

Assignment- 4

Name	AJAY KOUSHIK K N
Team ID	PNT2022TMID01682
Project Name	Plasma Donor Application

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

The screenshot displays the Docker Playground interface. On the left, a sidebar shows a clock at 03:57:32, a 'CLOSE SESSION' button, and an 'Instances' section with a search icon and a settings gear. Below this is a '+ ADD NEW INSTANCE' button and a list of instances, including one named '192.168.0.8' with a 'root' user. The main area shows a terminal window for a session titled 'cddvkasm0_cddvkym0qau000a07j5g'. The terminal displays the IP '192.168.0.8', memory usage '1.24% (49.52MB / 3.99GB)', and CPU usage '0.31%'. It also shows an SSH command: 'ssh ip:172-18-0-22-cddvkasm0qau000a07j5g@directlabs.pla'. Below the terminal, there are 'DELETE' and 'EDITOR' buttons. The terminal output shows a warning about a sandbox environment, followed by the command 'docker pull hello-world' and its execution. The output indicates that the image was pulled successfully from the library/hello-world:latest tag. The final command shown is 'docker run hello-world'.

```
03:57:32
CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.8
root

cddvkasm0_cddvkym0qau000a07j5g

IP: 192.168.0.8 OPEN PORT
Memory: 1.24% (49.52MB / 3.99GB) CPU: 0.31%
SSH: ssh ip:172-18-0-22-cddvkasm0qau000a07j5g@directlabs.pla

DELETE EDITOR

#####
# WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY discouraged. Any consequences of doing so are
# completely the user's responsibility.
#
# The PND team.
#####
[root@localhost ~]# docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
cddvkasm0qau000a07j5g: Pull complete
Digest: sha256:cdbb0272f4076e8361b647c6b162b66f32a60f2d5e7
Status: Downloaded newer image for hello-world:latest
[root@localhost ~]# docker run hello-world
```

03:57:05

CLOSE SESSION

Instances 🔑 ⚙️

+ ADD NEW INSTANCE

192.168.0.8 node1

cddvksm0_cddvkvm0qau000a07j5g

IP

192.168.0.8

OPEN PORT

Memory

1.26% (50.45MiB / 3.906GiB)

CPU

0.39%

SSH

ssh ip172-18-0-22-cddvksm0qau000a07j5g@direct.labs.pla 📄

DELETE

EDITOR

```

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/

[node1] (local) root@192.168.0.8 -
$

```

Activate Windows

Go to Settings to activate Windows.

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

desktopapplication. DOCKER FILE:

```

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"]

```

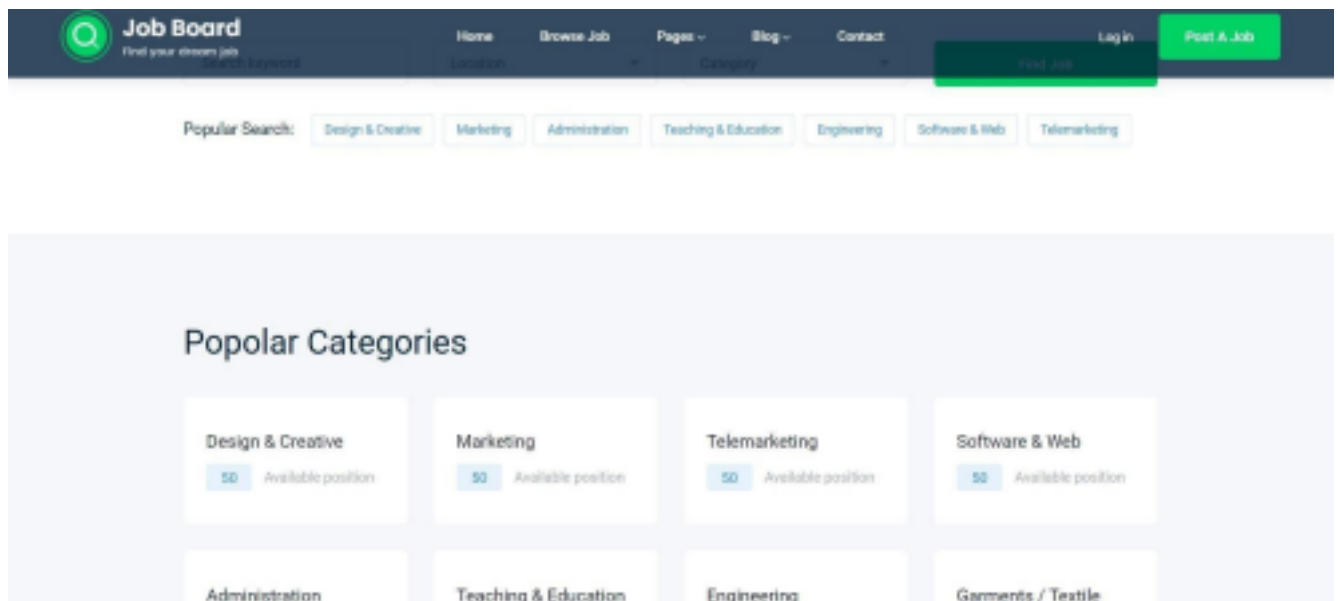
DEPLOYMENT OF JOBPOTAL APPLICATION:

The screenshot shows the Docker Desktop interface. On the left is a sidebar with navigation options: Containers, Images, Volumes, Dev Environments, Extensions, and Add Extensions. The main panel is titled 'Containers' and includes a toggle for 'Only show running containers' and a search bar. Below this is a table listing the containers:

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	agitated_neumann 918d20862639	icr.io/helloapp/bmctest	Exited (137)	49160:8080		
<input type="checkbox"/>	jolly_buring b62a07129d93	jobportalapplication:latest	Running	1234:8000	4 minutes ago	

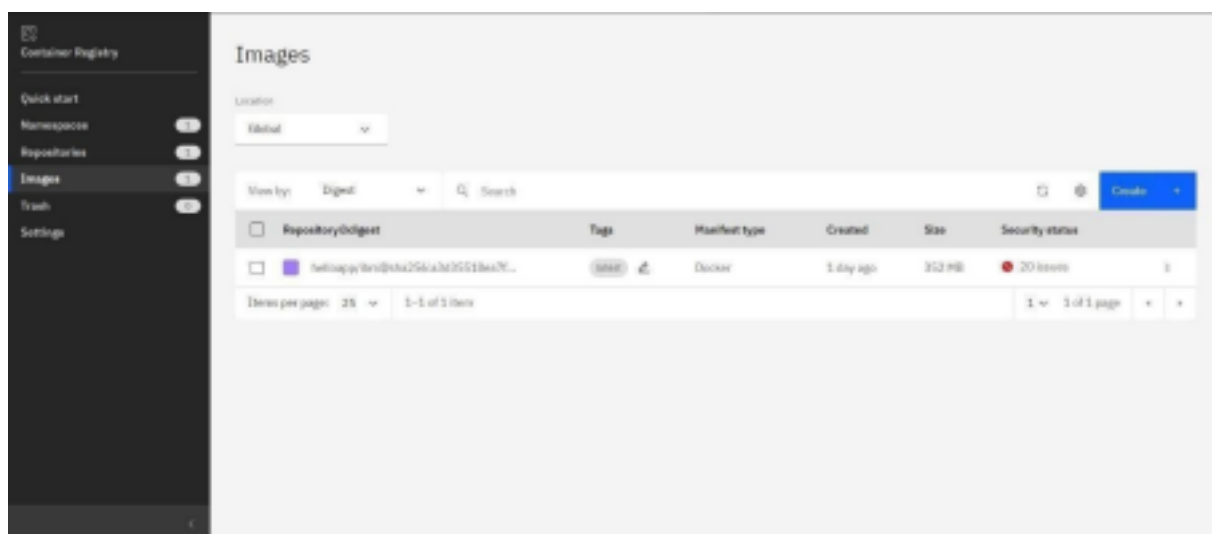
At the bottom right of the container list, it says 'Showing 2 items'. The bottom status bar shows 'RAM 3.66GB', 'CPU 0.57%', 'Connected to Hub', and 'v4.13.0'.

OUTPUT:

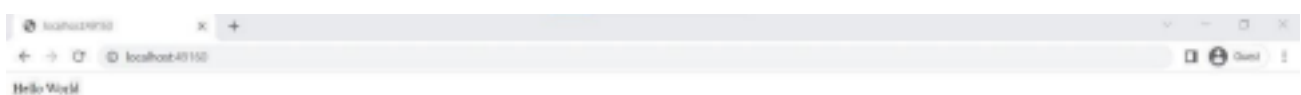


3.Create a IBM container registry and deploy hello-world app or job port

app.IBM CONTAINER REGISTRY DEPLOYMENT:

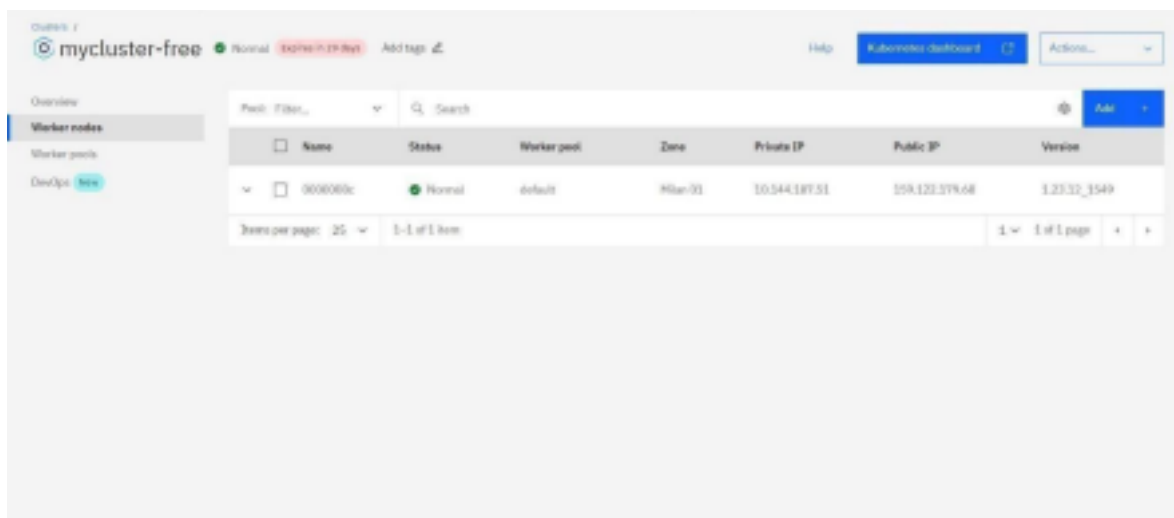


OUTPUT:



4. Create a Kubernetes cluster in IBM cloud and deploy hello world image or job portal image and also expose the same app to run in node port.

Creating Kubernetes cluster in IBM cloud and exposing node port:



The screenshot shows the 'mycluster-free' dashboard in IBM Cloud. The 'Worker nodes' tab is selected, displaying a table with one node. The node's status is 'Normal', and it is part of the 'default' worker pool. The table includes columns for Name, Status, Worker pool, Zone, Private IP, Public IP, and Version.

Name	Status	Worker pool	Zone	Private IP	Public IP	Version
0000060c	Normal	default	nlb-us-01	10.544.187.51	159.122.379.68	1.25.12_1549

Output:

