Assignment – 4

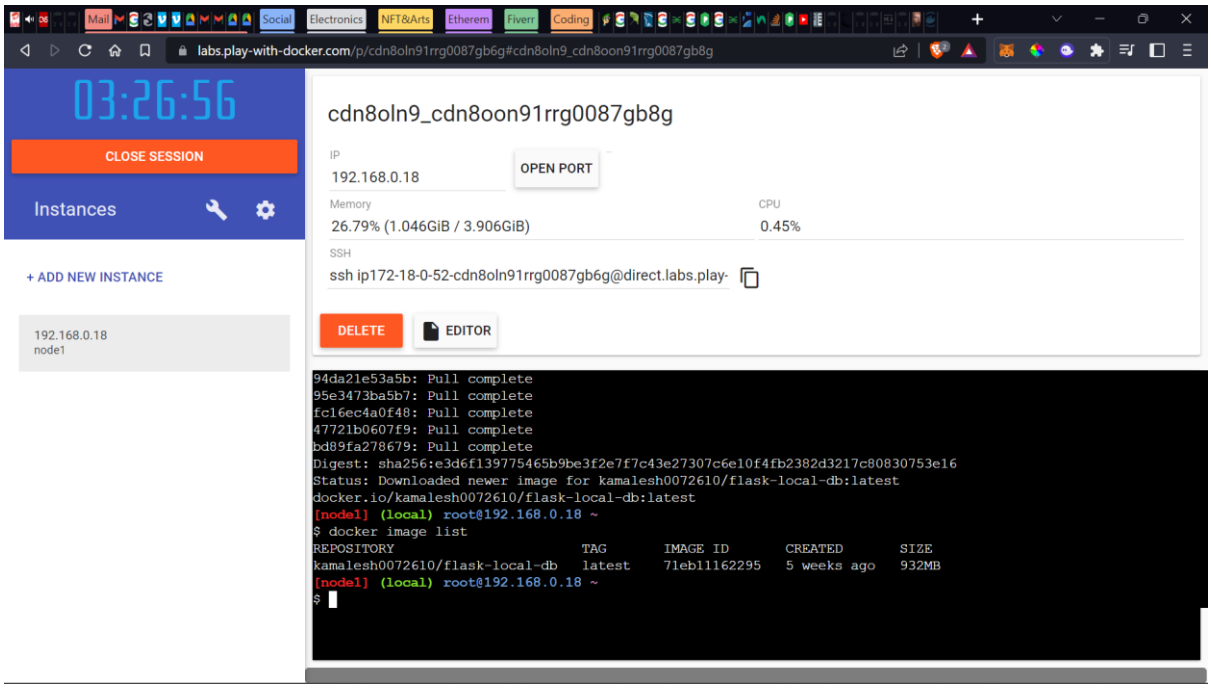
Cloud Application Development

Assignment Date	19 September 2022
Student Name	LOKESH N
Student Roll Number	211719106043
Maximum Mark	2 Marks

