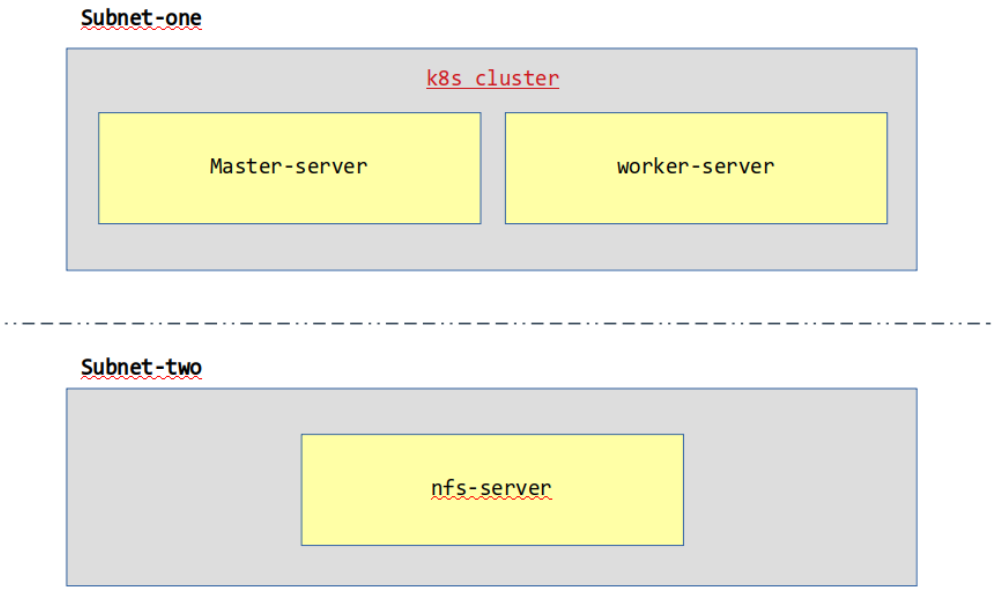
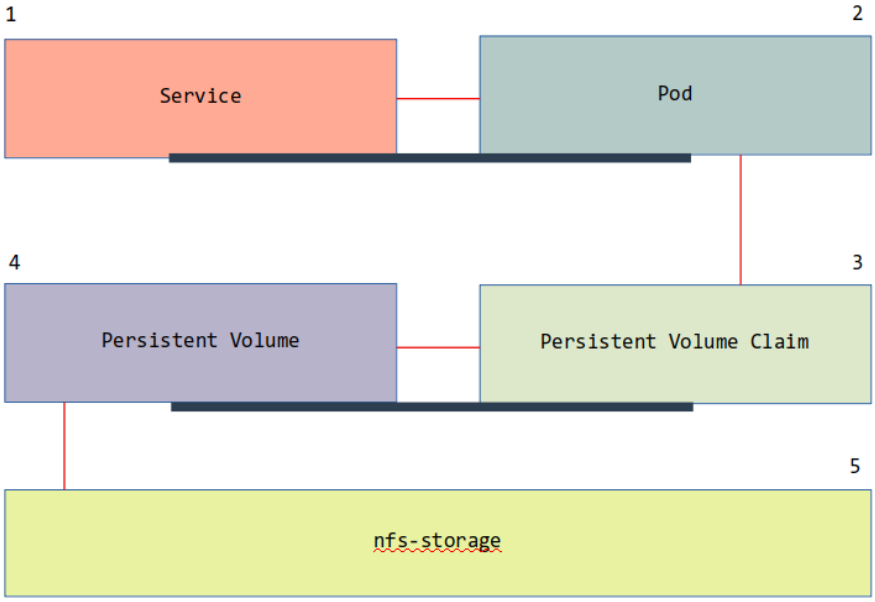
Nfs-server

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