

## ASSIGNMENT 4

### TEAM ID : PNT2022TMID17544

#### Question 1:

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



The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:57:32, a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button. Below that, a list of instances shows '192.168.0.8 root:1'. The main panel displays the instance details for 'cddvksm0\_cddvkvm0qau000a07j5g'. It shows the IP '192.168.0.8', a memory usage of '1.24% (49.52MiB / 3.906GiB)', and a CPU usage of '0.31%'. There's an 'OPEN PORT' button and an 'SSH' command: 'ssh ip172-18-0-22-cddvksm0qau000a07j5g@direct.labs.plr'. Below this are 'DELETE' and 'EDITOR' buttons. The terminal window is open, showing a warning message and the command 'docker pull hello-world' being executed. The output shows the image being pulled from Docker Hub.

```
WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY discouraged. Any consequences of doing so are
# completely the user's responsibilities.
#
# The PwD team.
#####
[redhat] (local) root@192.168.0.8 -
$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29718123e: Pull complete
Digest: sha256:c181fa77aefae0d7a671ab3ec3ee05414477c951ab1a6f352a06974245fe7
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
[redhat] (local) root@192.168.0.8 -
$ docker run hello-world
```



The screenshot shows the Docker Playground interface after running the 'hello-world' container. The clock now shows 03:57:05. The instance details are the same, but the CPU usage is now '0.39%'. The terminal window shows the output of the 'docker run hello-world' command, which includes a list of steps explaining how Docker pulled the image and ran the container. The output also includes a message about trying something more ambitious with 'docker run -it ubuntu bash' and links to Docker documentation.

```
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

[redhat] (local) root@192.168.0.8 -
$
```

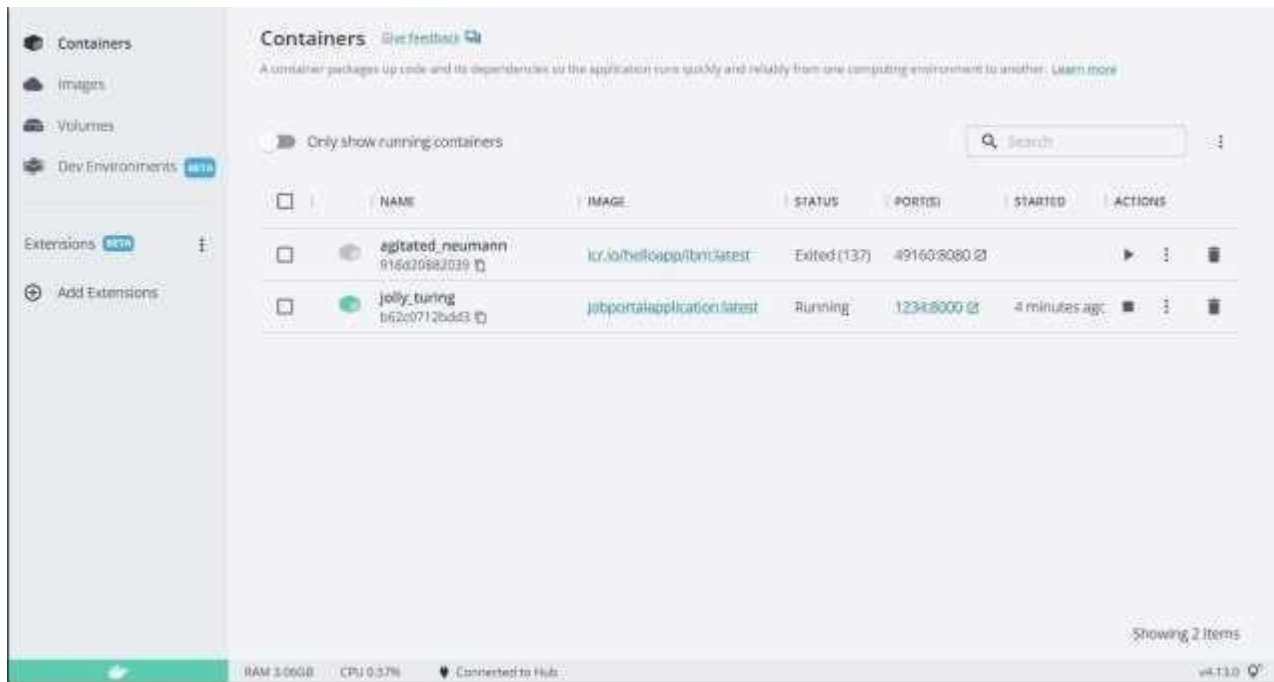
## Question 2:

