**ASSIGNMENT – 4**

**KUBERNETES, DOCKER**

|  |  |
| --- | --- |
| Assignment Date | 28 October 2022 |
| Student Name | Malavika.A.C |
| Student Roll Number | 921319104108 |
| Maximum Marks | 2 Mark |

**QESTION-1:**

Pull an image from Dockers hub and run it in Dockers playground

**SOLUTION:**

**STEPS:**

**STEP 1**-Login to Dockers hub and get an image

**STEP 2**-• Open Dockers playground

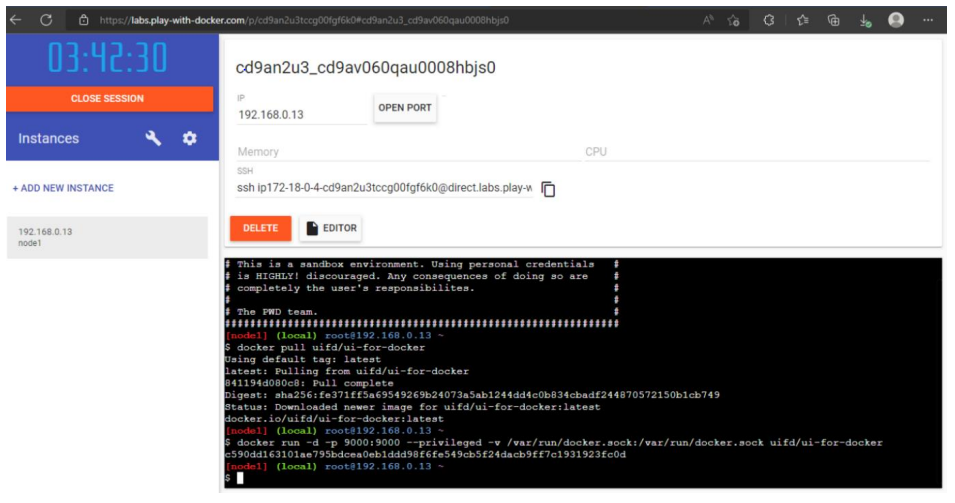
• Login with Dockers

• Create new instance

**STEP 3**-In the command prompt run the following:

$ docker pull hello-world //To pull an image from docker hub

$ docker run hello-world //To run the image in docker playground



**QUESTION-2:**

Create a Dockers file and deploy it in Dockers desktop application

**SOLUTION:**

**STEP 1:**

• Create a flask application

• Create a Dockerfile in the same folder

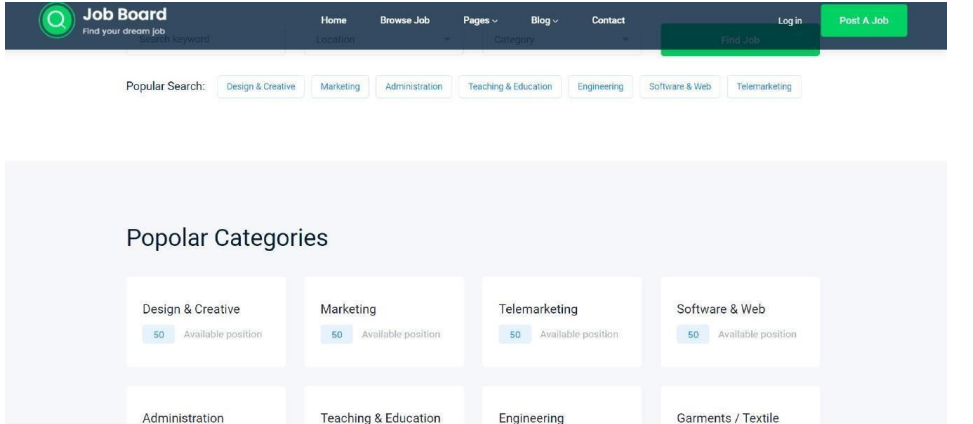
**STEP 2:**

Run the following commands to deploy it in docker desktop

$ docker build –t jobportal // to deploy all the folders to docker desktop

$ docker image //to show the list of images in docker desktop

$ docker container run –p 5000:5000 jobportal //to run



**Question-3:**

Create a IBM container registry and deploy hello world app or job portal app.

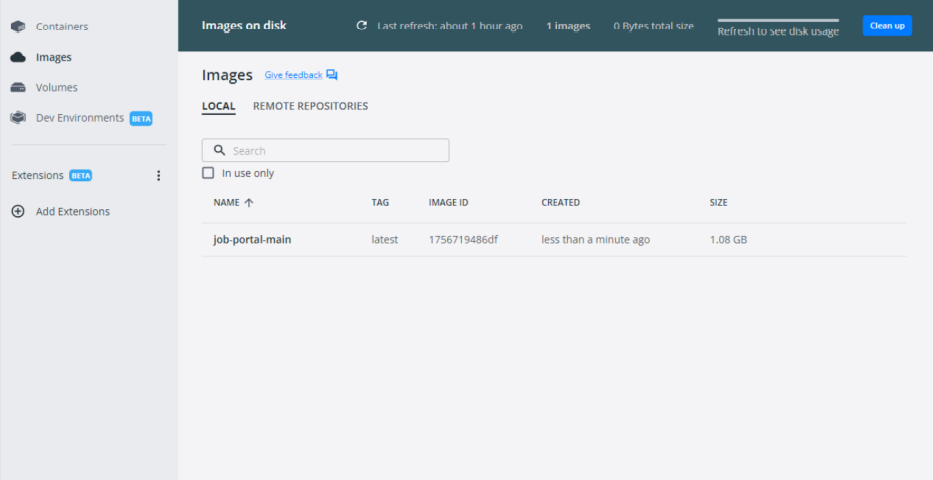
**Solution:**

● Log into IBM cloud

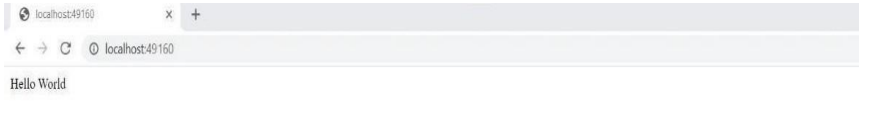
● Create a container registry

● Using IBM Cloud CLI, install the container registry plugin in our system

● Push our docker image into the created container registry using docker push so, our job portal app is deployed in the IBM container registry



**Output:**



**Question-4:**

Create a Kubernetes cluster in IBM cloud and deploy hello world image or jobportal image and also expose the same app to run in nodeport.

**Solution:**

● Log into IBM cloud

● Create a kubernete

● Using IBM Cloud CLI, install the ks plugin in our system

● Create a cluster in the kubernetes

● Now, go to the kubernetes dashboard where we need to create a service based on a yml file (given below)

● In that file, we have to mention which image we are going to use and the app name

● Take the public IP address and Nodeport since we exposed the flask app in nodeport

● Finally, we got the URL address where our flask app is hosted

**CODE:**

apiVersion: v1 kind:

Service metadata:

name: job-portal-app

spec: selector:

app: job-portal-app

ports: - port: 5000

type: NodePort

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: job-portal-app

labels:

app: job-portal-app

spec: selector:

matchLabels: app:

job-portal-app

replicas: 1

template:

metadata: labels:

app: job-portal-app

spec: containers:

name: job-portal-app

image: image\_name

ports:

containerPort: 5000

env:

name: DISABLE\_WEB\_APP

value: "false”

