

Assignment -4

Assignment Date	13 November 2022
Student Name	VIJAIBALAJI N
Student Roll Number	953619104026
Maximum Marks	2 Marks

1.Pull an Image from docker hub and run it in docker playground

The image shows two screenshots of the Docker Playground interface, demonstrating the process of pulling a Docker image from Docker Hub and running it in a Docker playground.

Top Screenshot: The interface shows a session titled "cdob9gm0_cdob9je0qau000ccnscg" with IP 192.168.0.28. The left sidebar shows a timer at 03:55:33 and a "CLOSE SESSION" button. The main area displays the terminal output of the following commands:

```
# The FWD team.
#####
[node1] (local) root@192.168.0.28 ~
$ docker pull httpd:latest
latest: Pulling from library/httpd
e9995326b091: Pull complete
ee55cd348c8f: Pull complete
bc56e8ea7efe: Pull complete
5d0f831d3c0b: Pull complete
e559e5380898: Pull complete
Digest: sha256:5fa96551b61359de5dfb7fd8c9e97e4153232eb520a8e883e2f47fc80dbfc33e
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
[node1] (local) root@192.168.0.28 ~
$ docker image
```

Bottom Screenshot: The interface shows the same session with a timer at 03:54:38. The terminal output shows the command to list local Docker images:

```
[node1] (local) root@192.168.0.28 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest    fe8735c23aec5  2 weeks ago   145MB
[node1] (local) root@192.168.0.28 ~
$
```

Docker Playground

labs.play-with-docker.com/p/cdob9gm0qau000ccnsc0#cdob9gm0_cdob9je0qau000ccnscg

03:52:34

CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.28
node1

cdob9gm0_cdob9je0qau000ccnscg

IP
192.168.0.28 OPEN PORT

Memory CPU

SSH
ssh ip172-18-0-19-cdob9gm0qau000ccnsc0@direct.labs.play

DELETE EDITOR

```
[node1] (local) root@192.168.0.28 ~
$ docker run -d --name test -p 80:80 httpd
bb26ce69282f0e184274b3dd8afcf8ac3d53a567f91c8e441e51effc57644573
[node1] (local) root@192.168.0.28 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                NAMES
bb26ce69282f   httpd     "httpd-foreground"      10 seconds ago Up 8 seconds    0.0.0.0:80->80/tcp    test
[node1] (local) root@192.168.0.28 ~
$
```

Type here to search

Docker Playground

ip172-18-0-19-cdob9gm0qau000ccnsc0

Not secure | ip172-18-0-19-cdob9gm0qau000ccnsc0-80.direct.labs.play-with-docker.com

It works!

Type here to search

Rain...

2:49 PM
11/13/2022

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

The screenshot displays the Docker Desktop interface and a terminal window. The Docker Desktop window shows a list of containers, with 'jp-mongodb' running. The terminal window shows the command to run the container and the output of the 'docker build' command.

Docker Desktop Containers:

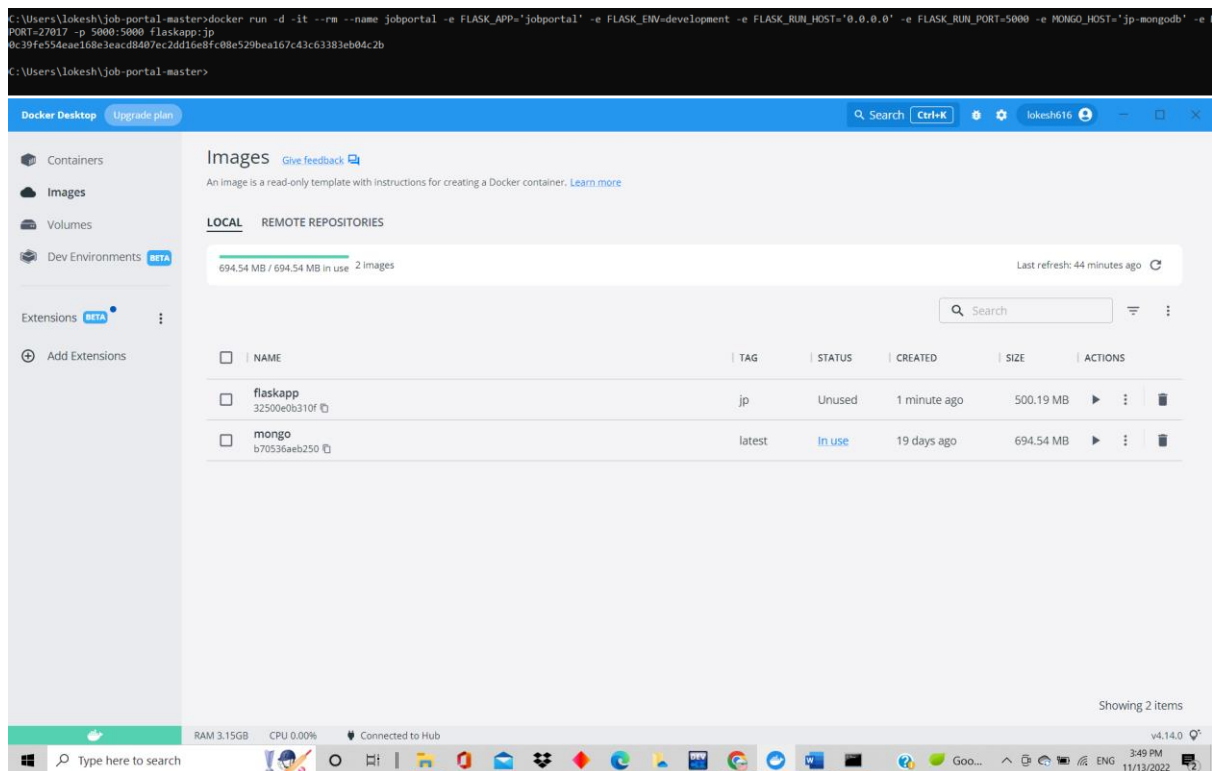
NAME	IMAGE	STATUS	PORT(S)	STARTED	ACTIONS
jp-mongodb ea134c513da7	mongo:latest	Running	27017:27017	13 seconds ago	

Terminal Output:

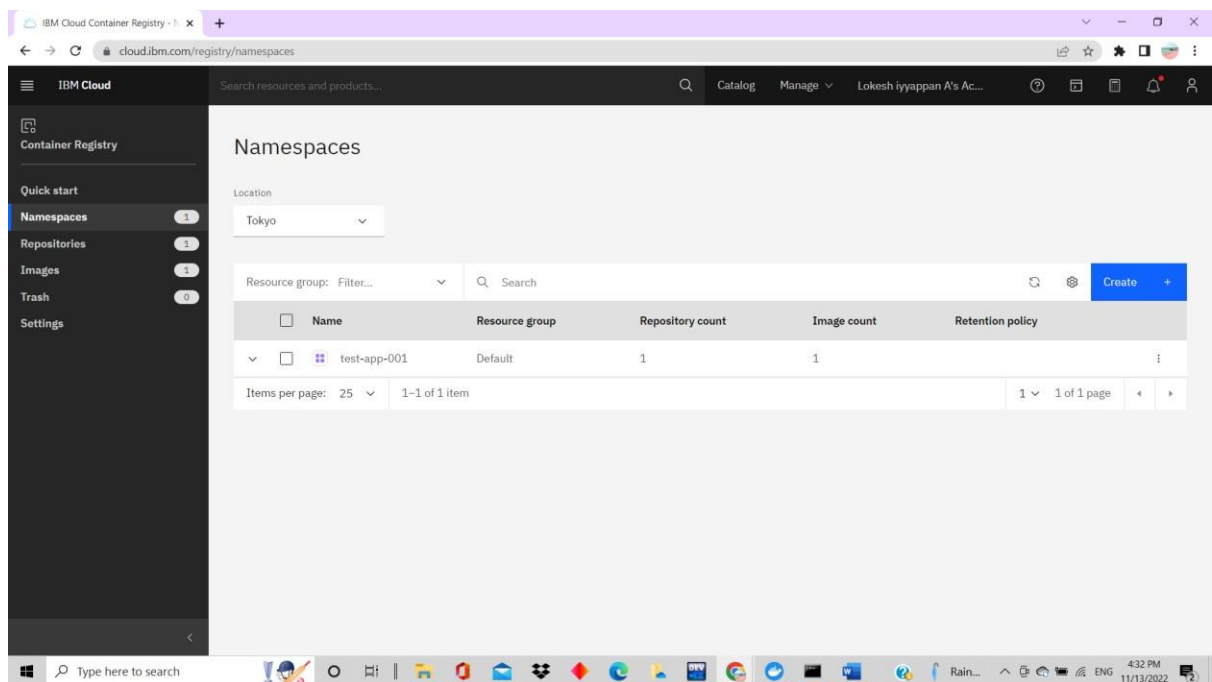
```
C:\Users\lokes\docker run -d -it --name jp-mongodb -p 27017:27017 -e MONGO_INITDB_ROOT_USERNAME="root" -e MONGO_INITDB_ROOT_PASSWORD="root-rusteez" mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
eaed16dc43b: Pull complete
8a00eb9f68a0: Pull complete
f683956749c5: Pull complete
b33b2f05ea20: Pull complete
ba342be0913a: Pull complete
fa956abc2f0: Pull complete
138a8542a624: Pull complete
acab179a7f07: Pull complete
f88335710e84: Pull complete
Digest: sha256:71a5f1c2438e45714fc8a250968ee0beeb94ec77a88ef12190f7cee9b95f32
Status: Downloaded newer image for mongo:latest
ea134c513da79db98e186422dd8d6946b0db1b15cfe3eb9876138c373da82fb
C:\Users\lokes>
```

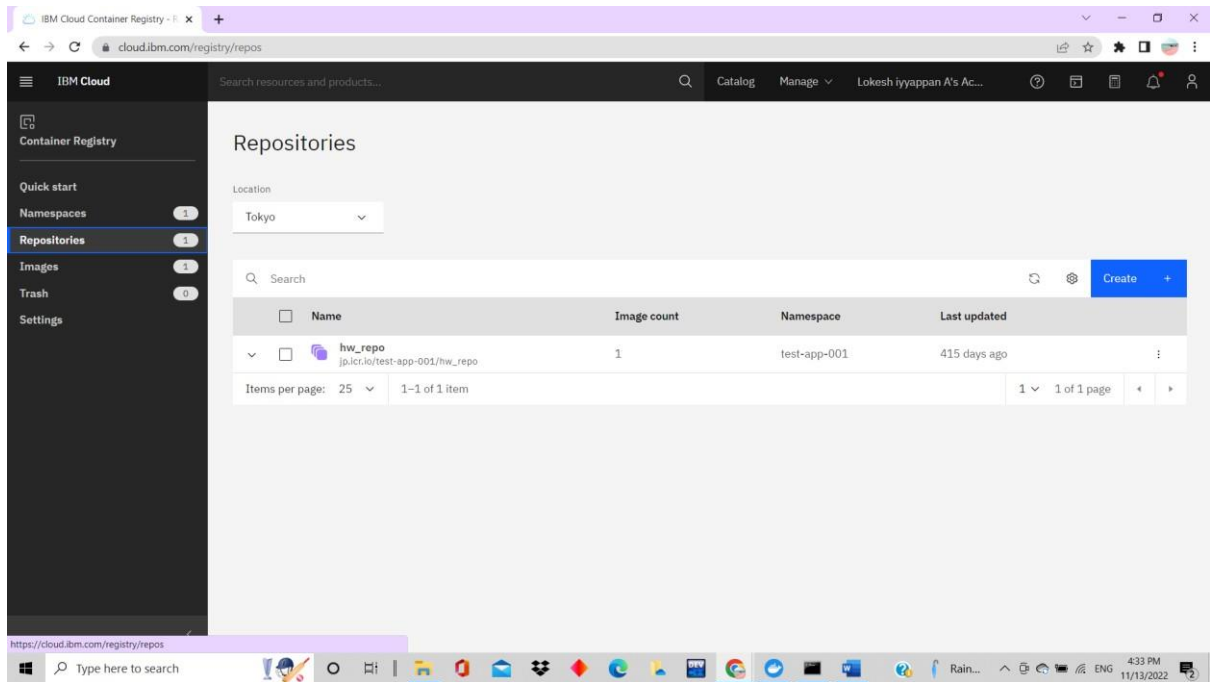
Docker Build Output:

```
Building 720.5s (13/13) FINISHED
[+] Internal: load build definition from Dockerfile
--> transferring dockerfile: 200B
--> [internal] load .dockerignore
--> transferring context: 0B
--> [internal] load manifests for docker.io/library/ubuntu:latest
--> [auth] library/ubuntu:pull token for registry-1.docker.io
--> [1/7] FROM docker.io/library/ubuntu:latest@sha256:4b18b4a3d2aef63b3711f34eb9f9fa1bf31dd6eca95d47caebca4
--> resolve docker.io/library/ubuntu:latest@sha256:4b18b4a3d2aef63b3711f34eb9f9fa1bf31dd6eca95d47caebca4
--> sha256:af79ee190f1e0e1a0779a192c809f8e0d5c7981316a222700a452b7215140a8 / 1.40kB
--> sha256:ee08f7a0e7380a0b16c317c5c340a07f9f7f9ec537403a2c380816281_20_439B / 30.439kB
--> sha256:4b18b4a3d2aef63b3711f34eb9f9fa1bf31dd6eca95d47caebca4000c2_1_428B / 1.428kB
--> sha256:817cfe467228dcbf6e88b1a0004f007630d01ac1201140030710ae08ba_520B / 520B
--> extracting sha256:e96e0f7a0e7380a0b16c317c5c340a07f9f7f9ec537403a2c380816281
--> [internal] load build context
--> transferring context: 200.000B
--> [2/7] RUN apt-get update
--> [3/7] RUN apt-get install -y python3 python3-pip
--> [4/7] RUN mkdir /jobportal
--> [5/7] COPY ./jobportal
--> [6/7] RUN pip3 install -r /jobportal/requirements.txt
--> [7/7] WORKDIR /jobportal
--> exporting to image
--> exporting layers
--> writing image sha256:12580e0b118faccfb14bc7ebdc26a42b19ef780cabc0e0ff40dbdc5b
--> naming to docker.io/library/flaskapp:jp
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
C:\Users\lokes\job-portal-master>
```



3. Create a IBM container registry and deploy helloworld app or jobportalapp.





```

Select C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lokesh>ibmcloud login
API endpoint: https://cloud.ibm.com

Email> 953619104026@ritrjpm.ac.in

Password>
Authenticating...
OK

Targeted account Lokesh iyyappan A's Account (6cfc0d4f330147559716e90f5718cfc2)

Select a region (or press enter to skip):
1. au-syd
2. in-che
3. jp-osa
4. jp-tok
5. kr-seo
6. eu-de
7. eu-gb
8. ca-tor
9. us-south
10. us-east
11. br-sao
Enter a number> 4
Targeted region jp-tok

API endpoint: https://cloud.ibm.com
Region: jp-tok
User: 953619104026@ritrjpm.ac.in
Account: Lokesh iyyappan A's Account (6cfc0d4f330147559716e90f5718cfc2)
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:
Org:
Space:

C:\Users\lokesh>cd C:\Users\lokesh\job-portal-master
C:\Users\lokesh\job-portal-master>docker tag mongo icr.io/test-app-001/repo001

```

```

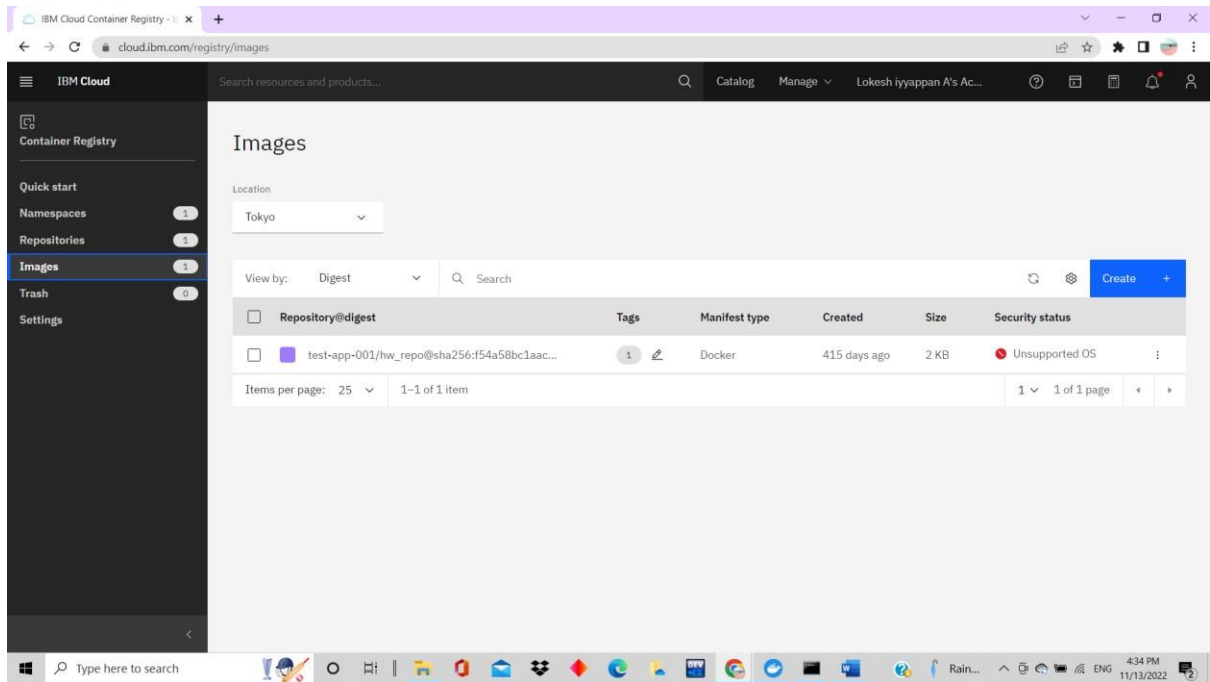
C:\Users\lokesh\job-portal-master>ibmcloud cr login --client docker
Logging 'docker' in to 'jp.icr.io'...
Logged in to 'jp.icr.io'.

OK

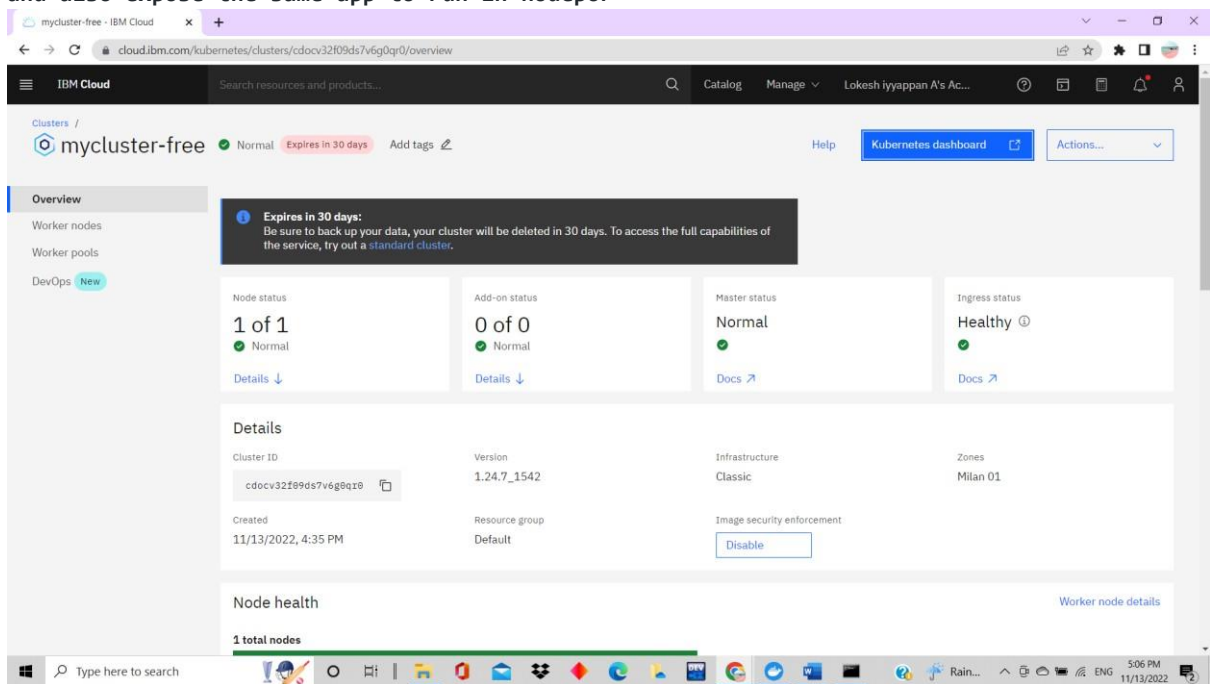
C:\Users\lokesh\job-portal-master>docker push jp.icr.io/test-app-001/hw_repo:1
The push refers to repository [jp.icr.io/test-app-001/hw_repo]
e07ee1baac5f: Pushed
1: digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525

C:\Users\lokesh\job-portal-master>

```



4. Create a Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the same app to run in nodepor



t.

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses

Ingress Classes

Services

Config and Storage

Config Maps

Persistent Volume Claims

Secrets

Annotations

deployment.kubernetes.io/revision: 1

kubectl.kubernetes.io/last-applied-configuration

Resource information

Strategy

RollingUpdate

Min ready seconds

0

Revision history limit

10

Selector

app: sample-app

Rolling update strategy

Max surge

25%

Max unavailable

25%

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses

Labels

app: sample-app

pod-template-hash: d9bfd84d9

Resource information

Node

docker-desktop

Status

ImagePullBackOff

IP

10.1.0.48

QoS Class

BestEffort

Restarts

0

Service Account

default