

ASSIGNMENT-4

ASSIGNMENT DATE	5 th NOVEMBER
STUDENT NAME	Irene Jerusha.P
TEAM ID	PNT2022PMID25781
TEAM NAME	Syntax Squids
MARKS	2 marks

Question 1:

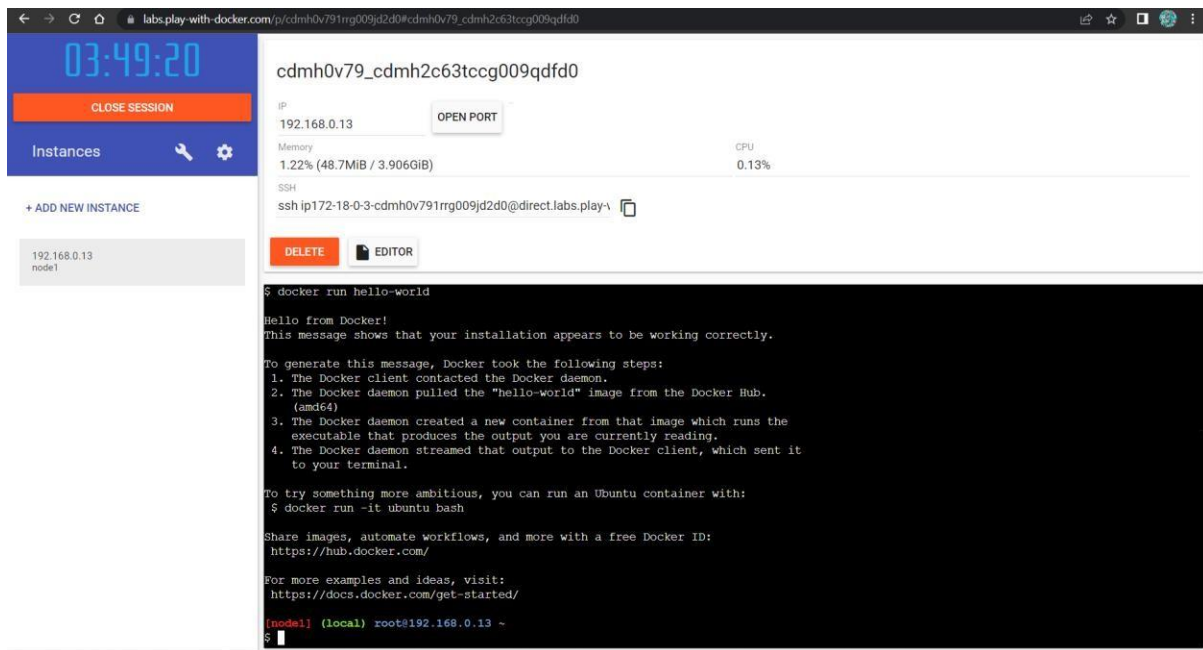
Pull an image from docker hub and run it in docker playground

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:51:49, a 'CLOSE SESSION' button, and a list of instances. The main area displays the details for an instance named 'cdmh0v79_cdmh2c63tccg009qdfd0'. It shows the IP address 192.168.0.13, memory usage (1.24% / 49.78MiB), and CPU usage (0.19%). Below this, there's a terminal window with the following output:

```
##### 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@192.168.0.13 ~
$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fcccceec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
(node1) (local) root@192.168.0.13 ~
$ 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:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
```



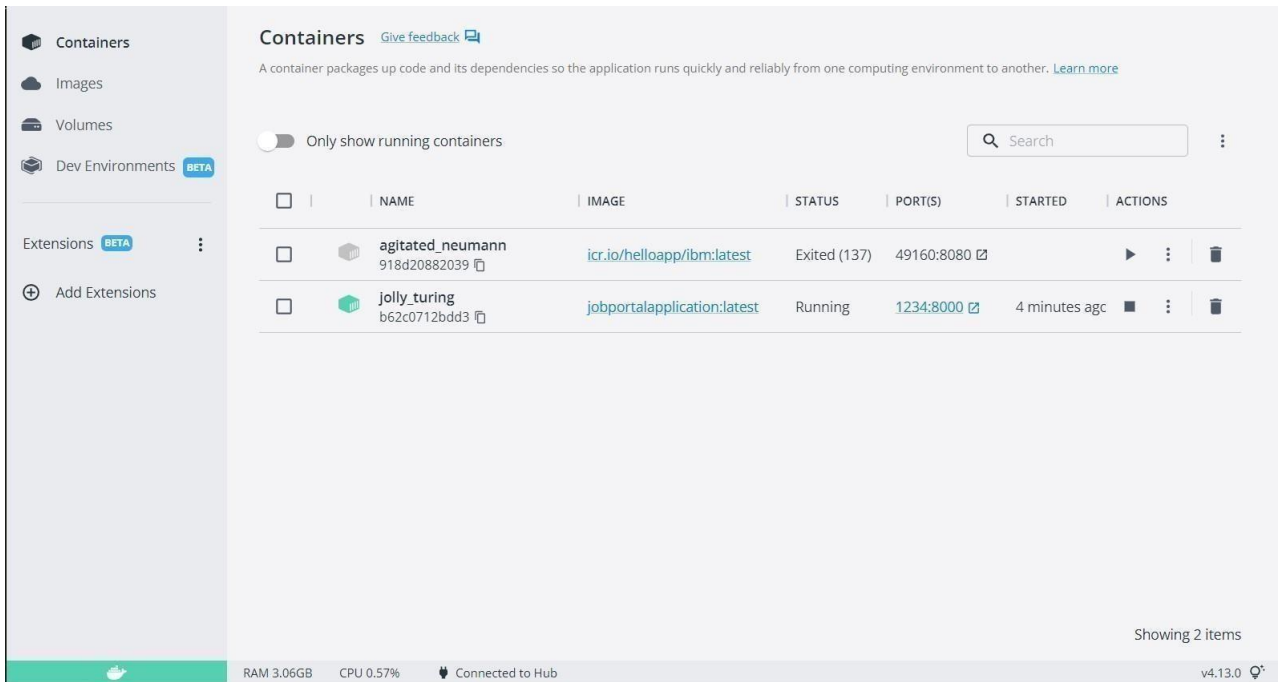
Question 2:

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

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 JOBPORTAL APPLICATION:

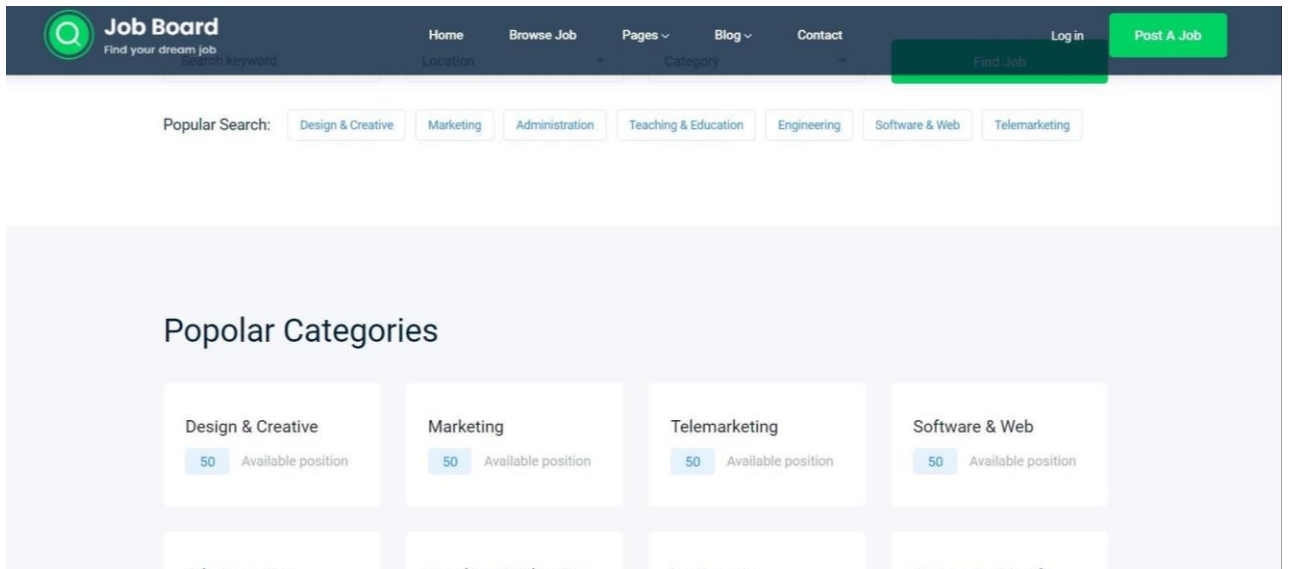


The screenshot shows the Docker Desktop interface. On the left is a sidebar with navigation options: Containers, Images, Volumes, Dev Environments (marked BETA), Extensions (marked BETA), and Add Extensions. The main area is titled 'Containers' and includes a toggle for 'Only show running containers' 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 area, it says 'Showing 2 items'. The bottom status bar shows 'RAM 3.06GB', 'CPU 0.57%', 'Connected to Hub', and 'v4.13.0'.

OUTPUT:



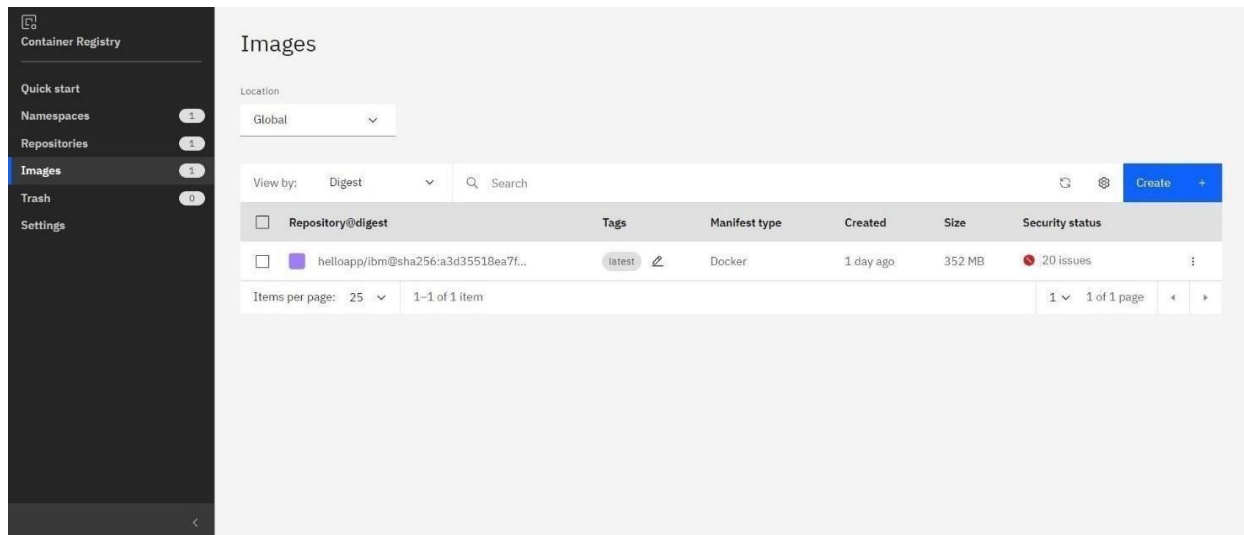
The screenshot shows the output of the Job Board application. The header is dark blue with the 'Job Board' logo and tagline 'Find your dream job'. Navigation links include Home, Browse Job, Pages, Blog, and Contact. There are buttons for 'Log in' and 'Post A Job'. A search bar is present with a 'Find Job' button. Below the header, there's a 'Popular Search:' section with buttons for Design & Creative, Marketing, Administration, Teaching & Education, Engineering, Software & Web, and Telemarketing. The main content area is titled 'Poplar Categories' (note the typo) and displays a grid of category cards. Each card shows the category name, a blue box with the number '50', and the text 'Available position'.

Category	Count	Status
Design & Creative	50	Available position
Marketing	50	Available position
Telemarketing	50	Available position
Software & Web	50	Available position
Administration		
Teaching & Education		
Engineering		
Others / Textile		

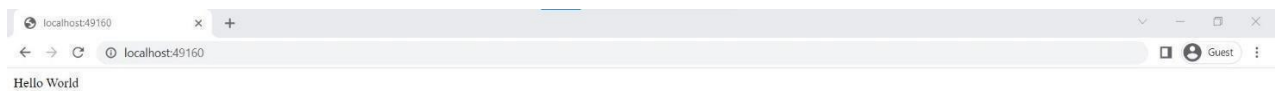
Question 3:

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

CONTAINER REGISTRY DEPLOYMENT:



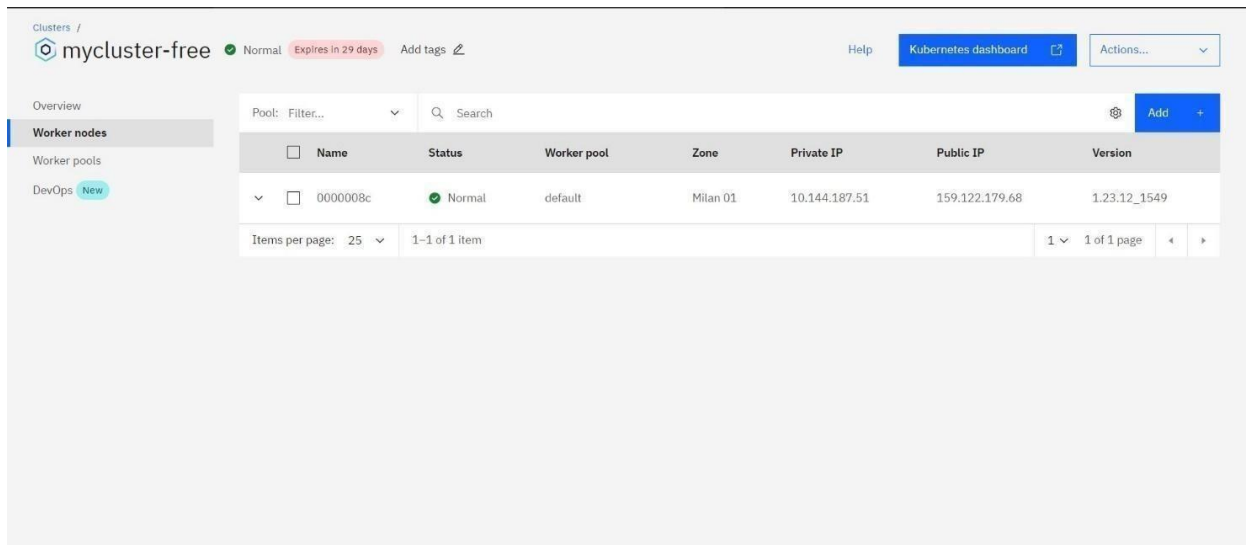
OUTPUT:



Question 4:

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

Creating Kubernetes cluster in IBM cloud and exposing node port:



OUTPUT:

