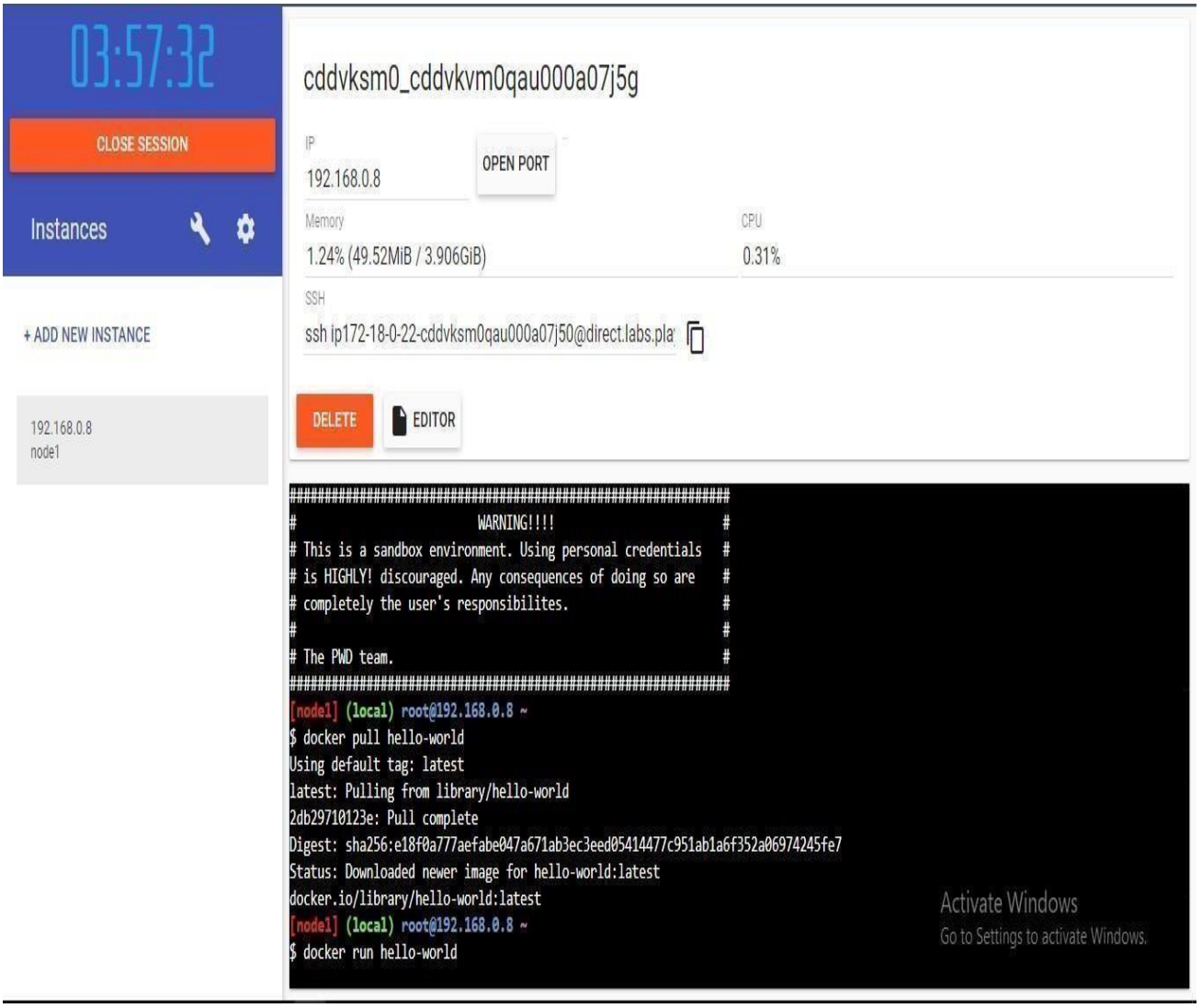
