Experiment -3.3

Student Name: Ayush Pandey **UID:** 22BDO10038
Branch: CSE-DevOps **Section/Group:** 22BCD-1(A)
Semester: 5 **Date of Performance:** 28-10-24
Subject Name: Docker and Kubernetes **Subject Code:** 22CSH-343

1. Aim/Overview of the practical:

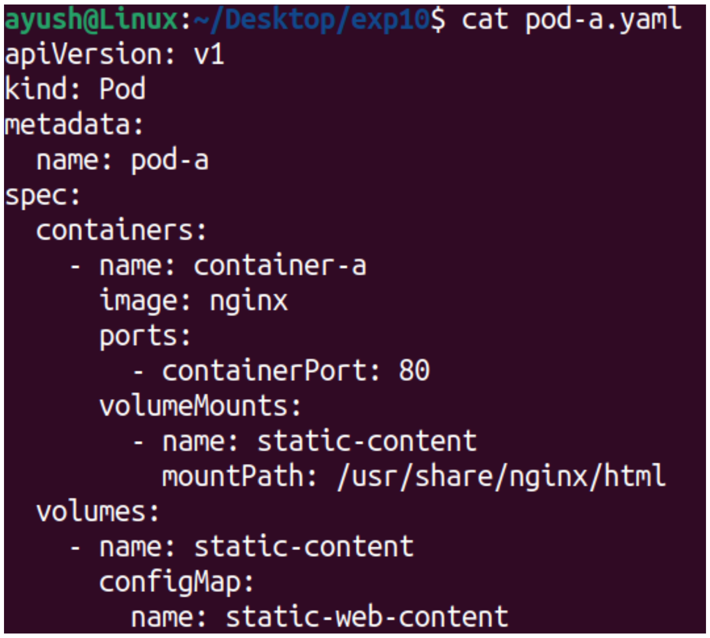
Deploying a Node.js Application on Kubernetes with IBM Containers.

2. Apparatus: PC, Docker Engine, Kubernetes, Minikube, Ubuntu Linux

3. Steps for experiment/practical:

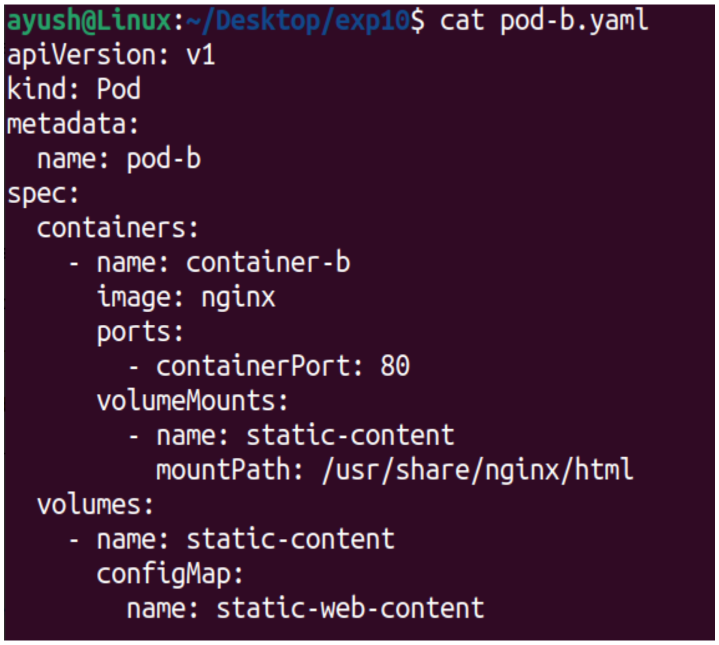
Step 1: Create YAML Manifests for the Pods

