Assignment 5

Name: Karanpreet Singh

Student Id: CT_CSI_DV_2394

Domain: DevOps

College Name: Chandigarh University

Date of Performance: 16-06-25 to 22-06-25

Stream: B.tech

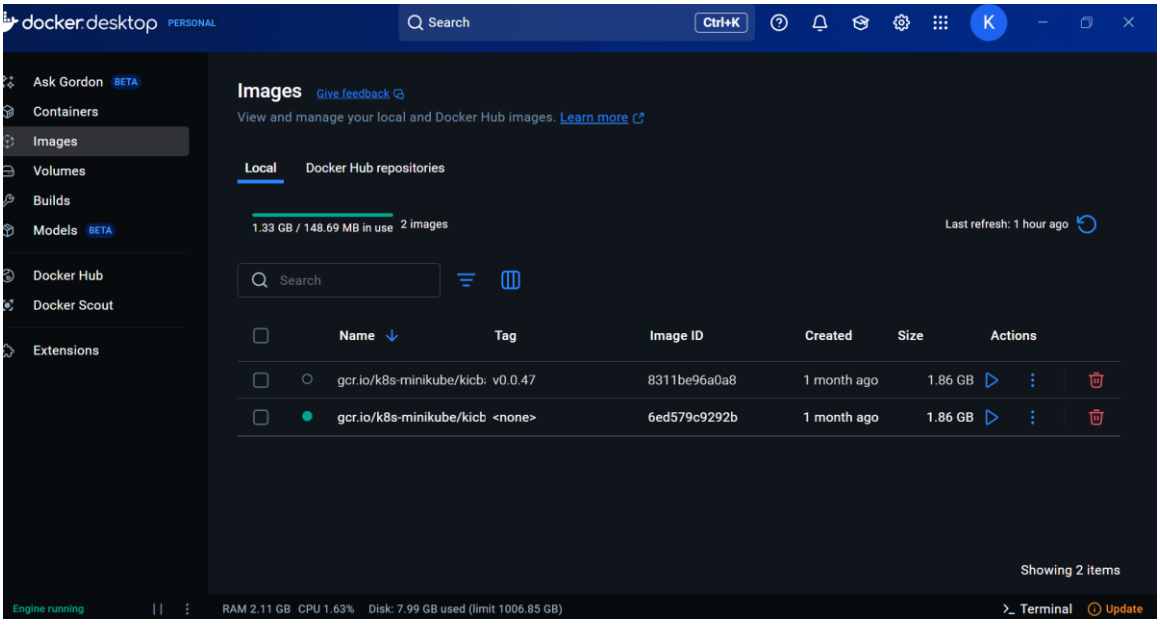
1. Create a Kubernetes cluster using minikube

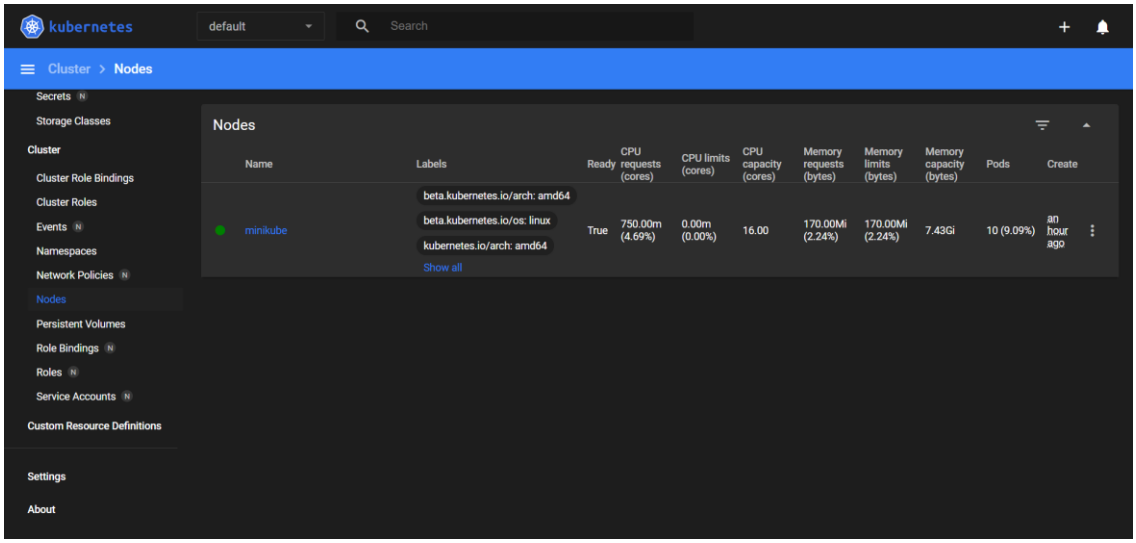
Start the Minikube cluster

minikube start --driver=docker

```
PS D:\celebal\assignment 5> minikube start --driver=docker
* minikube v1.36.0 on Microsoft Windows 11 Home Single Language 10.0.26100.4351 Build 26100.4351
* Using the docker driver based on user configuration
* Using Docker Desktop driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.47 ...
    > gcr.io/k8s-minikube/kicbase...:  502.26 MiB / 502.26 MiB  100.00% 11.31 M
* Creating docker container (CPUs=2, Memory=3900MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking
/proxy/
* Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS D:\celebal\assignment 5> kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   57s   v1.33.1
PS D:\celebal\assignment 5> minikube dashboard
* Enabling dashboard ...
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
* Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server
```





Create a deployment

kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0

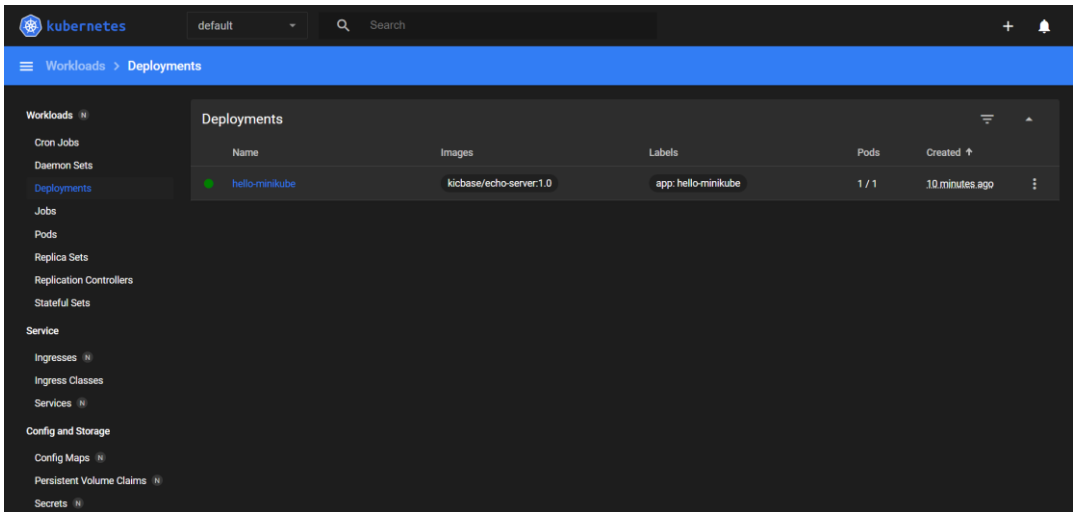
Expose the deployment as a service

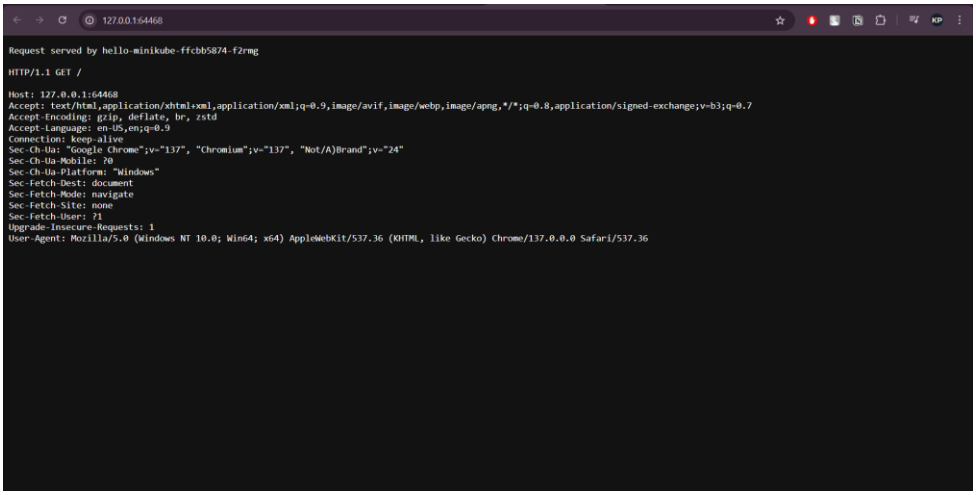
kubectl expose deployment hello-minikube --type=NodePort --port=8080

Open the app in browser

minikube service hello-minikube

```
PS D:\celebal\assignment 5> kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
deployment.apps/hello-minikube created
PS D:\celebal\assignment 5> kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
PS D:\celebal\assignment 5> minikube service hello-minikube
|-----------|----------------|-------------|---------------------------|
| NAMESPACE |      NAME      | TARGET PORT |            URL            |
|-----------|----------------|-------------|---------------------------|
| default   | hello-minikube |        8080 | http://192.168.49.2:30661 |
|-----------|----------------|-------------|---------------------------|
* Starting tunnel for service hello-minikube.
|-----------|----------------|-------------|------------------------|
| NAMESPACE |      NAME      | TARGET PORT |          URL           |
|-----------|----------------|-------------|------------------------|
| default   | hello-minikube |             | http://127.0.0.1:64468 |
|-----------|----------------|-------------|------------------------|
* Opening service default/hello-minikube in default browser...
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```





2. Create a Kubernetes cluster using kubeadm

AWS Security Group Configuration

Created a new Security Group (e.g., Kubernetes-Cluster-SG)

Allowed SSH (Port 22) and Kubernetes API access (Port 6443)

Attached this Security Group to all EC2 instances during launch

Node Preparation (Both Master & Workers)

Disabled Swap using sudo swapoff -a

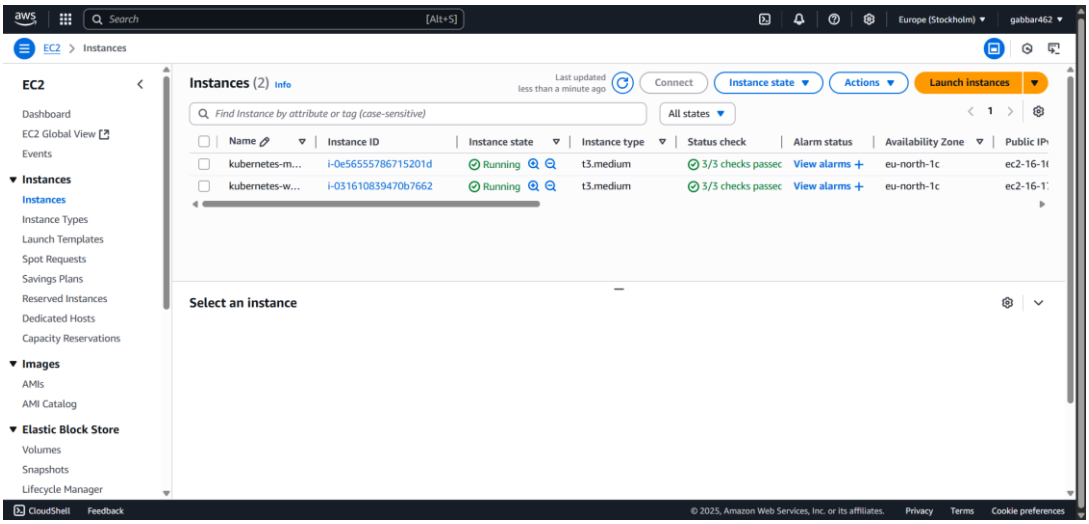
Enabled Kernel Modules and applied sysctl parameters for networking

Installed containerd as the container runtime

Configured containerd to use SystemdCgroup and set appropriate sandbox image

Installed Kubernetes components: kubelet, kubeadm, and kubectl

Marked them on hold to prevent version changes



Master Node Configuration

Initialized the cluster using sudo kubeadm init

```
ubuntu@ip-172-31-3-67:~$ sudo kubeadm init
I0928 12:07:49.750337    9073 version.go:251] remote version is much newer: v1.28.2; falling back to: st
able-1.20
[init] Using Kubernetes version: v1.20.15
[preflight] Running pre-flight checks
        [WARNING SystemVerification]: this Docker version is not on the list of validated versions: 24.0
.5. Latest validated version: 19.03
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
```

Set up the kubeconfig file for cluster administration

Installed Calico as the network plugin

Generated the kubeadm join command for worker nodes

```
ubuntu@ip-172-31-3-67:~$ mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-3-67:~$ kubectl get nodes
NAME              STATUS     ROLES                  AGE   VERSION
ip-172-31-3-67    NotReady   control-plane,master   73s   v1.20.0
ubuntu@ip-172-31-3-67:~$
```

Worker Node Configuration

Reset pre-flight checks with sudo kubeadm reset

Used the generated join command from the master node to join the cluster

Appended --cri-socket unix:///run/containerd/containerd.sock --v=5 for verbose output

```
ubuntu@ip-172-31-3-67:~$ kubectl apply -f https://github.com/weaveworks/weave
eave-daemonset-k8s.yaml
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.apps/weave-net created
ubuntu@ip-172-31-3-67:~$ sudo kubeadm token create --print-join-command
kubeadm join 172.31.3.67:6443 --token 4xjvuk.df8dlaue24ghad5l     --discovery
d629ebb300ed7533b8dd033c8532b9b15b0c1440265ada6be3c4c07036b453ff
ubuntu@ip-172-31-3-67:~$
```

Cluster Verification

Verified all nodes using kubectl get nodes from the master node

Ensured worker nodes were connected and running correctly

```
ubuntu@ip-172-31-3-67:~$ kubectl get nodes
NAME              STATUS   ROLES                  AGE     VERSION
ip-172-31-3-67    Ready    control-plane,master   5m44s   v1.20.0
ip-172-31-8-38    Ready    <none>                 25s     v1.20.0
ubuntu@ip-172-31-3-67:~$
```

```
ubuntu@ip-172-31-8-38:~$ sudo docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED              STATUS              PORTS     NA
MES
ba9a66e987ad   weaveworks/weave-npc   "/usr/bin/launch.sh"     About a minute ago   Up About a minute             k8
s_weave-npc_weave-net-f2mgq_kube-system_af00e0de-6f76-4c71-ad76-505c32c2c862_0
4bc5188b5723   weaveworks/weave-kube  "/home/weave/launch.…"   About a minute ago   Up About a minute             k8
s_weave_weave-net-f2mgq_kube-system_af00e0de-6f76-4c71-ad76-505c32c2c862_0
23134ec903f6   k8s.gcr.io/kube-proxy  "/usr/local/bin/kube…"   About a minute ago   Up About a minute             k8
s_kube-proxy_kube-proxy-9h4t2_kube-system_b11fb14c-853c-470e-8579-28b0670f6b0b_0
e833bcdd3232   k8s.gcr.io/pause:3.2   "/pause"                 About a minute ago   Up About a minute             k8
s_POD_weave-net-f2mgq_kube-system_af00e0de-6f76-4c71-ad76-505c32c2c862_0
fd106a863a02   k8s.gcr.io/pause:3.2   "/pause"                 About a minute ago   Up About a minute             k8
s_POD_kube-proxy-9h4t2_kube-system_b11fb14c-853c-470e-8579-28b0670f6b0b_0
ubuntu@ip-172-31-8-38:~$
```

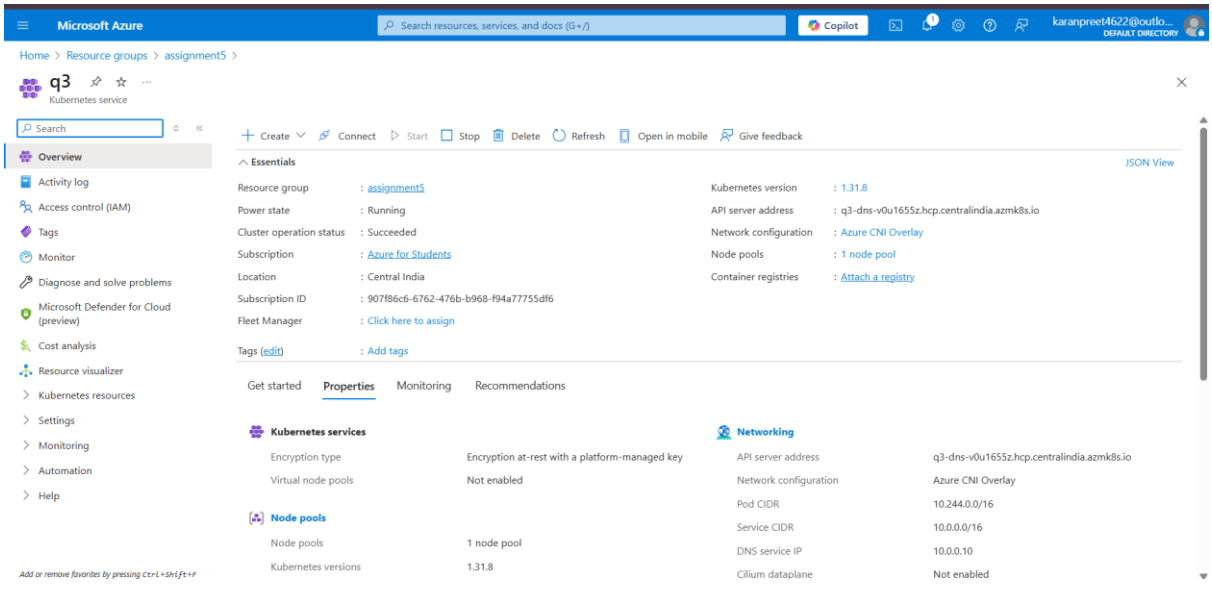
3. **Deploy an AKS cluster using the portal. Access the dashboard and create roles for multiple users**

   AKS Cluster Creation:

   Navigated to Kubernetes Services in the Azure Portal.

   Created a new cluster named q3 in the resource group assignment5.

   Chose default node pool settings and enabled RBAC for user role management.



   Cluster Access Configuration:

   Installed Azure CLI and kubectl locally.

   Executed the following command to connect to the AKS cluster:

   az aks get-credentials --resource-group assignment5 --name q3

