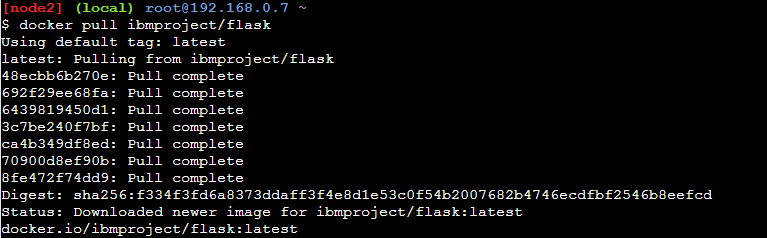
**Assignment -4**

Docker and Kubernetes

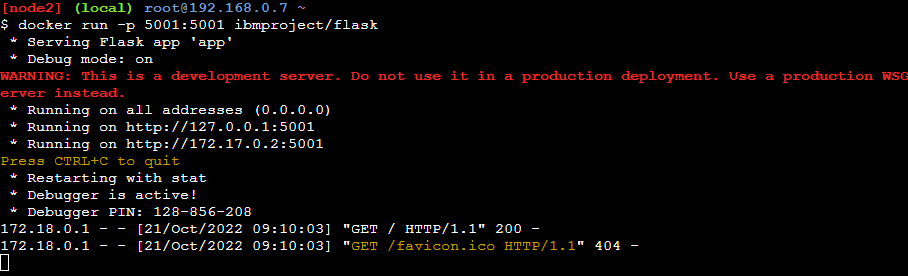
|  |  |
| --- | --- |
| Student Name | Jasmine Pramila I |
| Student Roll Number | 910019106016 |

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

Pull Image from docker hub



Running the image



Output



Flask program Containerized and run

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello():

return "welcome to the flask "

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host ='0.0.0.0', port = 5001, debug = True)

Dockerfile

FROM python:alpine3.7

COPY . /app

WORKDIR /app

RUN pip install -r requirements.txt

EXPOSE 5001

ENTRYPOINT [ "python" ]

CMD [ "app.py" ]

**2.Create a dockerfile and deploy it in docker desktop**

**Flask application**

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello():

return "welcome to the flask"

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host ='0.0.0.0', port = 5001, debug = True)

**Dockerfile**

FROM python:alpine3.7

COPY . /app

WORKDIR /app

RUN pip install -r requirements.txt

EXPOSE 5001

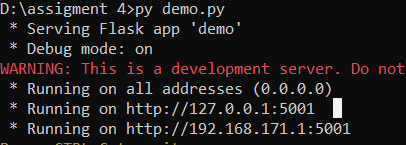
ENTRYPOINT [ "python" ]

CMD [ "demo.py" ]