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

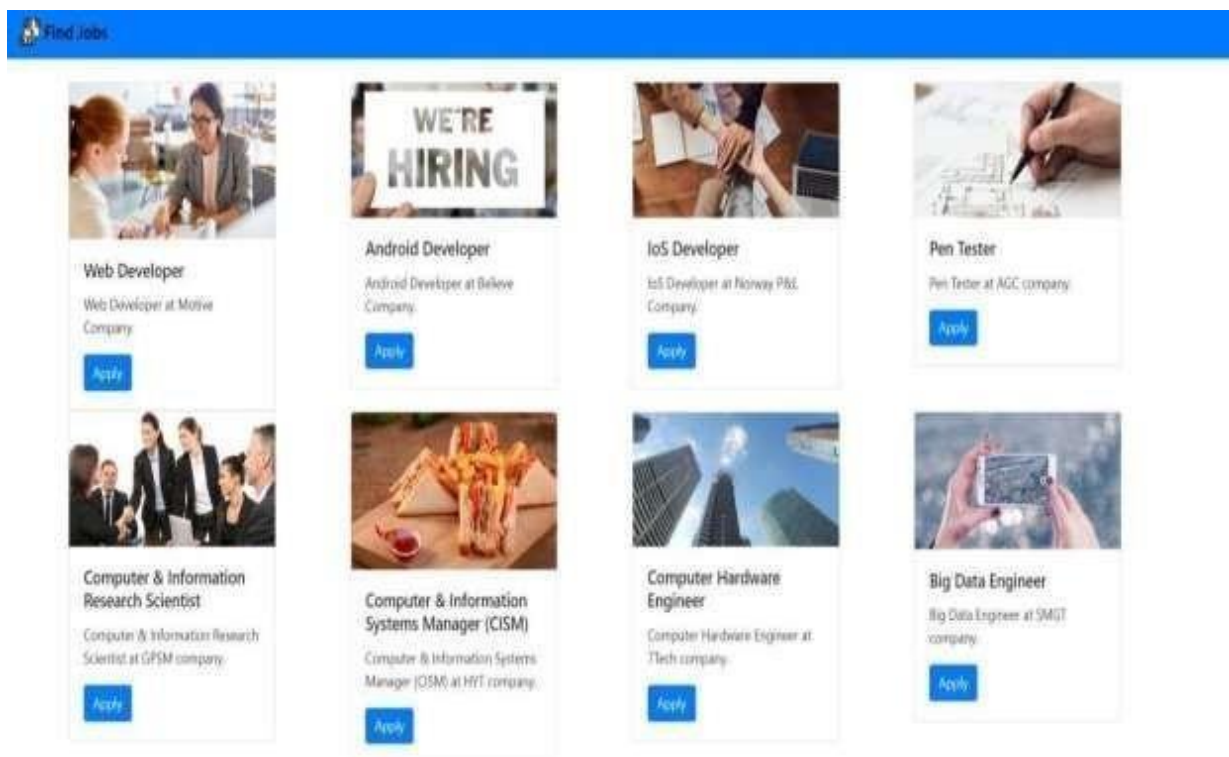
DOCKERFILE:

```
1 FROM python:3.8-buster
2
3 WORKDIR /app
4
5 COPY requirements.txt /app/
6
7 RUN pip install -r requirements.txt
8
9 COPY . /app/
10
11 RUN cp .env.dev.sample .env
12
13 EXPOSE 8000
14
15 RUN chmod +x entrypoint.sh
16
17 CMD ["sh", "entrypoint.sh"]
```