```
PS D:\celebal\assignment 5> az aks get-credentials --resource-group assignment5 --name q3
Merged "q3" as current context in C:\Users\hp\.kube\config
PS D:\celebal\assignment 5> kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/re
commended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
PS D:\celebal\assignment 5> kubectl create serviceaccount dashboard-admin-sa
serviceaccount/dashboard-admin-sa created
PS D:\celebal\assignment 5> kubectl create clusterrolebinding dashboard-admin-sa --clusterrole=cluster-admin --serviceac
count=default:dashboard-admin-sa
clusterrolebinding.rbac.authorization.k8s.io/dashboard-admin-sa created
```

Kubernetes Dashboard Deployment:

Deployed the Kubernetes Dashboard using:

kubectl apply -f

https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml

Dashboard Access Configuration:

Created a service account and assigned it admin privileges:

kubectl create serviceaccount dashboard-admin-sa

kubectl create clusterrolebinding dashboard-admin-sa --clusterrole=cluster-admin --serviceaccount=default:dashboard-admin-sa

Generated a token for dashboard login:

kubectl create token dashboard-admin-sa

Started the Kubernetes proxy using:

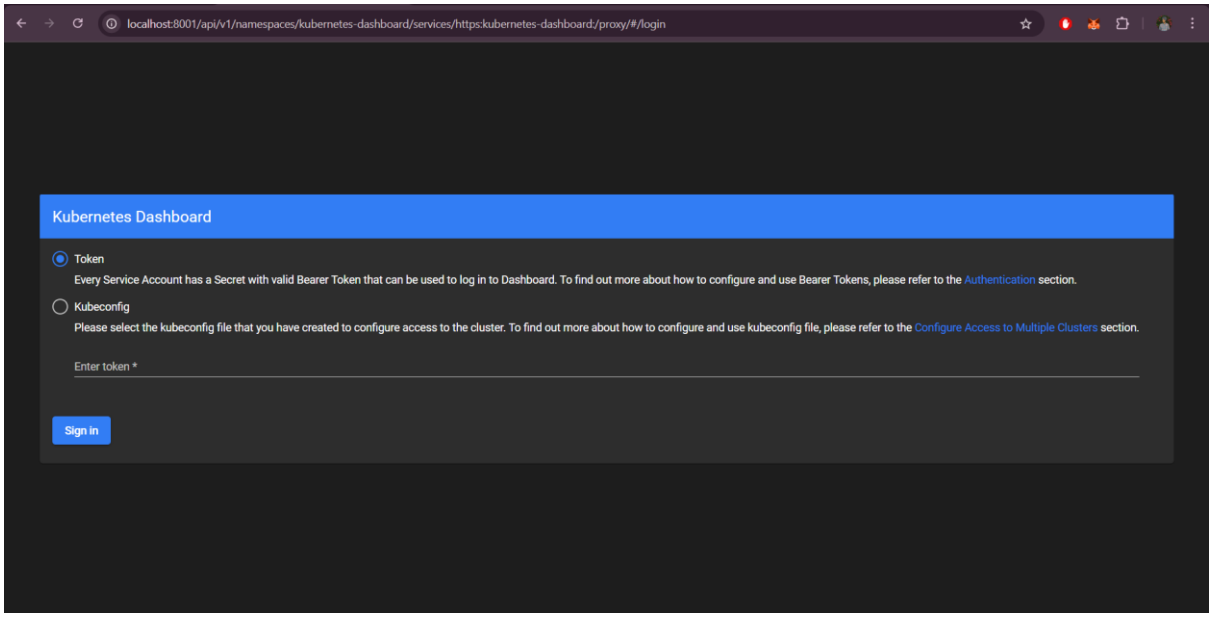
kubectl proxy

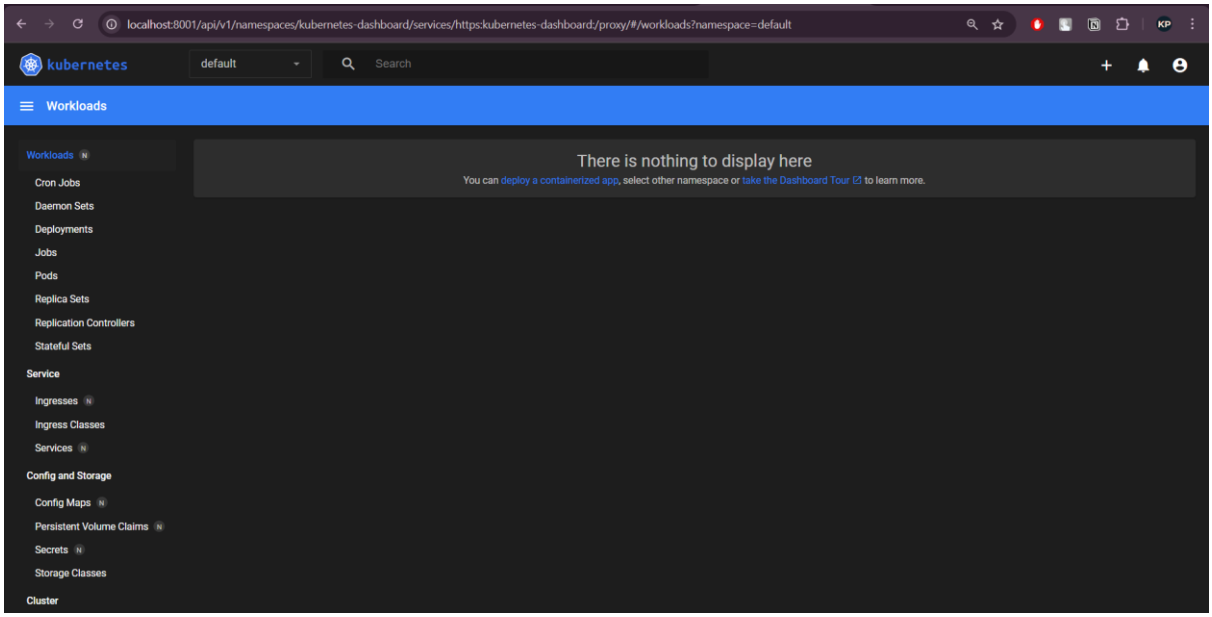
```
PS D:\celebal\assignment 5> kubectl create token dashboard-admin-sa
eyJhbGciOiJSUzI1NiIsImtpZCI6InFiTkVPREJBeGUtZ2NrYUxadXM0NlpjcEY1MURDcTg1azZqRjM0bk40eFUifQ.eyJhdWQiOlsiaHR0cHM6Ly9jZW50c
mFsaW5kaWEub2ljLnByb2QtYWtzLmF6dXJlLmNvbS8zZmE3Yzk3Ni1mMzY1LTRkZWItOTI0Yy1mOWJmNjdmNDYyZjIvN2RmMTQ3YTUtYjcwZi00ZTNkLThlY
