

# Module 7: Kubernetes Assignment -5

## Tasks To Be Performed:

1. Use the previous deployment
2. Deploy an NGINX deployment of 3 replicas
3. Create an NGINX service of type ClusterIP
4. Create an ingress service/ Apache to Apache service/ NGINX to NGINX service

## Solution:

1. Use the previous deployment
2. Deploy an NGINX deployment of 3 replicas



```
awscli.kubernetes.io/last-applied-configuration: |
  ("apiVersion": "apps/v1", "kind": "Deployment", "metadata": {"annotations": {}, "labels": {"app": "nginx"}, "name": "nginx-deployment", "namespace": "default"}, "spec": {"replicas": 3, "selector": {"matchLabels": {"app": "nginx"}}, "template": {"metadata": {"labels": {"app": "nginx"}}, "spec": {"containers": [{"image": "nginx:latest", "name": "nginx", "ports": [{"containerPort": 80}]}]}})
creationTimestamp: "2024-06-16T20:40:20Z"
generation: 2
labels:
  app: nginx
name: nginx-deployment
namespace: default
resourceVersion: "5085"
uid: aal238b9-5e6c-409b-990b-7cdc0b1a09c3
spec:
  progressDeadlineSeconds: 600
  replicas: 3
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: nginx
-- INSERT --
```

i-0868263520b95d509 (K8S-M)  
PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

```
progressDeadlineSeconds: 600
replicas: 3
revisionHistoryLimit: 10
```

```
ubuntu@ip-172-31-4-255:~$ kubectl edit deployment nginx-deployment
deployment.apps/nginx-deployment edited
ubuntu@ip-172-31-4-255:~$
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

```
ubuntu@ip-172-31-4-255:~$ kubectl get deployment nginx-deployment
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    3/3     3             3           45m
ubuntu@ip-172-31-4-255:~$
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

### 3. Create an NGINX service of type ClusterIP


```
ubuntu@ip-172-31-4-255:~$ kubectl edit service nginx-service
service/nginx-service edited
ubuntu@ip-172-31-4-255:~$ kubectl get service nginx-service
NAME                TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)    AGE
nginx-service       ClusterIP   10.105.225.162  <none>       80/TCP     15m
ubuntu@ip-172-31-4-255:~$
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

### 4. Create an ingress service/ Apache to Apache service/ NGINX to NGINX service

A} Create an ingress service

 **kubernetes**

DocumentationKubernetes BlogTrainingPartnersCommunityCase StudiesVersionsEnglishSearch this site

Search this site

- Documentation
- Getting started
- Concepts
  - Overview
  - Cluster Architecture
  - Containers
  - Workloads
  - Services, Load Balancing, and Networking
    - Service
    - Ingress**
    - Ingress Controllers
    - Gateway API
    - EndpointSlices
    - Network Policies
    - DNS for Services and Pods
    - IPv4/IPv6 dual-stack
    - Topology Aware Routing
    - Networking on Windows
    - Service ClusterIP allocation

## The Ingress resource

A minimal Ingress resource example:

```
service/networking/minimal-ingress.yaml

apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: minimal-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  ingressClassName: nginx-example
  rules:
  - http:
      paths:
      - path: /testpath
        pathType: Prefix
        backend:
          service:
            name: test
            port:
              number: 80
```

- ingress API reference
- ingressClass API reference
- Edit this page
- Create child page
- Create an issue
- Print entire section

### Terminology

- What is Ingress?
- Prerequisites
- The Ingress resource
- Ingress rules
- DefaultBackend
- Resource backends
- Path types
- Examples
- Hostname wildcards
- Ingress class
- IngressClass scope
- Deprecated annotation
- Default IngressClass

```
error: error parsing ingress.yaml: error converting yaml to
ubuntu@ip-172-31-4-255:~$ sudo nano ingress.yaml
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

```
aws | Services | Search
GNU nano 7.2
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress-service
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  # ingressClassName: nginx-example
  rules:
  - host: example.com
  - http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: my-service
            port:
              number: 80
```

```
ubuntu@ip-172-31-4-255:~$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/my-ingress-service created
ubuntu@ip-172-31-4-255:~$
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

## B) Apache to Apache service

```
ubuntu@ip-172-31-4-255:~$ sudo nano apache.yaml
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

```
aws Services Search [Alt+S]
GNU nano 7.2 apache.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: apache-deployment
  labels:
    app: apache
spec:
  replicas: 2
  selector:
    matchLabels:
      app: apache
  template:
    metadata:
      labels:
        app: apache
    spec:
      containers:
        - name: apache
          image: apache:latest
          ports:
            - containerPort: 81
---
```

```
aws | Services | Search
GNU nano 7.2
selector:
  matchLabels:
    app: apache
template:
  metadata:
    labels:
      app: apache
  spec:
    containers:
      - name: apache
        image: apache:latest
        ports:
          - containerPort: 81
--
apiVersion: v1
kind: Service
metadata:
  name: apache-service
spec:
  selector:
    app: apache
  type: NodePort
  ports:
    - protocol: TCP
      port: 81
      targetPort: 80
G Help      ^O Write Out  ^W Where Is   ^K Cut
X Exit      ^R Read File  ^\ Replace    ^U Paste
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

```
ubuntu@ip-172-31-4-255:~$ kubectl apply -f apache.yaml
deployment.apps/apache-deployment created
service/apache-service created
ubuntu@ip-172-31-4-255:~$
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

### 3} NGINX to NGINX service

```
ubuntu@ip-172-31-4-255:~$ sudo nano nginx.yaml
ubuntu@ip-172-31-4-255:~$
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

```
aws Services Search
GNU nano 7.2
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:latest
        ports:
        - containerPort: 82
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Ex  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Ju

```
aws | Services | Search
GNU nano 7.2
selector:
  matchLabels:
    app: nginx
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
      - name: nginx
        image: nginx:latest
        ports:
          - containerPort: 82
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  type: NodePort
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80

^G Help      ^O Write Out  ^W Where Is   ^K Cut
^X Exit      ^R Read File  ^\ Replace    ^U Paste
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255

```
ubuntu@ip-172-31-4-255:~$ kubectl apply -f nginx.yaml
deployment.apps/nginx-deployment configured
service/nginx-service configured
ubuntu@ip-172-31-4-255:~$
```

i-0868263520b95d509 (K8S-M)

PublicIPs: 54.241.63.85 PrivateIPs: 172.31.4.255