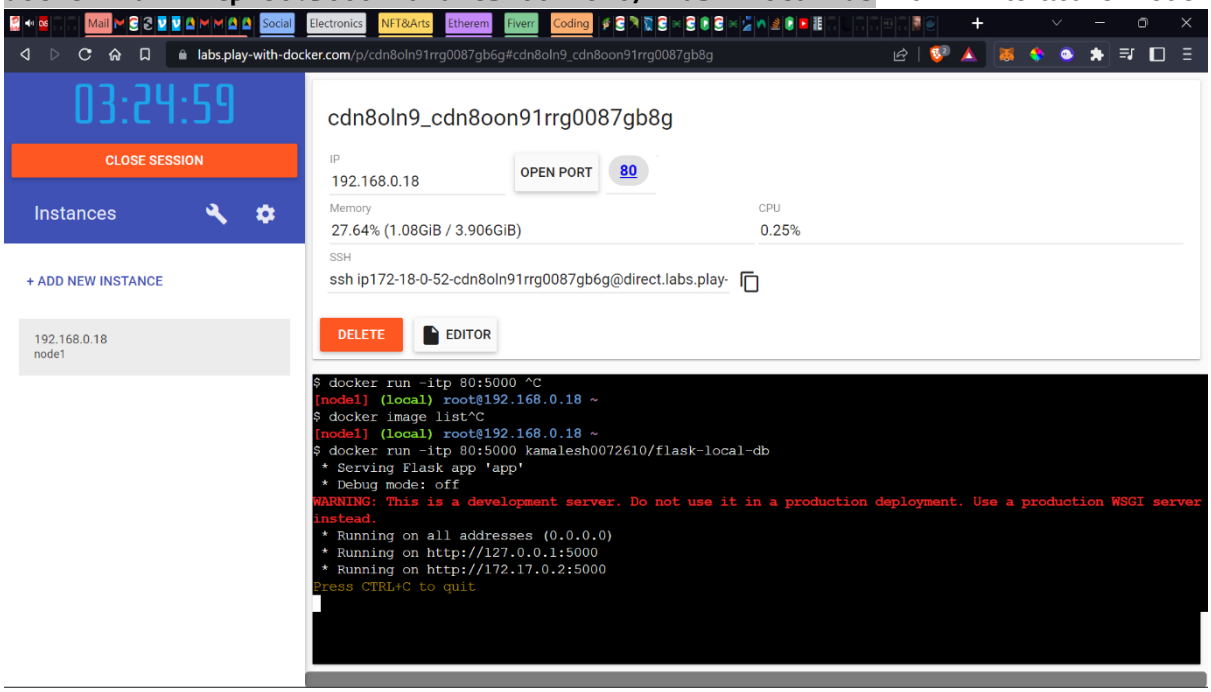
1.Pull an Image from docker hub and run it in docker playground.

Pushed my own Image to Docker Hub and used that for this assignment.

```
docker pull kamalesh0072610/flask-local-db:latest
docker image list
```

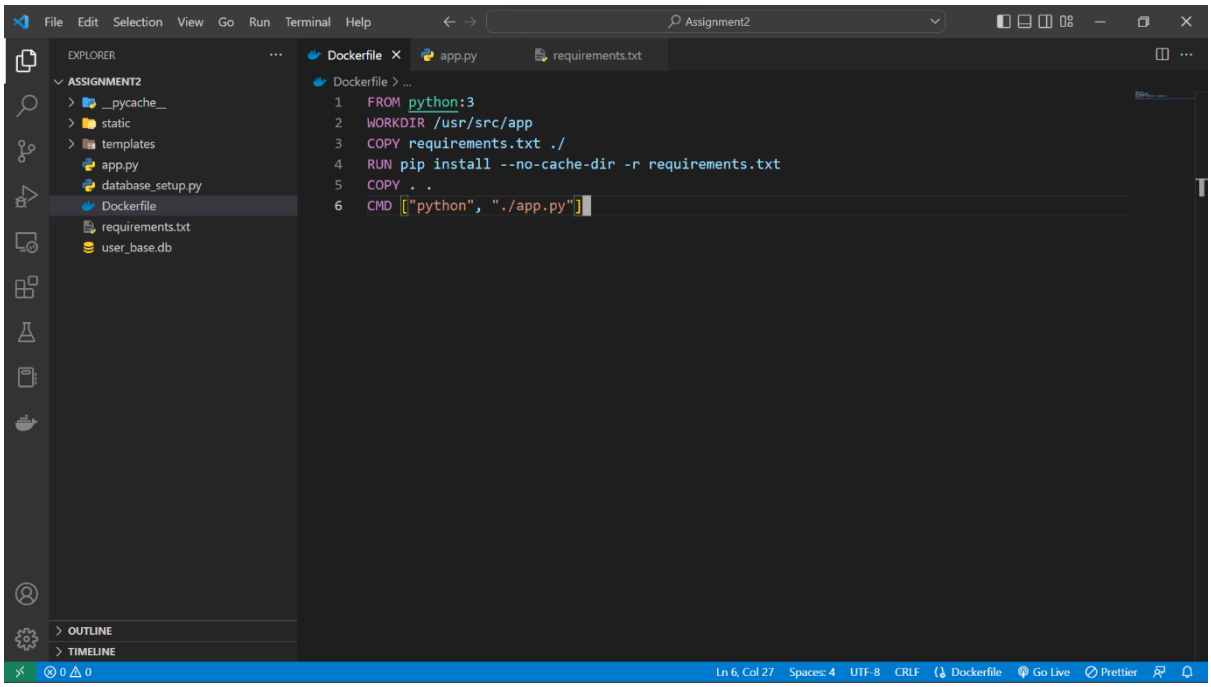


`docker run -itp 80:5000 kamalesh0072610/flask-local-db`– run in interactive mode.

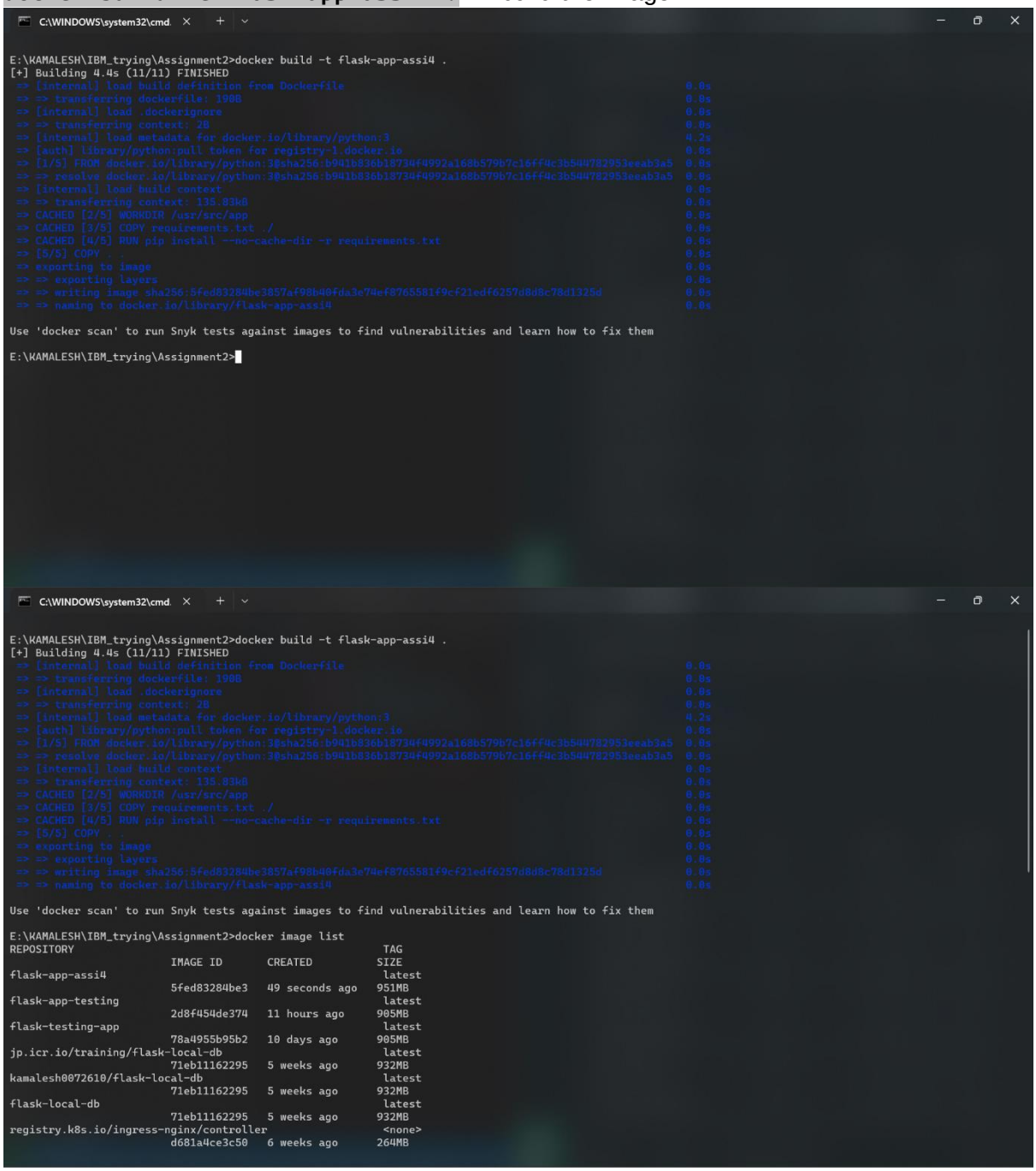


2.Create a dockerfile for the job portal / flask application and deploy it in Docker desktop application.

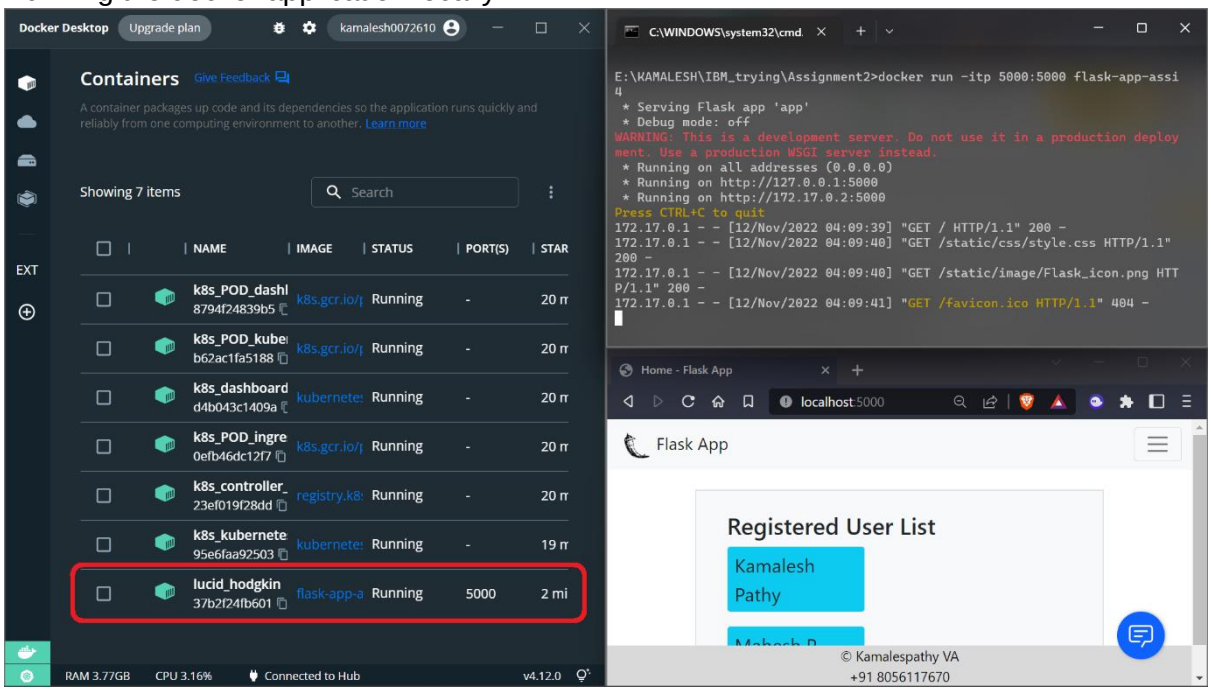
I've used the flask application used for assignment 2 for this assignment.



`docker build -t flask-app-assi4 .` – build the image



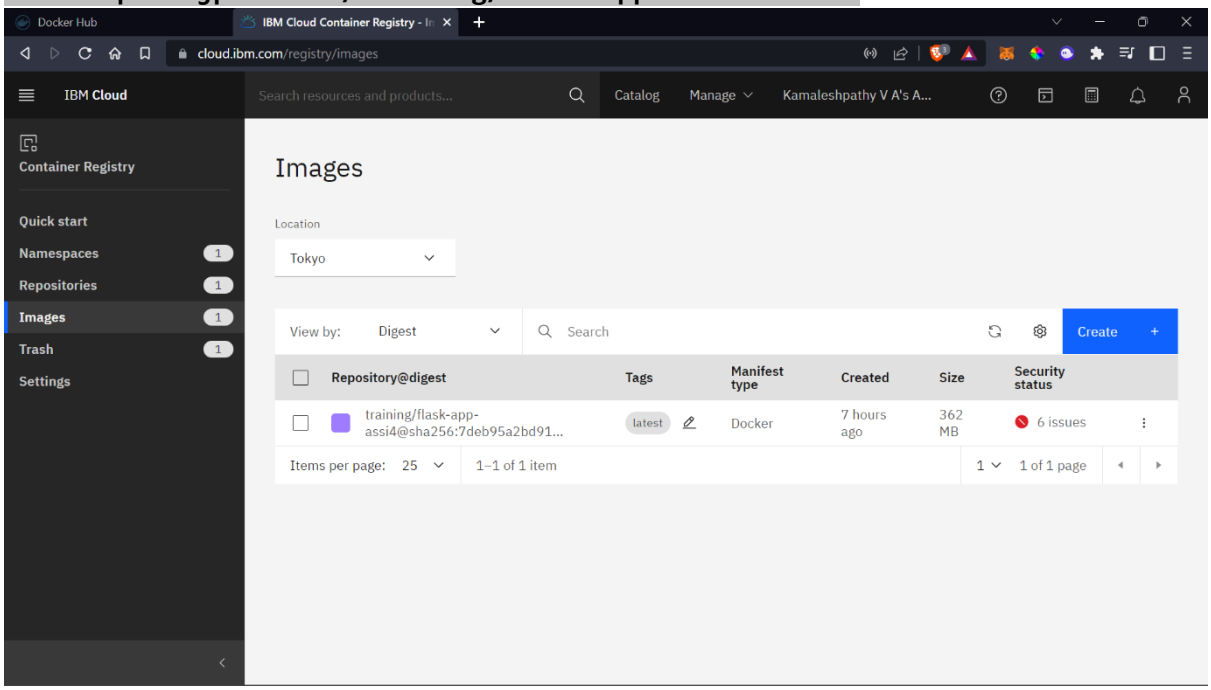
Running the docker application locally.



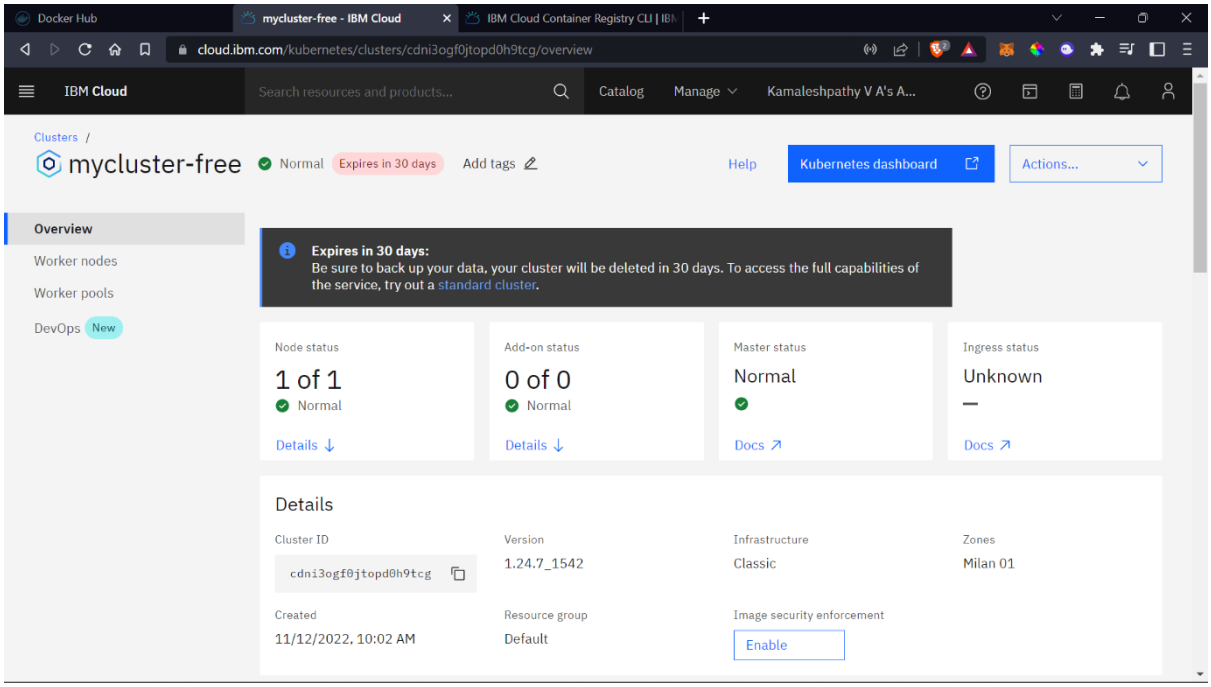
3. Create a IBM container registry and push docker image of flask application or job portal app.

Pushed the image to ibm container registry.

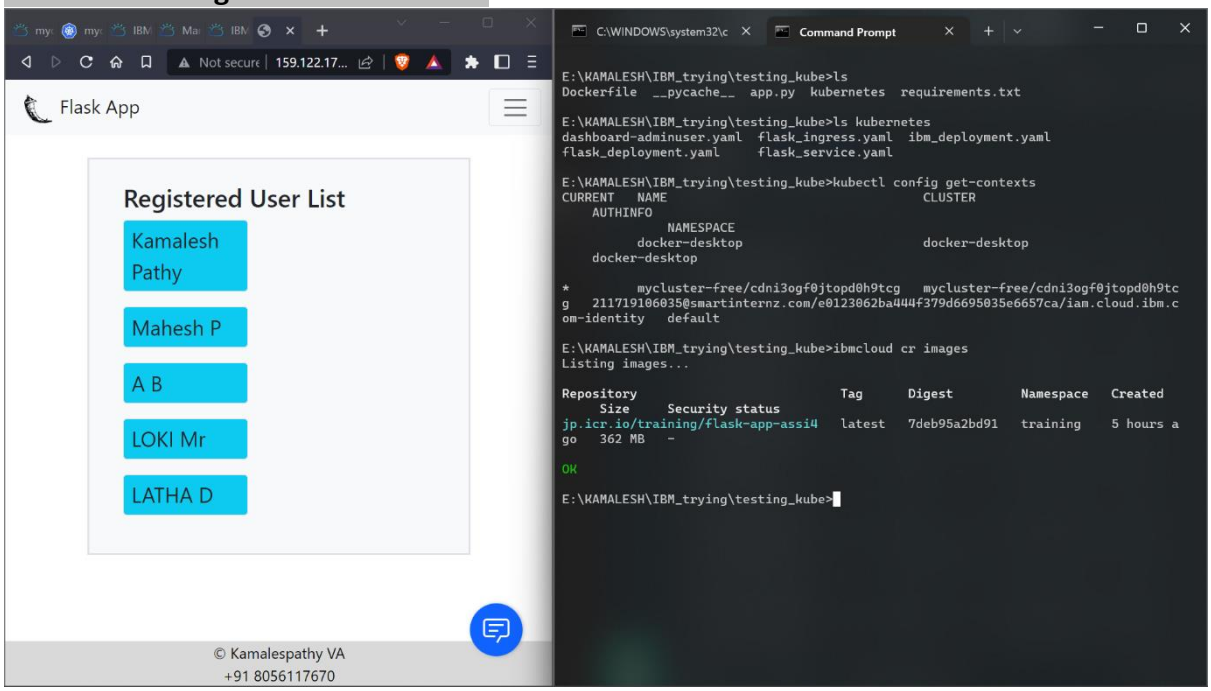
```
ibmcloud login
ibmcloud plugin install container-registry -r “IBM Cloud”
ibmcloudcr namespace-add training
ibmcloudcr login
docker tag flask-app-assig4jp.icr.io/training/flask-app-assi4:latest
docker push jp.icr.io/training/flask-app-assi4:latest
```



4. Create a Kubernetes cluster in IBM cloud and deploy flask application image or job portal image and also expose the same app to run in nodeport.




```
ibmcloud plugin install container-service
ibmcloud ks cluster config --cluster cdni3ogf0jtopd0h9tcg
kubectl config current-context
```



ibm_deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-app

spec:
  replicas: 5
  selector:
   matchLabels:
    app: flask-app
  template:
   metadata:
    labels:
     app: flask-app

   spec:
    containers:
    - name: flask-app-container
      image: jp.icr.io/training/flask-app-assi4
      imagePullPolicy: Always
      ports:
      - containerPort: 5000
        protocol: TCP
```

flask_service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: flask-app-service
spec:
  type: ClusterIP
  ports:
    - port: 5000
  selector:
    app: flask-app
```

flask_ingress.yaml

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: flask-app-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/ssl-redirect: "false"

spec:
  # ingressClassName: nginx
  rules:
    - http:
```

```
        paths:
          - backend:
              service:
                name: flask-app-service
                port:
                  number: 5000
            path: /
            pathType: Prefix
```

```
kubectl apply -f kubernetes/ibm_deployment.yaml
kubectl apply -f kubernetes/flask_service.yaml
kubectl apply -f kubernetes/flask_ingress.yaml
kubectl expose deployment flask-app --type=NodePort --name=flask-app
```