KUBERNETES ASSIGNMENT – 2

Name: Vikram
Assignment: Expose NGINX Deployment Using NodePort Service

Problem Statement

Using the existing Kubernetes cluster and NGINX deployment:

1. Create a NodePort Service for NGINX
2. Expose the application
3. Verify access from a browser

TASK 1: Use the Existing Deployment

Verify the deployment created in Assignment-1:

kubectl get deployments

kubectl get pods -o wide

Ensure assignment1-deployment is running with 3 replicas.

```
ubuntu@ip-10-0-15-63:~$ kubectl get deployments
kubectl get pods -o wide
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
assignment1-deployment   3/3     3            3           5m40s
NAME                                     READY   STATUS    RESTARTS   AGE     IP           NODE              NOMINATED NODE   READINESS GATES
assignment1-deployment-766fc4584d-dvb74   1/1     Running   0          5m40s   10.244.0.6   ip-10-0-15-63   <none>           <none>
assignment1-deployment-766fc4584d-nff89   1/1     Running   0          5m40s   10.244.0.4   ip-10-0-15-63   <none>           <none>
assignment1-deployment-766fc4584d-r264v   1/1     Running   0          5m40s   10.244.0.5   ip-10-0-15-63   <none>           <none>
ubuntu@ip-10-0-15-63:~$
```

TASK 2: Create NodePort Service YAML

Create service file:

```
nano nginx-service.yaml
```

Paste the following:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30007
```

Apply the service:

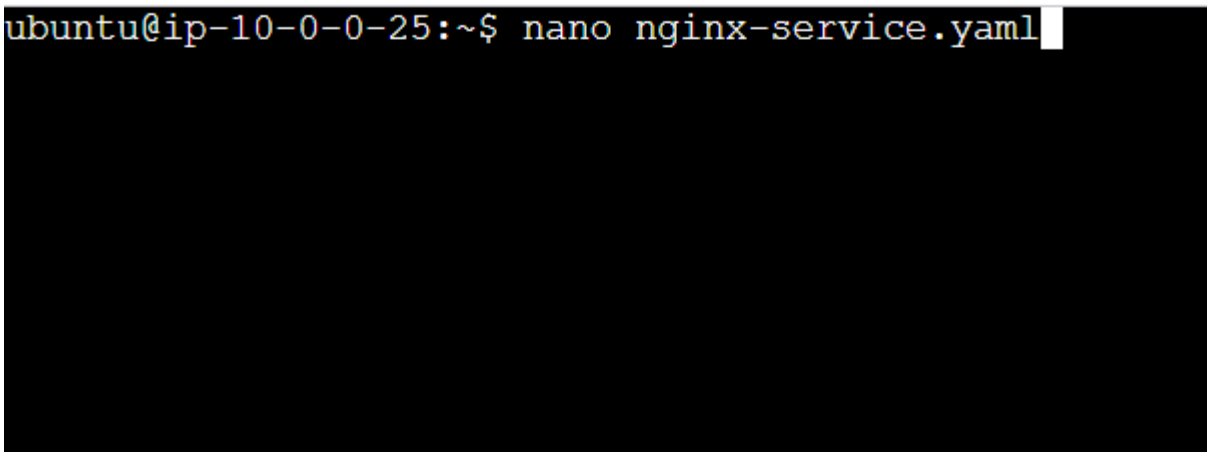
```
kubectl apply -f nginx-service.yaml
```

Verify service:

```
kubectl get services
```

You should see:

```
nginx-nodeport   NodePort   ...   80:30007/TCP
```



```
  GNU nano 7.2
apiVersion: v1
kind: Service
metadata:
  name: nginx-nodeport
spec:
  type: NodePort
  selector:
    app: nginx
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30007
```

```
ubuntu@ip-10-0-0-25:~$ kubectl apply -f nginx-service.yaml
service/nginx-nodeport created
ubuntu@ip-10-0-0-25:~$
```

```
ubuntu@ip-10-0-0-25:~$ kubectl get services
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP        100m
nginx-nodeport   NodePort    10.102.74.144   <none>        80:30007/TCP   24s
ubuntu@ip-10-0-0-25:~$ kubectl get nodes -o wide
NAME           STATUS   ROLES           AGE    VERSION    INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION    CONTAINER-RUNTIME
ip-10-0-0-25   Ready    control-plane   101m   v1.29.15   10.0.0.25     <none>        Ubuntu 24.04.3 LTS   6.14.0-1015-aws   containerd://1.7.28
ip-10-0-5-25   Ready    <none>          96m    v1.29.15   10.0.5.25     <none>        Ubuntu 24.04.3 LTS   6.14.0-1015-aws   containerd://1.7.28
ip-10-0-6-87   Ready    <none>          96m    v1.29.15   10.0.6.87     <none>        Ubuntu 24.04.3 LTS   6.14.0-1015-aws   containerd://1.7.28
ubuntu@ip-10-0-0-25:~$
```

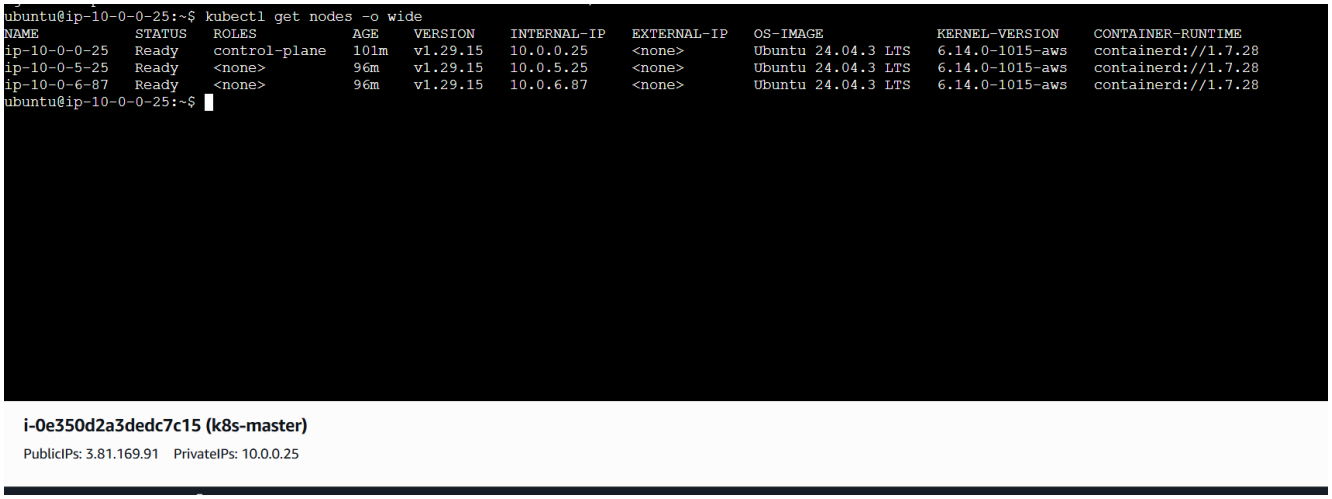
i-0e350d2a3dedc7c15 (k8s-master)

PublicIPs: 3.81.169.91 PrivateIPs: 10.0.0.25

TASK 3: Verify in Browser

Get worker node public IP:

kubectl get nodes -o wide



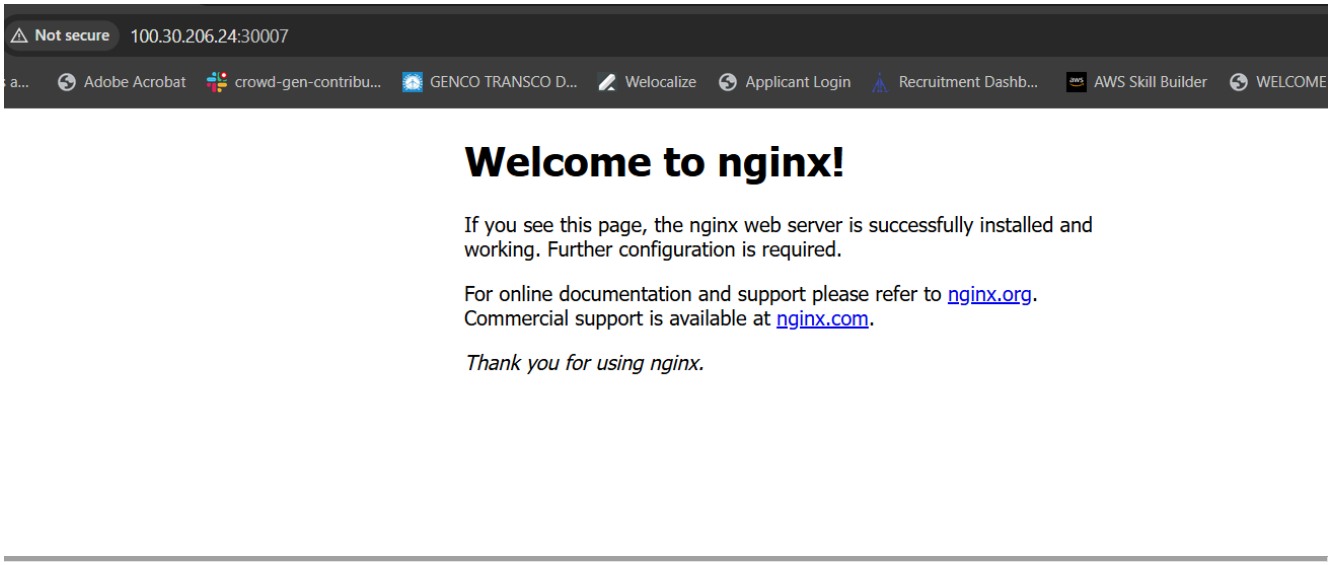
Open in browser:

http://<WORKER_NODE_PUBLIC_IP>:30007

Note: add port 30007 tcp in security group

Expected result:

NGINX Welcome Page displayed



Verification Commands

kubectl get pods

kubectl get svc

kubectl describe svc nginx-nodeport

```
ubuntu@ip-10-0-15-63:~$ kubectl get pods
kubectl get svc
kubectl describe svc nginx-nodeport
NAME                                     READY   STATUS    RESTARTS   AGE
assignment1-deployment-766fc4584d-dvb74   1/1     Running   0          25m
assignment1-deployment-766fc4584d-nff89   1/1     Running   0          25m
assignment1-deployment-766fc4584d-r264v   1/1     Running   0          25m
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP        50m
nginx-nodeport   NodePort    10.104.208.124   <none>        80:30007/TCP   18m
Name:                     nginx-nodeport
Namespace:                default
Labels:                   <none>
Annotations:              <none>
Selector:                 app=nginx
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.104.208.124
IPs:                      10.104.208.124
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  30007/TCP
Endpoints:                10.244.0.4:80,10.244.0.5:80,10.244.0.6:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
```

i-0ec6451fd14969f99 (k8s-master)

Conclusion

Successfully exposed the NGINX deployment using NodePort Service and verified accessibility from the browser.

✓ Used existing deployment

✓ Created NodePort service

✓ Exposed NGINX application

✓ Verified external access