

Kubernetes In 30 Days challenge :-

Day 9 :-

- ① Usage of commands & arguments in k8's pod (i.e. container),
- ② Kubernetes volume significance and various types.

commands & arguments in k8's Manifest file :-

- You can define commands and arguments in definition file using "command" field in the config. file.
- To define arguments we use "args" field in the file
- commands and arguments are cannot be modified once pods are created.

```
apiVersion: v1
kind: Pod
metadata:
  name: command-demo
  labels:
    purpose: demonstrate-command
spec:
  containers:
    - name: command-demo-container
      image: debian
      command: ["printenv"]
      args: ["HOSTNAME", "KUBERNETES_PORT"]
      restartPolicy: OnFailure
```

→ Explore above file, `printenv` cmd will take args as `HOSTNAME` and `KUBERNETES_PORT` and will be ran on container inside the pod.

→ Here you go for sequence of cmds to follow.

① create Pod def. file using above file.

② `kubectl create -f <pod-definition>.yaml`

③ `kubectl get pod`

here you'll get Pod as 'completed' status - which represents that command execution is done on container.

④ `kubectl logs pod command-demo`

Here you'll get the o/p something like this :

`command-demo`

`tcp://10.3.240.1:443`

(which are Hostname & KB port)

② Volumes in Kubernetes :-

- Pods are mortal and they tend to die at anytime. The data inside Pod is also lost if Pod is dead. Even if New Pod is up it is with fresh space.
- Sometimes we want to have shared files among containers in a Pod But it will also requires volume to enable that.

Both of These Problems will addressed by the concept of "volume"

- ① KB supports various types of volumes and Pod can have many number of volumes attached.
- ② Ephemeral volumes (Pod's own vol) are bind to Pod's lifecycle But persistent volume will be available outside the Pod's lifecycle.

Volume field in Manifest file will be specified in

- specs.volumes → volumes should be provided to pods.

- specs.containers(*). volumeMounts

- ↳ container path should be mentioned here

checkout below mentioned Manifest file for more details :

```
apiVersion: v1
kind: Pod
metadata:
  name: test-pd
spec:
  containers:
    - image: registry.k8s.io/test-webserver
      name: test-container
      volumeMounts:
        - mountPath: /test-pd
          name: test-volume
  volumes:
    - name: test-volume
      hostPath:
        # directory location on host
        path: /data
        # this field is optional
        type: Directory
```

→ This is an Example for hostPath volume type

Few Volume types of Kubernetes:

- ① azureFile CSI Migration
- ② configMap
- ③ emptyDir
- ④ hostPath
- ⑤ iSCSI
- ⑥ local
- ⑦ nfs
- ⑧ persistentVolumeClaim
- ⑨ secret etc...

That's all about volumes.