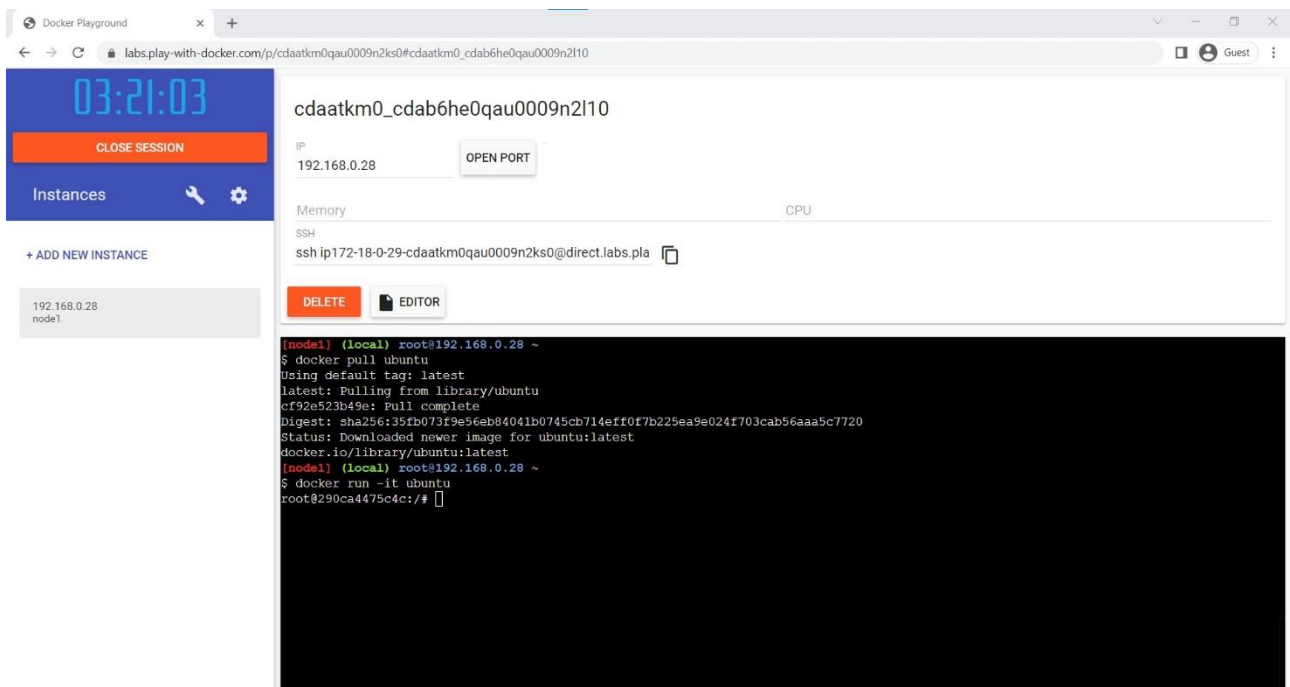


Assignment - 4

Assignment Date	24 October 2022
Student Name	Subramanian K
Student Roll Number	111519205047
Maximum 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 03:21:03, a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button. Below that, a list of instances shows '192.168.0.28 node1'. The main area displays the instance 'cdaatk0_cdab6he0qau0009n2l10' with its IP '192.168.0.28' and an 'OPEN PORT' button. Below this, there are 'DELETE' and 'EDITOR' buttons. The terminal window shows the following commands and output:

```
[node1] (local) root@192.168.0.28 ~
$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
cf92e523b49e: Pull complete
Digest: sha256:35fb073f9e56eb84041b0745cb714eff0f7b225ea9e024f703cab56aaa5c7720
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
[node1] (local) root@192.168.0.28 ~
$ docker run -it ubuntu
root@290ca4475c4c:/#
```

Question 2:

Create a docker file for the jobportal 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:

Containers

Images

Volumes

Dev Environments BETA

Extensions BETA

Add Extensions

Containers Give feedback

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

☐ Only show running containers

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

Showing 2 items

RAM 3.06GB

CPU 0.57%

Connected to Hub

v4.13.0

OUTPUT:

The screenshot shows a web browser window with the URL `localhost:1234/en/search/?position=&location=Bangladesh`. The application has a header with a logo, navigation links (HOME, JOBS, CREATE RESUME/CV, ABOUT US, REGISTER, LOGIN), and a language selector. The main content area is titled "FIND A JOB YOU WILL LOVE" and contains two input fields: "POSITION" (with placeholder text "Position you are looking for") and "LOCATION" (with placeholder text "Any particular location?"). A search button with a magnifying glass icon is to the right of the location field. Below the search area, a message states "WE HAVE FOUND 10 JOBS". A table displays the first two results:

Job Title	Company	Location	Posted
LEARN LARAVEL WITH VUE2	ggnhfh	Dhaka, Bangladesh	Posted 3 years, 6 months
SOFTWARE ENGINEER	Reve System	Dhaka, Bangladesh	Posted 3 years, 6 months

Question 3:

Create a IBM container registry and deploy helloworld app or jobportapp.

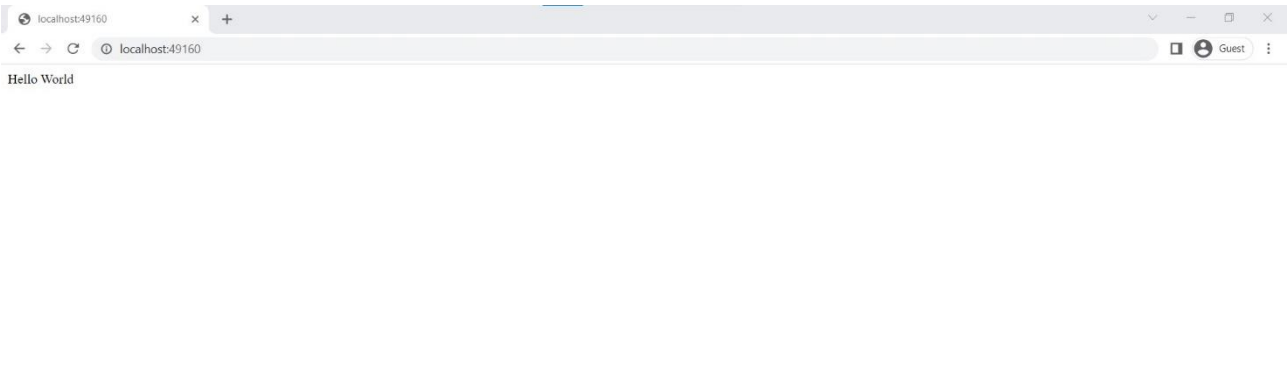
IBM CONTAINER REGISTRY DEPLOYMENT:

The screenshot shows the IBM Container Registry console. The left sidebar contains navigation links: Container Registry, Quick start, Namespaces (1), Repositories (1), Images (1), Trash (0), and Settings. The main content area is titled "Images" and shows a list of images. The "Location" dropdown is set to "Global". The "View by" dropdown is set to "Digest". The search bar is empty. The table below shows the following image:

Repository@digest	Tags	Manifest type	Created	Size	Security status
helloapp/ibm@sha256:a3d35518ea7f...	latest	Docker	1 day ago	352 MB	20 issues

At the bottom of the table, it says "Items per page: 25" and "1-1 of 1 item".

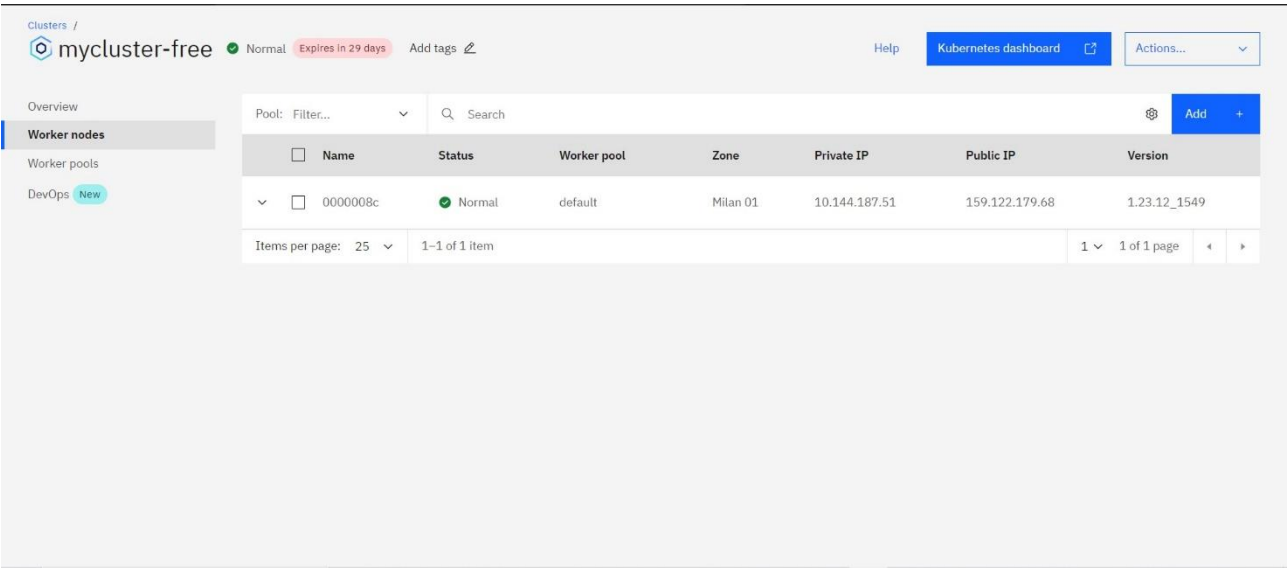
OUTPUT:



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:

