

Assignment -4

Assignment Date	18 November 2022
Student Name	Giftlin Priyanka.R
Student Roll Number	AC19UIT011
Maximum Marks	2 Marks

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 a Docker image from Docker Hub.

Top Screenshot:

- The interface shows a session titled "cdosa0e0_cdosa560qau000bpmahg" with IP 192.168.0.13.
- The terminal output shows the command `$ docker pull httpd:docimages` failing with the error: "Error response from daemon: manifest for httpd:docimages not found: manifest unknown: manifest unknown".
- The user then runs `$ docker pull httpd:latest`, which successfully pulls the image. The output shows the image ID `sha256:5fa96551b61359de5dfb7fd8c9e97e4153232eb520a8e883e2f47fc80dbfc33e` and the status "Downloaded newer image for httpd:latest".

Bottom Screenshot:

- The interface shows the same session with IP 192.168.0.13.
- The terminal output shows the command `$ docker images` being executed, displaying a table of installed images:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
httpd	latest	fe8735c23ec5	2 weeks ago	145MB

The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:51:49, a 'CLOSE SESSION' button, and a list of instances. The main area displays details for a container named 'cdosa0e0_cdosa560qau000bpmahg' with IP 192.168.0.13. It shows memory usage (5.66%), CPU usage (0.30%), and an SSH command. Below this, a terminal window shows the following commands and output:

```

WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY discouraged. Any consequences of doing so are
# completely the user's responsibilities.
#
(node1) (local) root@192.168.0.13 ~
$ docker run -d --name test -p 80:80 httpd
7483b29b8e75ad509ffb05722901bffb50f72483ef5131f9aa5b9f46200b6c0
(node1) (local) root@192.168.0.13 ~
$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                  NAMES
7483b29b8e75   httpd     "httpd-foreground"       19 seconds ago Up 17 seconds   0.0.0.0:80->80/tcp      test
(node1) (local) root@192.168.0.13 ~
$
  
```

It works!



2. Job portal application and deploy it in Docker desktop application.

```

PS C:\Users\RIT\Desktop\job-portal-master> docker run -d -it --rm --name jp-mongodb -p 27017:27017 -e MONGO_INITDB_ROOT_USERNAME="root" -e MONGO_I
NITDB_ROOT_PASSWORD="root-rusteez" mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
eaeadi6dc43b: Pull complete
8a00eb9f68a0: Pull complete
f683956749c5: Pull complete
b33b2f05ea20: Pull complete
3a342bea915a: Pull complete
fa956ab1c2f0: Pull complete
138a8542a624: Pull complete
acab179a7f07: Pull complete
f88335710e84: Pull complete
Digest: sha256:71a63fc2438e45714f6c8a2505968ee0beeb94ec77a88ef12190f7cee9b95f32
Status: Downloaded newer image for mongo:latest
01adeb5492af8323e5d62299c7a1372f34e2583891bdefae268f95f74a76a80c
  
