A picture containing clipart

Description generated with very high confidence

KUBERNETES MODULE 4

Hands-on: 1

[**support@intellipaat.com**](mailto:support@intellipaat.com)

**+91-7022374614**

**US: 1-800-216-8930(Toll Free)**

**Creating a Volume**

**Operation 1:** Write a yaml file for the volume that you are creating inside a pod

apiVersion: v1

kind: Pod

metadata:

name: sharevol

spec:

containers:

- name: c1

image: centos:7

command:

- "bin/bash"

- "-c"

- "sleep 10000"

volumeMounts:

- name: xchange

mountPath: "/tmp/xchange"

- name: c2

image: centos:7

command:

- "bin/bash"

- "-c"

- "sleep 10000"

volumeMounts:

- name: xchange

mountPath: "/tmp/data"

volumes:

- name: xchange

emptyDir: {}

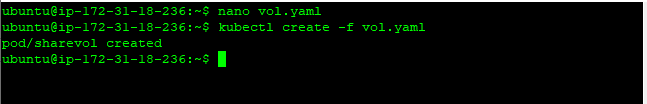
**Operation 2:** Go to command line and create a yaml file and paste the above created specs.

**$ nano <file\_name>.yaml**

Once done hit Ctrl+s and then Ctrl+x to exit

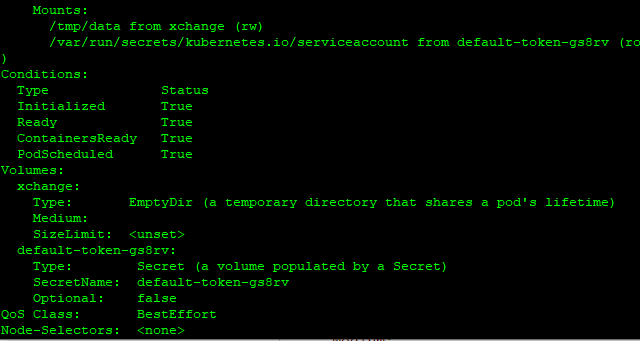
**Operation 3:** next thing to do is to create the yaml file.

**$ kubectl create -f <file\_name>.yaml**



**Operation 4:** to check if your volume got created run the following command

**kubectl describe pod <pod name>**



**Operation 5:** exec into one of the containers in the pod, c1, check the volume mount and generate some data

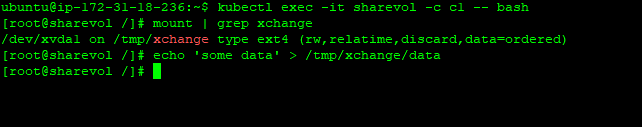
**$ kubectl exec -it sharevol -c c1 -- bash**

**[root@sharevol /]# mount | grep xchange**

**[root@sharevol /]# echo 'some data' > /tmp/xchange/data**

**Operation 5:** To see the output go to the logs of the pod by running the following command

**$ kubectl logs <pod name>**



**Operation 6:** now exec into c2, the second container running in the pod, we can see the volume mounted at /tmp/data and are able to read the data created in the previous step:

**$ kubectl exec -it sharevol -c c2 -- bash**

**[root@sharevol /]# mount | grep /tmp/data**

**[root@sharevol /]#cat /tmp/data/data**