zctNzgwNmZmNDNkY2E4LyIsImh0dHBzOi8vcTMtZG5zLXYwdTE2NTV6LmhjcC5jZW50cmFsaW5kaWEuYXptazhzLmlvIiwiXCJxMy1kbnMtdjB1MTY1NXoua
GNwLmNlbnRyYWxpbmRpYS5hem1rOHMuaW9cIiJdLCJleHAiOjE3NTA1NDU1MzIsImlhdCI6MTc1MDU0MTkzMiwiaXNzIjoiaHR0cHM6Ly9jZW50cmFsaW5ka
WEub2ljLnByb2QtYWtzLmF6dXJlLmNvbS8zZmE3Yzk3Ni1mMzY1LTRkZWItOTI0Yy1mOWJmNjdmNDYyZjIvN2RmMTQ3YTUtYjcwZi00ZTNkLThlYzctNzgwN
mZmNDNkY2E4LyIsImp0aSI6IjlhMTdmMDM3LWM0OWItNDhlYS04NjYzLTQzY2VkYTQxYjFmYSIsImt1YmVybmV0ZXMuaW8iOnsibmFtZXNwYWNlIjoiZGVmY
XVsdCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJkYXNoYm9hcmQtYWRtaW4tc2EiLCJ1aWQiOiJiZGZkMDMyOS1mZTA4LTRlMDMtYTYyOS03YzFjNTM4N
2JlZTUifX0sIm5iZiI6MTc1MDU0MTkzMiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6ZGFzaGJvYXJkLWFkbWluLXNhIn0.VAFkT84O0
QesrAIUiYBZQbfUpB2tT2sUpfwAA2uXHVjjFpTpNSoyl1TEXWhoP9ocXaShY39nijiEfTOqqmt-IwgjeoedwYaVPfJnT7zahQ7Si7F4_kRb_zlmb621HoOW2
3edZ8Simir8ruq0hcCfByCj_40MLdaK0ve12rL0PP4S3RbnSoYLGyhywJrJO6zA7KTmb9htIPiw3RLrcTP3kyMHZnpPvJqOVCQVKuqNcPdc8K_2XCFrdGQx5
A-Upbia3TqUoOlrrfqczMAZHnrXItUtLtvdGtJU63nETPYtyVSD9XeKGodlgFX887wi2JhLrNA_vGLq3Exn2p-pNkdgJd627BrZpKOrOG4VaJIJZMq_1Iwyu
_fDkYeIjOy0g34VhMgbTsl5W3CXehC_jUfEiakO9du06dleKLVEHu9KhDX5XRt6kESIZ2K3lEOiaGg5hhqksbESL-j_3sCWyGiqcjdLJV3jtJeyeSgU4pAEJ
6TVkYm5DixTOl_oKLP4aZXMxNq4nHhP9TLjAj0gh1jPQWCeYJ-xkRtYOEsNtwIy_ajPikwBV4nJD869axEUkWfh6zPeZ2OO3O1w12mKX2VgT1ajW3jprTFEz
9Wd6yBcJbNrOmQCTe7B-PXj82r6qk1AHpjf4jDffi1-b_px9givHs0JeCAJ1x14d9ih93fRwkw
PS D:\celebal\assignment 5> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

Accessed the dashboard via:

http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/

Used the generated token to log in successfully.





RBAC for Multiple Users:

Created and applied RBAC roles to manage permissions for different users using YAML files (e.g., Role, RoleBinding, ClusterRole, and ClusterRoleBinding) depending on namespace-level or cluster-wide access requirements.

4. **Deploy a microservice application on AKS cluster and access it using public internet**

Created a Resource Group:

az group create --name assignment5 --location centralindia

Provisioned an AKS Cluster:

az aks create --resource-group assignment5 --name aks-cluster --node-count 1 --enable-addons monitoring --generate-ssh-keys

```
PS D:\celebal\assignment 5> az group create --name assignment5 --location centralindia
{
  "id": "/subscriptions/907f86c6-6762-476b-b968-f94a77755df6/resourceGroups/assignment5",
  "location": "centralindia",
  "managedBy": null,
  "name": "assignment5",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
PS D:\celebal\assignment 5> az aks create --resource-group assignment5 --name aks-cluster --node-count 1 --ena
ble-addons monitoring --generate-ssh-keys
{
  "aadProfile": null,
  "addonProfiles": {
    "omsagent": {
      "config": {
        "logAnalyticsWorkspaceResourceID": "/subscriptions/907f86c6-6762-476b-b968-f94a77755df6/resourceGroups
/DefaultResourceGroup-CID/providers/Microsoft.OperationalInsights/workspaces/DefaultWorkspace-907f86c6-6762-47
6b-b968-f94a77755df6-CID",
        "useAADAuth": "true"
      },
      "enabled": true,
      "identity": null
    }
```

Connected to the AKS Cluster:

az aks get-credentials --resource-group assignment5 --name aks-cluster

```
PS D:\celebal\assignment 5> az aks get-credentials --resource-group assignment5 --name aks-cluster
Merged "aks-cluster" as current context in C:\Users\hp\.kube\config
PS D:\celebal\assignment 5> kubectl get nodes
NAME                                STATUS   ROLES    AGE     VERSION
aks-nodepool1-28661980-vmss000000   Ready    <none>   3m55s   v1.31.8
```

Deployed the Microservice:

Created a deployment.yaml using a valid Docker image (e.g., nginx).

Applied it using:

kubectl apply -f deployment.yaml

Exposed the Service Publicly:

Created a service.yaml of type LoadBalancer.

Applied it using:

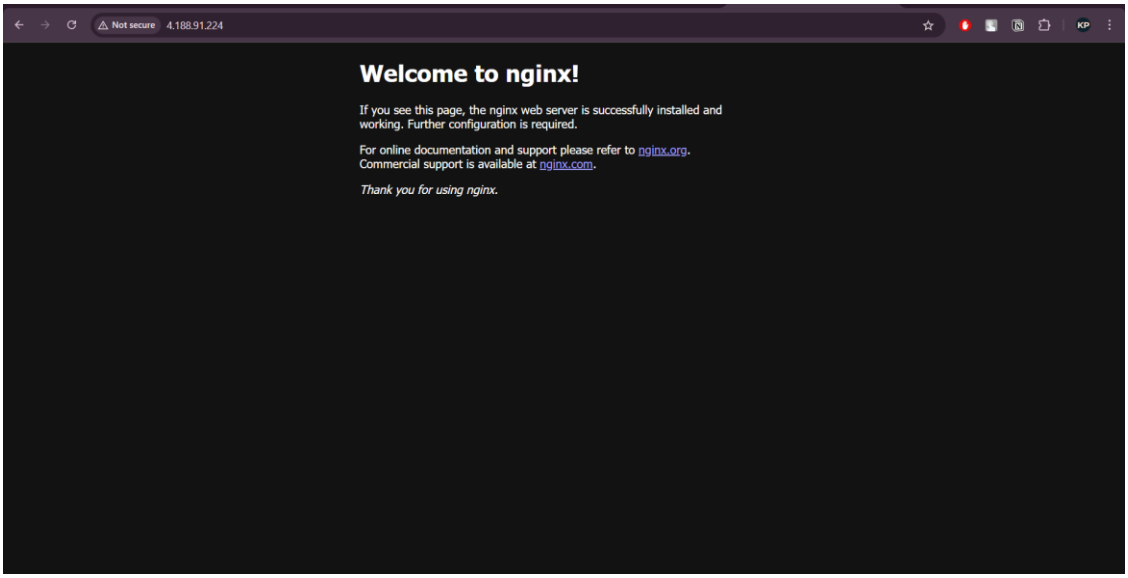
kubectl apply -f service.yaml

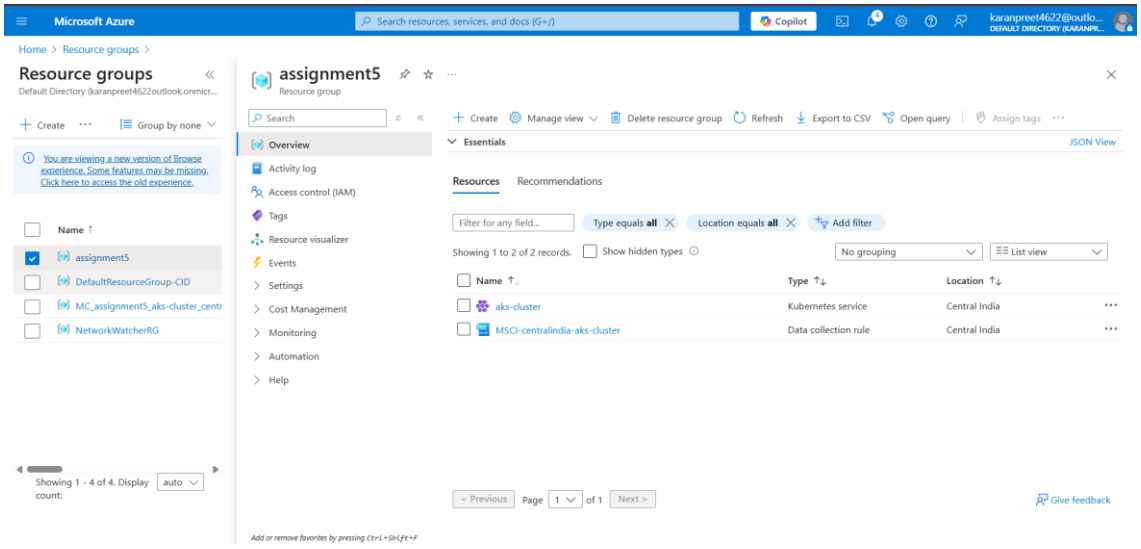
```
PS D:\celebal\assignment 5\q4> kubectl apply -f deployment.yaml
deployment.apps/test-nginx created
PS D:\celebal\assignment 5\q4> kubectl apply -f service.yaml
service/test-service created
PS D:\celebal\assignment 5\q4> kubectl get service test-service
NAME           TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
test-service   LoadBalancer   10.0.7.23     <pending>     80:32456/TCP   6s
PS D:\celebal\assignment 5\q4> kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
microservice-app-75fccb5597-fxvfh   0/1     InvalidImageName   0          5m41s
microservice-app-75fccb5597-jp6lc   0/1     InvalidImageName   0          5m41s
test-nginx-64cfc656b4-csf7g         1/1     Running            0          30s
PS D:\celebal\assignment 5\q4> kubectl get service test-service
NAME           TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)        AGE
test-service   LoadBalancer   10.0.7.23     4.188.91.224   80:32456/TCP   31s
PS D:\celebal\assignment 5\q4>
```

Accessed the Application:

Retrieved the public IP using:

kubectl get service





5. **Expose services in the cluster with node port, cluster IP, load balancer**

Deployment Creation

A deployment named my-app was created using the NGINX image. It ensures that multiple replicas of the pod are running and managed.

ClusterIP Service

This service exposes the app internally within the cluster. It allows pod-to-pod communication.

```
PS D:\celebal\assignment 5\q5> kubectl apply -f deployment.yaml
deployment.apps/my-app created
PS D:\celebal\assignment 5\q5> kubectl get pods
NAME                                READY   STATUS             RESTARTS   AGE
microservice-app-75fccb5597-fxvfh   0/1     InvalidImageName   0          8m40s
microservice-app-75fccb5597-jp6lc   0/1     InvalidImageName   0          8m40s
my-app-86d5bc587d-vsh2r             1/1     Running            0          5s
my-app-86d5bc587d-wfw7h             1/1     Running            0          5s
test-nginx-64cfc656b4-csf7g         1/1     Running            0          3m29s
PS D:\celebal\assignment 5\q5> kubectl apply -f clusterip-service.yaml
service/my-clusterip-service created
PS D:\celebal\assignment 5\q5> kubectl get svc my-clusterip-service
NAME                  TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
my-clusterip-service  ClusterIP   10.0.130.123   <none>        80/TCP    4s
```

NodePort Service

This exposes the application **externally via a port on the node IP**. It maps an internal port to a port accessible via Minikube IP.

```
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)
kubernetes     ClusterIP   10.96.0.1       <none>        443/TCP
h
nodeport-svc   NodePort    10.96.113.194   <none>        80:30001/TCP
```

LoadBalancer Service

This service is meant to expose the application using a cloud provider-managed external IP. Since I used Minikube, I simulated a LoadBalancer by running: minikube tunnel

```
PS D:\celebal\assignment 5\q5> kubectl apply -f loadbalancer-service.yaml
service/my-loadbalancer-service created
PS D:\celebal\assignment 5\q5> minikube tunnel
* Tunnel successfully started

* NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessi
ble ...

! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more inform
ation, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-requ
ires-root-permission
* Starting tunnel for service my-loadbalancer-service.
```

```
PS D:\celebal\assignment 5\q5> kubectl get svc my-loadbalancer-service
NAME                     TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
my-loadbalancer-service  LoadBalancer   10.102.65.10   127.0.0.1     80:32220/TCP   24s
```