KUBERNETES ASSIGNMENT

Exercise 1: Deploy an Nginx Pod

Objective: Deploy a simple Nginx pod and access it.

1. Start a Kubernetes cluster (Minikube or other cluster):

```
master@master-vm:~$ curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  119M  100  119M    0     0  13.8M      0  0:00:08  0:00:08 --:--:-- 18.0M
master@master-vm:~$ chmod +x minikube
master@master-vm:~$ sudo mv minikube /usr/local/bin/
master@master-vm:~$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
master@master-vm:~$ sudo snap install minikube --classic
Warning: flag --classic ignored for strictly confined snap minikube

minikube 0.8.0 from Felix Winterhalter (blackclaws) installed
```

2. Create an Nginx pod:

```
master@master-vm: ~
master@master-vm:~$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
master@master-vm:~$ minikube start
😄  minikube v1.35.0 on Ubuntu 20.04
✨  Unable to pick a default driver. Here is what was considered, in preference order:
    ▪ docker: Not healthy: "docker version --format {{.Server.Os}}-{{.Server.Version}}:{{.Server.Platform.Name}}" exit status 1: p
ermission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdo
cker.sock/v1.45/version": dial unix /var/run/docker.sock: connect: permission denied
    ▪ docker: Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker' <https://docs.dock
er.com/engine/install/linux-postinstall/>
💡  Alternatively you could install one of these drivers:
    ▪ kvm2: Not installed: exec: "virsh": executable file not found in $PATH
    ▪ podman: Not installed: exec: "podman": executable file not found in $PATH
    ▪ qemu2: Not installed: exec: "qemu-system-x86_64": executable file not found in $PATH
    ▪ virtualbox: Not installed: unable to find VBoxManage in $PATH

❌  Exiting due to DRV_NOT_HEALTHY: Found driver(s) but none were healthy. See above for suggestions how to fix installed drivers.

master@master-vm:~$ kubectl run nginx-pod --image=nginx --restart=Never
pod/nginx-pod created
```

3. Verify the pod is running:

```
master@master-vm:~$ kubectl get pods
NAME                     READY   STATUS    RESTARTS   AGE
nginx-5869d7778c-g5r4s   1/1     Running   0          2m11s
nginx-pod                1/1     Running   0          72s
```

4. Check pod details:

```
master@master-vm:~$ kubectl describe pod nginx-pod
Name:             nginx-pod
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Thu, 13 Mar 2025 12:38:21 +0530
Labels:           run=nginx-pod
Annotations:      <none>
Status:           Pending
IP:               10.244.0.4
IPs:
  IP:  10.244.0.4
Containers:
  nginx-pod:
    Container ID:
    Image:          nginx
    Image ID:
    Port:           <none>
    Host Port:      <none>
    State:          Waiting
      Reason:       ErrImagePull
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-v5x2p (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       False
```

5. Delete the pod:

```
master@master-vm:~$ kubectl delete pod nginx-pod
pod "nginx-pod" deleted
```

Exercise 2: Create a Nginx Deployment and scale it.

Objective: Create a Nginx deployment and scale it up.

Step 1: **Create a deployment with Nginx:**

```
master@master-vm:~$ kubectl create deployment nginx-deployment --image=nginx
deployment.apps/nginx-deployment created
```

Step 2: Check the deployment:

```
master@master-vm:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   1/1     1            1           25s
```

Step 3: Scale the deployment to 3 replicas:

```
master@master-vm:~$ kubectl scale deployment nginx-deployment --replicas=3
deployment.apps/nginx-deployment scaled
```

Step 4: Check the running pods:

```
master@master-vm:~$ kubectl get pods -o wide
NAME                               READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
nginx-deployment-6cfb98644c-lm2rw  1/1     Running   0          4m3s  10.244.0.21  minikube   <none>           <none>
nginx-deployment-6cfb98644c-qhscg  1/1     Running   0          2m7s  10.244.0.22  minikube   <none>           <none>
nginx-deployment-6cfb98644c-rmwfp  1/1     Running   0          2m7s  10.244.0.23  minikube   <none>           <none>
```

Step 5: Delete the deployment:

```
master@master-vm:~$ kubectl delete deployment nginx-deployment
deployment.apps "nginx-deployment" deleted
```



```
master@master-vm:~$ kubectl delete configmap app-config
configmap "app-config" deleted
```