**Persistent Storage Using NFS**

* OpenShift Container Platform clusters can be provisioned with [persistent storage](https://docs.openshift.com/container-platform/3.5/architecture/additional_concepts/storage.html#architecture-additional-concepts-storage) using NFS.
* Persistent volumes (PVs) and persistent volume claims (PVCs) provide a convenient method for sharing a volume across a project.
* While the NFS-specific information contained in a PV definition could also be defined directly in a pod definition, doing so does not create the volume as a distinct cluster resource, making the volume more susceptible to conflicts.

Note: NFS-spec info must be provided in PV definition but not pod definition

## **Provisioning**

* Storage must exist in the underlying infrastructure before it can be mounted as a volume in OpenShift Container Platform.
* Requirements to provision NFS volumes,

1. A list of NFS servers and
2. Export paths

Steps to mount the volume:

Step 1: You must first create an object definition for the PV:

*Example 1. PV Object Definition Using NFS*

**apiVersion: v1**

**kind: PersistentVolume**

**metadata:**

**name: pv0001 --(name of PV)**

**spec:**

**capacity:**

**storage: 5Gi –(amount of storage allocated)**

**accessModes:**

**- ReadWriteOnce –(label to match PVC and PV)**

**nfs:**

**path: /tmp --(path)**

**server: 172.17.0.2 –(host ip)**

**persistentVolumeReclaimPolicy: Recycle –(reclaim policy)**

PV creation:

Save the definition to a file, for example **nfs-pv.yaml**, and create the PV:

$ oc create -f nfs-pv.yaml

persistentvolume "pv0001" created

PV verification:

Verify that the PV was created:

# oc get pv

NAME LABELS CAPACITY ACCESSMODES STATUS CLAIM REASON AGE

pv0001 <none> 5368709120 RWO Available 31s

Step 2: The next step can be to create a persistent volume claim (PVC) which will bind to the new PV

*Example 2. PVC Object Definition*

**apiVersion: v1**

**kind: PersistentVolumeClaim**

**metadata:**

**name: nfs-claim1**

**spec:**

**accessModes:**

**- ReadWriteOnce --(provides security and acts as label to match to a PV)**

**resources:**

**requests:**

**storage: 1Gi –(This claim looks for PV offering == or > 1 Gi capacity)**

Save the definition to a file, for example **nfs-claim.yaml**, and create the PVC:

# oc create -f nfs-claim.yaml

**Pros**:

* Currently, only NFS and Hostoath support the ‘Recycle’ claim policy.

**Cons**:

🡪 NFS-specific information contained in a PV definition could also be defined directly in a pod definition, doing so does not create the volume as a distinct cluster resource, making the volume more susceptible to conflicts.

🡪Each NFS volume must be mountable by all schedulable nodes in the cluster.

### **iSCSI**

Pros:

Cons:

* Each iSCSI LUN must be accessible by all nodes in the cluster.
* iSCSI does not support the 'Recycle' reclaim policy.