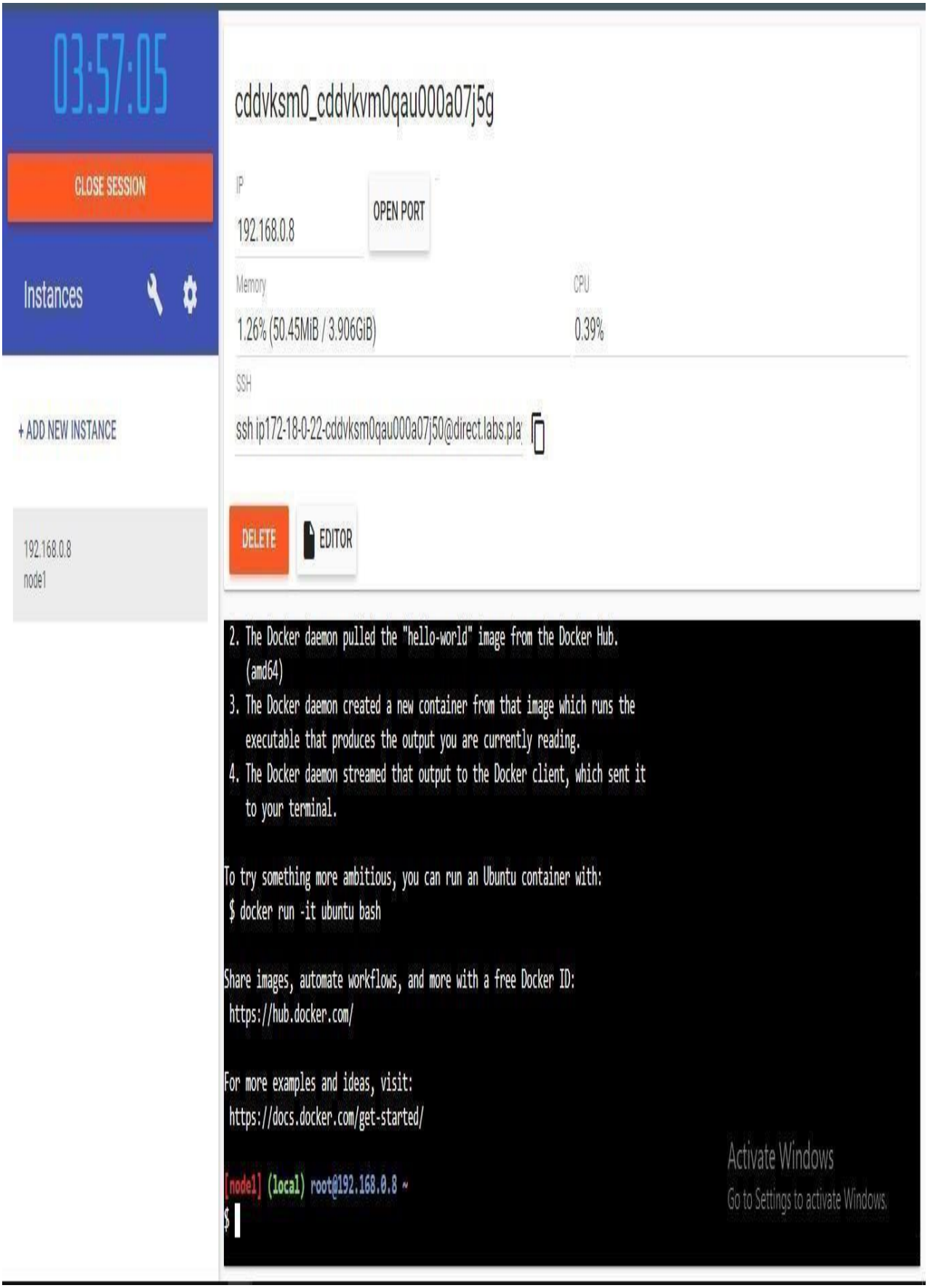
# Assignment - 5

| | |
|---|---|
| Assignment Date | 05 November 2022 |
| Student Name | Madhan R |
| Team ID | PNT2022TMID29967 |
| Maximum Marks | 2 Marks |

Question 1:

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

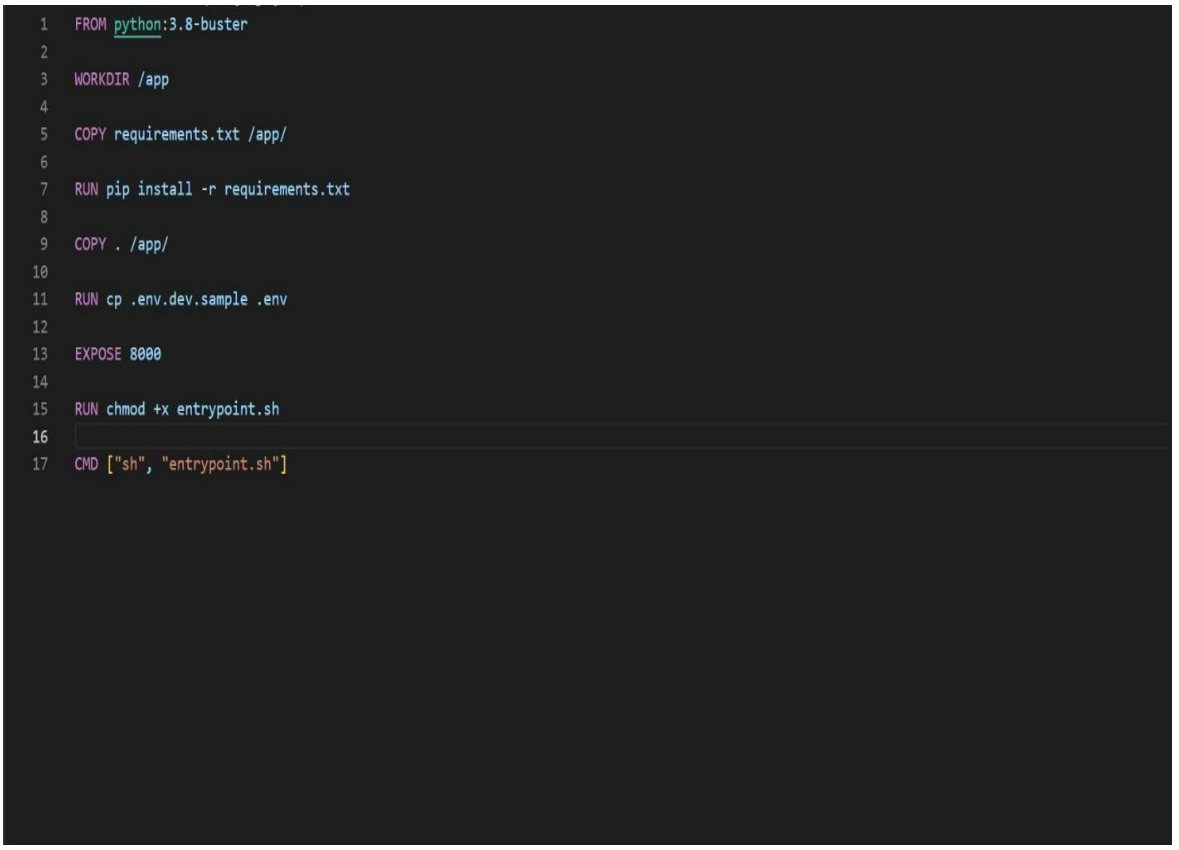



03:57:05
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.8
node1
cddvksm0_cddvkvm0qau000a07j5g
IP
192.168.0.8
OPEN PORT
Memory
1.26% (50.45MiB / 3.906GiB)
CPU
0.39%
SSH
ssh ip172-18-0-22-cddvksm0qau000a07j50@direct.labs.pla
DELETE
EDITOR
2. The Docker daemon pulled the "hello-world" image from the 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.

