**How to create dynamic NFS SC from KCLI installer VM**

Pre-condition:

If pv is already created with exportfs -rav listed

If Deployed DynaNFS on different ENV then pv.yaml setup must prepare and create base on DO322 student guide.

git clone https://github.com/kubernetes-sigs/nfs-subdir-external-provisioner/

oc edit configs.imageregistry/cluster

apiVersion: imageregistry.operator.openshift.io/v1

kind: Config

metadata:

name: cluster

spec:

httpSecret: d89ed8cb6766cfd01abe9e6e32b351bfce523c4272b4d32ed215799997ec30ee169206938f5329f28f67af17f6e4ca2cd46a1c4aa5afad28e143ba5bacbfd9c8

logLevel: Normal

managementState: **Managed**

observedConfig: null

operatorLogLevel: Normal

proxy: {}

replicas: 1

requests:

read:

maxWaitInQueue: 0s

write:

maxWaitInQueue: 0s

rolloutStrategy: RollingUpdate

storage:

pvc:

claim: image-registry-storage

[root@cnfdf06-installer deploy]# oc get po -n openshift-image-registry |grep image-regis

image-registry-6d578f8f5b-6jhq4 1/1 Running 0 35h

oc get pv|grep pv015

pv015 100Gi RWX Recycle Bound openshift-image-registry/image-registry-storage

**#check exportfs**

[root@cnfdf06-installer deploy]# exportfs -rva

exporting \*:/**pv015**

[root@cnfdf06-installer deploy]# oc describe pv pv015

Name: pv015

Labels: <none>

Annotations: pv.kubernetes.io/bound-by-controller: yes

Finalizers: [kubernetes.io/pv-protection]

StorageClass:

Status: Bound

Claim: openshift-image-registry/image-registry-storage

Reclaim Policy: Recycle

Access Modes: RWX

VolumeMode: Filesystem

Capacity: 100Gi

Node Affinity: <none>

Message:

Source:

Type: NFS (an NFS mount that lasts the lifetime of a pod)

**Server: 192.168.128.146**

**Path: /pv015**

ReadOnly: false

Events: <none>

[root@cnfdf06-installer deploy]# ls -lZ /**pv015**\*

drwxrwxrwx. 2 root root **system\_u:object\_r:container\_file\_t:s0** 21 Nov 19 15:33 nfs-dynamic-namespace-test-claim-pvc-c02ed08b-6c90-42f8-b0ce-20c6c91e865a

[root@cnfdf06-installer ~]# stat /pv015

File: /pv015

Size: 87 Blocks: 0 IO Block: 4096 directory

Device: fc01h/64513d Inode: 25244829 Links: 3

Access: (0777/drwxrwxrwx) Uid: ( 0/ root) Gid: ( 0/ root)

Context: system\_u:object\_r:container\_file\_t:s0

#create NS/Project

oc create namespace nfs-dynamic-namespace

oc create -f rbac.yaml

oc create role use-scc-hostmount-anyuid --verb=use --resource=scc --resource-name=hostmount-anyuid -n nfs-dynamic-namespace

oc project nfs-dynamic-namespace

oc adm policy add-role-to-user use-scc-hostmount-anyuid -z nfs-client-provisioner --role-namespace='nfs-dynamic-namespace'

oc create -f deployment.yaml

oc get po

oc create -f class.yaml

oc get sc

oc create -f test-claim.yaml

oc get pvc

oc reate test-pod.yaml

oc create -f test-pod.yaml

[root@cnfdf06-installer deploy]# oc get po

NAME READY STATUS RESTARTS AGE

**nfs-client-provisioner-57488d9968-kqmpm 1/1 Running 0 70m**

test-pod 0/1 Completed 0 69m

[root@cnfdf06-installer deploy]# oc get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE

test-claim Bound pvc-c02ed08b-6c90-42f8-b0ce-20c6c91e865a 1Mi RWX managed-nfs-storage 70m

[root@cnfdf06-installer deploy]# oc get sc

NAME PROVISIONER RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE

managed-nfs-storage nfs-dynamic-provisioner Retain Immediate false 70m

Deployment.yaml:

apiVersion: apps/v1

kind: Deployment

metadata:

name: nfs-client-provisioner

labels:

app: nfs-client-provisioner

namespace: nfs-dynamic-namespace

spec:

replicas: 1

strategy:

type: Recreate

selector:

matchLabels:

app: nfs-client-provisioner

template:

metadata:

labels:

app: nfs-client-provisioner

spec:

serviceAccountName: nfs-client-provisioner

containers:

- name: nfs-client-provisioner

image: k8s.gcr.io/sig-storage/nfs-subdir-external-provisioner:v4.0.2

volumeMounts:

- name: nfs-client-root

mountPath: /persistentvolumes

env:

- name: PROVISIONER\_NAME

value: nfs-dynamic-provisioner

- name: NFS\_SERVER

value: 192.168.128.146

- name: NFS\_PATH

value: /pv015

volumes:

- name: nfs-client-root

nfs:

server: 192.168.128.146

path: /pv015

**rbac.yaml:**

* Update namespace and provision name to match with other files

nfs-dynamic-namespace

apiVersion: v1

kind: ServiceAccount

metadata:

name: nfs-client-provisioner

# replace with namespace where provisioner is deployed

namespace: nfs-dynamic-namespace

---

kind: ClusterRole

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: nfs-client-provisioner-runner

rules:

- apiGroups: [""]

resources: ["nodes"]

verbs: ["get", "list", "watch"]

- apiGroups: [""]

resources: ["persistentvolumes"]

verbs: ["get", "list", "watch", "create", "delete"]

- apiGroups: [""]

resources: ["persistentvolumeclaims"]

verbs: ["get", "list", "watch", "update"]

- apiGroups: ["storage.k8s.io"]

resources: ["storageclasses"]

verbs: ["get", "list", "watch"]

- apiGroups: [""]

resources: ["events"]

verbs: ["create", "update", "patch"]

---

kind: ClusterRoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: run-nfs-client-provisioner

subjects:

- kind: ServiceAccount

name: nfs-client-provisioner

# replace with namespace where provisioner is deployed

namespace: nfs-dynamic-namespace

roleRef:

kind: ClusterRole

name: nfs-client-provisioner-runner

apiGroup: rbac.authorization.k8s.io

---

kind: Role

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: leader-locking-nfs-client-provisioner

# replace with namespace where provisioner is deployed

namespace: nfs-dynamic-namespace

rules:

- apiGroups: [""]

resources: ["endpoints"]

verbs: ["get", "list", "watch", "create", "update", "patch"]

---

kind: RoleBinding

apiVersion: rbac.authorization.k8s.io/v1

metadata:

name: leader-locking-nfs-client-provisioner

# replace with namespace where provisioner is deployed

namespace: nfs-dynamic-namespace

subjects:

- kind: ServiceAccount

name: nfs-client-provisioner

# replace with namespace where provisioner is deployed

namespace: nfs-dynamic-namespace

roleRef:

kind: Role

name: leader-locking-nfs-client-provisioner

apiGroup: rbac.authorization.k8s.io

[root@cnfdf06-installer deploy]# cat class.yaml

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: managed-nfs-storage

annotation:

**storageclass.kubernetes.io/is-default-class: "true"**

provisioner: nfs-dynamic-provisioner

reclaimPolicy: Retain

parameters:

archiveOnDelete: "false"

**Dynamic-nfs FILES:**

<https://drive.google.com/file/d/1KxTyrUvvm223HiBLuYs3aKkHhY56MjEb/view?usp=sharing>