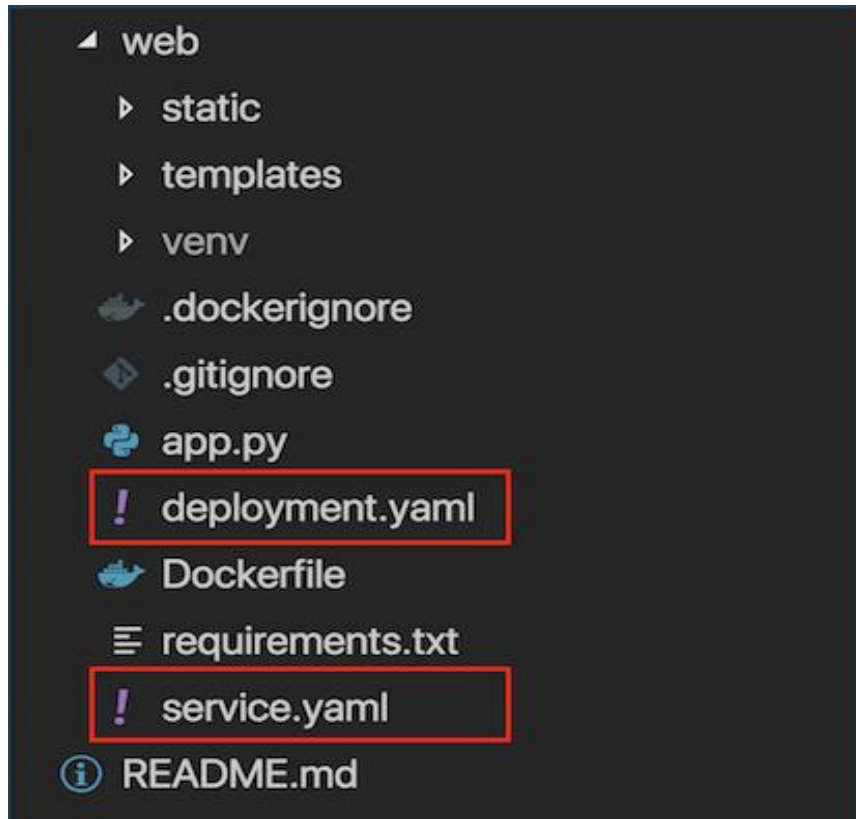


DEPLOY IN KUBERNETES CLUSTER

TEAM ID : PNT2022TMID22368

CREATE CONFIGURATION FILES FOR KUBERNETES

STEP 1: Once the image is successfully uploaded to the private registry, go to your project directory and create two files: deployment.yaml and service.yaml.



STEP 2 : In the deployment.yaml file, paste this 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 the service.yaml file, paste this code:

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

DEPLOY YOUR APPLICATION TO KUBERNETES :

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

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

STEP 2 : Set the context for the cluster in your CLI.

a. Get the command to set the environment variable and download the Kubernetes configuration files.

```
ibmcloud cs cluster-config cluster_kunal
```

b. Set the KUBECONFIG environment variable. Copy the output from the previous command and paste it in your terminal. The command output should look similar to the following.

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

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

```
kubectl get nodes
```

STEP 4 : Create the deployment.

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

STEP 5 : Create the service.

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

STEP 6 : Look at the Kubernetes dashboard from the IBM Kubernetes Service overview 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

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-5cd96cf8bc-d6n6x	10.47.79.201	Running	0	5 minutes	0	19,382 Mi

Replica Sets

Name	Labels	Pods	Age	Images
flask-node-deployment-5cd96cf8bc	app: flasknode 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 provider: kubernetes	172.21.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	a minute
flask-node-deployment	-	172.21.104.14	flask-node-deployment:5000 TCP flask-node-deployment:0 TCP	-	a minute

Config and Storage

STEP 7 : Finally, go to your browser and ping the Public IP of your worker node.

