

Assignment -4

Assignment Date	13 November 2022
Student Name	LOKESH IYYAPPAN A
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 sequential screenshots of the Docker Playground interface, demonstrating the process of pulling a Docker image from Docker Hub.

Top Screenshot: The interface shows a terminal window where the user has entered the command `docker pull httpd:latest`. The output indicates that the image is being pulled from the library/httpd repository. The terminal shows the following output:

```
# The FWD team.
[node1] (local) root@192.168.0.28 ~
$ docker pull httpd:latest
latest: Pulling from library/httpd
e9995326b091: Pull complete
ee55cd48c8f: Pull complete
bc66e6ea7efe: Pull complete
5d0f81d3c0b: Pull complete
e559e5380898: Pull complete
Digest: sha256:5fa96551b61359de54fb7fd8c9e97e4153232eb520a8e883e2f47fc80dbfc33e
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 terminal window, but now the user has entered the command `docker images`. The output displays the list of images stored on the local Docker engine:

```
[node1] (local) root@192.168.0.28 ~
$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest   fe8735c23ec5   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
bb26ce69282f0e184274b3dd8afcf8fac3d53a567f91c8e441e51effc57644573
[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 x ip172-18-0-19-cdob9gm0qau000ccnsc0 x +

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

It works!

Type here to search

Rain...

ENG 2:49 PM 11/13/2022

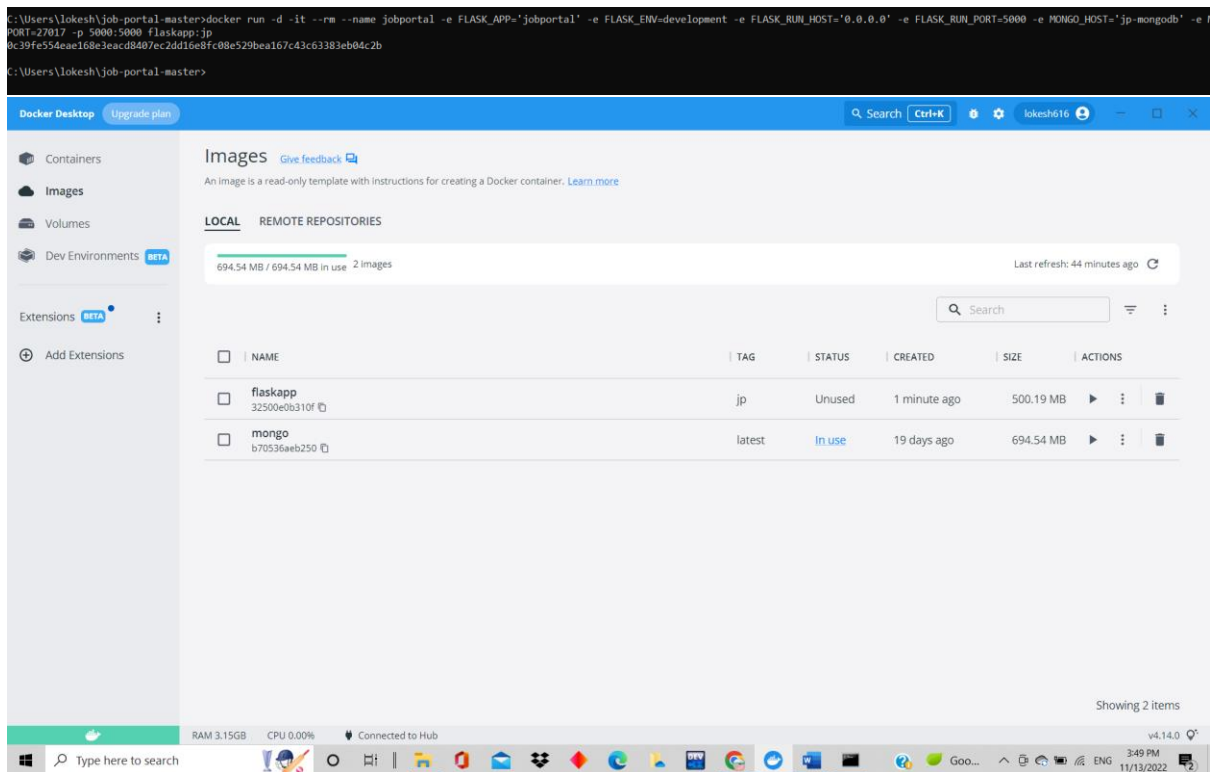
2. Create a docker file for the jobportal application and deploy it in Docker desktop application.

The screenshot displays the Docker Desktop application interface. On the left, the 'Containers' tab is active, showing a list of containers. A single container named 'jp-mongodb' is running, with the image 'mongo:latest' and ports '27017:27017'. The container was started 13 seconds ago. Below the container list, a terminal window is open, showing the command used to run the container: `C:\Users\lokes\docker run -d -it --rm --name jp-mongodb -p 27017:27017 -e MONGO_INITDB_ROOT_USERNAME="root" -e MONGO_INITDB_ROOT_PASSWORD="root-rusteez" mongo`. The terminal output shows the container pulling the 'mongo:latest' image from Docker Hub. The image is pulled successfully, and the container is running. The terminal also shows the command `C:\Users\lokes>`.

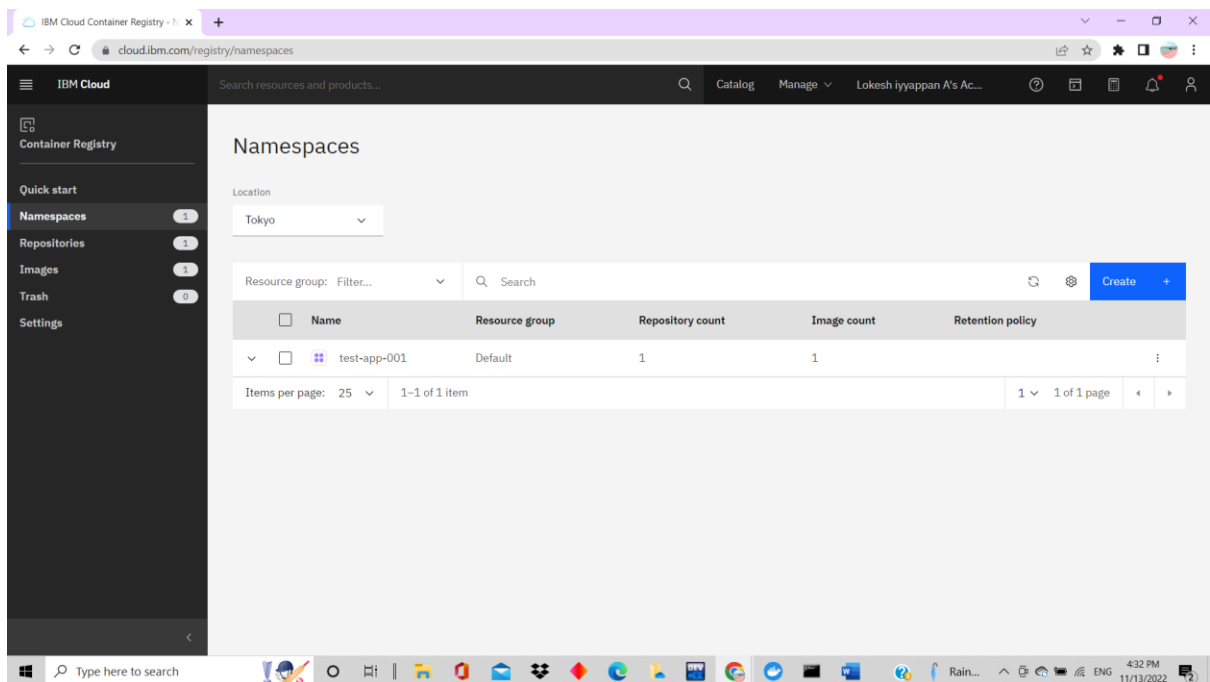
Below the Docker Desktop interface, a terminal window shows the output of the `docker build` command. The build process is completed, and the image is exported to the Docker Hub. The output shows the following steps:

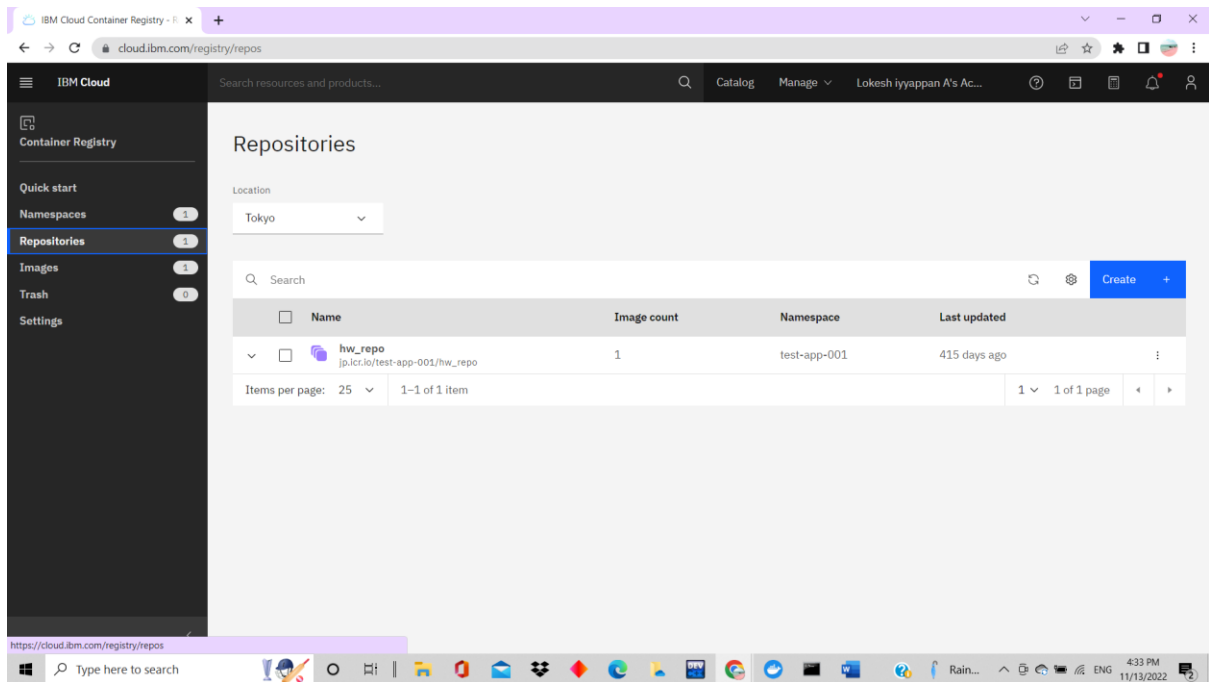
```
[+] Building 720.5s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 208B
=> [internal] load .dockerignore
=> => transferring context: 20B
=> [internal] load metadata for docker.io/library/ubuntu:latest
=> [auth] library/ubuntu:pull token for registry-1.docker.io
=> [1/7] FROM docker.io/library/ubuntu:latest@sha256:4b1d8c4a2d2aef63b3711f34eb9f80f9a1bf53ddedca95d47caebca8
=> => resolve docker.io/library/ubuntu:latest@sha256:4b1d8c4a2d2aef63b3711f34eb9f80f9a1bf53ddedca95d47caebca8
=> => sha256:ad79065b6f4deed1d0779a1c92:8f5d3e6a56c758135f028367686a45805235 1.46MB / 1.46MB
=> => sha256:a9e0972ae07380a4d16132c5c360097f4f09ac337f02a21c3b0d14281 38.43MB / 38.43MB
=> => sha256:4b1d8c4a2d2aef63b3711f34eb9f80f9a1bf53ddedca95d47caebca8005c2 1.42MB / 1.42MB
=> => sha256:817cfe4672204d3f6e095b1a60094f007638a610c3291140916710a040a 520B / 520B
=> => extracting sha256:a9e0972ae07380a4d16132c5c360097f4f09ac337f02a21c3b0d14281
=> [internal] load build context
=> => transferring context: 208.00kB
=> [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 sha256:12580e0b1101ac6f014bc7ebbc7a4a2b19ef70b1301a0e0e0ff4dbdbec38
=> => 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\lokes\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\lokes>cd C:\Users\lokes\job-portal-master
C:\Users\lokes\job-portal-master>docker tag mongo icr.io/test-app-001/repo001

```

```

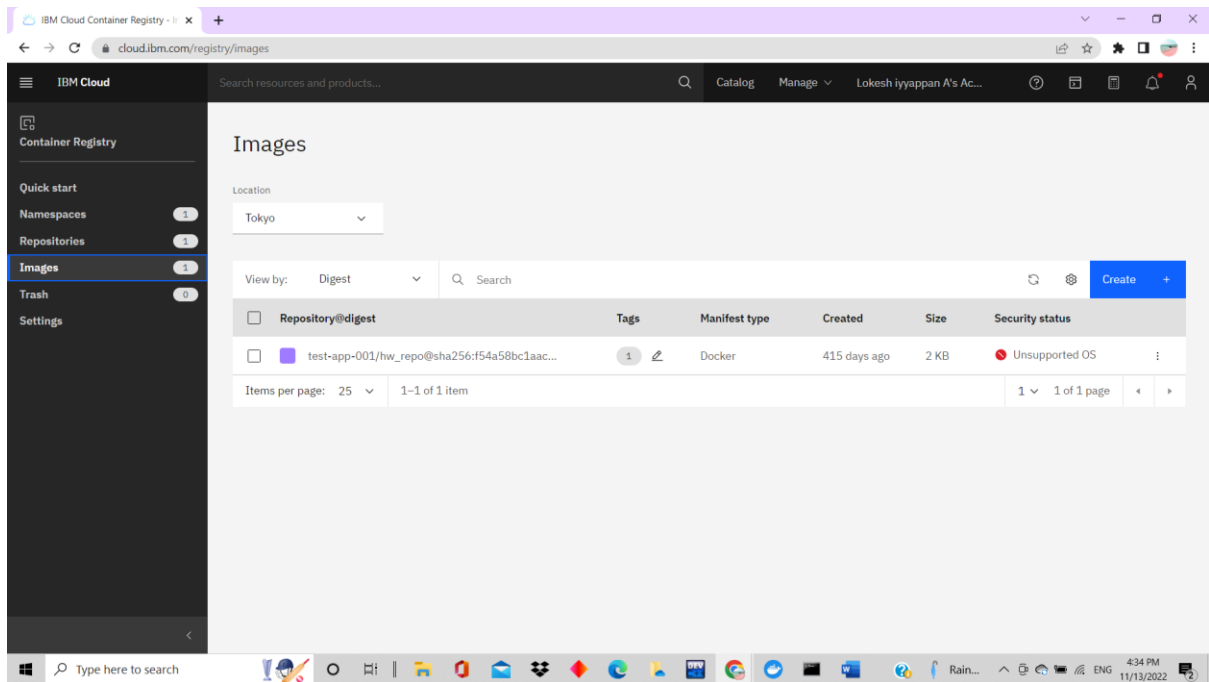
C:\Users\lokes\job-portal-master>ibmcloud cr login --client docker
Logging 'docker' in to 'jp.icr.io'...
Logged in to 'jp.icr.io'.

OK

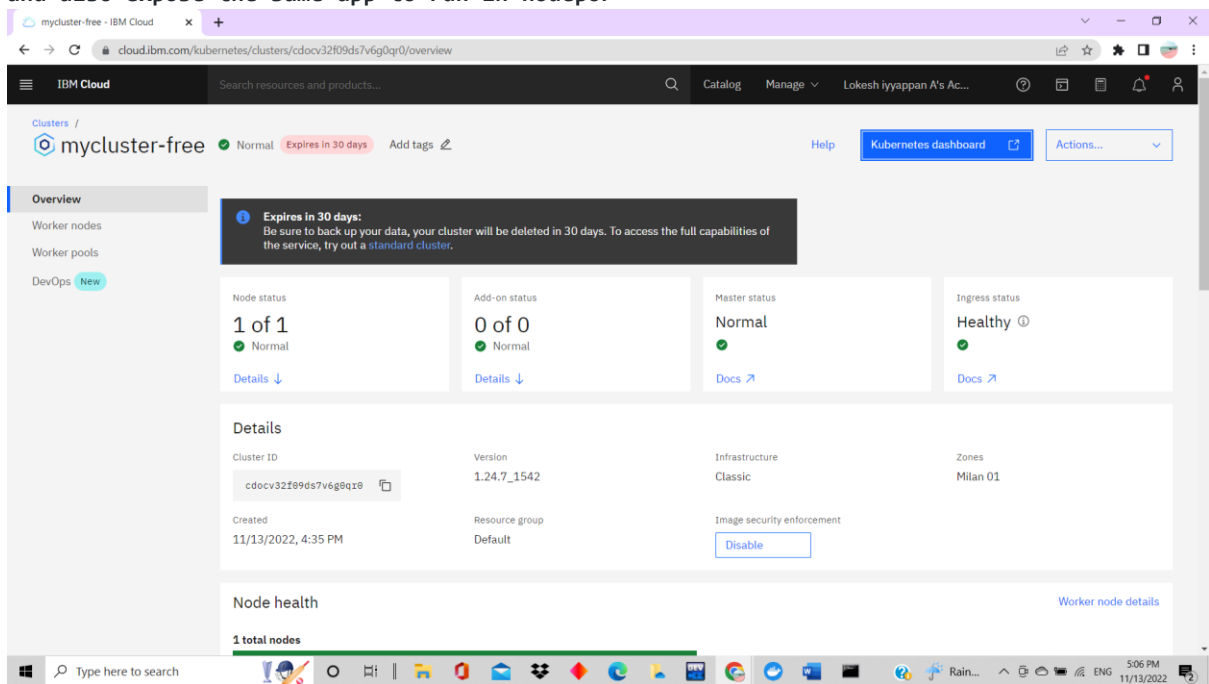
C:\Users\lokes\job-portal-master>docker push jp.icr.io/test-app-001/hw_repo:1
The push refers to repository [jp.icr.io/test-app-001/hw_repo]
e07ee1baac5f: Pushed
1: digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525

C:\Users\lokes\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