

## Assignment - 4

Student Name	Suryaprakash P
Student Roll Number	73771914181

### Question-1:

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

### Solution:

- Pull an image uifd/ui-for-docker from the docker hub
- This image is used for viewing and managing the docker engine
- Use docker pull image\_name and docker run -it image\_name commands to
- run the above image in the Docker Playground

The screenshot shows the Docker Playground interface in a web browser. The top bar displays the time 03:58:11 and a 'CLOSE SESSION' button. Below this, there's a sidebar with 'Instances' and a '+ ADD NEW INSTANCE' button. The main area shows a terminal session for an instance named 'cdmuqov9\_cdmuqpv91rrg009jdp1g'. The terminal output shows the user running 'docker images' and then 'docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker'. The output indicates that the image 'uifd/ui-for-docker:latest' was pulled from Docker Hub. The terminal also shows the user running 'docker images' again, listing the pulled image.

```
# This is a sandbox environment. Using personal credentials #
# is HIGHLY! discouraged. Any consequences of doing so are #
# completely the user's responsibilities. #
#
# The FWD team. #
#####
[node1] (local) root@192.168.0.8 ~
$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
[node1] (local) root@192.168.0.8 ~
$ docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
Unable to find image 'uifd/ui-for-docker:latest' locally
latest: Pulling from uifd/ui-for-docker
841194d080c8: Pull complete
Digest: sha256:fe371ff5a69549269b24073a5ab1244dd4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
079a3d708e31c90dd6d551833fa13816a6b1c4bbff759d5c720ca7afd8c50888
[node1] (local) root@192.168.0.8 ~
$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
uifd/ui-for-docker latest 965940f98fa5 6 years ago 8.1MB
[node1] (local) root@192.168.0.8 ~
$
```

## Question-2

Create a docker file for the jobportal application and deploy it in Docker desktop application.

### Solution:

- Create a docker file for build and deploy flask app.
- Use docker build -t image\_name . in the current directory to start building the
- docker image and deploy in our local docker
- Use docker run -p 5000:5000 image\_name to run in local system

### CODE

```
FROM ubuntu/apache2
```

```
FROM python
```

```
COPY ./requirements.txt /flaskApp/requirements.txt
```

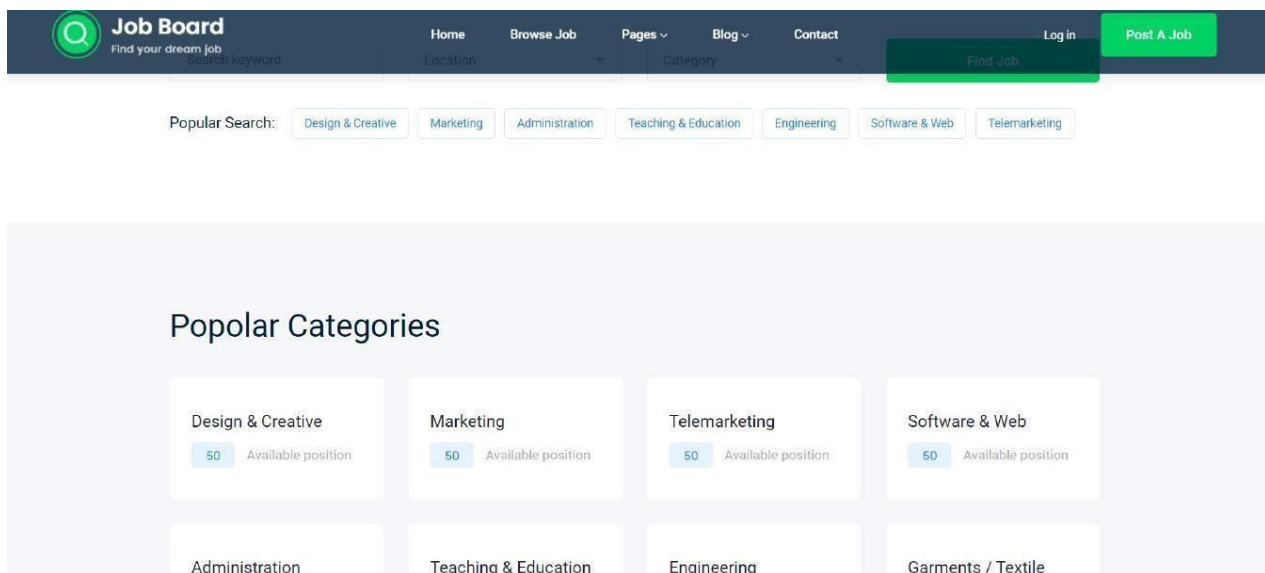
```
WORKDIR /flaskApp
```

```
RUN pip install -r requirements.txt
```

```
COPY . /flaskApp
```

```
ENTRYPOINT [ "python" ]
```

```
CMD ["app.py" ]
```