1. **pod-a.yaml**



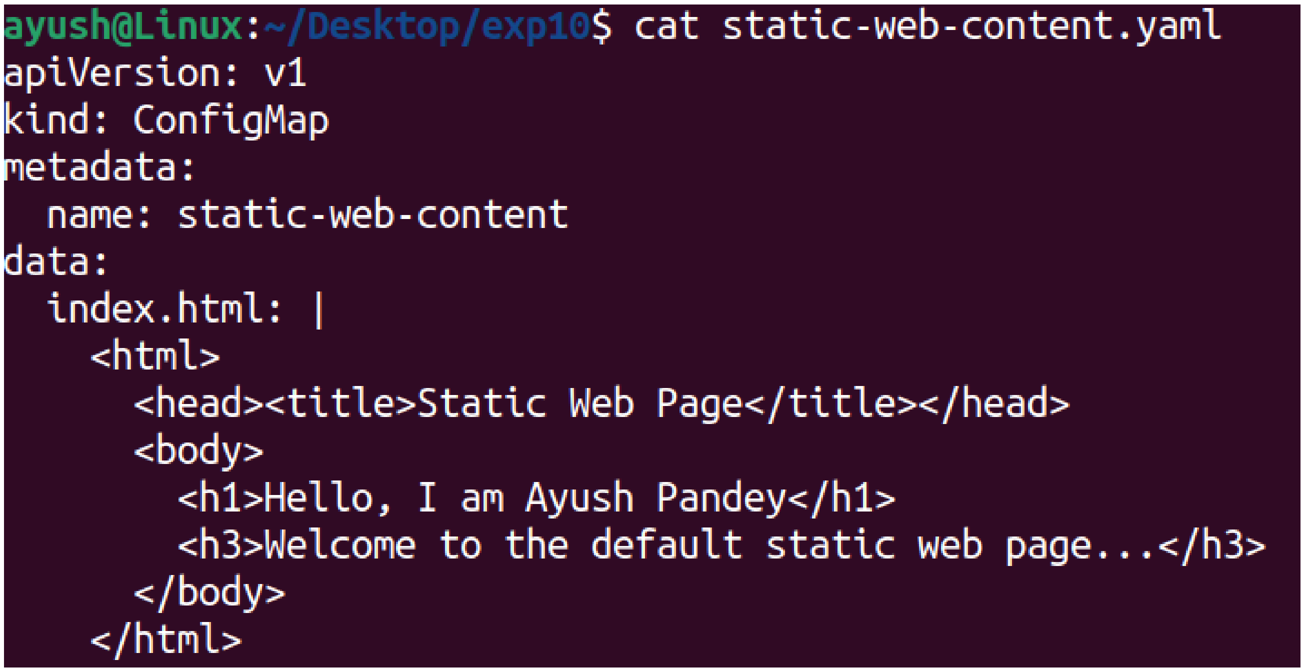
```
ayush@Linux:~/Desktop/exp10$ cat pod-a.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod-a
spec:
  containers:
    - name: container-a
      image: nginx
      ports:
        - containerPort: 80
      volumeMounts:
        - name: static-content
          mountPath: /usr/share/nginx/html
  volumes:
    - name: static-content
      configMap:
        name: static-web-content
```

2. **pod-b.yaml**



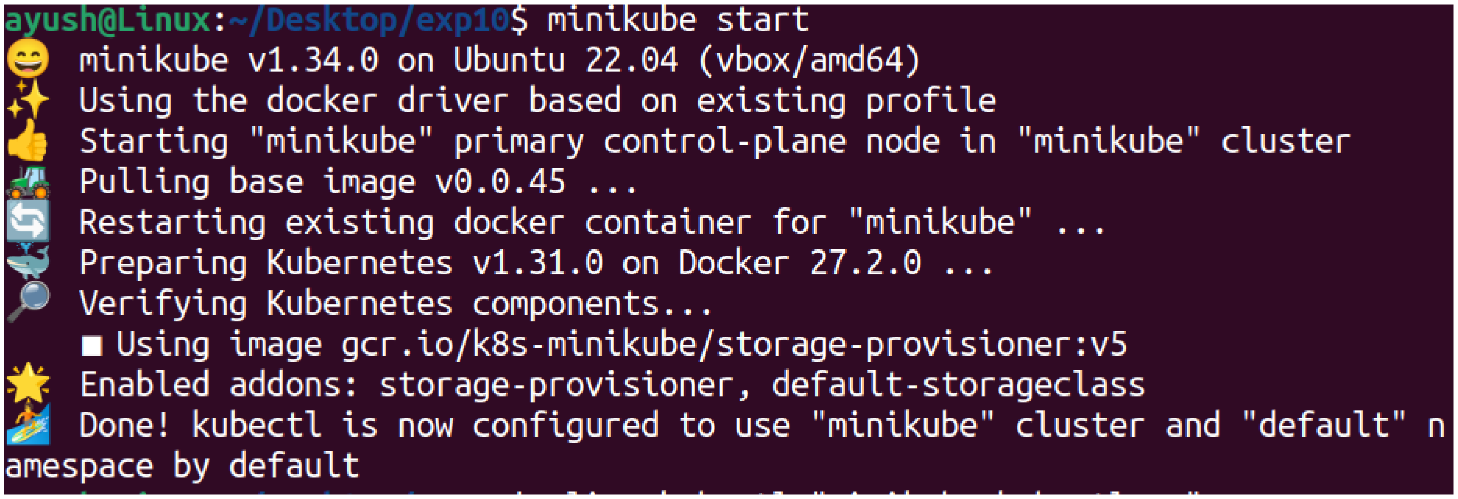
```
ayush@Linux:~/Desktop/exp10$ cat pod-b.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod-b
spec:
  containers:
    - name: container-b
      image: nginx
      ports:
        - containerPort: 80
      volumeMounts:
        - name: static-content
          mountPath: /usr/share/nginx/html
  volumes:
    - name: static-content
      configMap:
        name: static-web-content
```

3. static-web-content.yaml

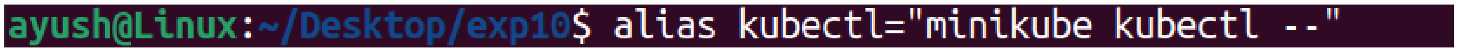


```
ayush@Linux:~/Desktop/exp10$ cat static-web-content.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: static-web-content
data:
  index.html: |
    <html>
      <head><title>Static Web Page</title></head>
      <body>
        <h1>Hello, I am Ayush Pandey</h1>
        <h3>Welcome to the default static web page...</h3>
      </body>
    </html>
```

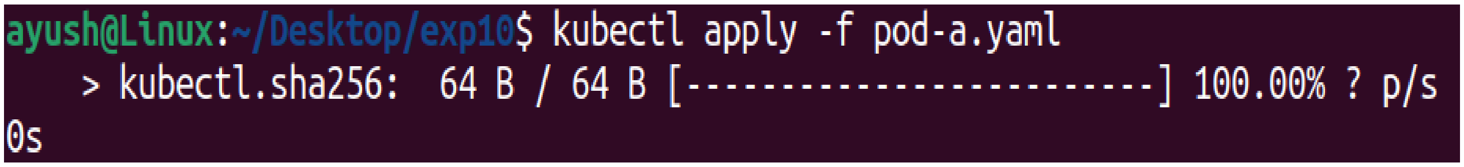
Step 2: Apply the YAML Manifests to Create Pods and ConfigMap



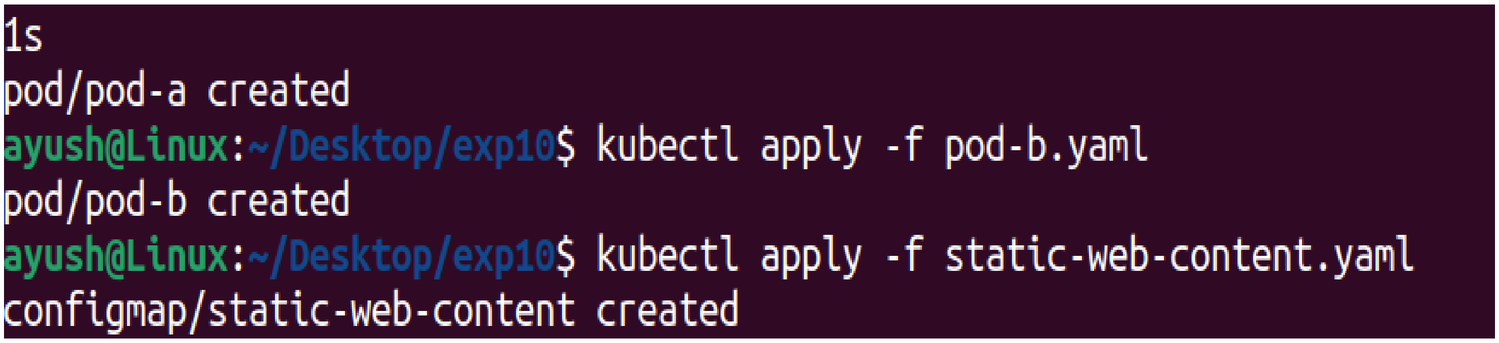
```
ayush@Linux:~/Desktop/exp10$ minikube start
😄  minikube v1.34.0 on Ubuntu 22.04 (vbox/amd64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.45 ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" n
amespace by default
```



```
ayush@Linux:~/Desktop/exp10$ alias kubectl="minikube kubectl --"
```

