

Day - 9.

- 1> Usage of Commands and Arguments in Kubernetes pod (i.e container)
- 2> Kubernetes volume significance and various type.

→ 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 argument 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: ["prntenv"]

args: ["HOSTNAME", "KUBERNETES_PORT"]

restartPolicy: OnFailure

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

→ Here you go for sequence of commands to follow.

1) Create pod def. file using above file.

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

3) `kubectl get pod`

Here you will get pod as "Completed" status - which represents that command execution is done on container.

4) `kubectl logs pod command-demo.`

Here you will get the output something

like this:

Command-demo

`tcp://10.3.240.1:443`

(which are `hostname` & `kubernetes 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) Kubernetes 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` → Volume should be provided to pods.

- * `Specs.Containers(*)`, `VolumeMounts`

- ↳ Container path should be mentioned here.

→ checkout below mentioned manifest 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

Volume:

- name: test-volume

hostPath:

path: /data

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 Volume.