Using PersistentVolumes

Relevant Documentation

- Persistent Volumes

Exam Tips

- A PersistentVolume defines a storage resource.
- A PersistentVolumeClaim defines a request to consume a storage resource.
- PersistentVolumeClaims automatically bind to a PersistentVolume that meets their criteria.
- Mount a PersistentVolumeClaim to a container like a regular volume.

Lesson Reference

Note: In the previous lesson, we created a file in the directory `/etc/hostPath` on both of our worker nodes. This lesson assumes that these files are still in place.

Log in to the **control plane node**.

Create a PersistentVolume that mounts the data from the host.

```
vi hostpath-pv.yml
```

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: hostpath-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
  - ReadWriteOnce
  storageClassName: slow
  hostPath:
    path: /etc/hostPath
    type: Directory
```

```
kubectl apply -f hostpath-pv.yml
```

Check the status of the PersistentVolume.

```
kubectl get pv
```

Create a PersistentVolumeClaim that will bind to the PersistentVolume.

```
vi hostpath-pvc.yml
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: hostpath-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 200Mi
  storageClassName: slow
```

```
kubectl apply -f hostpath-pvc.yml
```

Check the status of the PersistentVolumeClaim to verify that is has bound itself to the PersistentVolume. The `STATUS` should be `Bound`, and the `VOLUME` should show the name of the PersistentVolume.

```
kubectl get pvc hostpath-pvc
```

Create a Pod that mounts the PersistentVolumeClaim.

```
vi pv-pod-test.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: pv-pod-test
spec:
  restartPolicy: OnFailure
  containers:
  - name: busybox
    image: busybox:stable
    command: ['sh', '-c', 'cat /data/data.txt']
    volumeMounts:
    - name: pv-host-data
      mountPath: /data
  volumes:
  - name: pv-host-data
    persistentVolumeClaim:
      claimName: hostpath-pvc
```

```
kubectl apply -f pv-pod-test.yml
```

Check the Pod status.

```
kubectl get pod pv-pod-test
```

Check the Pod logs. You should see output from the PersistentVolume data indicating which worker node the Pod is running on.

```
kubectl logs pv-pod-test
```