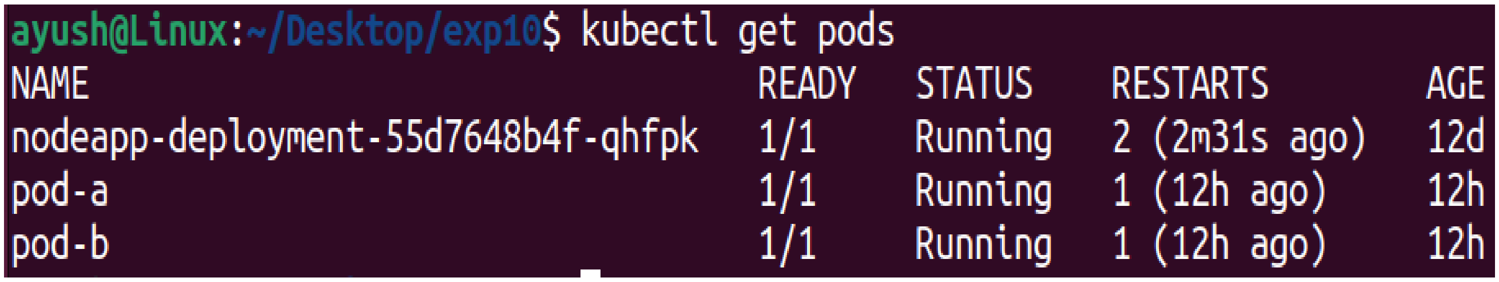


```
ayush@Linux:~/Desktop/exp10$ kubectl apply -f pod-a.yaml
    > kubectl.sha256:  64 B / 64 B [------------------------] 100.00% ? p/s
0s
```



```
1s
pod/pod-a created
ayush@Linux:~/Desktop/exp10$ kubectl apply -f pod-b.yaml
pod/pod-b created
ayush@Linux:~/Desktop/exp10$ kubectl apply -f static-web-content.yaml
configmap/static-web-content created
```

