

Assignment - 4

Assignment Date	24 October 2022
Student Name	Jagabathi Babu
Student Roll Number	111519205013
Maximum Marks	2 Marks

Question 1:

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

The image shows two screenshots of the Docker Playground interface, demonstrating the steps to pull and run a Docker image.

Top Screenshot: The interface shows a session titled "cdu5fae3_cdu5fcu3tccg00cskckg" with IP 192.168.0.28. The terminal output shows the following commands and results:

```
WARNING!!!!
# 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@192.168.0.28 ~
$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fccceec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
[node1] (local) root@192.168.0.28 ~
$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
```

Bottom Screenshot: The same interface shows the continuation of the terminal output, listing the steps Docker took to generate the message:

```
1. The Docker client contacted the Docker daemon.
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@192.168.0.28 ~
$
```

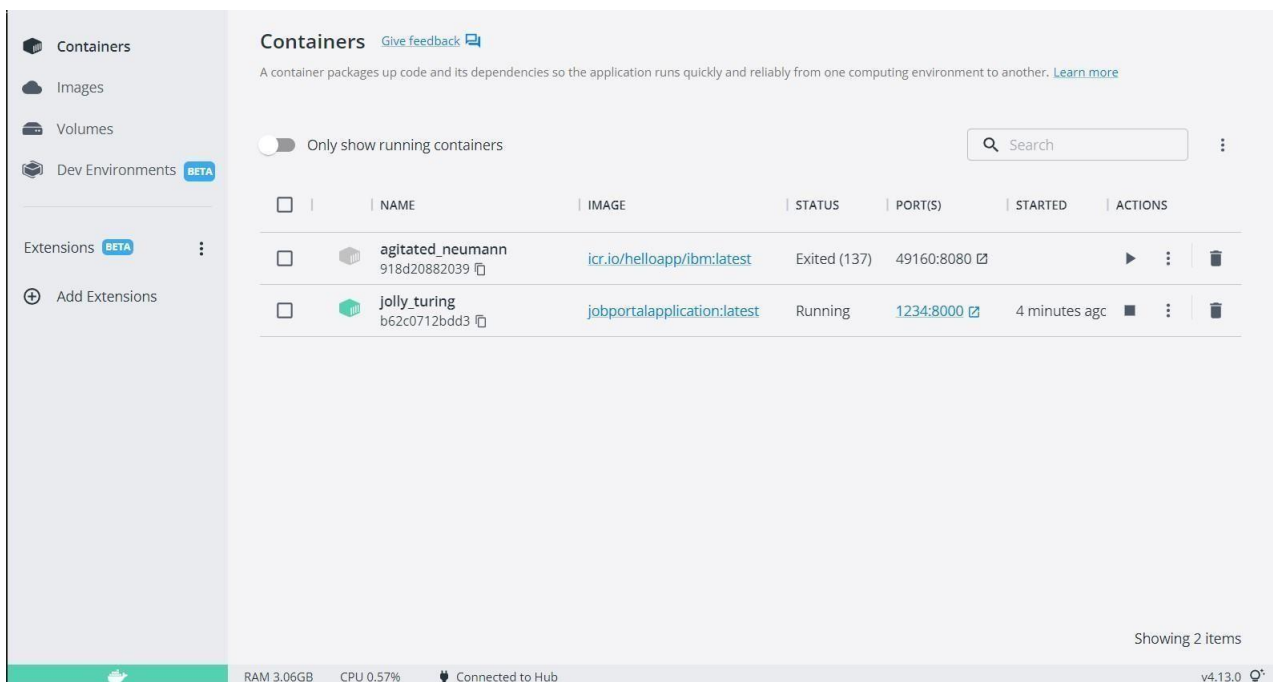
Question 2:

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:

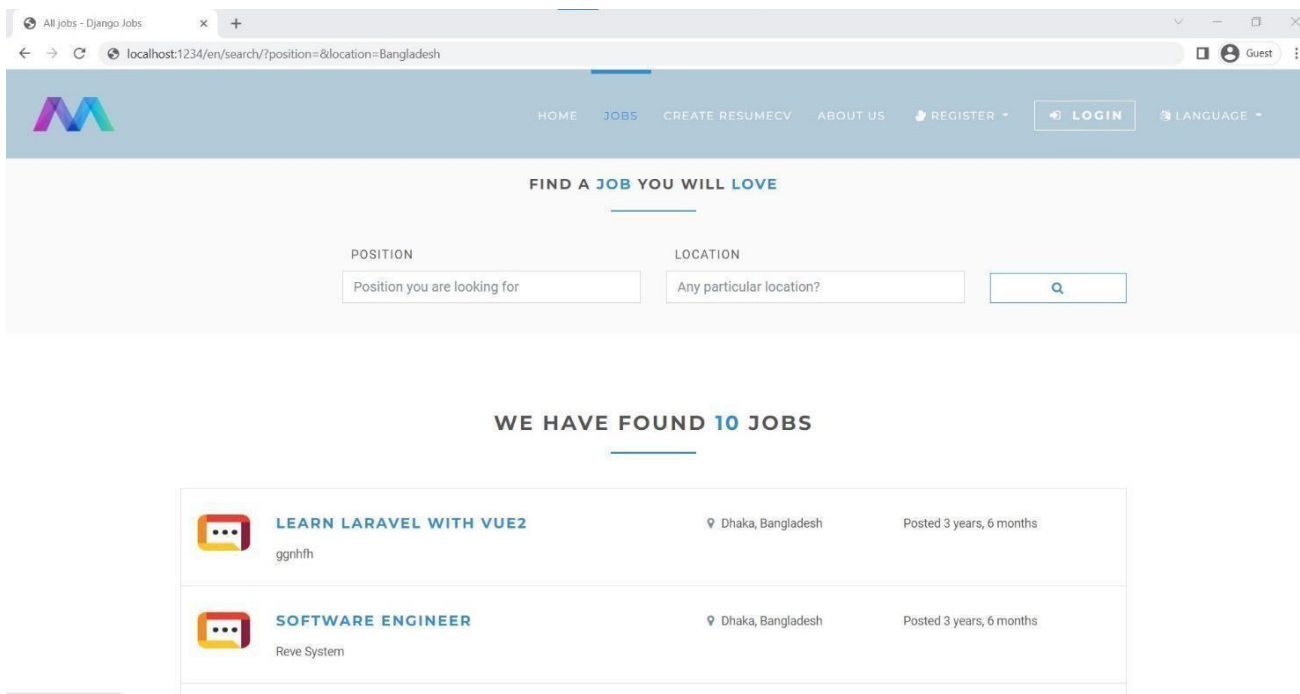


The screenshot shows the Docker Desktop interface. On the left is a sidebar with navigation options: Containers, Images, Volumes, Dev Environments (marked BETA), Extensions (marked BETA), and Add Extensions. The main panel is titled 'Containers' and includes a toggle for 'Only show running containers'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

	NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
<input type="checkbox"/>	agitated_neumann 918d20882039	icr.io/helloapp/ibm:latest	Exited (137)	49160:8080		
<input type="checkbox"/>	jolly_turing b62c0712bdd3	jobportalapplication:latest	Running	1234:8000	4 minutes ago	

At the bottom right of the main panel, it says 'Showing 2 items'. The bottom status bar shows system information: RAM 3.06GB, CPU 0.57%, Connected to Hub, and version v4.13.0.

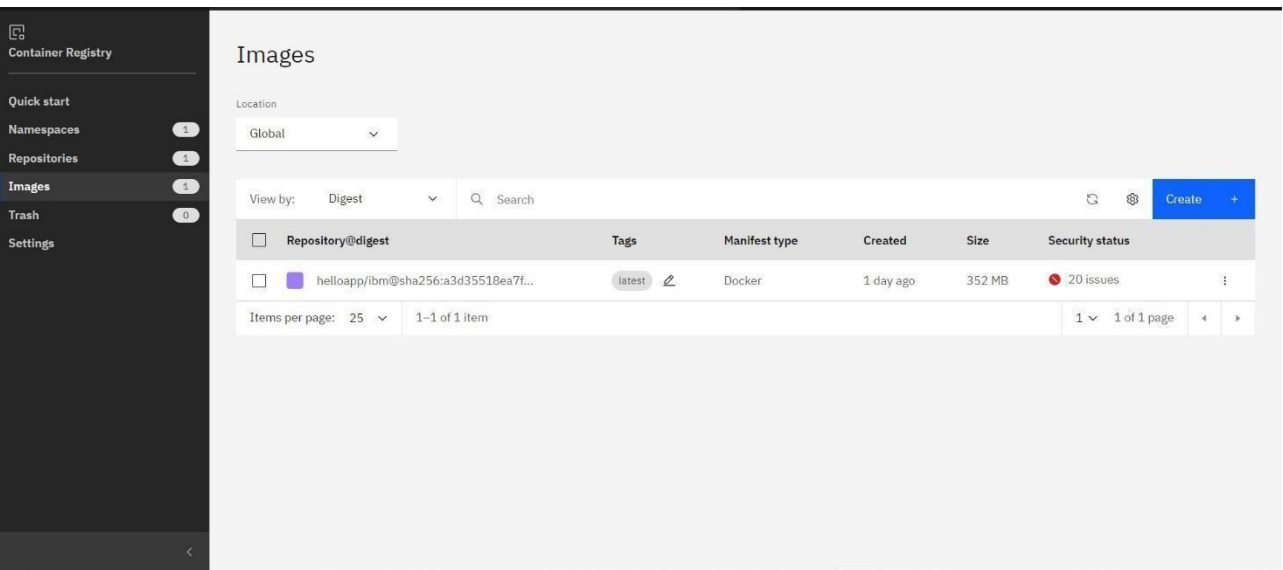
OUTPUT:



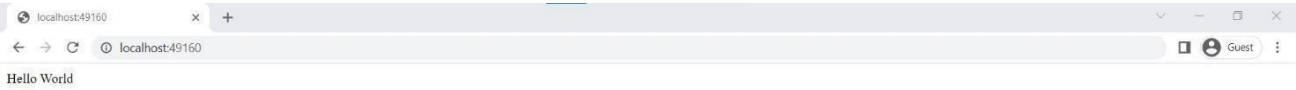
Question 3:

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

IBM CONTAINER REGISTRY DEPLOYMENT:



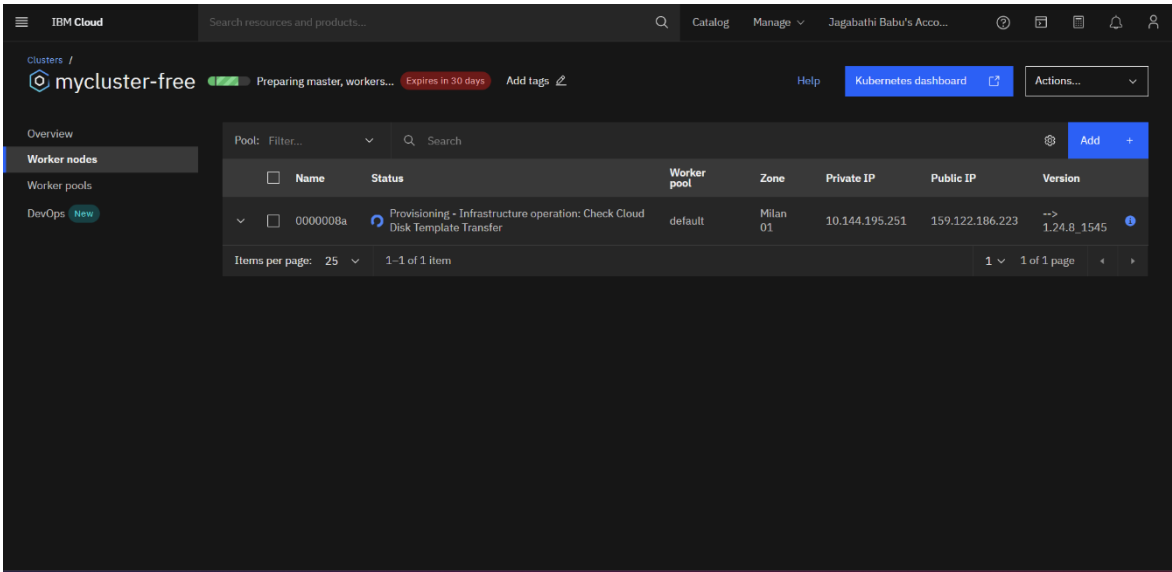
OUTPUT:



Question 4:

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

Creating Kubernetes cluster in IBM cloud and exposing node port:



OUTPUT:

