**Deploy NGINX with a Hello World Page**

Once your AKS cluster is ready, deploy NGINX inside the subnet.

**1. Create an NGINX Deployment**

kubectl create deployment nginx-hello --image=nginx --replicas=1

**2. Expose NGINX as a Service**

kubectl expose deployment nginx-hello --type=LoadBalancer --name=nginx-service --port=80 --target-port=80

This will create an external IP for the NGINX service.

**3. Verify the Service**

kubectl get svc nginx-service

Example output:

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

nginx-service LoadBalancer 10.0.0.5 20.102.45.67 80:32000/TCP 2m

The EXTERNAL-IP (e.g., **20.102.45.67**) is your public endpoint.

**Step 3: Modify NGINX to Show "Hello World"**

By default, NGINX serves its default page. To serve a custom page:

**1. Create a ConfigMap for the Hello World Page**

kubectl create configmap nginx-html --from-literal=index.html='<h1>Hello World from Azure AKS!</h1>'

**2. Update the Deployment to Use the ConfigMap**

Create a new YAML file (nginx-deployment.yaml):

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-hello

spec:

replicas: 1

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx

volumeMounts:

- name: html

mountPath: /usr/share/nginx/html

volumes:

- name: html

configMap:

name: nginx-html

Apply the updated deployment:

kubectl apply -f nginx-deployment.yaml

**Step 4: Access the Hello World Page**

Once the deployment is updated, open a web browser or use curl:

curl http://<EXTERNAL-IP>

Example:

curl http://20.102.45.67

Expected output:

<h1>Hello World from Azure AKS!</h1>