Question 2:

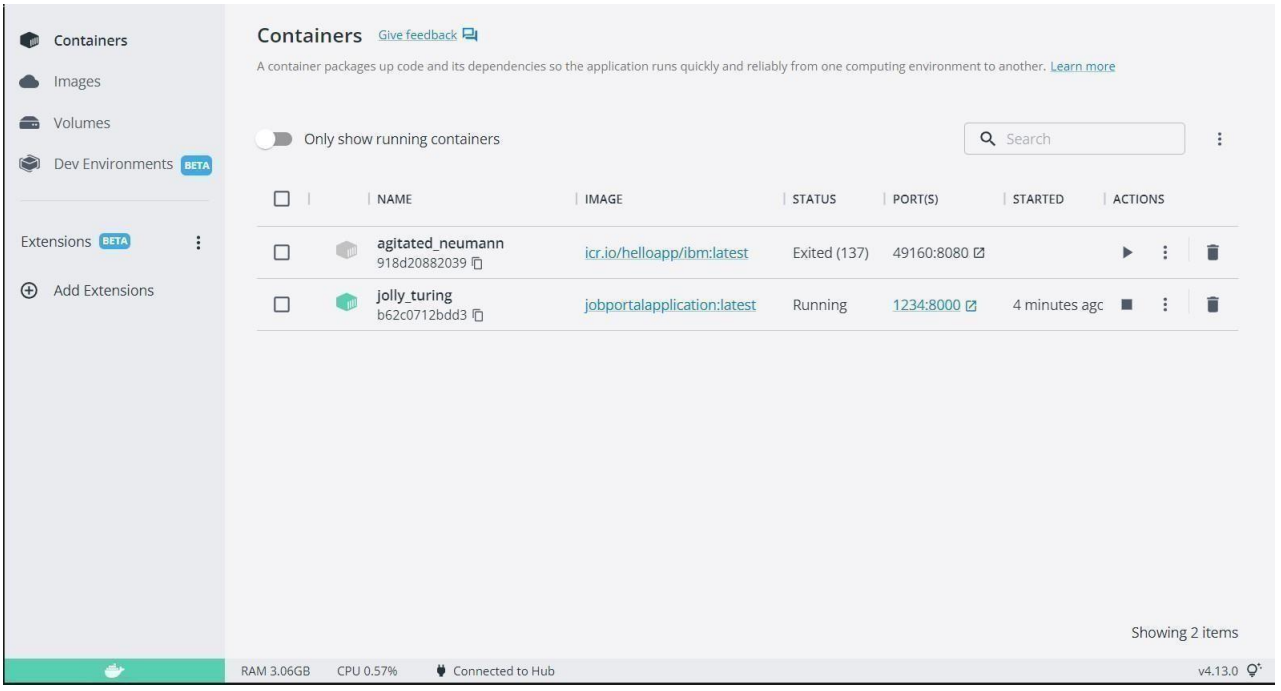
Create a docker file for the jobportal application and deploy it in Docker desktop application.

DOCKERFILE:

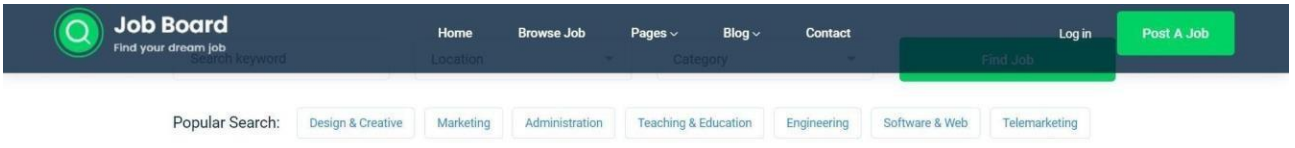


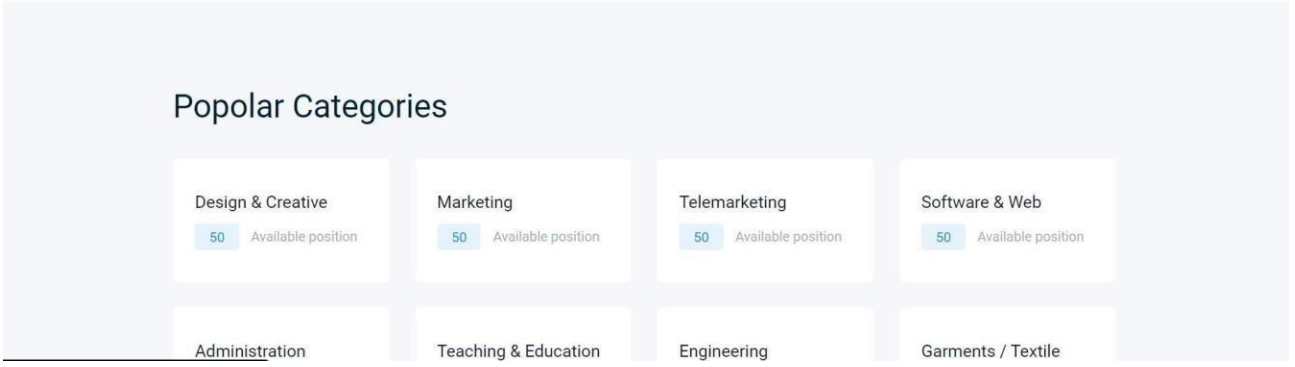
```
FROM python:3.8-buster

WORKDIR /app

COPY requirements.txt /app/

RUN pip install -r requirements.txt

COPY . /app/

RUN cp .env.dev.sample .env

EXPOSE 8000

RUN chmod +x entrypoint.sh

CMD ["sh", "entrypoint.sh"]
```

DEPLOYMENT OF JOBPORTAL APPLICATION:




OUTPUT:

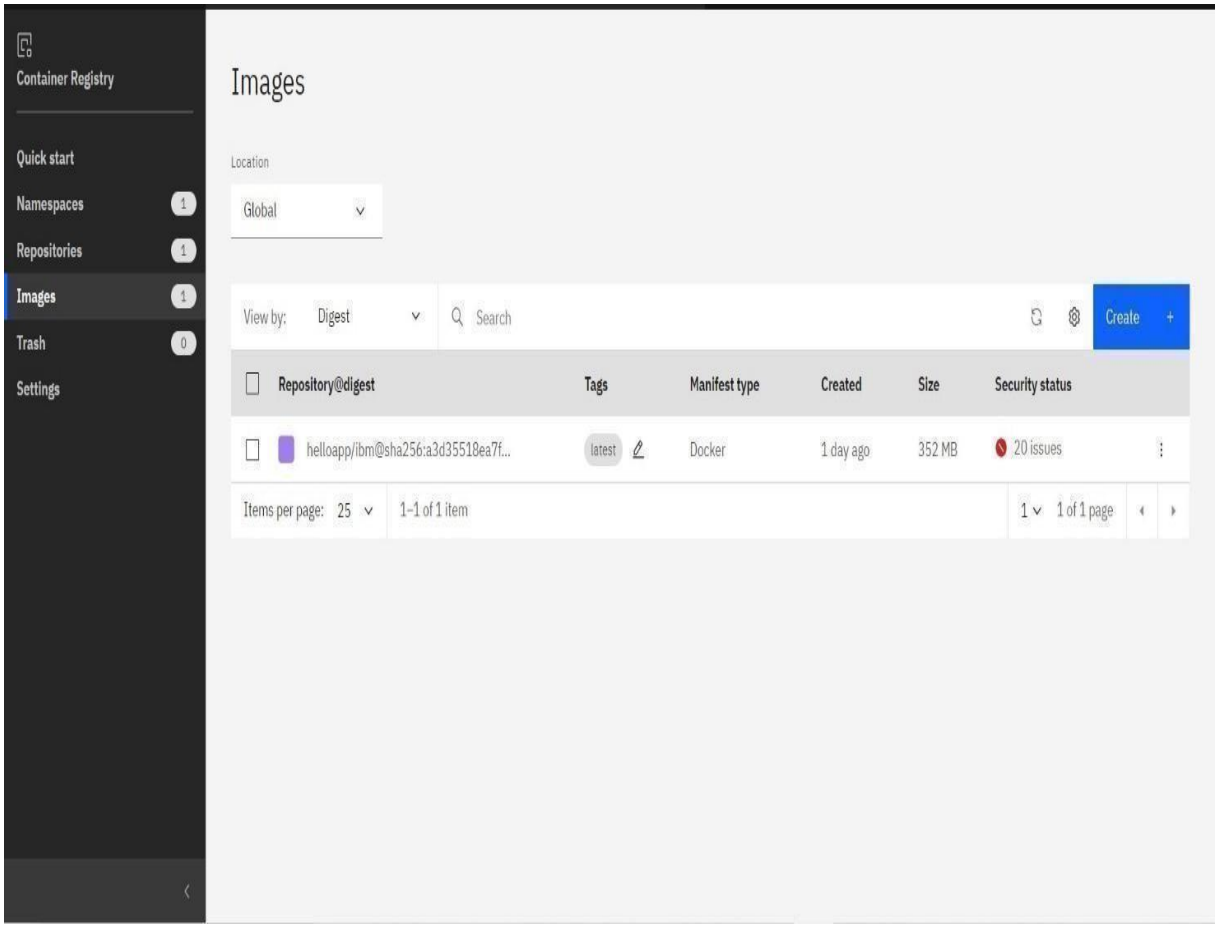


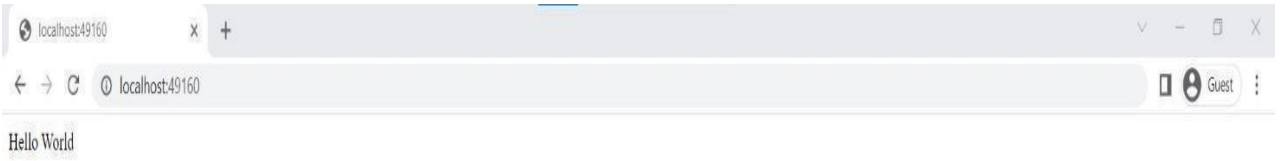
Question 3:

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

IBM CONTAINER REGISTRY DEPLOYMENT:



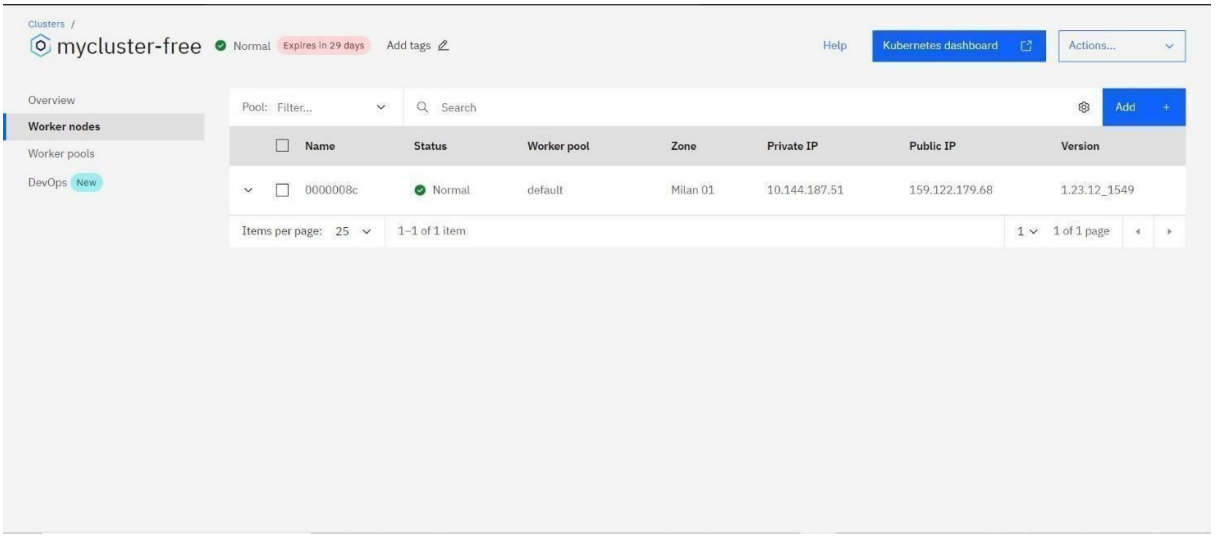

OUTPUT:



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: