumes:

- name: nfsvol

persistentVolumeClaim:

claimName: nfs-pvc

---

apiVersion: v1

kind: Service

metadata:

name: web-service

labels:

role: web-service

spec:

selector:

name: nginx-nfs-pod

type: NodePort

ports:

- port: 80

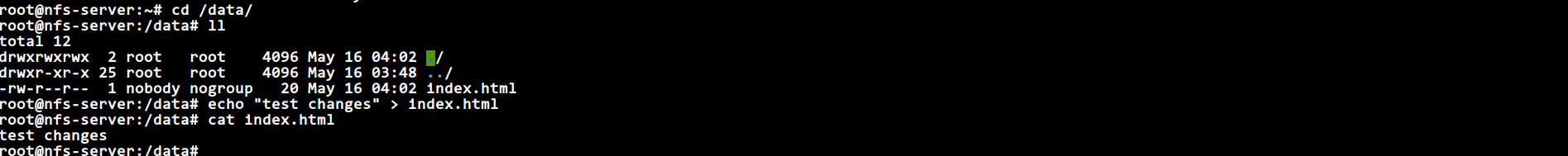
nodePort: 32001

# curl [master-server-ip]:32001

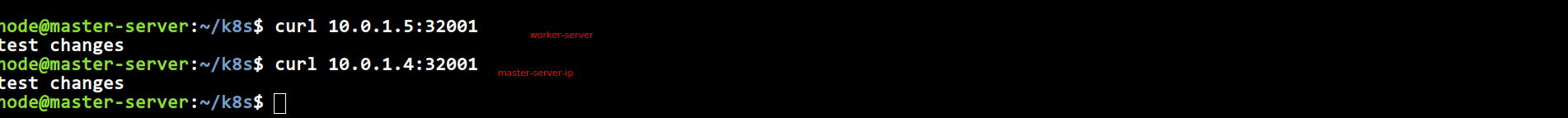
# curl [worker-server-ip]:32001

verify this

login nfs-server



verify this



Configure with multi replicaset

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx

spec:

strategy:

type: RollingUpdate #Bydefault (RollingUpdate/Recreate)

rollingUpdate:

maxSurge: 2

maxUnavailable: 1

revisionHistoryLimit: 4

paused: false

replicas: 3

minReadySeconds: 10

selector:

matchLabels:

name: nginx-nfs-pod

template:

metadata:

name: web

labels:

name: nginx-nfs-pod

spec:

containers:

- name: nginx-nfs-pod

image: nginx

ports:

- name: web

containerPort: 80

volumeMounts:

- name: nfsvol

mountPath: /usr/share/nginx/html

volumes:

- name: nfsvol

persistentVolumeClaim:

claimName: nfs-pvc

---

apiVersion: v1

kind: Service

metadata:

name: web-service

labels:

role: web-service

spec:

selector:

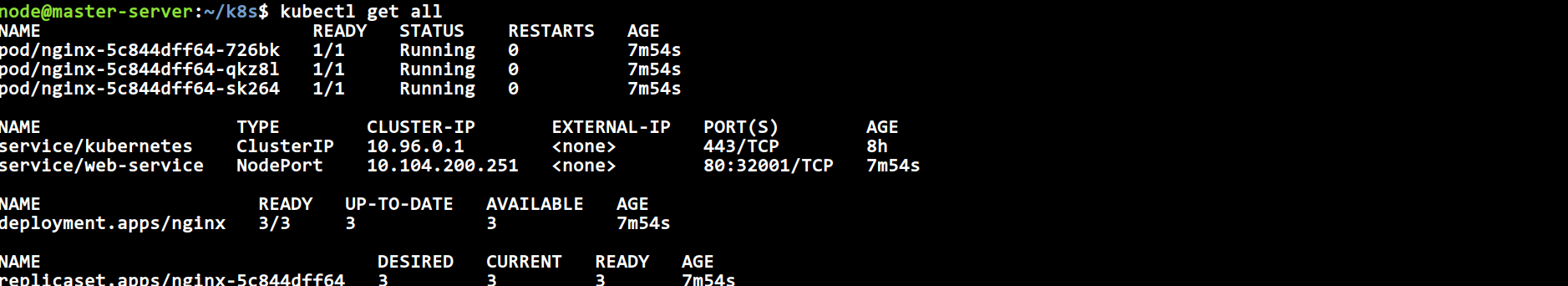
name: nginx-nfs-pod

type: NodePort

ports:

- port: 80

nodePort: 32001



check access log for each containers

