Using K8s Volumes

Relevant Documentation

- Volumes

Lesson Reference

Create a Pod that uses a hostPath volume to store data on the host.

```
vi volume-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: volume-pod
spec:
  restartPolicy: Never
  containers:
  - name: busybox
    image: busybox
    command: ['sh', '-c', 'echo Success! > /output/success.txt']
    volumeMounts:
    - name: my-volume
      mountPath: /output
  volumes:
  - name: my-volume
    hostPath:
      path: /var/data
```

```
kubectl create -f volume-pod.yml
```

Check which worker node the pod is running on.

```
kubectl get pod volume-pod -o wide
```

Log in to that host and verify the contents of the output file.

```
cat /var/data/success.txt
```

Create a multi-container Pod with an emptyDir volume shared between containers.

```
vi shared-volume-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: shared-volume-pod
spec:
  containers:
  - name: busybox1
```

```
    image: busybox
    command: ['sh', '-c', 'while true; do echo Success! > /output/output.txt; sleep 5; done']
    volumeMounts:
    - name: my-volume
      mountPath: /output
  - name: busybox2
    image: busybox
    command: ['sh', '-c', 'while true; do cat /input/output.txt; sleep 5; done']
    volumeMounts:
    - name: my-volume
      mountPath: /input
  volumes:
  - name: my-volume
    emptyDir: {}
```

```
kubectl create -f shared-volume-pod.yml
```

Check the container log for busybox2. You should see the data that was generated by busybox1.

```
kubectl logs shared-volume-pod -c busybox2
```