

Kubernetes In 30 Days challenge:-

#Day 6:-

- Read the caption for installation guide of minikube and make sure minikube & kubectl are installed in the Laptop.
- Run below cmds to confirm :

```
# minikube version
```

```
# kubectl version
```

① Info about PODs

② Basic kubectl cmds

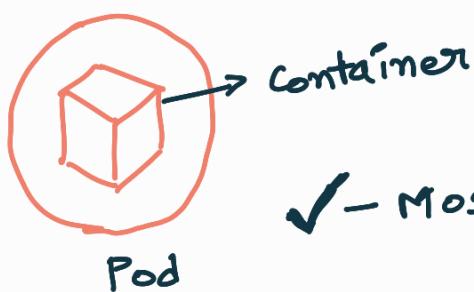
③ Running a test pod & Accessing

① Pod:- It is the basic unit of k8 cluster.

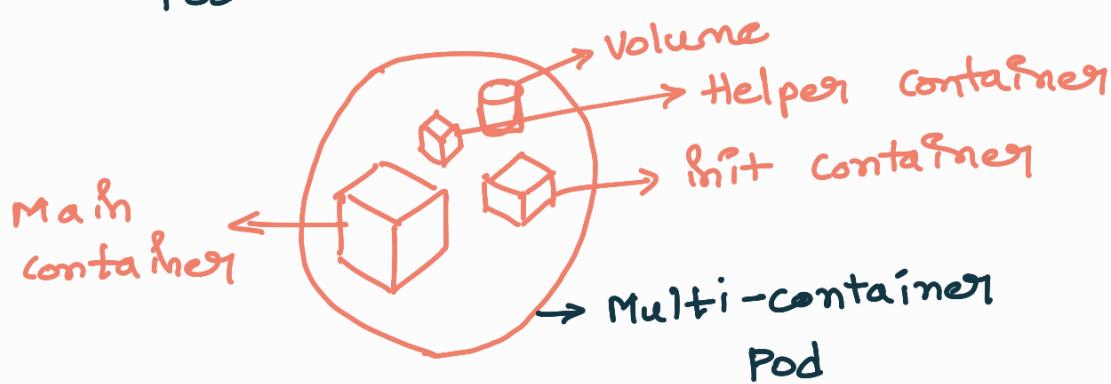
Pod represents a process running in the cluster.

→ one pod - one container is the most common use-case.

→ Pod is an Extra cover to container



✓ - Most use case → k8 manages the Pod rather than Container



→ These containers are tightly-coupled

→ To avail high availability horizontal
Pod scaling.

② Basic kubectl commands:-

```
# kubectl create -f <definition-file>.yaml
```

```
# kubectl get pods
```

```
# kubectl get nodes
```

```
# kubectl get svc
```

```
# kubectl get all --all
```

```
# kubectl get ns
```

```
# kubectl describe pod <pod-name>
```

```
# kubectl get pod <pod-name> -o yaml
```

```
# kubectl get pod -o wide
```

```
# kubectl logs <pod-name>
```

```
# kubectl run <pod-name> --image=nginx
```

```
# kubectl apply -f <updated-def-file>.yaml
```

Now Follow the below steps to test minikube

cluster :-

① step1 - start minikube cluster

```
$ minikube start
* minikube v1.32.0 on Microsoft Windows 11 Home Single Language 10.0.22631.2861 Build 22631.2861
* Automatically selected the virtualbox driver
* Starting control plane node minikube in cluster minikube
* Creating virtualbox VM (CPUs=2, Memory=2200MB, Disk=20000MB) ...
! This VM is having trouble accessing https://registry.k8s.io
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

② create a deployment using below command :

```
PS C:\WINDOWS\system32> kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
deployment.apps/hello-minikube created
PS C:\WINDOWS\system32> kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
PS C:\WINDOWS\system32> kubectl get po -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS   AGE
default       hello-minikube-7f54cff968-zjpjr   1/1     Running   0          17s
kube-system   coredns-5dd5756b68-vlm4h         1/1     Running   0          69s
kube-system   etcd-minikube                     1/1     Running   0          80s
kube-system   kube-apiserver-minikube          1/1     Running   0          83s
kube-system   kube-controller-manager-minikube  1/1     Running   0          80s
kube-system   kube-proxy-x9hh5                  1/1     Running   0          69s
kube-system   kube-scheduler-minikube          1/1     Running   0          82s
kube-system   storage-provisioner             1/1     Running   0          76s
PS C:\WINDOWS\system32> kubectl get po
NAME           READY   STATUS    RESTARTS   AGE
hello-minikube-7f54cff968-zjpjr   1/1     Running   0          21s
PS C:\WINDOWS\system32> kubectl get sv
error: the server doesn't have a resource type "sv"
PS C:\WINDOWS\system32> kubectl get svc
NAME        TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)        AGE
hello-minikube  NodePort  10.98.134.87  <none>        8080:30859/TCP  39s
kubernetes  ClusterIP  10.96.0.1     <none>        443/TCP       105s
```

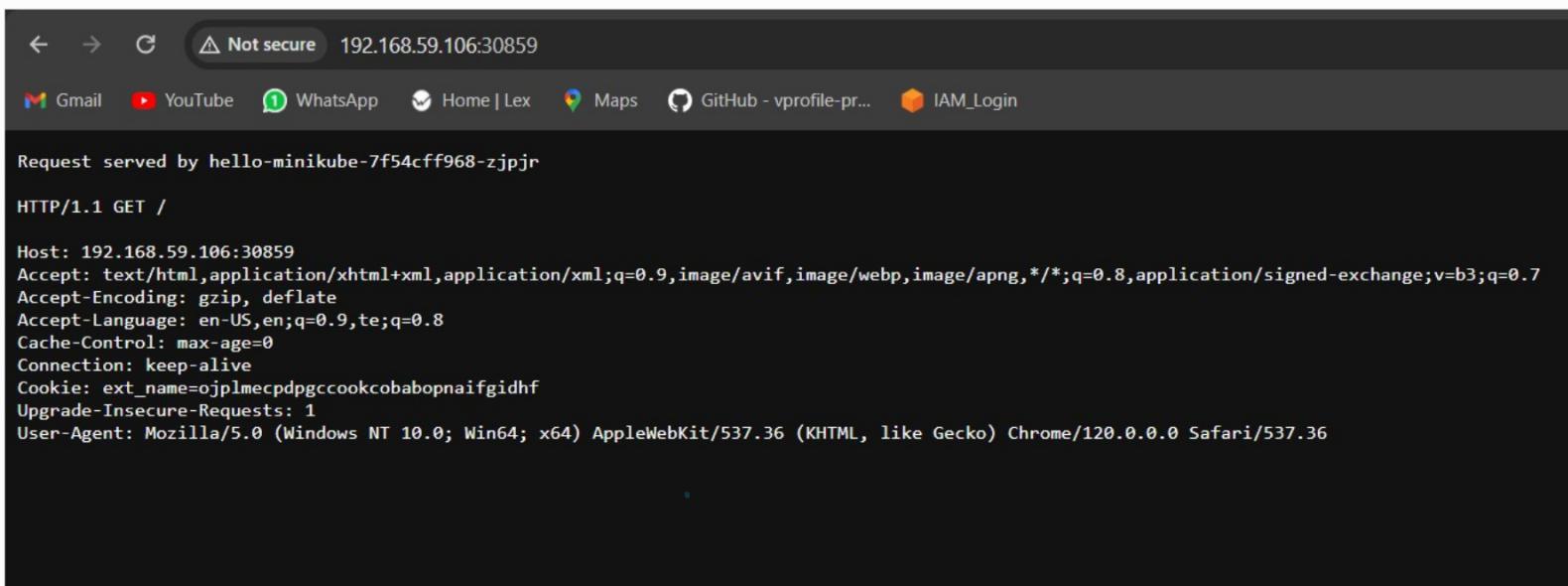
Apply followed cmd to expose Node Port to access URL.

`http://<Node-IP>:<Node Port>`

③ To find the NodePort , run the below cmd . checkout the below s.s.

```
PS C:\WINDOWS\system32> kubectl describe svc hello-minikube
Name:           hello-minikube
Namespace:      default
Labels:         app=hello-minikube
Annotations:    <none>
Selector:       app=hello-minikube
Type:          NodePort
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.98.134.87
IPs:           10.98.134.87
Port:          <unset>  8080/TCP
TargetPort:    8080/TCP
NodePort:      <unset>  30859/TCP
Endpoints:     10.244.0.3:8080
Session Affinity: None
External Traffic Policy: Cluster
Events:        <none>
PS C:\WINDOWS\system32> minikube ip
192.168.59.106
```

④ Now Try to access the page using
`http://<Node-IP>:<NodePort>`

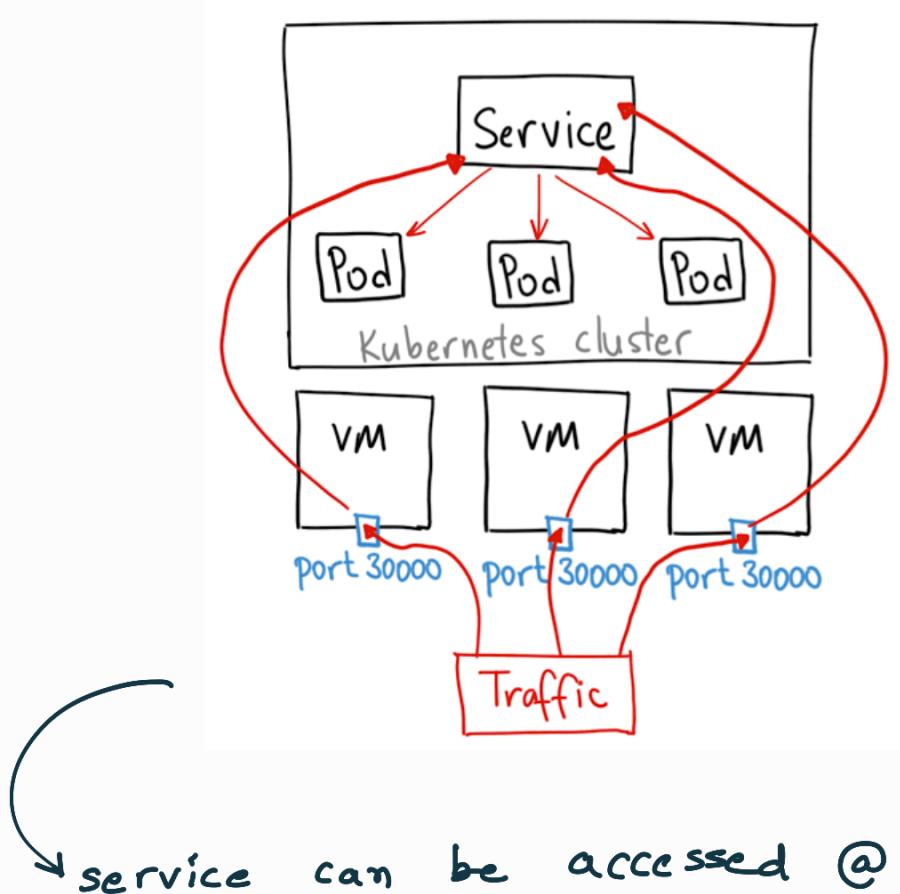


⑤ Validation is successful. Now cleanup the all resources using below cmd:

```
PS C:\WINDOWS\system32> minikube delete --all
* Deleting "minikube" in virtualbox ...
* Removed all traces of the "minikube" cluster.
* Successfully deleted all profiles
```

Bonus :-

NodePort working



Any NodeIP: 30000

This is similar to Docker Networking.
NodePort is one of 3 services of K8s.