Lesson 07 Demo 02

Mounting Pod Files to Host with hostPath

Objective: To create a hostPath volume to mount files from a pod onto the file system of the host node

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster should already be set up (refer to the steps in Lesson 02, Demo 01 for guidance).

Steps to be followed:

1. Create a pod using hostPath
2. Create files within the pod
3. Access files on other nodes

Step 1: Create a pod using hostPath

1.1 To create a hostPath volume, draft the following YAML code and save it in the **hostpath.yaml** file:

```
apiVersion: v1
kind: Pod
metadata:
  name: httpd-vol
spec:
  containers:
  - image: docker.io/httpd
    name: httpd-container
    volumeMounts:
      - mountPath: /data
        name: httpd-volume
  volumes:
  - name: httpd-volume
    hostPath:
      path: /tmp/data
```

```
apiVersion: v1
kind: Pod
metadata:
  name: httpd-vol
spec:
  containers:
  - image: docker.io/httpd
    name: httpd-container
    volumeMounts:
      - mountPath: /data
        name: httpd-volume
  volumes:
  - name: httpd-volume
    hostPath:
      path: /tmp/data



~
~
~
~
~
~
~
:wq
```

```
labsuser@master:~$ kubectl get nodes
NAME                       STATUS   ROLES           AGE   VERSION
master.example.com         Ready    control-plane   16h   v1.28.2
worker-node-1.example.com  Ready    <none>          16h   v1.28.2
worker-node-2.example.com  Ready    <none>          16h   v1.28.2
labsuser@master:~$ vi hostpath.yaml
labsuser@master:~$ cat hostpath.yaml
apiVersion: v1
kind: Pod
metadata:
  name: httpd-vol
spec:
  containers:
  - image: docker.io/httpd
    name: httpd-container
    volumeMounts:
      - mountPath: /data
        name: httpd-volume
  volumes:
  - name: httpd-volume
    hostPath:
      path: /tmp/data



labsuser@master:~$
```

1.2 Use the following command to create the pod with the hostPath:
kubectl apply -f hostpath.yaml

```
labsuser@master:~$ kubectl apply -f hostpath.yaml
pod/httpd-vol created
labsuser@master:~$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
httpd-vol   1/1     Running   0          11s
labsuser@master:~$
```

Upon execution, a pod with a hostPath volume will be created.

Step 2: Create files within the pod

2.1 Start a shell session inside the **httpd-vol** pod using the command:
kubectl exec -it httpd-vol -- bash

```
labsuser@master:~$ kubectl exec -it httpd-vol -- bash
root@httpd-vol:/usr/local/apache2#
```

Note: You are now inside the **httpd-vol** pod.

2.2 Create a directory inside the container and add multiple files to it so other nodes can access them later. Run the following commands:
cd /data/
touch file{1..10}.txt

```
labsuser@master:~$ kubectl exec -it httpd-vol -- bash
root@httpd-vol:/usr/local/apache2# cd /data/
root@httpd-vol:/data# touch file{1..10}.txt
root@httpd-vol:/data#
```

2.3 List the files in the **/data** directory using:
ls

```
root@httpd-vol:/data# touch file{1..10}.txt
root@httpd-vol:/data# ls
file1.txt  file10.txt  file2.txt  file3.txt  file4.txt  file5.txt  file6.txt  file7.txt  file8.txt  file9.txt
root@httpd-vol:/data#
```

```
root@httpd-vol:/data# ls
file1.txt  file10.txt  file2.txt  file3.txt  file4.txt  file5.txt  file6.txt  file7.txt  file8.txt  file9.txt
root@httpd-vol:/data# exit
exit
labsuser@master:~$
```

Note: Exit the shell using the **exit** command

Step 3: Access files on other nodes

3.1 To determine on which node a pod is running, use the command:
kubectl get pods -o wide

```
labsuser@master:~$ kubectl get pods -o wide
NAME        READY   STATUS    RESTARTS   AGE     IP               NODE                     NOMINATED NODE   READINESS GATES
httpd-vol   1/1     Running   0          6m34s   192.168.232.193  worker-node-2.example.com  <none>           <none>
labsuser@master:~$
```

If, for instance, the **httpd-vol** pod is running on **worker-node-2**, switch to the terminal of **worker-node-2**.

```
labsuser@worker-node-2:~$
```

3.2 Navigate to the **/tmp/** directory:
cd /tmp/
ls

```
labsuser@worker-node-2:~$ cd /tmp/
labsuser@worker-node-2:/tmp$ ls
data
dcv-pcscd-0
snap-private-tmp
ssh-XXXXXX2si0i6
systemd-private-01fd5ed2d6a14eb08e2b213a7bf93411-ModemManager.service-utd3ZF
systemd-private-01fd5ed2d6a14eb08e2b213a7bf93411-chrony.service-ZvyBYn
systemd-private-01fd5ed2d6a14eb08e2b213a7bf93411-colord.service-7hlIkh
systemd-private-01fd5ed2d6a14eb08e2b213a7bf93411-power-profiles-daemon.service-SvO3Jm
systemd-private-01fd5ed2d6a14eb08e2b213a7bf93411-systemd-logind.service-IMClmv
systemd-private-01fd5ed2d6a14eb08e2b213a7bf93411-systemd-oomd.service-KIsSfC
systemd-private-01fd5ed2d6a14eb08e2b213a7bf93411-systemd-resolved.service-OZPEvQ
systemd-private-01fd5ed2d6a14eb08e2b213a7bf93411-upower.service-oH5aku
tmp.O1wmCqwomo
tracker-extract-3-files.1001
labsuser@worker-node-2:/tmp$
```

3.3 Navigate to the **data** directory using the **cd** command:
cd data

```
labsuser@worker-node-2:/tmp$ cd data
labsuser@worker-node-2:/tmp/data$
```

3.4 Confirm the files created in the previous step are present on **worker-node-2**:
ls

```
labsuser@worker-node-2:/tmp$ cd data
labsuser@worker-node-2:/tmp/data$ ls
file1.txt  file10.txt  file2.txt  file3.txt  file4.txt  file5.txt  file6.txt  file7.txt  file8.txt  file9.txt
labsuser@worker-node-2:/tmp/data$
```

Due to the utilization of the hostPath volume, all the files can now be accessed on **worker-node-2**.

By following these steps, you have successfully demonstrated how to use a hostPath volume to mount files from a pod to the host node's file system.