# Volume PV

1. Create an index.html file on your Node

$ minikube ssh  
  
docker$ sudo mkdir /mnt/data  
  
docker$ sudo sh -c "echo 'Hello from Kubernetes storage' \  
 > /mnt/data/index.html"  
  
docker$ cat /mnt/data/index.html  
Hello from Kubernetes storage  
  
docker$ exit

1. Create a PersistentVolume

# volumes/pv-volume.yaml  
apiVersion: v1  
kind: PersistentVolume  
metadata:  
 name: task-pv-volume  
 labels:  
 type: local  
spec:  
 storageClassName: manual  
 capacity:  
 storage: 10Mi  
 accessModes:  
 - ReadWriteOnce  
 hostPath:  
 path: "/mnt/data"

$ kubectl apply -f volumes/pv-volume.yaml  
persistentvolume/task-pv-volume created  
  
$ kubectl get pv task-pv-volume  
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE  
task-pv-volume 10Mi RWO Retain Available manual 43s

1. Create a PersistentVolumeClaim using the following spec:

# volumes/pv-claim.yaml  
apiVersion: v1  
kind: PersistentVolumeClaim  
metadata:  
 name: task-pv-claim  
spec:  
 storageClassName: manual  
 accessModes:  
 - ReadWriteOnce  
 resources:  
 requests:  
 storage: 3Mi

Create the PersistentVolumeClaim:

$ kubectl apply -f volumes/pv-claim.yaml  
persistentvolumeclaim/task-pv-claim created

Look again at the PersistentVolume. Now the output shows a STATUS of Bound.

$ kubectl get pv task-pv-volume  
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE  
task-pv-volume 10Mi RWO Retain Bound my/task-pv-claim manual 4m15s

Look at the PersistentVolumeClaim. The output shows that the PersistentVolumeClaim is bound to your PersistentVolume task-pv-volume.

$ kubectl get pvc task-pv-claim  
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE  
task-pv-claim Bound task-pv-volume 10Mi RWO manual 104s

1. Create a Pod using the following spec:

# volumes/pv-pod.yaml  
apiVersion: v1  
kind: Pod  
metadata:  
 name: task-pv-pod  
spec:  
 volumes:  
 - name: task-pv-storage  
 persistentVolumeClaim:  
 claimName: task-pv-claim  
 containers:  
 - name: task-pv-container  
 image: nginx  
 ports:  
 - containerPort: 80  
 name: "http-server"  
 volumeMounts:  
 - mountPath: "/usr/share/nginx/html"  
 name: task-pv-storage

Create the Pod:

$ kubectl apply -f volumes/pv-pod.yaml  
pod/task-pv-pod created

Verify that the container in the Pod is running:

$ kubectl get pod task-pv-pod  
NAME READY STATUS RESTARTS AGE  
task-pv-pod 1/1 Running 0 69s

Get a shell to the container running in your Pod:

$ kubectl exec -it task-pv-pod -- /bin/bash  
pod# curl http://localhost/  
Hello from Kubernetes storage

1. Clean up:

$ kubectl delete pod task-pv-pod  
$ kubectl delete pvc task-pv-claim  
$ kubectl delete pv task-pv-volume  
  
$ minikube ssh  
docker$ sudo rm /mnt/data/index.html  
docker$ sudo rmdir /mnt/data  
docker$ exit

## Solution

1. Create an index.html file on your Node

$ minikube ssh  
  
docker@minikube:~$ sudo mkdir /mnt/data  
  
docker@minikube:~$ sudo sh -c "echo 'Hello from Kubernetes storage' > /mnt/data/index.html"  
  
docker@minikube:~$ cat /mnt/data/index.html  
Hello from Kubernetes storage  
  
docker@minikube:~$ exit  
logout

1. Create a PersistentVolume

$ kubectl apply -f volumes/pv-volume.yaml  
persistentvolume/task-pv-volume created  
  
$ kubectl get pv task-pv-volume  
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE  
task-pv-volume 10Mi RWO Retain Available manual 5s

1. Create a PersistentVolumeClaim:

$ kubectl apply -f volumes/pv-claim.yaml  
persistentvolumeclaim/task-pv-claim created

Look again at the PersistentVolume. Now the output shows a STATUS of Bound.

$ kubectl get pv task-pv-volume  
NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS REASON AGE  
task-pv-volume 10Mi RWO Retain Bound msuslov/task-pv-claim manual 48s

Look at the PersistentVolumeClaim. The output shows that the PersistentVolumeClaim is bound to your PersistentVolume task-pv-volume.

$ kubectl get pvc task-pv-claim  
NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE  
task-pv-claim Bound task-pv-volume 10Mi RWO manual 38s

1. Create a Pod :

$ kubectl apply -f volumes/pv-pod.yaml  
pod/task-pv-pod created

Verify that the container in the Pod is running:

$ kubectl get pod task-pv-pod  
NAME READY STATUS RESTARTS AGE  
task-pv-pod 1/1 Running 0 7s

Get a shell to the container running in your Pod:

$ kubectl exec -it task-pv-pod -- /bin/bash  
  
root@task-pv-pod:/# curl http://localhost/  
Hello from Kubernetes storage  
  
root@task-pv-pod:/# exit  
exit

1. Clean up:

$ kubectl delete pod/task-pv-pod pvc/task-pv-claim pv/task-pv-volume  
pod "task-pv-pod" deleted  
persistentvolumeclaim "task-pv-claim" deleted  
persistentvolume "task-pv-volume" deleted  
  
$ minikube ssh  
docker@minikube:~$ sudo rm -rf /mnt/data  
docker@minikube:~$ exit  
logout