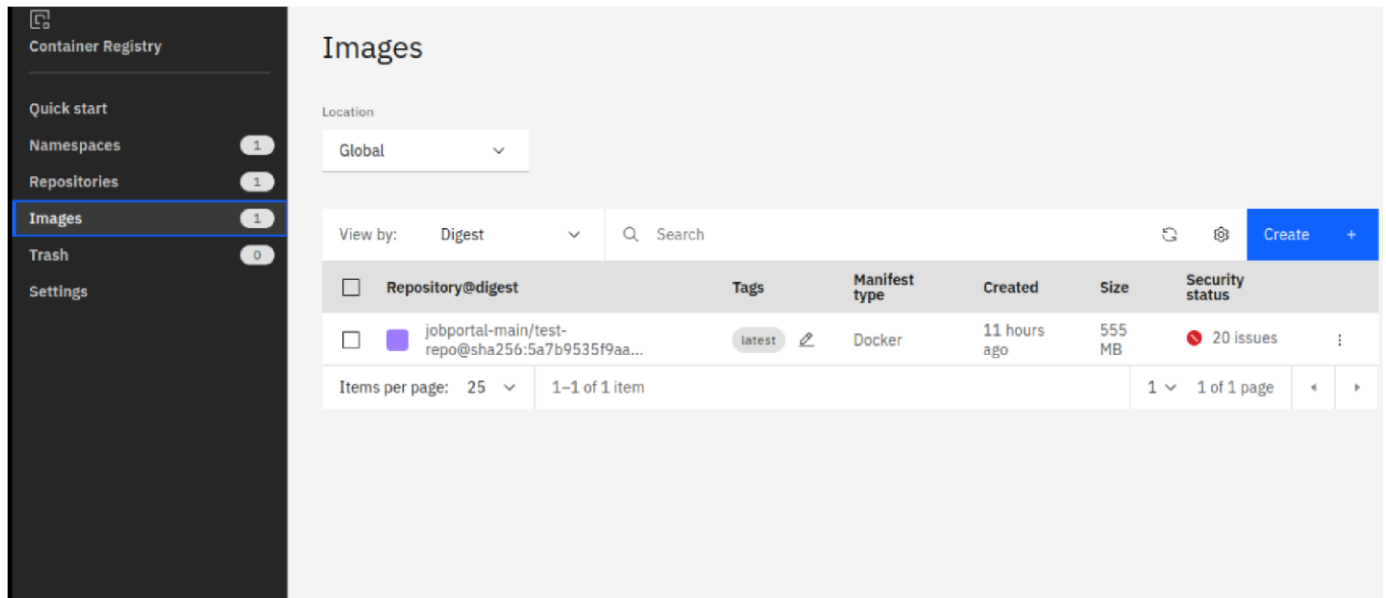


### Question-3:

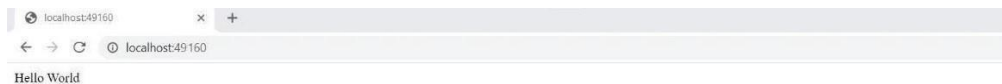
Create a IBM container registry and deploy hello world app or job portal app.

#### 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: "HELLO  
WORLD"



#### Question-4:

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

#### Solution:

- Log into IBM cloud
- Create a kubernetes
- 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
- yaml 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"

kubernetes

default

Search

+

Create

Service

Ingresses

Services

Config and Storage

Config Maps

Persistent Volume Claims

Secrets

Storage Classes

Cluster

Cluster Role Bindings

Cluster Roles

Events

Namespaces

Network Policies

Nodes

Create from input

Create from file

Create from form

Select YAML or JSON file specifying the resources to deploy to the currently selected namespace. [Learn more](#)

Choose YAML or JSON file

Upload

Cancel

Kubernetes clusters

Resource group: Filter...

Location: Filter...

Search

Create cluster +

Name	State	Location	Worker count	Created	Version	Infrastructure
jaga-cluster	Normal	Amsterdam 03	1	Expires in 30 days	1.23.12_1546	Classic

Items per page: 25

1-1 of 1 item

11 of 1 page