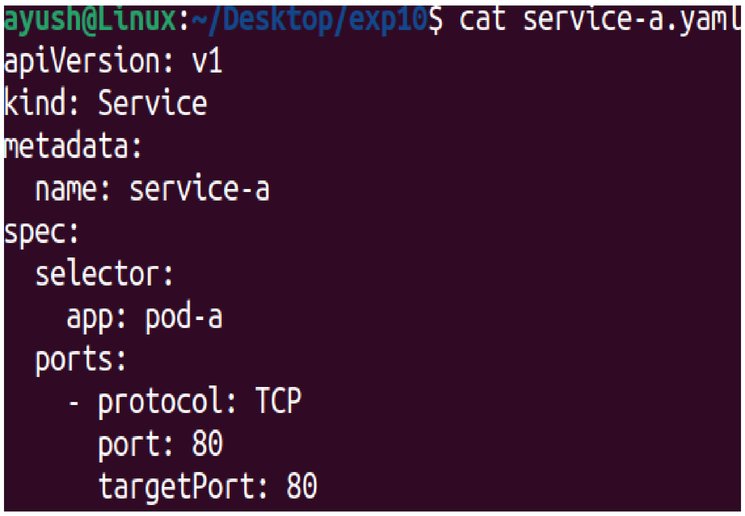
Step 3: Check the Status of the Pods



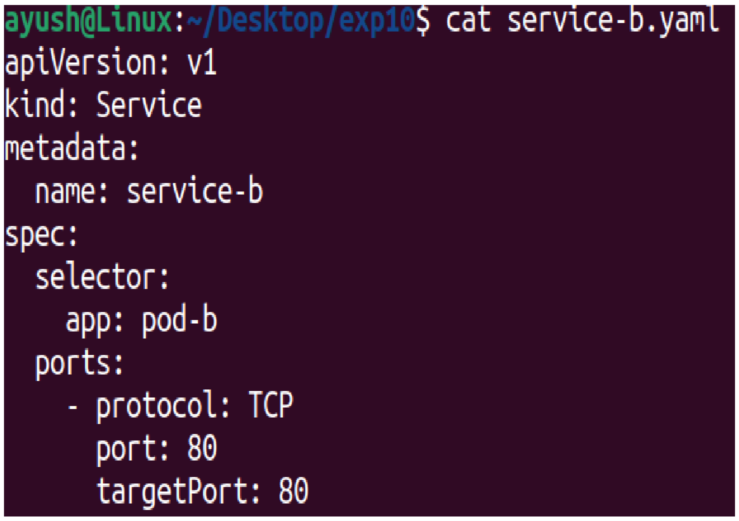
```
ayush@Linux:~/Desktop/exp10$ kubectl get pods
NAME                                 READY   STATUS    RESTARTS        AGE
nodeapp-deployment-55d7648b4f-qhfpk  1/1     Running   2 (2m31s ago)   12d
pod-a                                1/1     Running   1 (12h ago)     12h
pod-b                                1/1     Running   1 (12h ago)     12h
```

Step 4: Enable Communication Between Pods

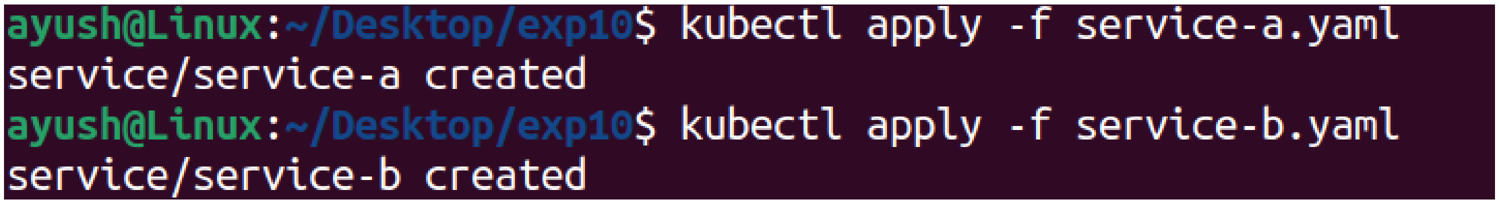
1. **service-a.yaml**
2. **service-b.yaml**



```
ayush@Linux:~/Desktop/exp10$ cat service-a.yaml
apiVersion: v1
kind: Service
metadata:
  name: service-a
spec:
  selector:
    app: pod-a
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```



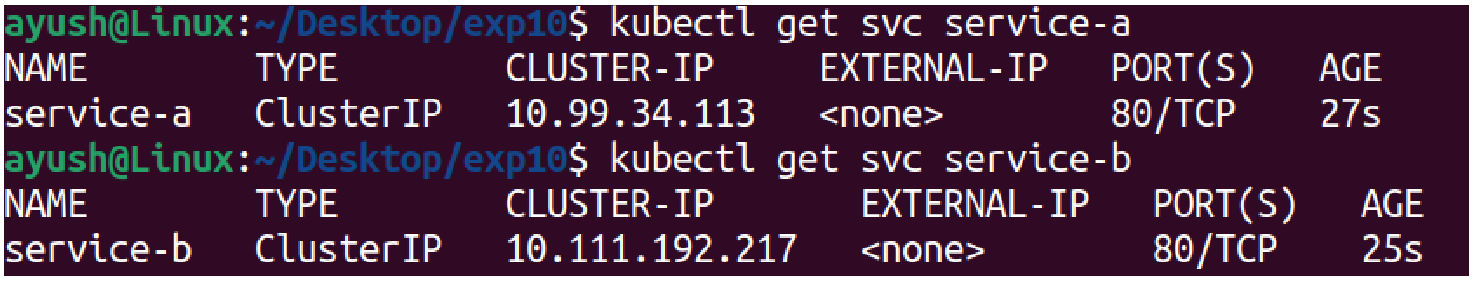
```
ayush@Linux:~/Desktop/exp10$ cat service-b.yaml
apiVersion: v1
kind: Service
metadata:
  name: service-b
spec:
  selector:
    app: pod-b
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```



```
ayush@Linux:~/Desktop/exp10$ kubectl apply -f service-a.yaml
service/service-a created
ayush@Linux:~/Desktop/exp10$ kubectl apply -f service-b.yaml
service/service-b created
```

