

EXPERIMENT 4

MANORATH ITAL

D10A/19

Aim

To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

Steps

- 1) In your Kubernetes cluster check if all nodes are connected.

\$ kubectl get nodes

```
ubuntu@master-node:~$ kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
master-node   Ready     control-plane   54m   v1.31.1
worker1       Ready     <none>         36m   v1.31.1
ubuntu@master-node:~$
```

- 2) Create deploy.yaml file

\$ sudo nano deploy.yaml

- 3) Copy nginx-deployment.yaml file from Kubernetes site and paste in above file.

- 4) Create deployment

\$ kubectl apply -f

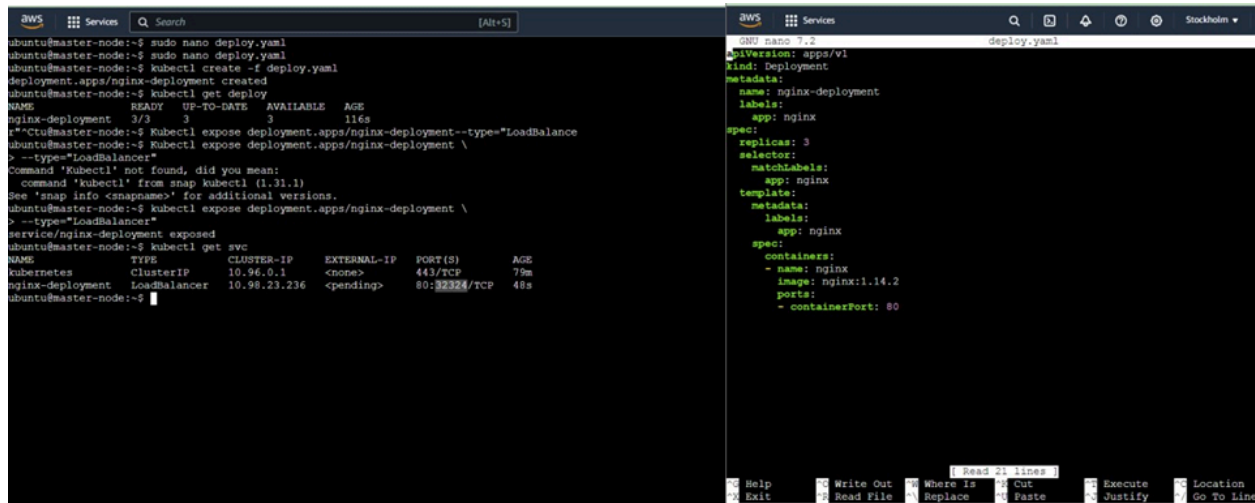
- 5) Check if deployment is created and all pods are running

\$ kubectl get deploy.po

6) Expose the app to internet using load balancer service.

\$ kubectl expose deployment.apps/nginx-deployment --type= "load balancer"

7) Kubectl get svc



The image consists of two side-by-side terminal screenshots. The left terminal shows the process of creating a deployment and exposing it as a service. The right terminal shows the content of the deployment.yaml file.

```
ubuntu@master-node:~$ sudo nano deploy.yaml
ubuntu@master-node:~$ kubectl create -f deploy.yaml
deployment.apps/nginx-deployment created
ubuntu@master-node:~$ kubectl get deploy
NAME                    READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment        3/3     3             3            116s
ubuntu@master-node:~$ kubectl expose deployment.apps/nginx-deployment \
--type="LoadBalancer"
Command 'Kubectl' not found, did you mean:
  command 'Kubectl' from snap kubectl (1.31.1)
See 'snap info <name>' for additional versions.
ubuntu@master-node:~$ kubectl expose deployment.apps/nginx-deployment \
--type="LoadBalancer"
service/nginx-deployment exposed
ubuntu@master-node:~$ kubectl get svc
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes          ClusterIP           10.96.0.1        <none>            443/TCP          79m
nginx-deployment    LoadBalancer       10.98.23.236     <pending>        80:32324/TCP     48s
ubuntu@master-node:~$
```

```
GNU nano 7.2 deploy.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

8) deployed our Nginx server on our EC2 instance.