**Requirement.txt**

Flask

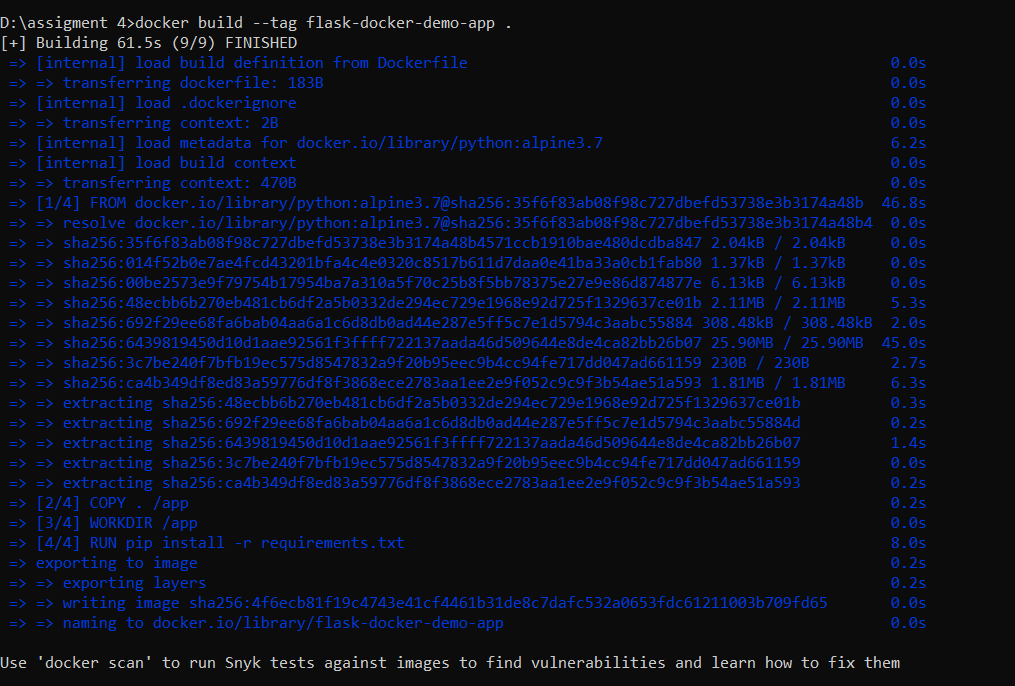
Running Flask app



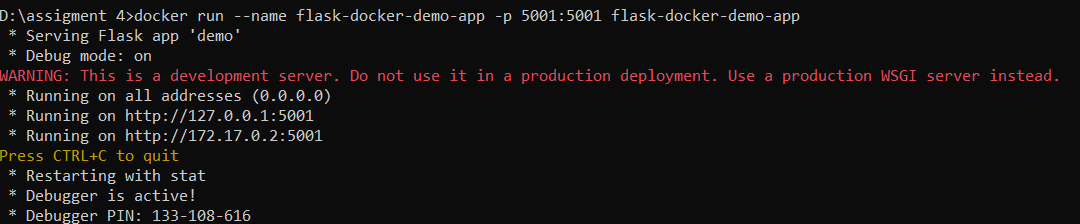
Flask app output



**Build docker image**

****

**Run docker image**

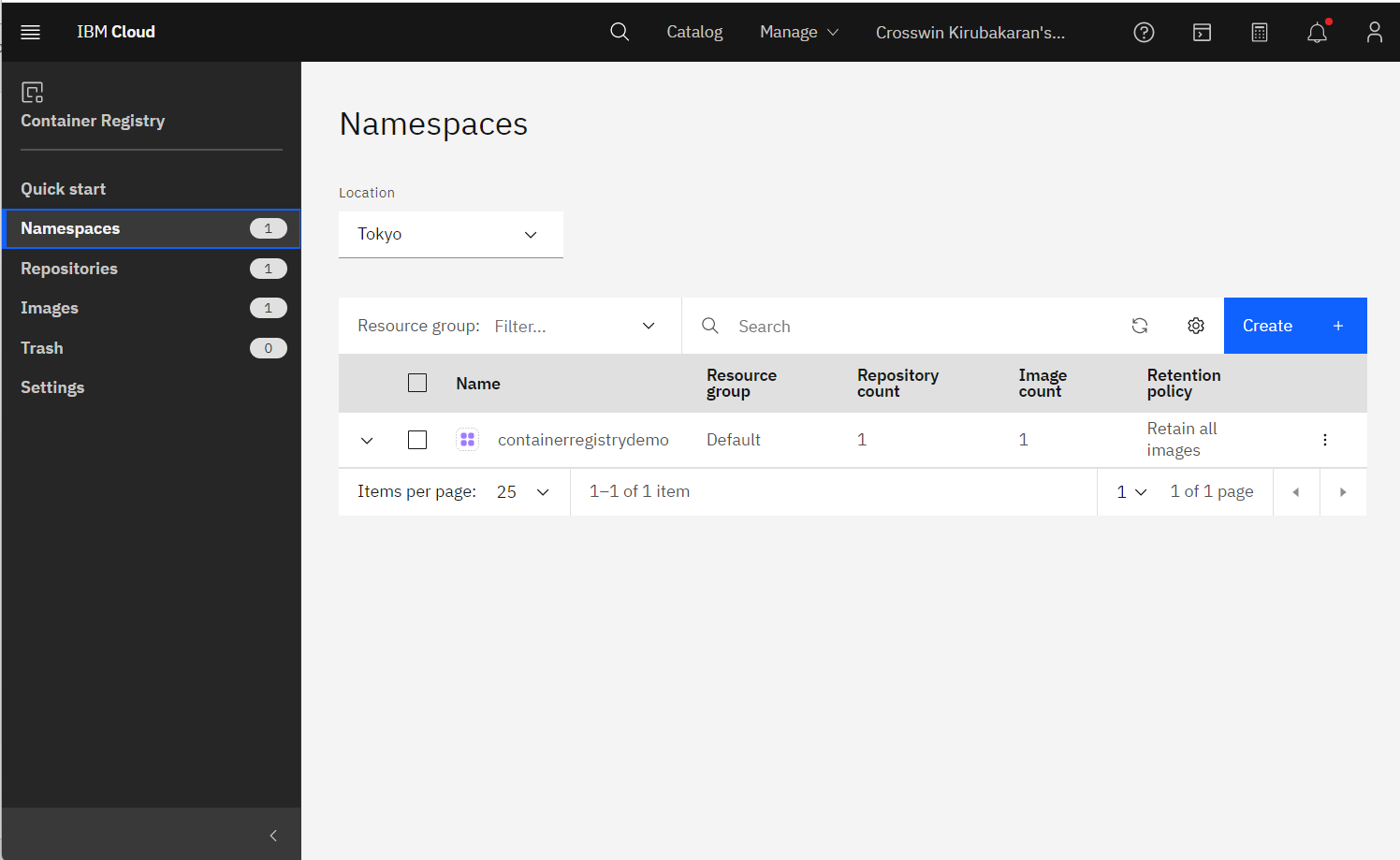
****

Container output

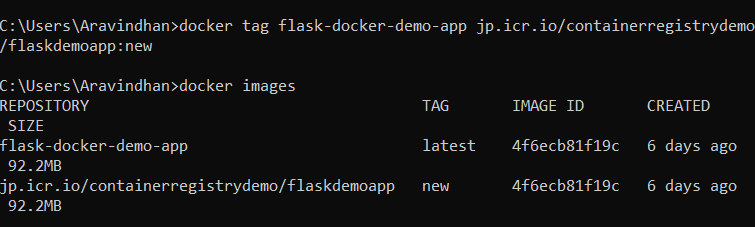


3.Create IBM container registry and push flask app

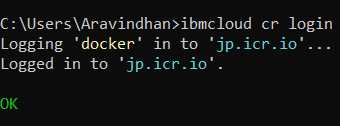
Create an namespace –(containerregistrydemo)



Change docker image name



Login to container registry



Push the image to container registry

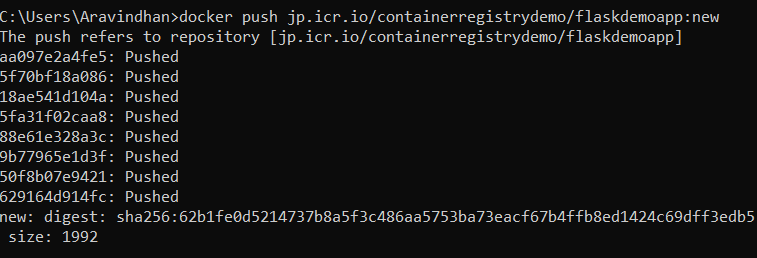
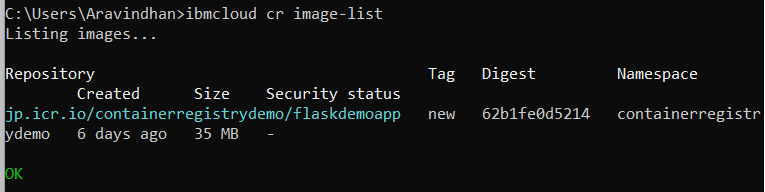


Image list in container registry



4.Deploy the flask app in IBM Kubernetes cluster and expose it in nodeport

Deployment.yaml file

apiVersion: apps/v1

kind: Deployment

metadata:

name: flasknode

spec:

replicas: 2

selector:

matchLabels:

app: flasknode

template:

metadata:

labels:

app: flasknode

spec:

containers:

- name: flasknode

image: jp.icr.io/containerregistrydemo/flaskdemoapp:new

imagePullPolicy: Always

ports:

- containerPort: 5001

Service.yaml file

apiVersion: v1

kind: Service

metadata:

name: flasknode

spec:

ports:

- port: 5001

targetPort: 5001

selector:

app: flasknode

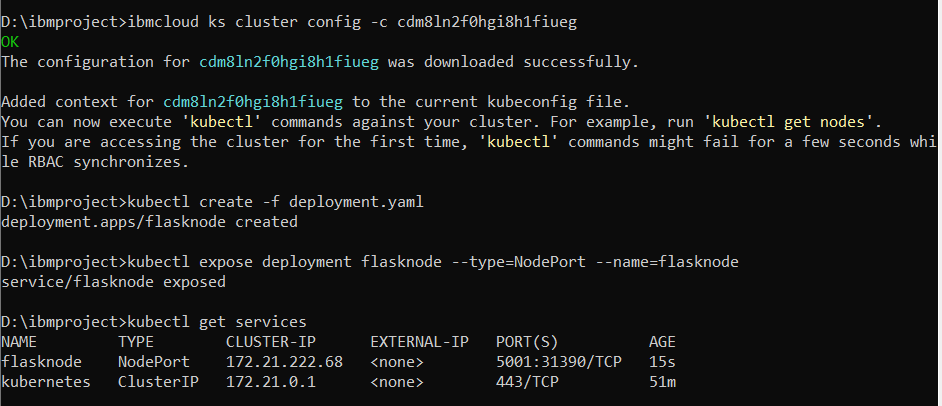
protocol: TCP

1.connect to IBM ks cluster

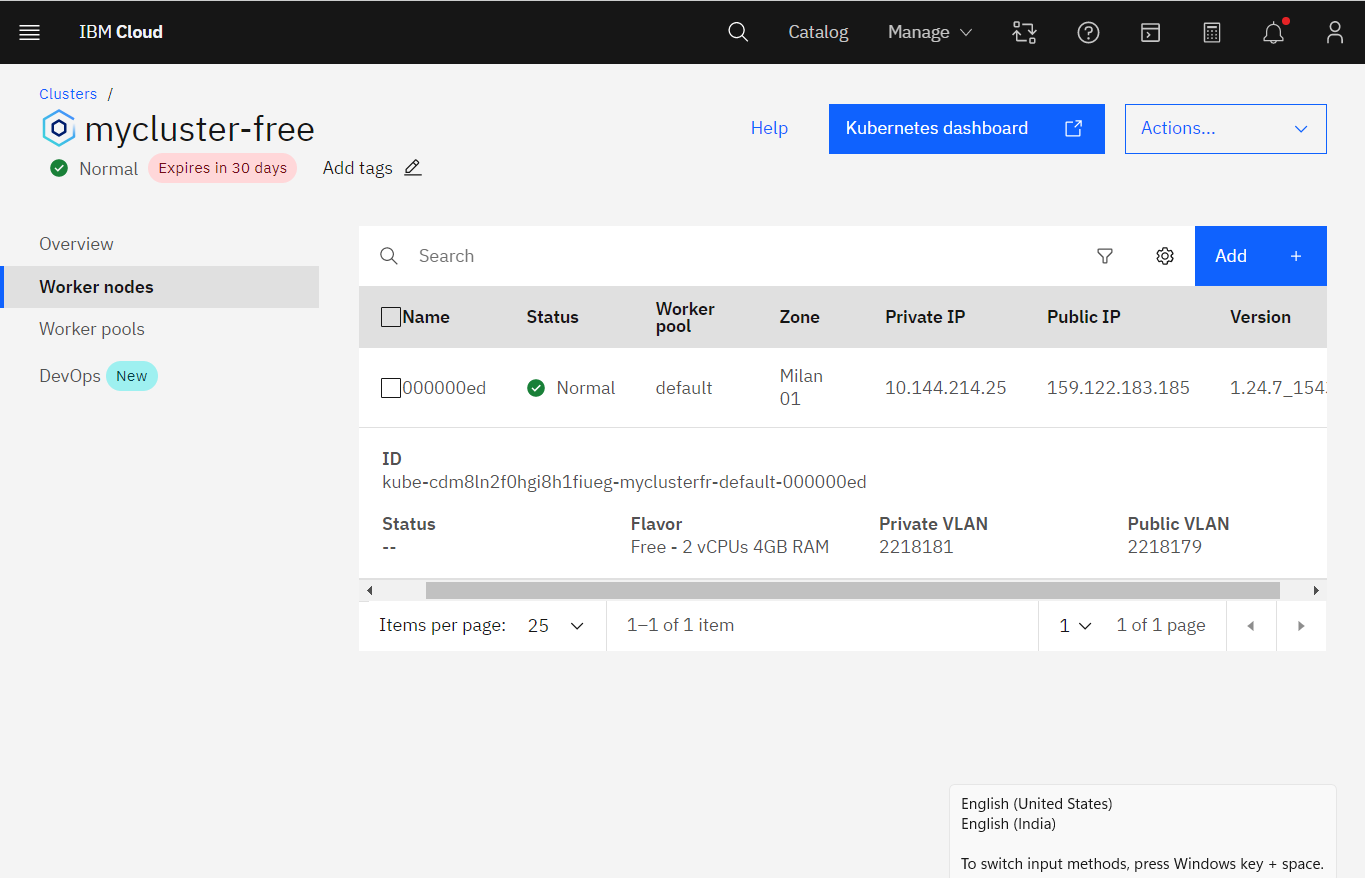
2 create deployment

3. create a service

4. get services to get the service port



Public ip is available in Kubernetes cluster dashboard



Output

