

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: {}
```

Operations 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 save & exit

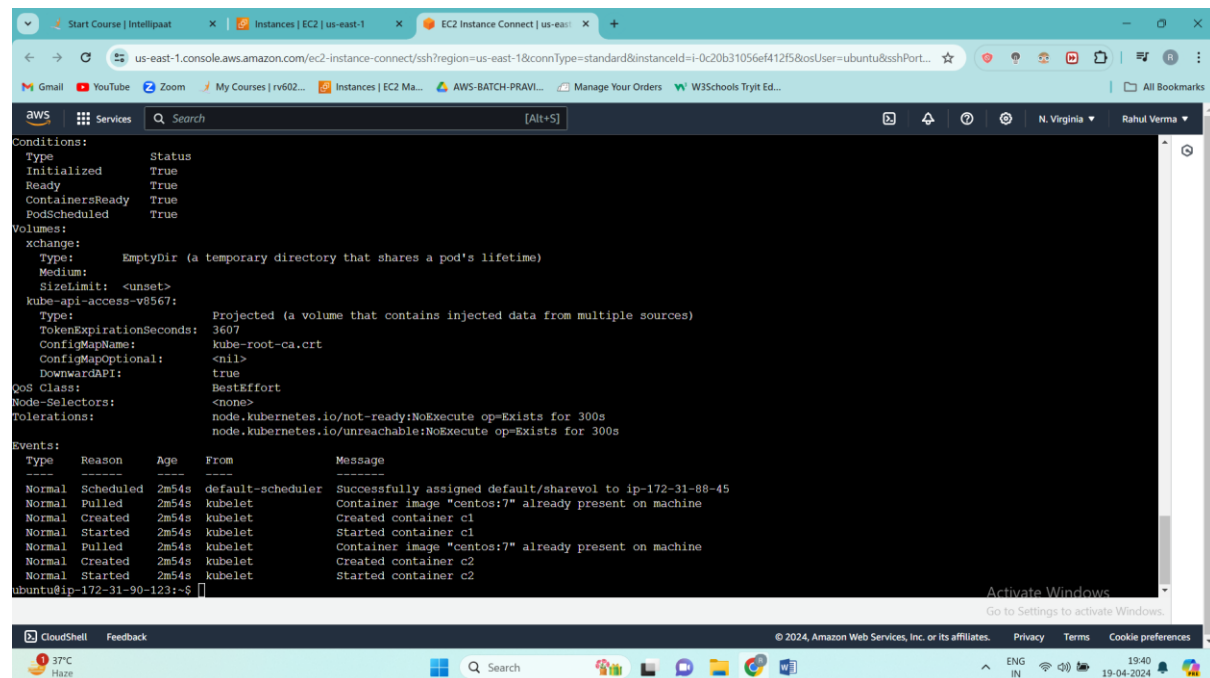
>next thing to do is to create the yaml file.

```
kubectl create -f <file name>
```

```
ubuntu@ip-172-31-90-123:~$ nano vol.yaml
ubuntu@ip-172-31-90-123:~$ kubectl create -f vol.yaml
pod/sharevol created
ubuntu@ip-172-31-90-123:~$
```

Operation 3: to check if your volume got created run the following command

```
kubectl describe pod <pod name>
```



```
Conditions:
  Type             Status
  Initialized       True
  Ready            True
  ContainersReady   True
  PodScheduled      True
Volumes:
  xchange:
    Type: EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
    SizeLimit: <unset>
  kube-api-access-v8567:
    Type: Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName: kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI: true
QoS Class: BestEffort
Node-Selectors: <none>
Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
              node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason      Age    From          Message
  ----     ------      -
  Normal   Scheduled   2m54s  default-scheduler  Successfully assigned default/sharevol to ip-172-31-88-45
  Normal   Pulled      2m54s  kubelet        Container image "centos:7" already present on machine
  Normal   Created     2m54s  kubelet        Created container c1
  Normal   Started     2m54s  kubelet        Started container c1
  Normal   Pulled      2m54s  kubelet        Container image "centos:7" already present on machine
  Normal   Created     2m54s  kubelet        Created container c2
  Normal   Started     2m54s  kubelet        Started container c2
ubuntu@ip-172-31-90-123:~$
```

Operation 4: exec into one of the containers in the pod, c1, check the volume mount and generate some data

```
kubect1 exec -it sharevol -c c1 --bash
```

```
mount | grep xchange
```

```
ubuntu@ip-172-31-90-123:~$ kubect1 exec -it sharevol -c c1 -- bash
[root@sharevol /]# mount | grep xchange
/dev/root on /tmp/xchange type ext4 (rw,relatime,discard)
[root@sharevol /]#
```

Now write-

```
echo "Kubernetes is the best" > /tmp/xchange/data
```

```
[root@sharevol /]# mount | grep xchange
/dev/root on /tmp/xchange type ext4 (rw,relatime,discard)
[root@sharevol /]# echo "Kubernetes is the best" > /tmp/xchange/data
[root@sharevol /]#
```

Now press ctrl+d to exit

Operation 5: 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:

```
kubect1 exec -it sharevol -c c2 -- bash
```

```
mount | grep /tmp/data
```

```
[root@sharevol /]# exit
ubuntu@ip-172-31-90-123:~$ kubect1 exec -it sharevol -c c2 -- bash
[root@sharevol /]# mount | grep /tmp/data
/dev/root on /tmp/data type ext4 (rw,relatime,discard)
[root@sharevol /]#
```

Now we are on /tmp/data, so to read the file from /tmp/xchange will use the following command

```
cat /tmp/data/data
```

```
[root@sharevol /]# cat /tmp/data/data
"Kubernetes is the best"
[root@sharevol /]#
```

So this is working and you can see how the data is being share across different containers inside a one pod and it's reachable for each of them.