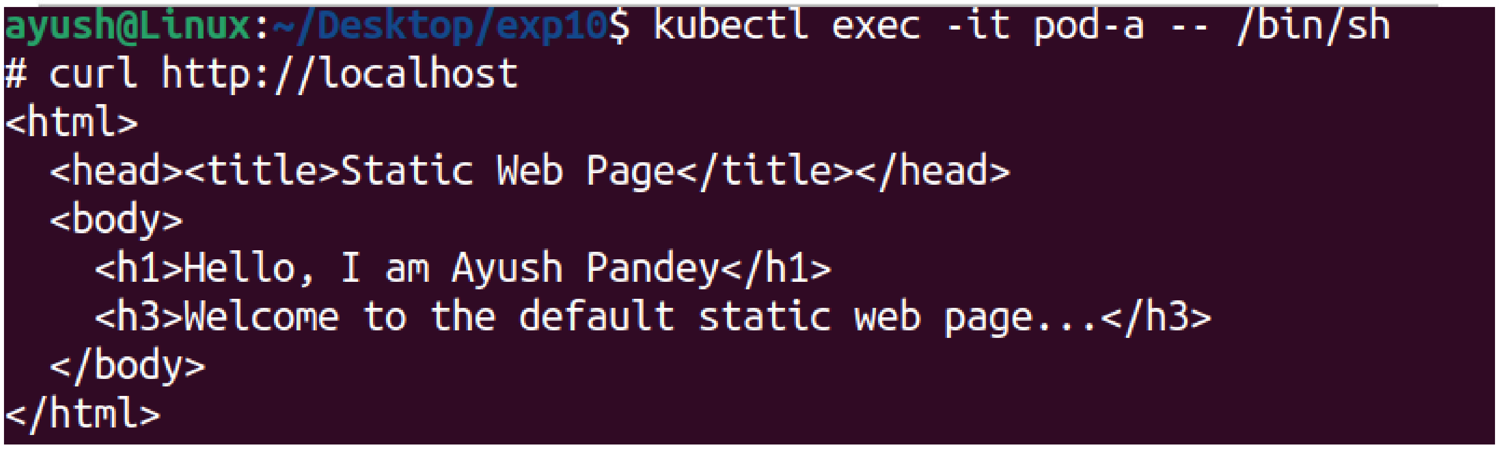
Step 5: Verify Communication

1. **Check the ClusterIP of the services:**



```
ayush@Linux:~/Desktop/exp10$ kubectl get svc service-a
NAME        TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
service-a   ClusterIP   10.99.34.113   <none>        80/TCP    27s
ayush@Linux:~/Desktop/exp10$ kubectl get svc service-b
NAME        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
service-b   ClusterIP   10.111.192.217   <none>        80/TCP    25s
```

2. **Access the services using the following command to ensure that they serve the static web content:**



```
ayush@Linux:~/Desktop/exp10$ kubectl exec -it pod-a -- /bin/sh
# curl http://localhost
<html>
  <head><title>Static Web Page</title></head>
  <body>
    <h1>Hello, I am Ayush Pandey</h1>
    <h3>Welcome to the default static web page...</h3>
  </body>
</html>
```

Kubernetes ensures immutability by maintaining the existing pods until the new ones are ready with the updated content.

Learning outcomes (What I have learnt):

1. I have learnt the concept of containerization and virtualization.

2. I have learnt about orchestration and orchestration tools.

3. I have learnt about Kubernetes and its architecture.

4. I have learnt the purpose of using microservice architecture over monolithic.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			