## DEPLOYMENT OF JOBPORTAL APPLICATION:



## OUTPUT:

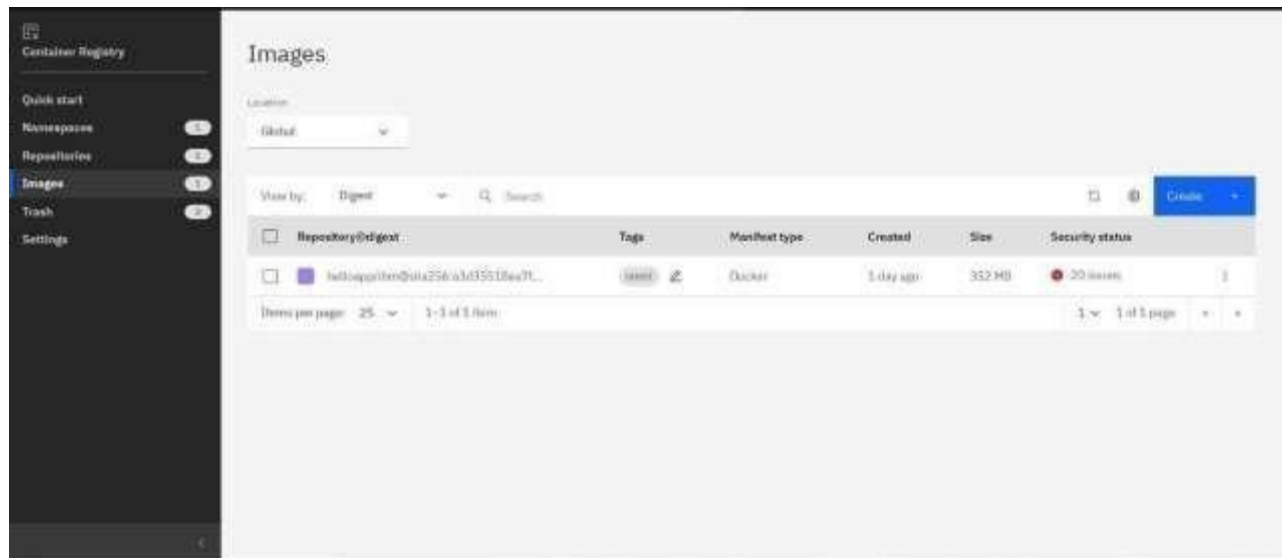


### Question 3:

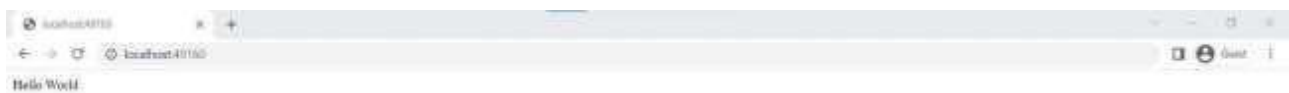
Create a IBM container registry and deploy hello world

appor jobportapp.IBM CONTAINER REGISTRY

DEPLOYMENT:

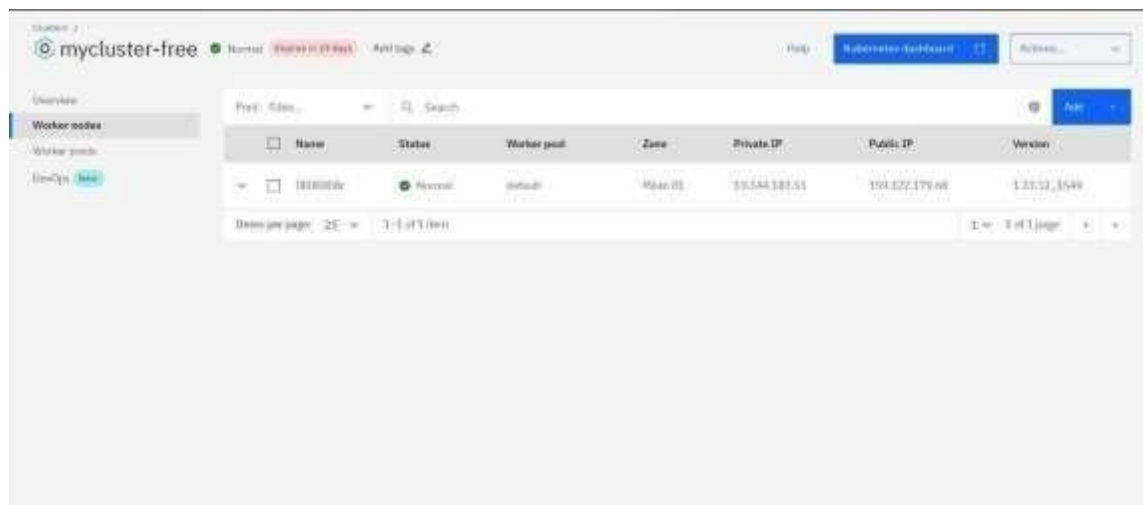


OUTPUT:



#### 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.  
CREATING KUBERNETES CLUSTER IN IBM CLOUD AND EXPOSING NODEPORT:



OUTPUT:

