

# [20] Persistent Volume Claim

## Pod

## Application with Persistent Volume Claim

- Create a new PersistentVolumeClaim
  - Name: `app-volume`
  - Storage Class: `app-hostpath-sc`
  - Capacity: `10Mi`
- Create a new pod which mounts the PersistentVolumeClaim as a volume
  - Name: `web-server-pod`
  - Image: `nginx`
  - Mount Path: `/usr/share/nginx/html`
- Configure the new pod to have `ReadWriteMany` access on the volume.
- `node.kubernetes.io`: k8s

## Reference

docs persistent volume → PersistentVolumeClaim

[Persistent Volumes](#)

docs persistent volume → claims as volumes

[Persistent Volumes](#)



```
[user@console ~]$ kubectl config use-context k8s
```

```
[user@console ~]$ vi app-volume-pvc.yaml
```

```
#docs Persistent volume -> PersistentVolumeClaims -> PVC
```

```
apiVersion: v1
```

```
kind: PersistentVolumeClaim
```

```
metadata:
```

```
  name: app-volume
```

```
spec:
```

```
  accessModes:
```

```
    - ReadWriteMany
```

```
  volumeMode: Filesystem
```

```
  resources:
```

```
    requests:
```

```
      storage: 10Mi
```

```
  storageClassName: app-hostpath-sc
```

```
:wq
```

```
[user@console ~]$ kubectl apply -f app-volume-pvc.yaml
```

```
[user@console ~]$ kubectl get pvc
```

```
[user@console ~]$ vi web-server-pod.yaml
```

```
#docs persistent volume -> Claims as volumes -> PVC
```

```
apiVersion: v1
```

```
kind: Pod
```

```
metadata:
```

```
  name: web-server-pod
```

```
spec:
```

```
  containers:
```

```
    - name: nginx
```

```
      image: nginx
```

```
      volumeMounts:
```

```
        - mountPath: "/usr/share/nginx/html"
```

```
    name: mypd
volumes:
  - name: mypd
    persistentVolumeClaim:
      claimName: app-volume
```

:wq

```
[user@console ~]$ kubectl apply -f web-server-pod.yaml
```

```
[user@console ~]$ kubectl get pods
```

```
[user@console ~]$ kubectl describe pod web-server-pod
```

## 0000 (610)

kubectl config use-context k8s Create a new PersistentVolumeClaim:

Name: pv-volume

Class: app-hostpath-sc

Capacity: 10Mi Create a new Pod which mounts the PersistentVolumeClaim as a volume: Name: web-server-pod Image: nginx Mount path: /usr/share/nginx/html Configure the new Pod to have ReadWriteMany access on the volume. Finally, using kubectl edit or kubectl patch expand the PersistentVolumeClaim to a capacity of 70Mi and record that change.

- 00

## 0000 (130)

List all persistent volumes sorted by capacity, saving the full kubectl output to /opt/KUCC00102/volume\_list.

Use kubectl 's own functionality for sorting the output, and do not manipulate it any further

- 

```
kubectrl get pv --sort-by=.spec.capacity.storage > /opt/KUCC00102/volume_list
```

## (16)

Create a persistent volume with name app-data, of capacity 2Gi and access mode ReadWriteMany.

The type of volume is hostPath and its location is /srv/app-data.

- 

```
vi pv.yaml

apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 2Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: slow
  hostPath:
    path: /srv/app-data

:wq
```

```
kubectrl apply -f pv.yaml
```

## (55)

Task Create a persistent volume with name app-data , of capacity 1Gi and access mode ReadOnlyMany. The type of volume is hostPath and its location is /srv/app-data .

- 

```
vi pv.yaml

apiVersion: v1
kind: PersistentVolume
metadata:
  name: app-data
spec:
  capacity:
    storage: 1Gi
  volumeMode: Filesystem
  accessModes:
    - ReadOnlyMany
  persistentVolumeReclaimPolicy: Recycle
  storageClassName: slow
  hostPath:
    path: /srv/app-data

:wq
```

```
kubectl apply -f pv.yaml
```

---

Revision #1

Created 31 May 2023 00:15:16 by 

Updated 20 June 2023 13:20:12 by 