

Assignment - 4

Question 1:

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

The image displays two screenshots of the Docker Playground interface, showing the process of pulling and running a Docker container.

Top Screenshot:

- Time:** 03:57:32
- Buttons:** CLOSE SESSION, Instances, + ADD NEW INSTANCE
- Instance:** 192.168.0.8 node1
- Instance Details:**
 - IP: 192.168.0.8
 - Memory: 1.24% (49.52MiB / 3.906GiB)
 - CPU: 0.31%
 - SSH: ssh ip172-18-0-22-cddvksm0qau000a07j50@direct.labs.pla
- Terminal 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 PMD 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:e18f0a777aefabe047a671ab3ec3eed05414477c951ab1a6f352a06974245fe7
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
```

Bottom Screenshot:

- Time:** 03:57:05
- Buttons:** CLOSE SESSION, Instances, + ADD NEW INSTANCE
- Instance:** 192.168.0.8 node1
- Instance Details:**
 - IP: 192.168.0.8
 - Memory: 1.26% (50.45MiB / 3.906GiB)
 - CPU: 0.39%
 - SSH: ssh ip172-18-0-22-cddvksm0qau000a07j50@direct.labs.pla
- Terminal 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 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

Search

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

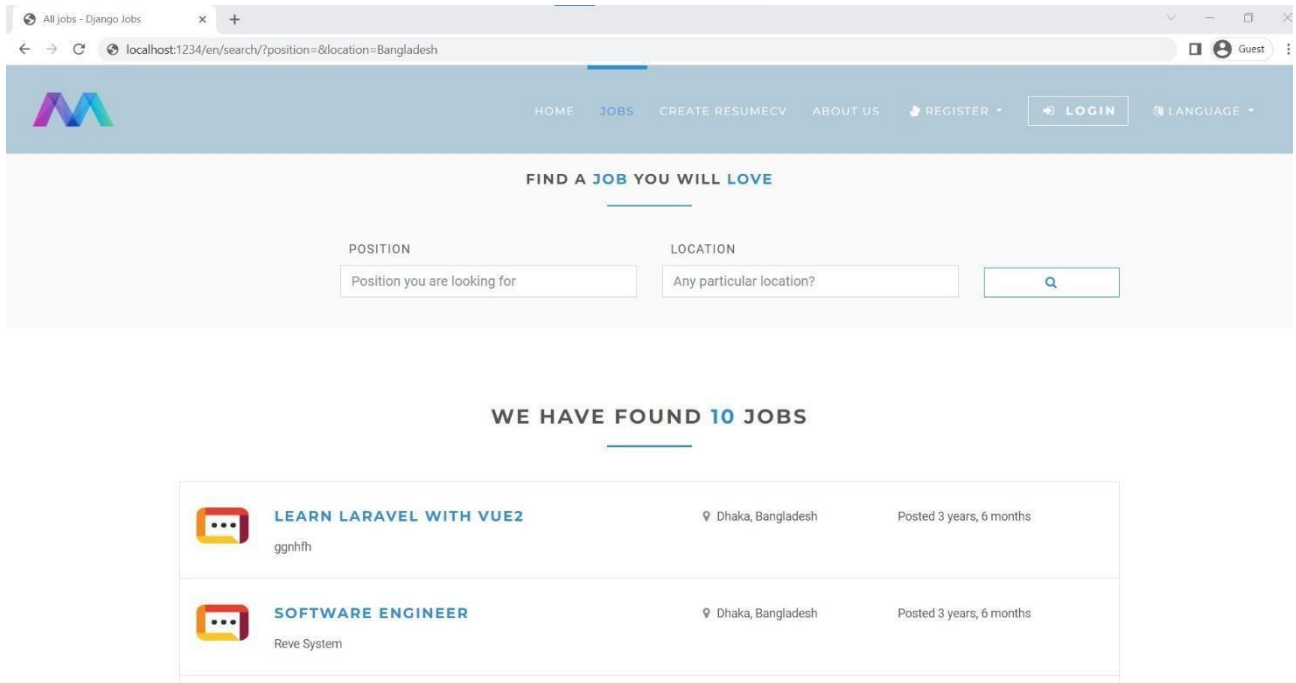
Showing 2 items

RAM 3.06GB CPU 0.57%

Connected to Hub

v4.13.0

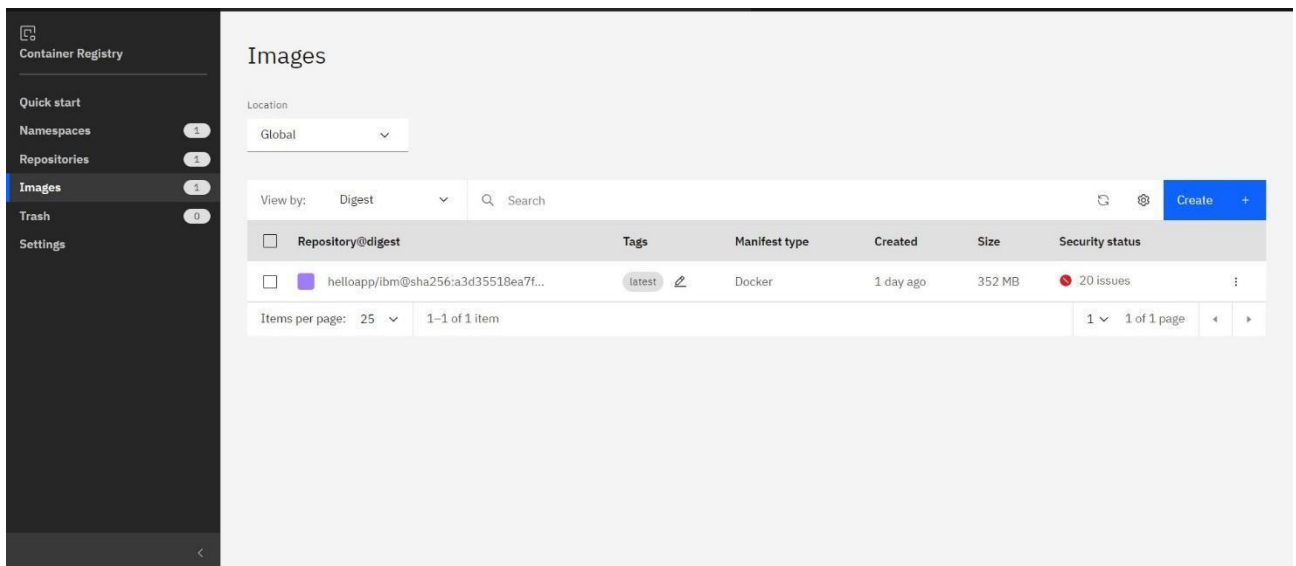
OUTPUT:



Question 3:

Create a IBM container registry and deploy 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:

A screenshot of the IBM Cloud Kubernetes Dashboard. The cluster is named 'mycluster-free' and is in a 'Normal' state, expiring in 29 days. The 'Worker nodes' tab is selected, showing a table with one node.

<input type="checkbox"/>	Name	Status	Worker pool	Zone	Private IP	Public IP	Version
<input type="checkbox"/>	0000008c	Normal	default	Milan 01	10.144.187.51	159.122.179.68	1.23.12_1549

Items per page: 25 1-1 of 1 item

Output:

