INVENTORY MANAGEMENT SYSTEM FOR RETAILERS

(Online Organic Store Management)

Domain: Cloud Application Development

Team Id: PNT2022MID10625

Team Leader: Arularasi. J – 191021013

Team Members:

Abinaya. D – 191021002

Abitha. N – 191021005

Dhanya Lakshmi. P – 191021022

Ezhilmathy. E - 191021027

DEPLOYMENT OF APP IN IBM CLOUD:

STEP 1: Containerize the App

STEP 2: Upload the image to IBM Registry

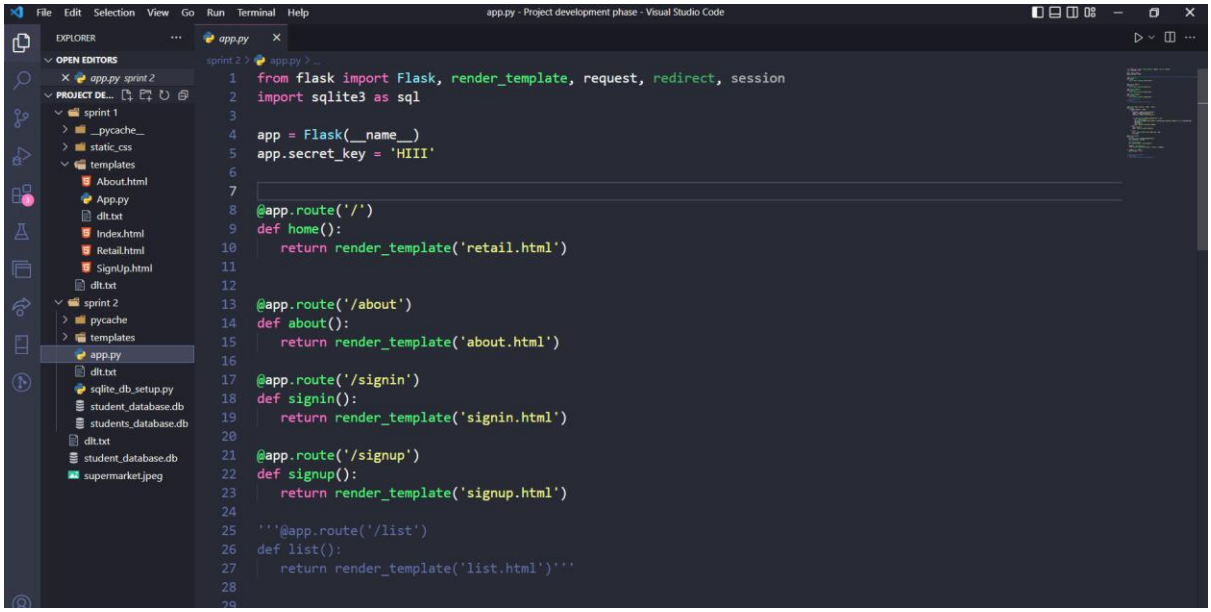
STEP 3: Deploy in Kubernates Cluster

STEP 1 : CONTAINERIZE THE APP:

Docker Image Creation

Make Project Folder

Open your terminal and make a folder for your flask application

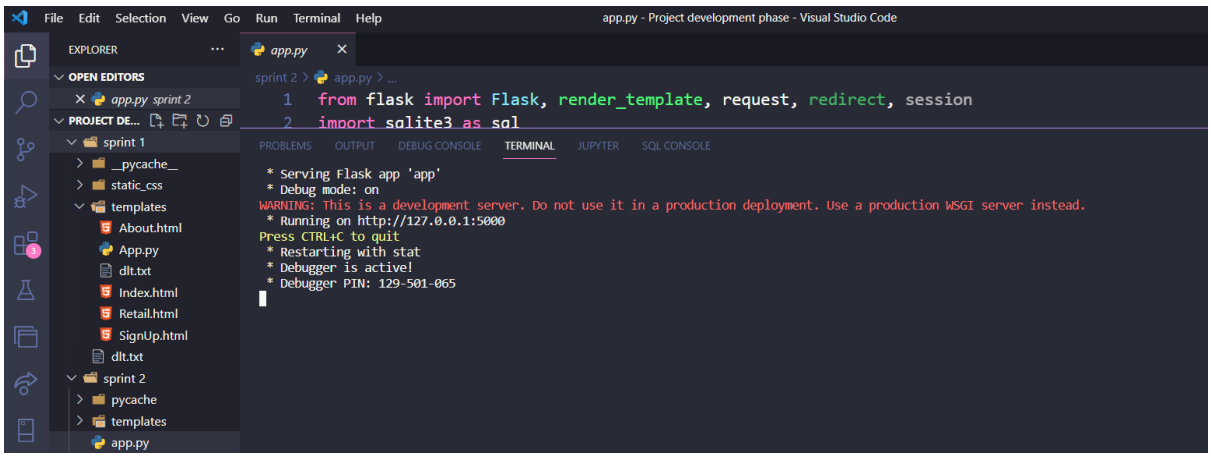


Insert the code to the Dockerfile:

```
FROM python:alpine3.7
COPY . /app
WORKDIR /app
RUN pip install -r requirements.txt
EXPOSE 5001
ENTRYPOINT [ "python" ]
CMD [ "app.py" ]
```

Test the Flask App:

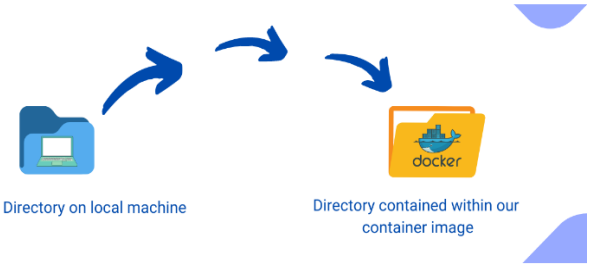
It should start our development server which comes with the flask on “http://0.0.0.0:5001/”.



Creating Docker image of the project:

```
sudo docker build --tag flask-docker-demo-app .
```

The above command will create an app with the tag flask-docker-demo-app.




```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                                    zsh
hiteshchoudhary~$docker build -t hiteshchoudhary/hey-python-flask:0.0.1.RELEASE .
[+] Building 7.2s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                  0.0s
 => => transferring dockerfile: 161B                                                  0.0s
 => [internal] load .dockerignore                                                     0.0s
 => => transferring context: 2B                                                       0.0s
 => [internal] load metadata for docker.io/library/python:3-alpine3.15                3.0s
 => [auth] library/python:pull token for registry-1.docker.io                         0.0s
 => [1/4] FROM docker.io/library/python:3-alpine3.15@sha256:d89ac9cefd2213b99a792fd8ec4f15c0297d9340a688b  0.0s
hiteshchoudhary~$docker container run -d -p 3000:3000 hiteshchoudhary/hey-python-flask:0.
Unable to find image 'hiteshchoudhary/hey-python-flask:0.' locally
^C
hiteshchoudhary~$docker container run -d -p 3000:3000 hiteshchoudhary/hey-python-flask:0.0.1.RELEASE

hiteshchoudhary~$
```

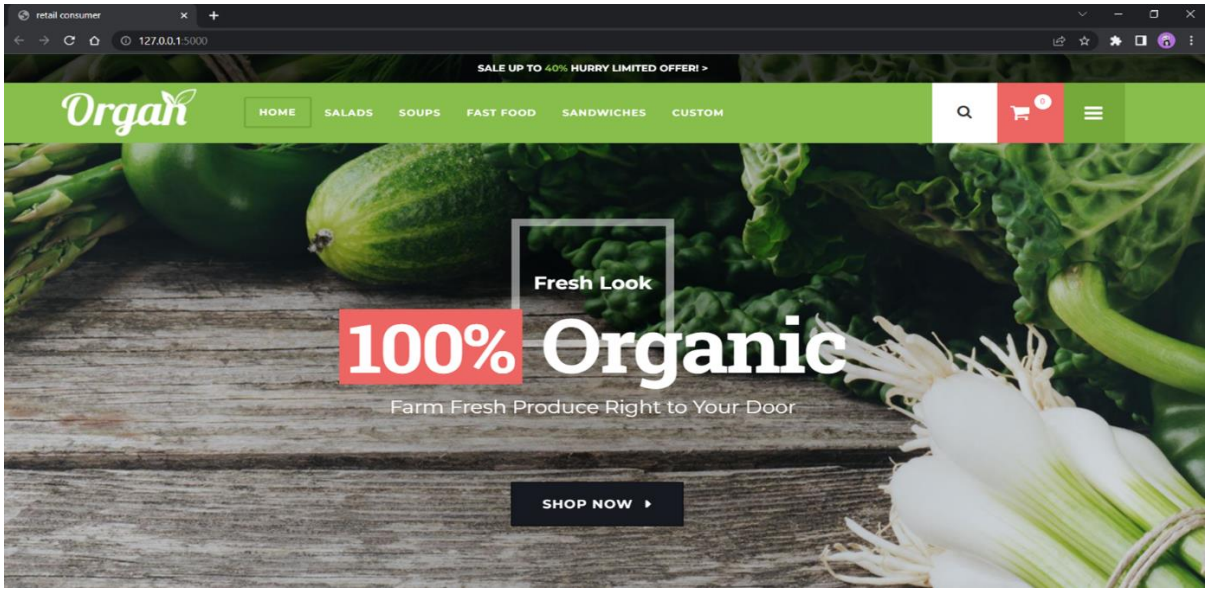
Run the docker image:

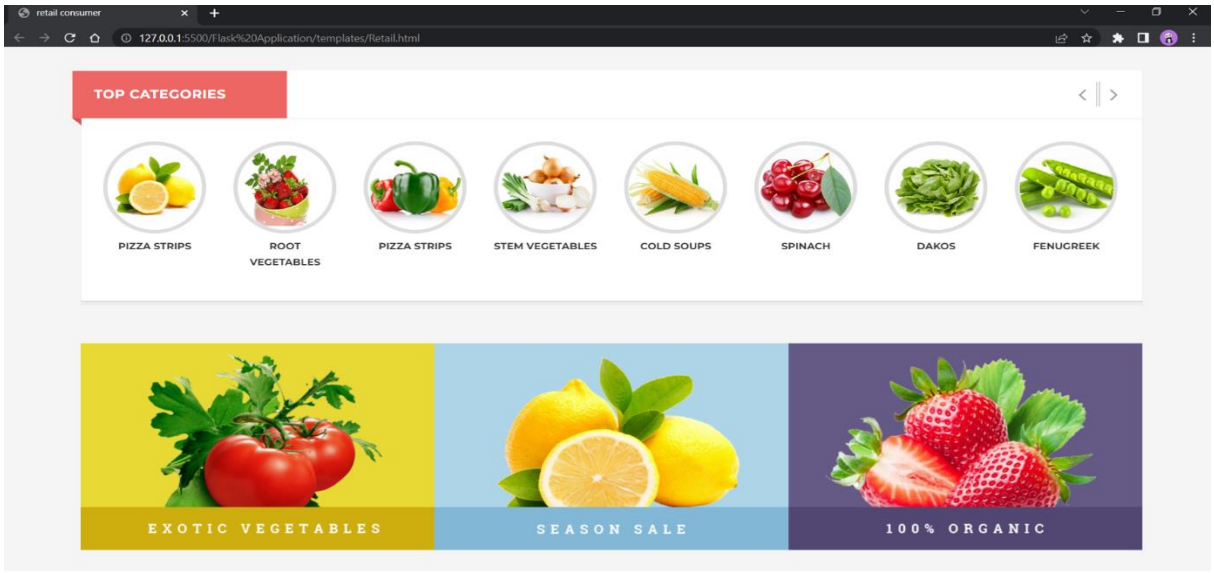
```
sudo docker run --name flask-docker-demo-app -p 5001:5001 flask-docker-demo-app
```

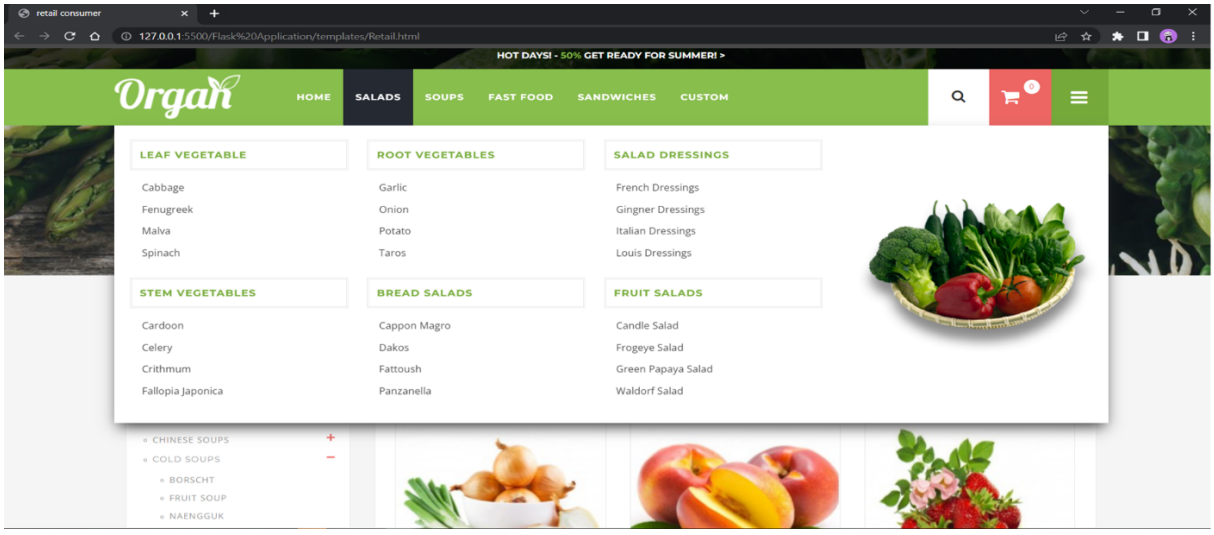
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                                    zsh
hiteshchoudhary~$docker container run -d -p 3000:3000 hiteshchoudhary/hey-python-flask:0.
Unable to find image 'hiteshchoudhary/hey-python-flask:0.' locally
^C
hiteshchoudhary~$docker container run -d -p 3000:3000 hiteshchoudhary/hey-python-flask:0.0.1.RELEASE

hiteshchoudhary~$docker container run -d -p 3000:3000 hiteshchoudhary/hey-python-flask:0.0.1.RELEASE
076f9da1074488b9cd7a76ca96cafb44fbba1e43b8ac6a9832ccba38aad74c7d
hiteshchoudhary~$
```

Test the App:

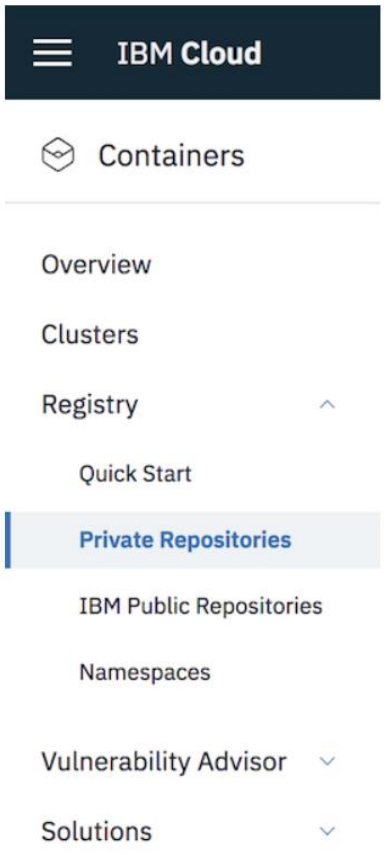

retail consumer
127.0.0.1:5000
SALE UP TO 40% HURRY LIMITED OFFER! >
Orgah
HOME
SALADS
SOUPS
FAST FOOD
SANDWICHES
CUSTOM
Fresh Look
100% Organic
Farm Fresh Produce Right to Your Door
SHOP NOW


retail consumer
127.0.0.1:5500/Flask%20Application/templates/Retail.html
TOP CATEGORIES
PIZZA STRIPS
ROOT VEGETABLES
PIZZA STRIPS
STEM VEGETABLES
COLD SOUPS
SPINACH
DAKOS
FENUGREEK
EXOTIC VEGETABLES
SEASON SALE
100% ORGANIC


retail consumer
127.0.0.1:5500/Flask%20Application/templates/Retail.html
HOT DAYS! - 50% GET READY FOR SUMMER! >
Orgah
HOME
SALADS
SOUPS
FAST FOOD
SANDWICHES
CUSTOM
LEAF VEGETABLE
Cabbage
Fenugreek
Malva
Spinach
ROOT VEGETABLES
Garlic
Onion
Potato
Taros
SALAD DRESSINGS
French Dressings
Gingner Dressings
Italian Dressings
Louis Dressings
STEM VEGETABLES
Cardoon
Celery
Crithmum
Fallopia Japonica
BREAD SALADS
Cappon Magro
Dakos
Fattoush
Panzanella
FRUIT SALADS
Candle Salad
Frogeye Salad
Green Papaya Salad
Waldorf Salad
CHINESE SOUPS
COLD SOUPS
BORSCHT
FRUIT SOUP
NAENGGUK

STEP 2 : UPLOAD THE IMAGE TO IBM CONTAINER REGISTERY:

1. From your account dashboard, go to IBM Cloud Kubernetes Service.
2. From the left navigation menu, select Private Repositories.



Install the Container Registry plug-in:

```
ibmcloud plugin install container-registry -r "IBM Cloud"
```

Log in to your IBM Cloud account:

```
ibmcloud login -a <cloud_foundary_end_point_for_the_region>
```

Name and create your namespace. Use this namespace for the rest of the Quick Start.

```
ibmcloud cr namespace-add <namespace>
```

Log your local Docker daemon into the IBM Cloud Container Registry.

```
ibmcloud cr login
```

Push the image:

```
kunals-mbp:web kunalmalhotra$ docker push registry.ng.bluemix.net/flask-node/app:latest
The push refers to repository [registry.ng.bluemix.net/flask-node/app]
a905410b27c1: Pushed
b96dea950728: Pushed
437e8db4a234: Pushed
ba9884d50644: Pushed
1989aa0f3739: Layer already exists
7bec9e49c283: Layer already exists
1172bcd1177f: Layer already exists
8eb4c3a69e64: Layer already exists
1fa8778eb779: Layer already exists
fa0c3f992cbd: Layer already exists
ce6466f43b11: Layer already exists
719d45669b35: Layer already exists
3b10514a95be: Layer already exists
latest: digest: sha256:5015254c21592b5ab08168707b74ddd763e97e80b59d9187afa2a80433b9d2ab size: 3061
kunals-mbp:web kunalmalhotra$
```

Verify that your image is in your private registry:

```
ibmcloud cr image-list
```

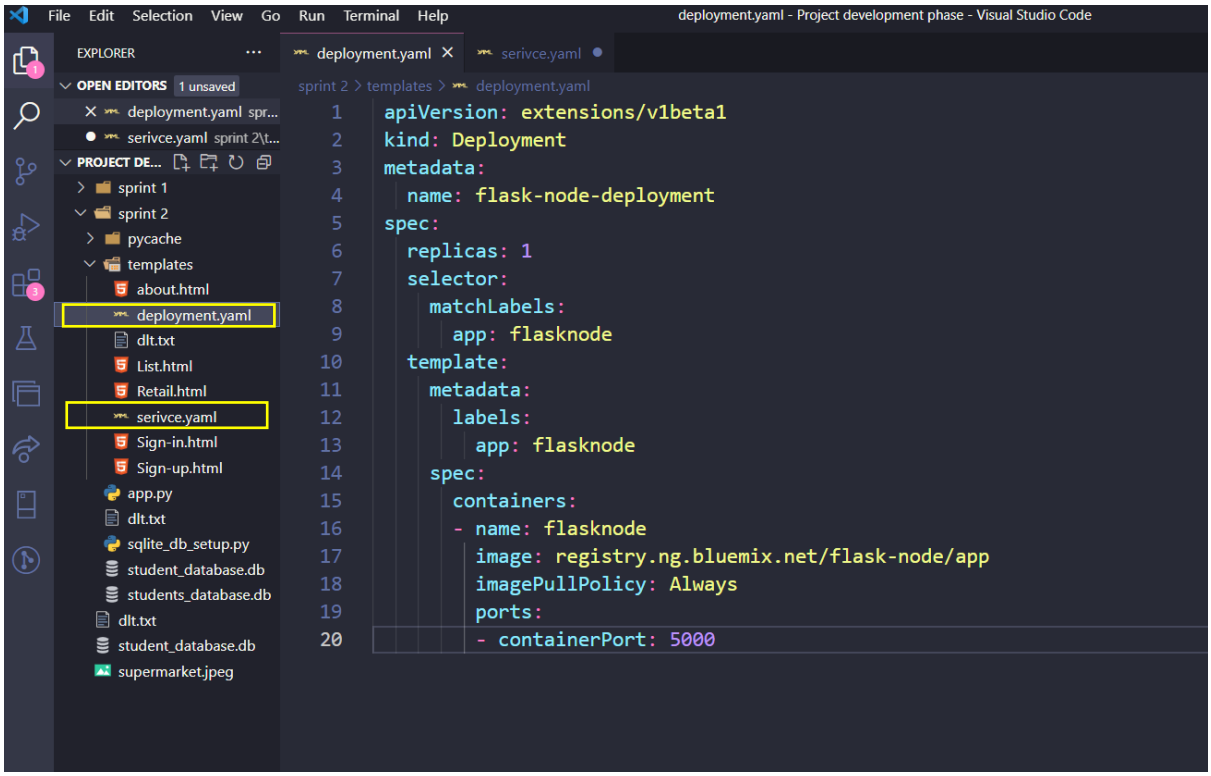
```
kunals-mbp:web kunalmalhotra$ ibmcloud cr image-list
Listing images...

REPOSITORY                                TAG      DIGEST         NAMESPACE    CREATED      SIZE     SECURITY STATUS
registry.ng.bluemix.net/flask-node/app    latest   b721dd768fe0   flask-node   1 day ago    366 MB   3 Issues

OK
kunals-mbp:web kunalmalhotra$
```

STEP 3: Deploy in Kubernates Cluster

Create Configuration files for Kubernates



Deploy Application to Kubernates:

```
ibmcloud cs region-set us-south
ibmcloud cs cluster-config cluster_kunal
```

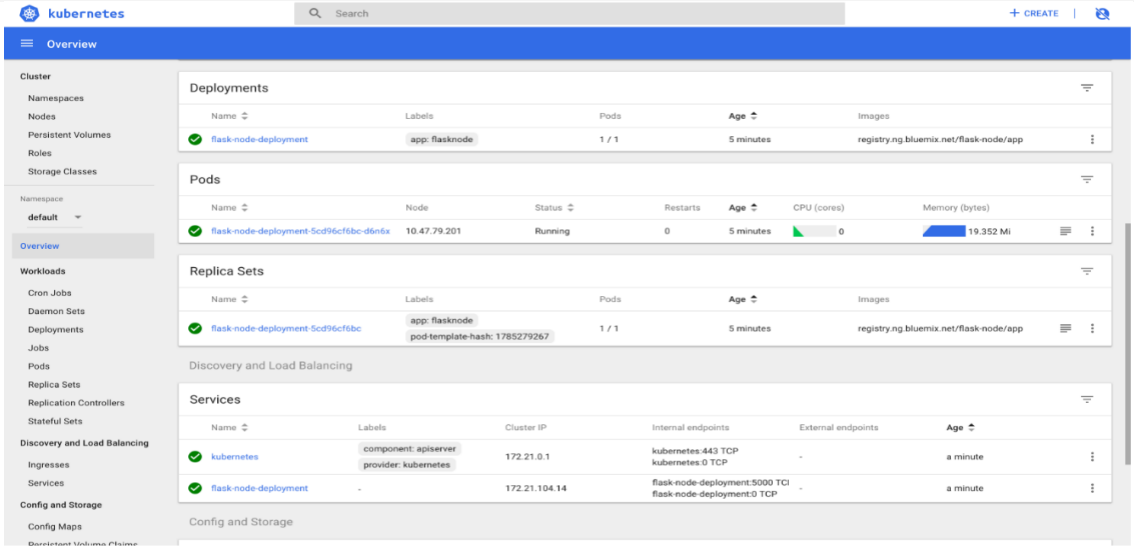
Set the KUBECONFIG environment variable. Copy the output from the previous command and paste it in your terminal.

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

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

```
kubectl get nodes
kubectl create -f deployment.yaml
kubectl create -f service.yaml
```

Look at the Kubernetes dashboard from the IBM Kubernetes Service overview page:



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