

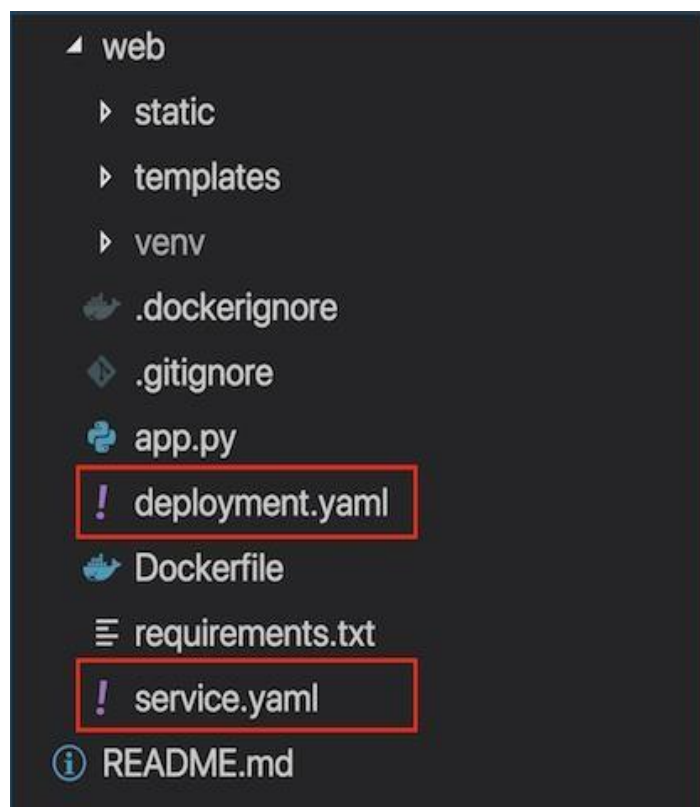
# Deployment of App in IBM Cloud

## Deploy On Kubernetes

|               |   |
|---------------|---|
| Date          | 19 NOV 2022   |
| Team ID       | PNT2022TMID03852  |
| Project Name  | Skill and Job Recommender   |
| Team Members: | Mark Franklin,Karthikeyan,Paul Nishanth<br>Ashwin,Vanni Venkatesh |

### Step 1 :

Go to the Project Repository then Create the Deployment.yaml and Service.yaml files.



## Step 2 :

In deployment.yaml file Type the Following Code.

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: flask-node-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: flasknode
  template:
    metadata:
      labels:
        app: flasknode
    spec:
      containers:
        - name: flasknode
          image: registry.ng.bluemix.net/flask-node/app
          imagePullPolicy: Always
          ports:
            - containerPort: 5000
```

## Step 3 :

In service.yaml file Type the Following Code.

```
apiVersion: v1
kind: Service
metadata:
  name: flask-node-deployment
spec:
  ports:
    - port: 5000
      targetPort: 5000
  selector:
    app: flasknode
```

## Step 4 :

Then follow the steps to deploy.

- 1.Target the IBM Cloud Kubernetes Service region where you want to work.

```
ibmcloud cs region-set us-south
```

- 2.Set the context for the cluster in your CLI.

```
ibmcloud cs cluster-config cluster_kunal> export  
KUBECONFIG=/Users/$USER/.bluemix/plugins/container-service/clusters/< cluster_name  
>/< cluster_configuration_file.yaml>
```

- 3.Verify that you can connect to your cluster by listing your worker nodes.

```
kubectl get nodes
```

- 4.Create the deployment.

```
kubectl create -f deployment.yaml
```

- 5.Create the service.

```
kubectl create -f service.yaml
```

## Step 5 :

Look the Kubernet Page.

kubernetes

+ CREATE

Overview

Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

Namespace

default

Overview

Workloads

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Discovery and Load Balancing

Ingresses

Services

Config and Storage

Config Maps

Persistent Volume Claims

Deployments

| Name                  | Labels         | Pods  | Age       | Images                                 |
|-----------------------|----------------|-------|-----------|--|
| flask-node-deployment | app: flasknode | 1 / 1 | 5 minutes | registry.ng.bluemix.net/flask-node/app |

Pods

| Name                                   | Node         | Status  | Restarts | Age       | CPU (cores) | Memory (bytes) |
|--|--------------|---------|----------|-----------|-------------|----------------|
| flask-node-deployment-5cd96cf6bc-d6n6x | 10.47.79.201 | Running | 0        | 5 minutes | 0           | 19.352 Mi      |

Replica Sets

| Name                             | Labels  | Pods  | Age       | Images                                 |
|----------------------------------|---|-------|-----------|--|
| flask-node-deployment-5cd96cf6bc | app: flasknode<br>pod-template-hash: 1785279267 | 1 / 1 | 5 minutes | registry.ng.bluemix.net/flask-node/app |

Discovery and Load Balancing

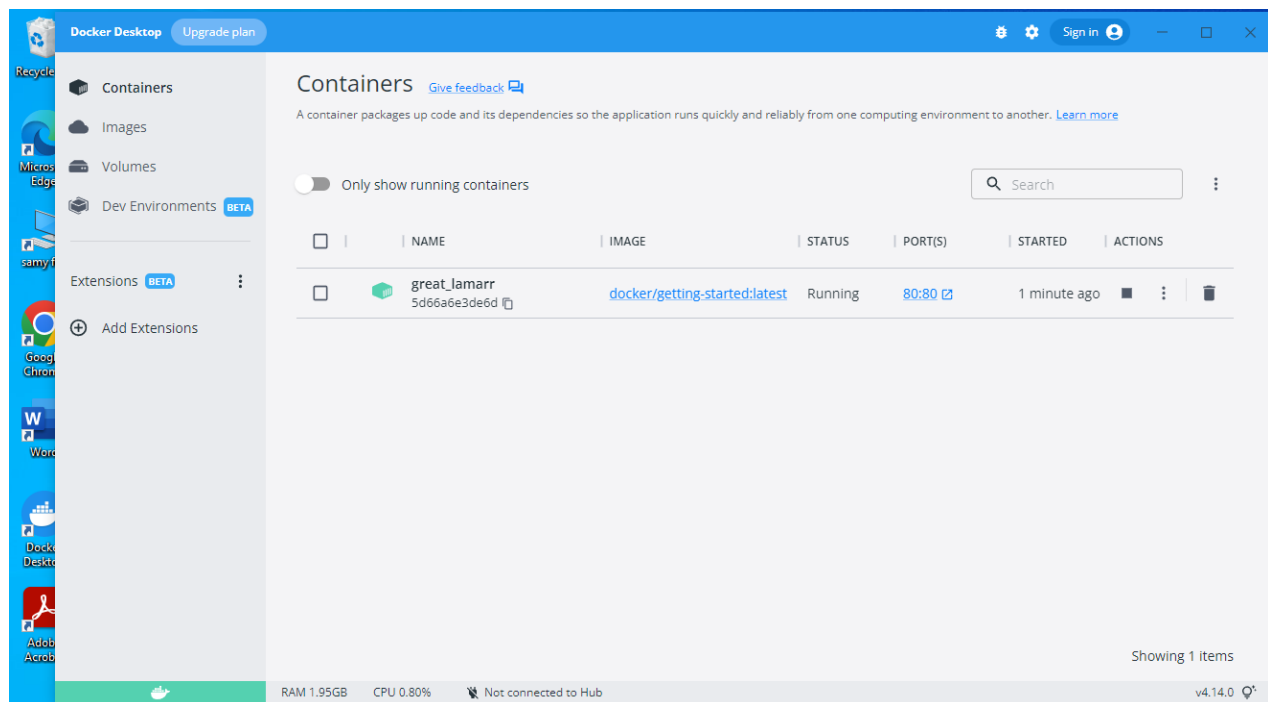
Services

| Name                  | Labels                                       | Cluster IP    | Internal endpoints  | External endpoints | Age      |
|-----------------------|--|---------------|---|--------------------|----------|
| kubernetes            | component: apiserver<br>provider: kubernetes | 172.21.0.1    | kubernetes:443 TCP<br>kubernetes:0 TCP                        | -                  | a minute |
| flask-node-deployment | -  | 172.21.104.14 | flask-node-deployment:5000 TCP<br>flask-node-deployment:0 TCP | -                  | a minute |

Config and Storage

## Step 6 :

Finally, go to the browser and ping the Public IP of your worker node.



## Step 7 :

Finally deploy the image on kubernetes Successfully.