**Kubernetes assignments**

Assignment1: Deploy a Kubernetes Cluster for 3 nodes ● Create a nginx deployment of 3 replicas

Assignment1 solution:

\*\*Launching an ec2 instance with t2.medium Instance type

sudo apt update

sudo apt install docker.io –y

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube\_latest\_amd64.deb

sudo dpkg -i minikube\_latest\_amd64.deb

minikube kubectl get no

apiVersion: apps/v1

kind: Deployment

metadata:

name: assignment1-deployment

labels:

app: nginx

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx

ports:

- containerPort: 80

minikube status

minikube start

sudo chmod 777 /var/run/docker.sock

minikube start

minikube kubectl get nodes

minikube kubectl get services

minikube kubectl get pods

sudo snap install kubectl –classic

kubectl get svc

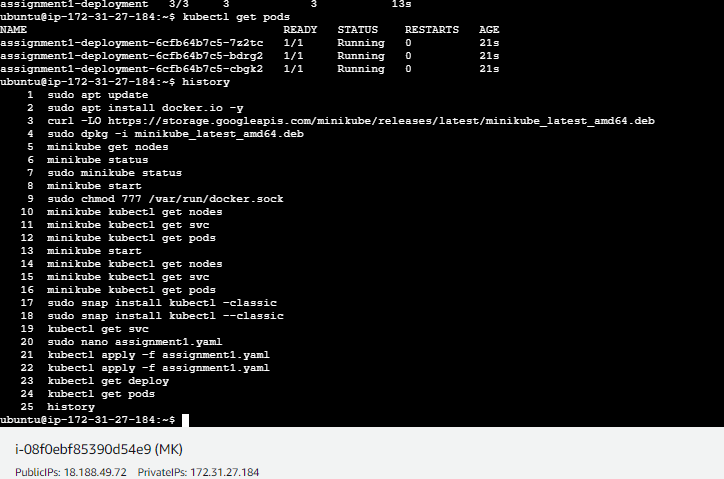
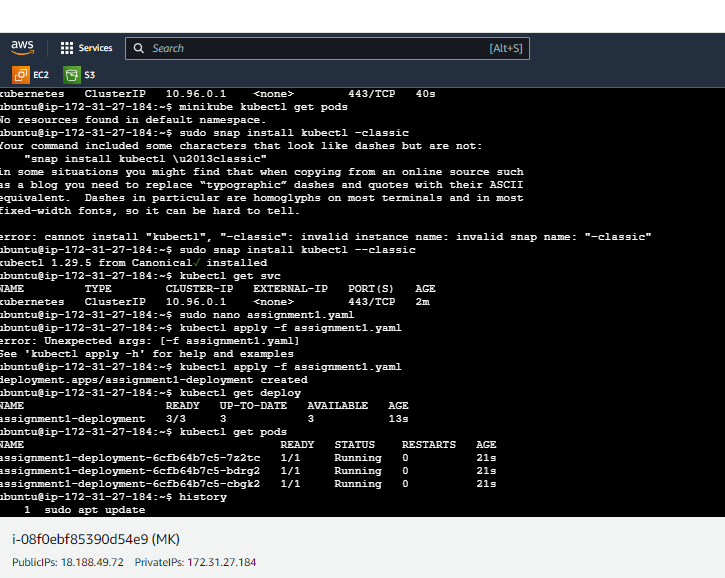
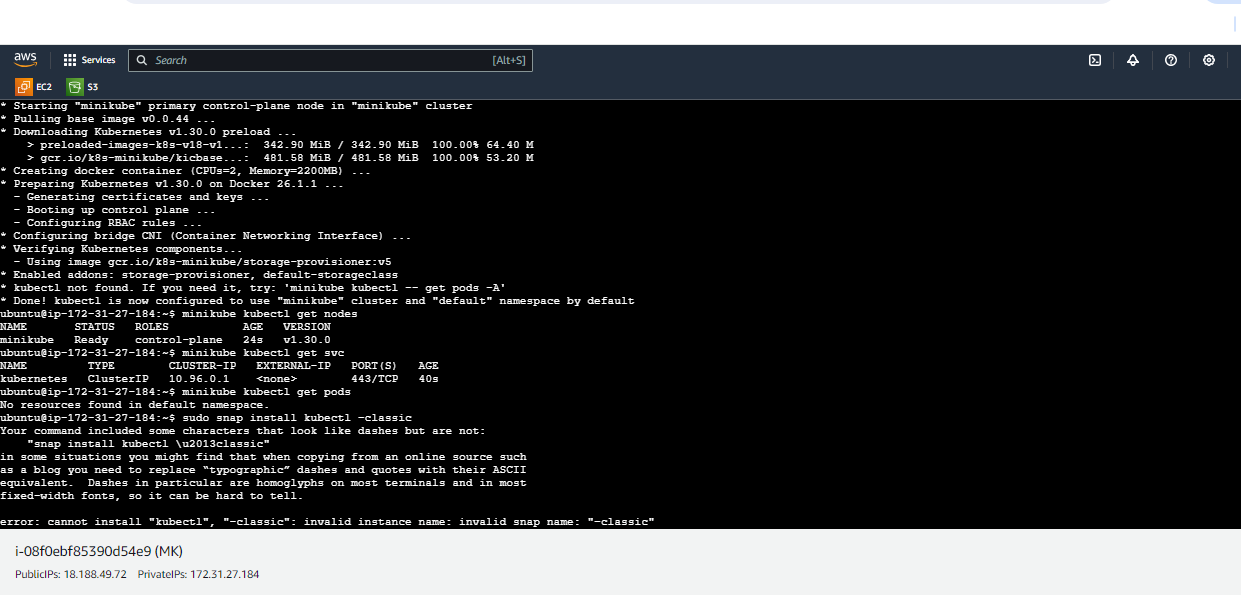
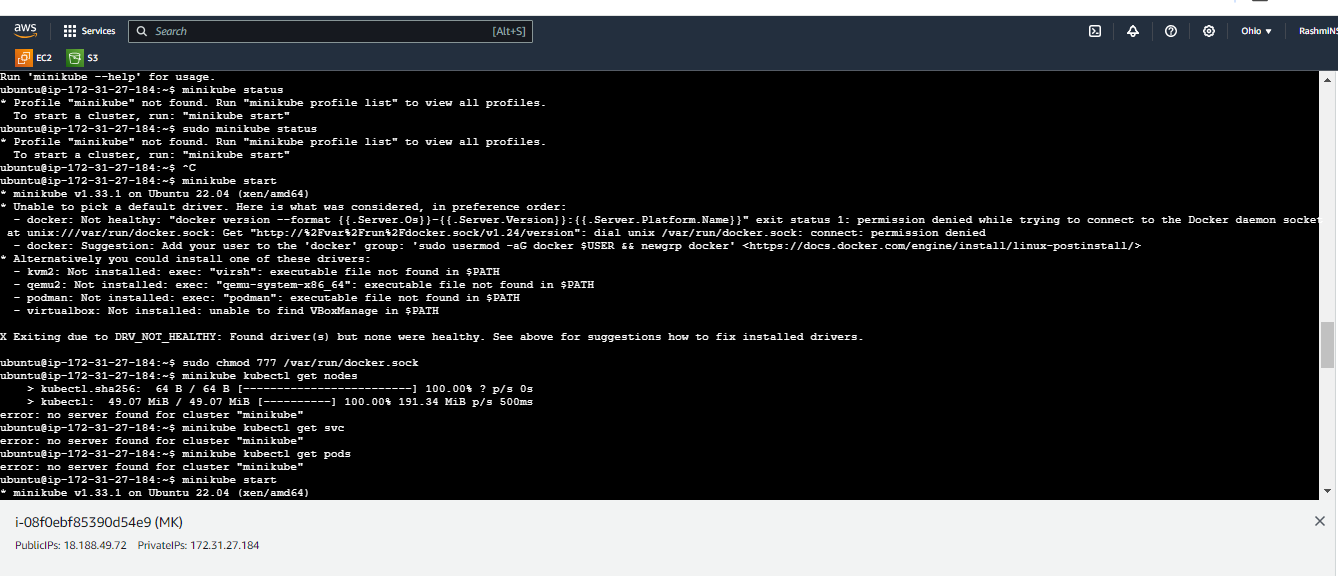
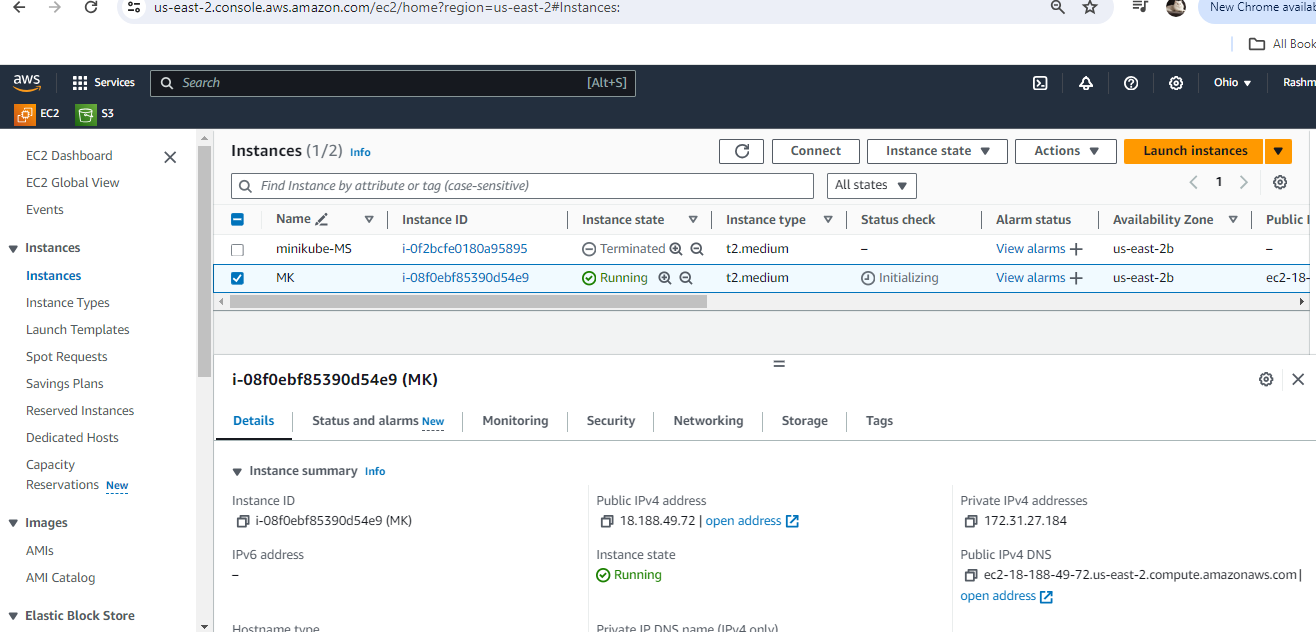
sudo nano assignment1.yaml

\*\* Use the Deployment file and paste here ->>

kubectl apply –f assignment1.yaml

kubectl get deploy

kubectl get pods



Assignment2: Use the previous deployment

● Create a service of type NodePort for nginx deployment

● Check the nodeport service on a browser to verify

Assignment2 solution:

cat assignment1.yaml

apiVersion: v1

kind: Service

metadata:

name: assignment2-service

spec:

type: NodePort

selector:

app: nginx

ports:

- port: 80

# By default and for convenience, the `targetPort` is set to

# the same value as the `port` field.

targetPort: 80

# Optional field

# By default and for convenience, the Kubernetes control plane

# will allocate a port from a range (default: 30000-32767)

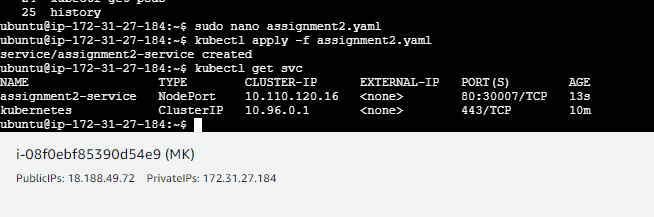
nodePort: 30007

sudo nano assignment2.yaml

\*\* Use the Service file and paste here ->>

kubectl apply –f assignment2.yaml

kubectl get svc



Assignment3: Use the previous deployment ● Change the replicas to 5 for the deployment

Assignment3 solution:

kubectl get pods

apiVersion: apps/v1

kind: Deployment

metadata:

name: assignment1-deployment

labels:

app: nginx

spec:

replicas: 5

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx

ports:

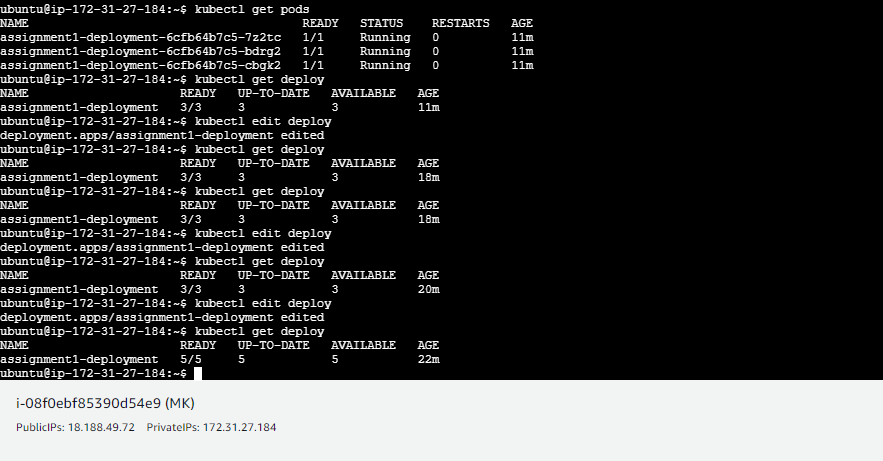
- containerPort: 80

kubectl get deploy

kubectl edit deploy

\*\* Edit Deployment file for replicas to 5

kubectl get deploy



Assignment4: Use the previous deployment

● Deploy an nginx deployment of 3 replicas

● Create an nginx service of type clusterip

Assignment4 solution:

apiVersion: v1

kind: Service

metadata:

name: kubernetes

spec:

type: ClusterIP

selector:

app: nginx

ports:

- port: 80

# By default and for convenience, the `targetPort` is set to

# the same value as the `port` field.

targetPort: 80

# Optional field

# By default and for convenience, the Kubernetes control plane

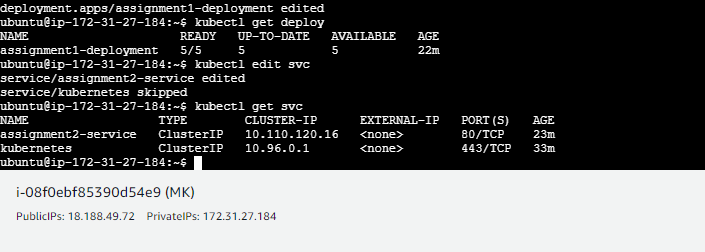
# will allocate a port from a range (default: 30000-32767)

nodePort: null

kubectl edit svc

\*\*Edit the NodePort Service file to ClusterIP

kubectl get svc



Assignment5: Create an ingress service /apache to apache service /nginx to nginx service

Assignment5 solution:

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: assignment5-ingress

annotations:

nginx.ingress.kubernetes.io/rewrite-target: /

spec:

ingressClassName: nginx-example

rules:

- http:

paths:

- path: /nginx

pathType: Prefix

backend:

service:

name: assignment2-service

port:

number: 80

minikube addons enable ingress

sudo nano assignment5.yaml

\*\* Add this Ingress service file ->>

kubectl get ing

kubectl apply –f assignment5.yaml

kubectl get ing

