

ASSIGNMENT - 4

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 03:51:29, a 'CLOSE SESSION' button, and a list of instances including '192.168.0.18 node1'. The main area displays the instance details for 'cduc2e3_cduc4e3tccg00csl39g', including its IP (192.168.0.18), memory, CPU, and an SSH command. Below this is a terminal window showing the execution of 'docker pull uifd/ui-for-docker' and 'docker run' commands, resulting in a successful container start.

The screenshot shows the 'UI For Docker' dashboard. It features a navigation bar with tabs for Dashboard, Containers, Containers Network, Images, Networks, Volumes, and Info. The 'Containers' tab is active, displaying a 'Running Containers' section with a list of containers (bold_matsumoto) and a 'Status' section with a donut chart showing the status of containers (Running, Stopped, Ghost). Below these are two line graphs: 'Containers created' and 'Images created', both showing a count of 1 over time.

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"]
```

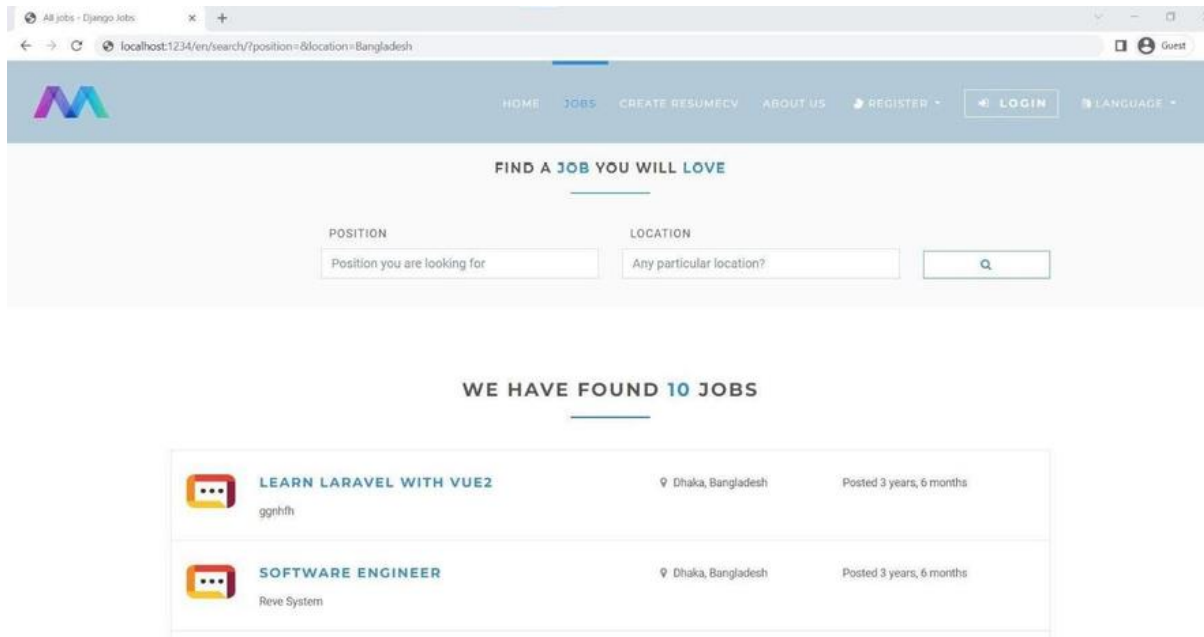
DEPLOYMENT OF JOBPORTAL APPLICATION:

The screenshot shows the Docker Desktop application 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 panel is titled 'Containers' and includes a 'Give feedback' link. Below the title is a description: 'A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)'. There is a toggle switch for 'Only show running containers' and a search bar. A table lists the containers:

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

At the bottom right, it says 'Showing 2 items'. The bottom status bar shows 'RAM 3.06GB', 'CPU 0.57%', 'Connected to Hub', and 'v4.13.0'.

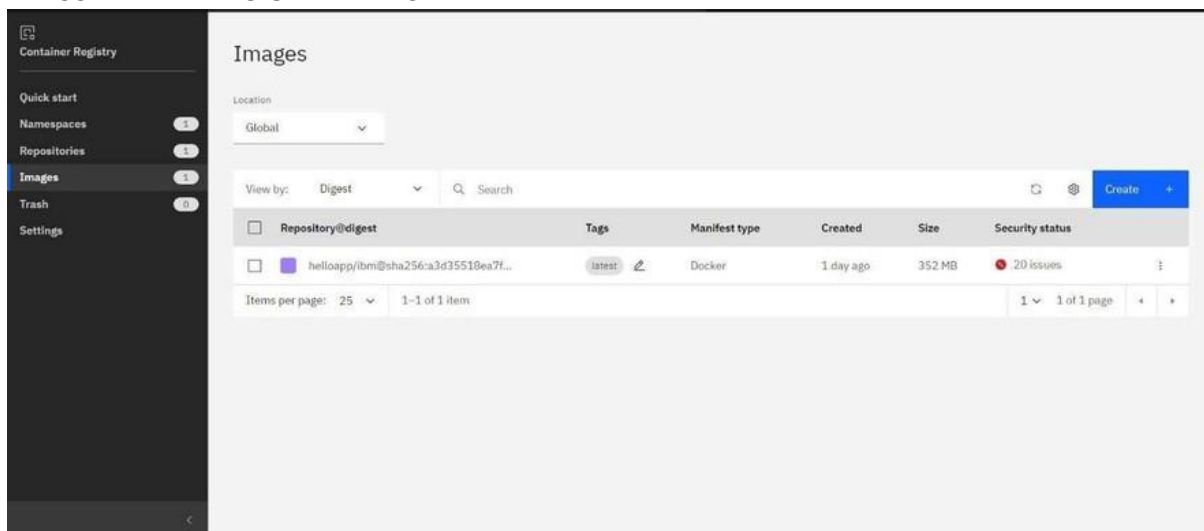
OUTPUT:



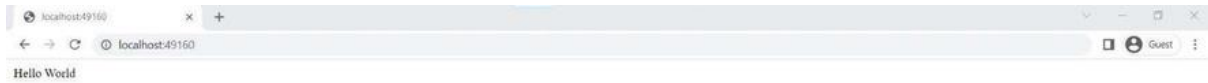
Question 3:

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

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 Service (K8S) dashboard. The page title is 'mycluster-free'. The left sidebar shows navigation options: Overview, Worker nodes (selected), Worker pools, and DevOps. The main content area displays a table of worker nodes. The table has columns: Name, Status, Worker pool, Zone, Private IP, Public IP, and Version. One node is listed with ID '0000008c', status 'Normal', and version '1.23.12_1549'. The table includes search and filter controls at the top and pagination at the bottom.

Name	Status	Worker pool	Zone	Private IP	Public IP	Version
0000008c	Normal	default	Milan 01	10.144.187.51	159.122.179.68	1.23.12_1549

Output:



Resume 1

BUILDER



Resume 2

BUILDER



CV

BUILDER