NFS configuration:

yum install nfs-utils.x86\_64 -y  
systemctl start nfs-server  
systemctl enable nfs-server  
systemctl status nfs-server

sudo mkdir -p /data/nfs-sc  
sudo chown -R nobody:nobody /data/nfs-sc  # Set ownership to nobody  
sudo chmod 777 /data/nfs-sc  # Set permissions to allow full access  
echo "/data/nfs-sc \*(rw,sync,no\_subtree\_check,no\_root\_squash,no\_wdelay)" | sudo tee -a /etc/exports  
sudo exportfs -rav  
sudo systemctl restart nfs-server  
sudo systemctl enable nfs-server  
sudo exportfs -v

mkdir /nfs

cat deployment.yaml  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: nfs-client-provisioner  
  labels:  
    app: nfs-client-provisioner  
  # replace with namespace where provisioner is deployed  
  namespace: nfs  
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: registry.k8s.io/sig-storage/nfs-subdir-external-provisioner:v4.0.2  
          volumeMounts:  
            - name: nfs-client-root  
              mountPath: /persistentvolumes  
          env:  
            - name: PROVISIONER\_NAME  
              value: nfs-storage  
            - name: NFS\_SERVER  
              value: NFS-IPxxxxx  
            - name: NFS\_PATH  
              value: /data/nfs-sc  
      volumes:  
        - name: nfs-client-root  
          nfs:  
            server: NFS-IPxxxxx  
            path: /data/nfs-sc

cat rbac.yaml  
apiVersion: v1  
kind: ServiceAccount  
metadata:  
  name: nfs-client-provisioner  
  # replace with namespace where provisioner is deployed  
  namespace: nfs  
---  
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  
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  
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  
subjects:  
  - kind: ServiceAccount  
    name: nfs-client-provisioner  
    # replace with namespace where provisioner is deployed  
    namespace: nfs  
roleRef:  
  kind: Role  
  name: leader-locking-nfs-client-provisioner  
  apiGroup: rbac.authorization.k8s.io

cat class.yaml  
apiVersion: storage.k8s.io/v1  
kind: StorageClass  
metadata:  
  name: nfs-client  
provisioner: nfs-storage  
parameters:  
  archiveOnDelete: "false"

  ----------------------------------------------------------------------------------------------------------------------------------------

To enable nfs pods

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

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

oc scale deploy nfs-client-provisioner -n nfs --replicas 1

 ------------------------------------------------------------------------------------------------------------------------------------------

Image registry config

Create the 'image-registry-storage' PVC by updating the Image Registry operator config by updating the management state to 'Managed' and adding 'pvc' and 'claim' keys in the storage key:

oc edit configs.imageregistry.operator.openshift.io

managementState: Managed  
storage:  
  pvc:  
    claim: # leave the claim blank

cat PV yaml

apiVersion: v1  
kind: PersistentVolume  
metadata:  
  name: registry-pv  
spec:  
  accessModes:  
    - ReadWriteMany  
  capacity:  
    storage: 100Gi  
  persistentVolumeReclaimPolicy: Retain  
  nfs:  
    path: /data/nfs-sc  
    server: 172.40.20.12

----------------------------------------------------------------------------------------------------------

How to check Intel VT or AMD-V hardware virtualization extensions enabled.

#lsmod | grep kvm  
kvm\_intel             409600  0  
kvm                  1138688  1 kvm\_intel  
irqbypass              16384  1 kvm

#egrep -c 'flags.\*(vmx|svm)' /proc/cpuinfo #cmd  
4

How to check NX (no execute) flag enabled.

#cat /proc/cpuinfo | grep -i nx #cmd

flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant\_tsc arch\_perfmon rep\_good nopl xtopology tsc\_reliable nonstop\_tsc cpuid tsc\_known\_freq pni pclmulqdq vmx ssse3 fma cx16 pcid sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx f16c rdrand hypervisor lahf\_lm abm 3dnowprefetch invpcid\_single ssbd ibrs ibpb stibp ibrs\_enhanced tpr\_shadow vnmi ept vpid ept\_ad fsgsbase tsc\_adjust bmi1 avx2 smep bmi2 erms invpcid avx512f avx512dq rdseed adx smap avx512ifma clflushopt clwb avx512cd sha\_ni avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves avx\_vnni avx512\_bf16 wbnoinvd arat avx512vbmi umip pku ospke avx512\_vbmi2 gfni vaes vpclmulqdq avx512\_vnni avx512\_bitalg avx512\_vpopcntdq rdpid cldemote movdiri movdir64b fsrm md\_clear serialize amx\_bf16 avx512\_fp16 amx\_tile amx\_int8 flush\_l1d arch\_capabilities  
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant\_tsc arch\_perfmon rep\_good nopl xtopology tsc\_reliable nonstop\_tsc cpuid tsc\_known\_freq pni pclmulqdq vmx ssse3 fma cx16 pcid sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx f16c rdrand hypervisor lahf\_lm abm 3dnowprefetch invpcid\_single ssbd ibrs ibpb stibp ibrs\_enhanced tpr\_shadow vnmi ept vpid ept\_ad fsgsbase tsc\_adjust bmi1 avx2 smep bmi2 erms invpcid avx512f avx512dq rdseed adx smap avx512ifma clflushopt clwb avx512cd sha\_ni avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves avx\_vnni avx512\_bf16 wbnoinvd arat avx512vbmi umip pku ospke avx512\_vbmi2 gfni vaes vpclmulqdq avx512\_vnni avx512\_bitalg avx512\_vpopcntdq rdpid cldemote movdiri movdir64b fsrm md\_clear serialize amx\_bf16 avx512\_fp16 amx\_tile amx\_int8 flush\_l1d arch\_capabilities  
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant\_tsc arch\_perfmon rep\_good nopl xtopology tsc\_reliable nonstop\_tsc cpuid tsc\_known\_freq pni pclmulqdq vmx ssse3 fma cx16 pcid sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx f16c rdrand hypervisor lahf\_lm abm 3dnowprefetch invpcid\_single ssbd ibrs ibpb stibp ibrs\_enhanced tpr\_shadow vnmi ept vpid ept\_ad fsgsbase tsc\_adjust bmi1 avx2 smep bmi2 erms invpcid avx512f avx512dq rdseed adx smap avx512ifma clflushopt clwb avx512cd sha\_ni avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves avx\_vnni avx512\_bf16 wbnoinvd arat avx512vbmi umip pku ospke avx512\_vbmi2 gfni vaes vpclmulqdq avx512\_vnni avx512\_bitalg avx512\_vpopcntdq rdpid cldemote movdiri movdir64b fsrm md\_clear serialize amx\_bf16 avx512\_fp16 amx\_tile amx\_int8 flush\_l1d arch\_capabilities  
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant\_tsc arch\_perfmon rep\_good nopl xtopology tsc\_reliable nonstop\_tsc cpuid tsc\_known\_freq pni pclmulqdq vmx ssse3 fma cx16 pcid sse4\_1 sse4\_2 x2apic movbe popcnt tsc\_deadline\_timer aes xsave avx f16c rdrand hypervisor lahf\_lm abm 3dnowprefetch invpcid\_single ssbd ibrs ibpb stibp ibrs\_enhanced tpr\_shadow vnmi ept vpid ept\_ad fsgsbase tsc\_adjust bmi1 avx2 smep bmi2 erms invpcid avx512f avx512dq rdseed adx smap avx512ifma clflushopt clwb avx512cd sha\_ni avx512bw avx512vl xsaveopt xsavec xgetbv1 xsaves avx\_vnni avx512\_bf16 wbnoinvd arat avx512vbmi umip pku ospke avx512\_vbmi2 gfni vaes vpclmulqdq avx512\_vnni avx512\_bitalg avx512\_vpopcntdq rdpid cldemote movdiri movdir64b fsrm md\_clear serialize amx\_bf16 avx512\_fp16 amx\_tile amx\_int8 flush\_l1d arch\_capabilities

#egrep -c 'flags.\*nx' /proc/cpuinfo  
4

Make you sc as default for virt

oc patch storageclass nfs-client -p '{"metadata": {"annotations": {"sstorageclass.kubevirt.io/is-default-virt-class": "true"}}}'