Summary

- Revision of previous class
- Difference between GB & GIB
  - GB=10^3
  - Gib= 2^{10}
- What is provisioner
  - Manifest file

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

  - Creating pod

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl create -f sc_pvc.yml
persistentvolumeclaim/mypvc created

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl  get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
 STORAGECLASS   REASON   AGE
pvc-f048bade-5627-483c-9bdf-c2a25bf1d446   1Gi        RWO            Delete           Bound    default/mypv
 standard               35s

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl  get pvc
NAME    STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mypvc   Bound    pvc-f048bade-5627-483c-9bdf-c2a25bf1d446   1Gi        RWO            standard       46s
```

  - A good practice is to mention the storage class name in the persistent volume claim
  - Provisioner
    - Provisioner: **K8s.io/minikube-hostpath** means the persistent volume created by the storage class is actually taking space from the minikube VM

```
C:\Users\Vimal Daga>kubectl  get  storageclass
NAME                 PROVISIONER                RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION

standard (default)   k8s.io/minikube-hostpath   Delete          Immediate           false
5m
```

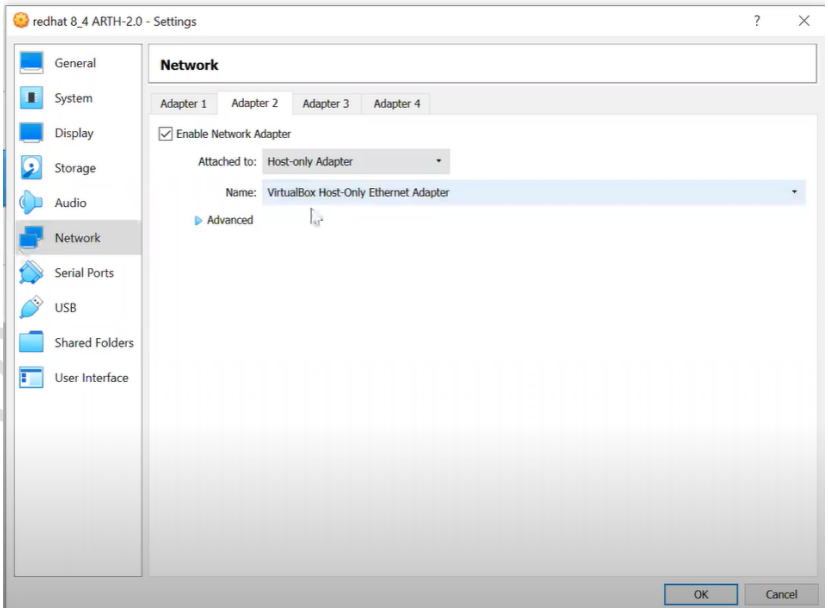
- From where to take storage is the duty of the storage class (Eg Cloud, HD)
- How to change the default storage class
    - Standard storage class is by default

```
C:\Users\Vimal Daga>kubectl  get  storageclass
NAME                 PROVISIONER                RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION
                                                                                    
standard (default)   k8s.io/minikube-hostpath   Delete          Immediate           false
5m
```

    - Editing storage class
        - Command:-kubectl edit sc standard

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"storage.k8s.io/v1","kind":"StorageClass","metadata":{"
    storageclass.kubernetes.io/is-default-class: "false"
  creationTimestamp: "2021-12-28T11:48:26Z"
  labels:
```

- A good practice is while creating persistent volume claims always give the provisioner name
- Practical:- NFS in storage class
    - Changing network setting



    - Setting NFS server
        - Command:-

```
[root@localhost ~]# mkdir /mynfs
[root@localhost ~]# vim /etc/exports
```

```
/mynfs   *(rw,no_root_squash)
~
~
~
~
~
~
~
```

```
[root@localhost ~]# systemctl restart nfs-server.service
[root@localhost ~]# systemctl enable  nfs-server.service
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service
 /usr/lib/systemd/system/nfs-server.service.
[root@localhost ~]# systemctl status   nfs-server.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor
  Drop-In: /run/systemd/generator/nfs-server.service.d
           └─order-with-mounts.conf
   Active: active (exited) since Tue 2021-12-28 22:34:19 IST; 18s ago
 Main PID: 3037 (code=exited, status=0/SUCCESS)
    Tasks: 0 (limit: 49341)
   Memory: 0B
   CGroup: /system.slice/nfs-server.service

Dec 28 22:34:19 localhost.localdomain systemd[1]: Starting NFS server and servi
Dec 28 22:34:19 localhost.localdomain systemd[1]: Started NFS server and servic
[root@localhost ~]# exportfs  -v
/mynfs          <world>(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,no_
oot_squash,no_all_squash)
[root@localhost ~]#
```

- Manifest file

```
kind: Deployment
apiVersion: apps/v1
metadata:
  name: nfs-client-provisioner
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nfs-client-provisioner
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: nfs-client-provisioner
    spec:
      containers:
        - name: nfs-client-provisioner
          image: k8s.gcr.io/sig-storage/nfs-subdir-external-
provisioner:v4.0.2
          volumeMounts:
            - name: nfs-client-root
              mountPath: /mekube
          env:
            - name: PROVISIONER_NAME
              value: k8s-sigs.io/nfs-subdir-external-
provisioner
```

```
            - name: NFS_SERVER
              value: 192.168.59.106
            - name: NFS_PATH
              value: /mynfs
      volumes:
        - name: nfs-client-root
          nfs:
            server: 192.168.59.106
            path: /mynfs
```

- Creating pod
  - Command:- Kubectl create -f (File name)

```
C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl create -f nfs.yml
deployment.apps/nfs-client-provisioner created

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get deploy
NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
nfs-client-provisioner   0/1     1            0           2s

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get pods
NAME                                      READY   STATUS    RESTARTS   AGE
nfs-client-provisioner-559cdcf56f-cdkhs   1/1     Running   0          7s

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get deploy
NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
nfs-client-provisioner   1/1     1            1           14s

C:\Users\Vimal Daga\Documents\Container2021-ws>
```

- Manifest file for Storage class

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: managed-nfs-storage
provisioner: k8s-sigs.io/nfs-subdir-external-provisioner # or choose another name,
parameters:
  pathPattern: "${.PVC.namespace}/${.PVC.annotations.nfs.io/storage-path}" # waits
  onDelete: delete
```

- Creating storage class

```
C:\Users\Vimal Daga\Documents\Container2021-ws>notepad sc_nfs.yml

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get pods
NAME                                      READY   STATUS    RESTARTS   AGE
nfs-client-provisioner-559cdcf56f-cdkhs   1/1     Running   0          4m10s

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl create -f sc_nfs.yml
storageclass.storage.k8s.io/managed-nfs-storage created

C:\Users\Vimal Daga\Documents\Container2021-ws>kubectl get sc
NAME                  PROVISIONER                                    RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVO
LUMEEXPANSION   AGE
managed-nfs-storage   k8s-sigs.io/nfs-subdir-external-provisioner   Delete          Immediate           false
                4s

C:\Users\Vimal Daga\Documents\Container2021-ws>
```

- Now pods will take storage from NFS storage

- A good practice is to mention the storage class name in the persistent volume claim

Important links :

- https://github.com/kubernetes-sigs/nfs-subdir-external-provisioner
- https://kubernetes.io/docs/concepts/storage/storage-classes/