

Assignment -4

Assignment Date	3 November 2022
Student Name	Bharathi R
Student Roll Number	720719110012
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 03:57:32, a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button. Below this, a list of instances shows '192.168.0.8 node1'. The main area displays the instance details for 'cddvksm0_cddvkvm0qau000a07j5g'. It shows the IP '192.168.0.8', memory usage '1.24% (49.52MiB / 3.906GiB)', and CPU usage '0.31%'. There are 'DELETE' and 'EDITOR' buttons. The terminal output shows a warning message, followed by the command 'docker pull hello-world' and its output, and then 'docker run hello-world'.

```
#####  
# 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. #  
#####  
[node1] (local) root@192.168.0.8 ~  
$ docker pull hello-world  
Using default tag: latest  
latest: Pulling from library/hello-world  
2db29710123e: Pull complete  
Digest: sha256:e18f0a77aefabe047a671ab3ec3eed05414477c951ab1a6f352a06974245fe7  
Status: Downloaded newer image for hello-world:latest  
docker.io/library/hello-world:latest  
[node1] (local) root@192.168.0.8 ~  
$ docker run hello-world
```

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:57:05, a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button. Below this, a list of instances shows '192.168.0.8 node1'. The main area displays the instance details for 'cddvksm0_cddvkvm0qau000a07j5g'. It shows the IP '192.168.0.8', memory usage '1.26% (50.45MiB / 3.906GiB)', and CPU usage '0.39%'. There are 'DELETE' and 'EDITOR' buttons. The terminal output shows a list of steps explaining the 'hello-world' container, followed by the command 'docker run -it ubuntu bash' and its output.

```
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 ~  
$
```

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 8080
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

☐

NAME

☐

agitated_neumann
918d20882039

icr.io/helloapp/ibm:latest

Exited (137)

49160:8080

☐

jolly_turing
b62c0712bdd3

jobportalapplication:latest

Running

1234:8000

4 minutes ago

Showing 2 items

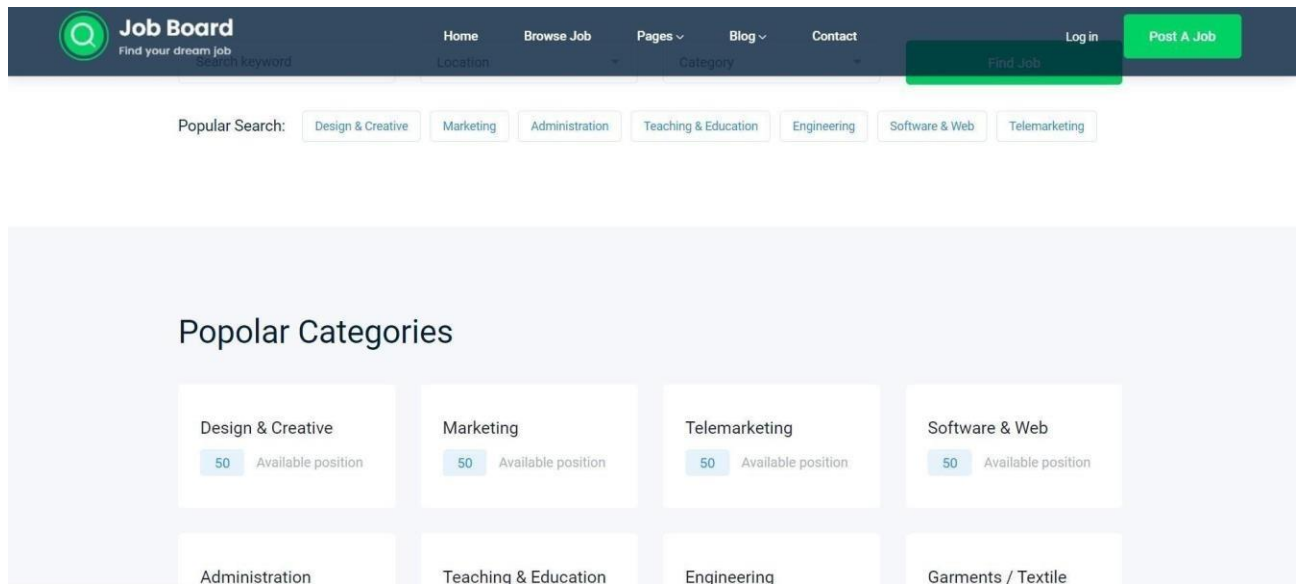
RAM 3.06GB

CPU 0.57%

Connected to Hub

v4.13.0

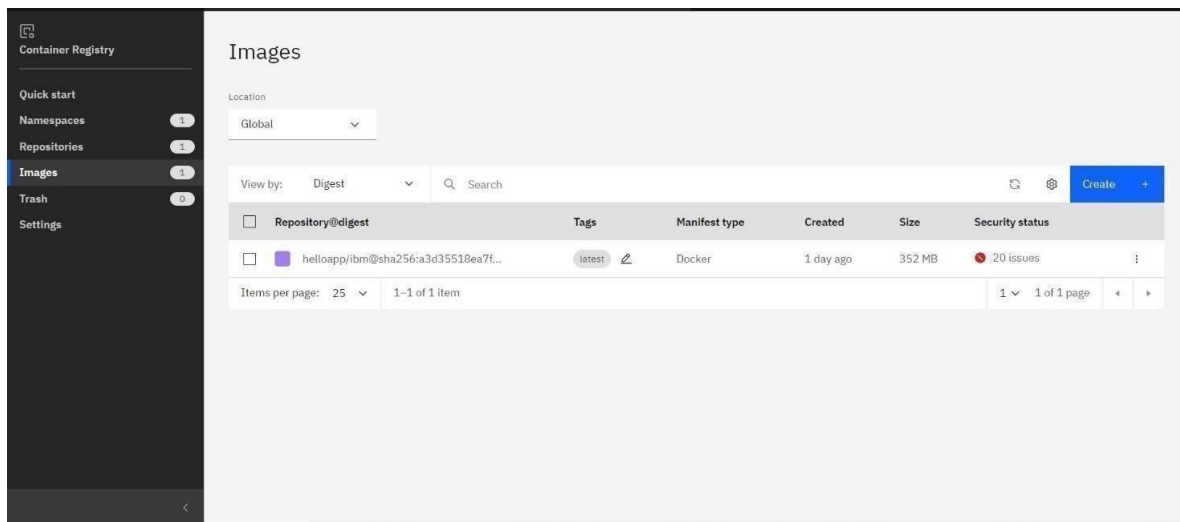
OUTPUT:



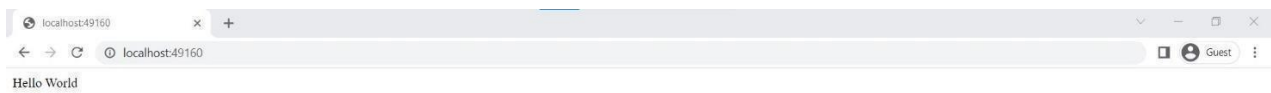
QUESTION-3:

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

CONTAINER REGISTRY DEPLOYMENT:



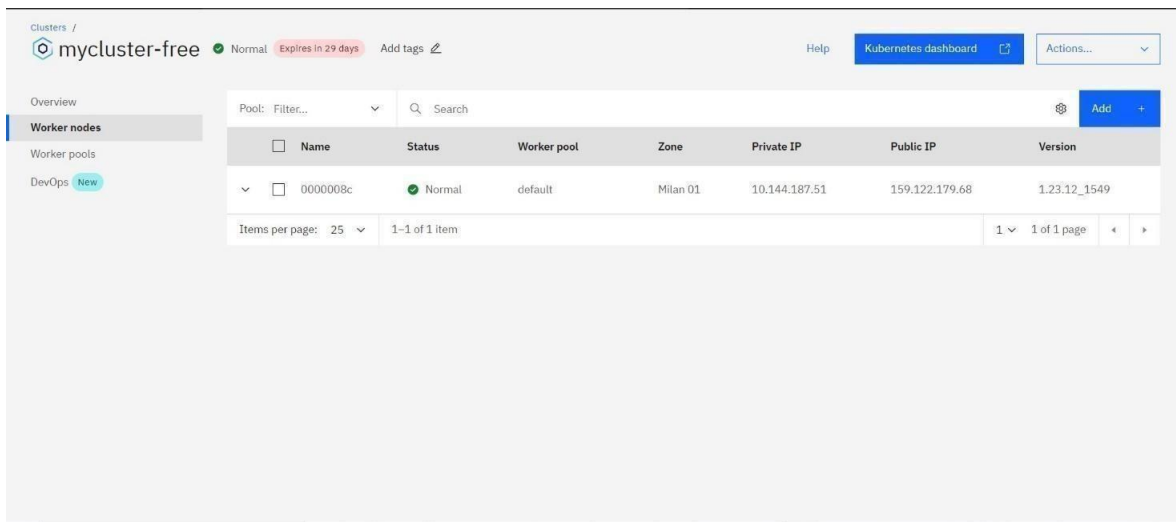
OUTPUT:



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 PART:



Output:

