1.create a pod for nginx with the pv and pvc:

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp1
  labels:
    name: myapp1
spec:
  containers:
  - name: myapp
    image: nginx
    volumeMounts:
     - mountPath: "/home"
       name: my-storage
  volumes:
    - name: my-storage
      persistentVolumeClaim:
        claimName: mypvc
```

Pvc:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  resources:
    requests:
      storage: 500Mi
  accessModes:
    - ReadWriteOnce
```

Pv:

```
root@kubemaster-vm:/home/kubemaster/jithu# cat pv.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mypv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt"
```

2.Connection between them&checked the connection in the pod describe

```
Volumes:
 my-storage:
   Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
   ClaimName:  mypvc
   ReadOnly:   false
 kube-api-access-vswkj:
```

Connection Blw pv and pvc:

```
root@kubemaster-vm:/home/kubemaster/jithu# kubectl get pv
NAME   CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM           STORAGECLASS   REASON   AGE
mypv   1Gi        RWO            Retain           Bound    default/mypvc                           30m
root@kubemaster-vm:/home/kubemaster/jithu# kubectl pvc
```

```
root@kubemaster-vm:/home/kubemaster/jithu# kubectl get pvc
NAME    STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mypvc   Bound    mypv     1Gi        RWO                           30m
```

```
root@kubemaster-vm:/home/kubemaster/jithu# kubectl get pv,pvc,pod
NAME                 CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM           STORAGECLASS   REASON   AGE
persistentvolume/mypv   1Gi        RWO            Retain           Bound    default/mypvc                           17s

NAME                          STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
persistentvolumeclaim/mypvc   Bound    mypv     1Gi        RWO                           14s

NAME         READY   STATUS    RESTARTS   AGE
pod/myapp1   1/1     Running   0          9s
root@kubemaster-vm:/home/kubemaster/jithu#
```

3.And tried to create a file in the path of the persistent volume mounted location, you can check in pv given /mnt/ and then it appears in the pod container that path stating that those two are connected:

Check the locations in the configuaration of pv and pod

Inside the container:

```
root@myapp1:/home# ls
data  file.txt
```

In pv location path:

```
root@kubemaster-vm:/mnt# ls
file.txt
```

As I runned the nginx the index.html Is inside the pod:

```
root@kubemaster-vm:/home/kubemaster/jithu# kubectl exec -it myapp1 -- /bin/bash
root@myapp1:/# cd /home
root@myapp1:/home# ls
data  file.txt
root@myapp1:/home# cd ..
root@myapp1:/# ls
bin  boot  dev  docker-entrypoint.d  docker-entrypoint.sh  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@myapp1:/# cd /mnt
root@myapp1:/mnt# ls
root@myapp1:/mnt# cd ..
root@myapp1:/# cd usr/share/nginx/html/
root@myapp1:/usr/share/nginx/html# ls
50x.html  index.html
root@myapp1:/usr/share/nginx/html#
```

Cat the index.html inside the pod

```
root@myapp1:/usr/share/nginx/html# cat index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```