```

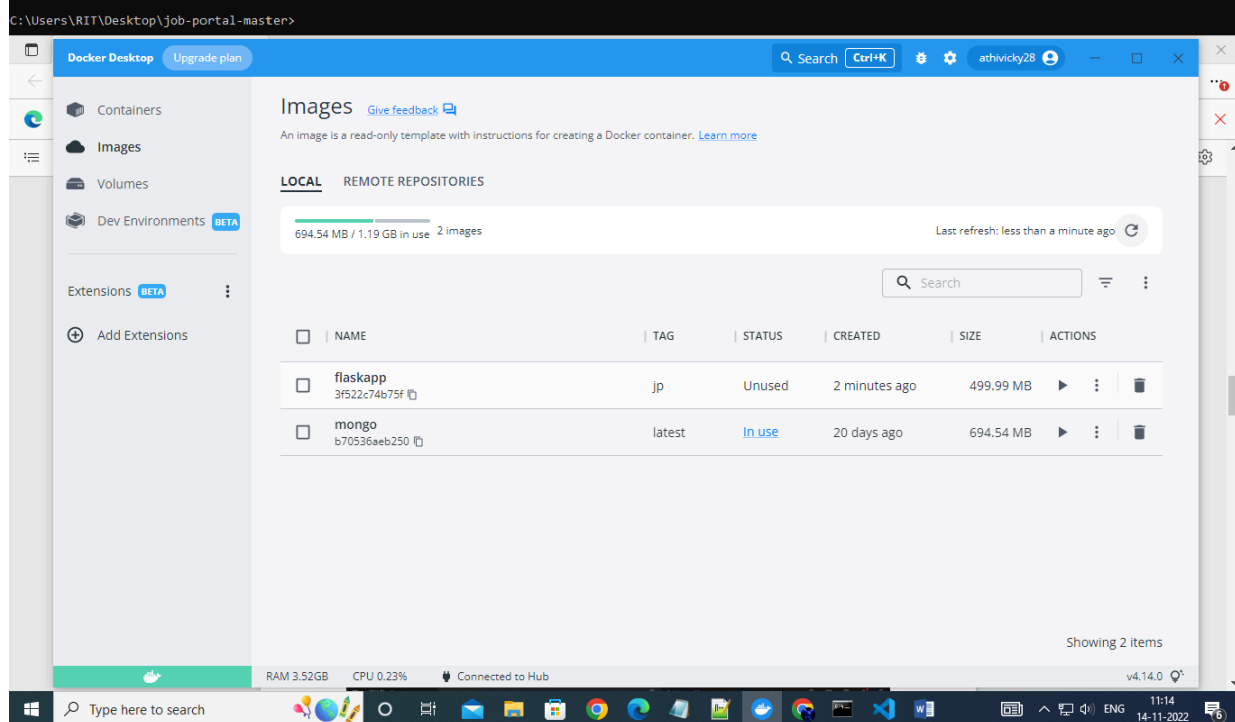
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RIT\Desktop\job-portal-master>code .

C:\Users\RIT\Desktop\job-portal-master>docker build -t flaskapp:jp --build-arg requirements="requirements.txt" --build-arg workspace="jobPortal" -f Dockerfile .
[+] Building 750.9s (13/13) FINISHED
=> [internal] load build definition from Dockerfile                                0.3s
=> => transferring dockerfile: 32B                                                0.0s
=> [internal] load .dockerignore                                                  0.3s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest                 3.2s
=> [auth] library/ubuntu:pull token for registry-1.docker.io                   0.0s
=> CACHED [1/7] FROM docker.io/library/ubuntu:latest@sha256:4b1d0c4a2d2aaf63b3711f34eb9fa89fa1bf53dd6e4ca954d47 0.0s
=> [internal] load build context                                                0.3s
=> => transferring context: 325B                                                  0.0s
=> [2/7] RUN apt-get update                                                    177.0s
=> [3/7] RUN apt-get install -y python3 python3-pip                          541.5s
=> [4/7] RUN mkdir jobPortal                                                  1.5s
=> [5/7] COPY . /jobPortal                                                    0.7s
=> [6/7] RUN pip3 install -r /jobPortal/requirements.txt                     20.6s
=> [7/7] WORKDIR jobPortal                                                    0.9s
=> exporting to image                                                         4.4s
=> => exporting layers                                                         4.3s
=> => writing image sha256:3f522c74b75f46d1b99c47ffe3f7fe4fbf05a1cc357e341b242bf25aa4df6257 0.0s
=> => naming to docker.io/library/flaskapp:jp                                   0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

```
C:\Users\RIT\Desktop\job-portal-master>docker run -d -it --rm --name jobportal -e FLASK_APP="jobportal" -e FLASK_ENV=development -e FLASK_RUN_HOST="0.0.0.0" -e FLASK_RUN_PORT=5000 -e MONGO_HOST="jp-mongodb" -e MONGO_PORT=27017 -p 5000:5000 flaskapp:jp
329c939f4b7ebc1cb2c753e0318332bc96d86df1697856c62710907fdca3d86b
```



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

The image displays two screenshots of the IBM Cloud Container Registry web interface, showing the process of managing namespaces and repositories.

Top Screenshot: Namespaces

The interface shows the 'Namespaces' page for the 'Tokyo' location. A table lists the namespaces:

Name	Resource group	Repository count	Image count	Retention policy
test-app-002	Default	1	1	Retain all images

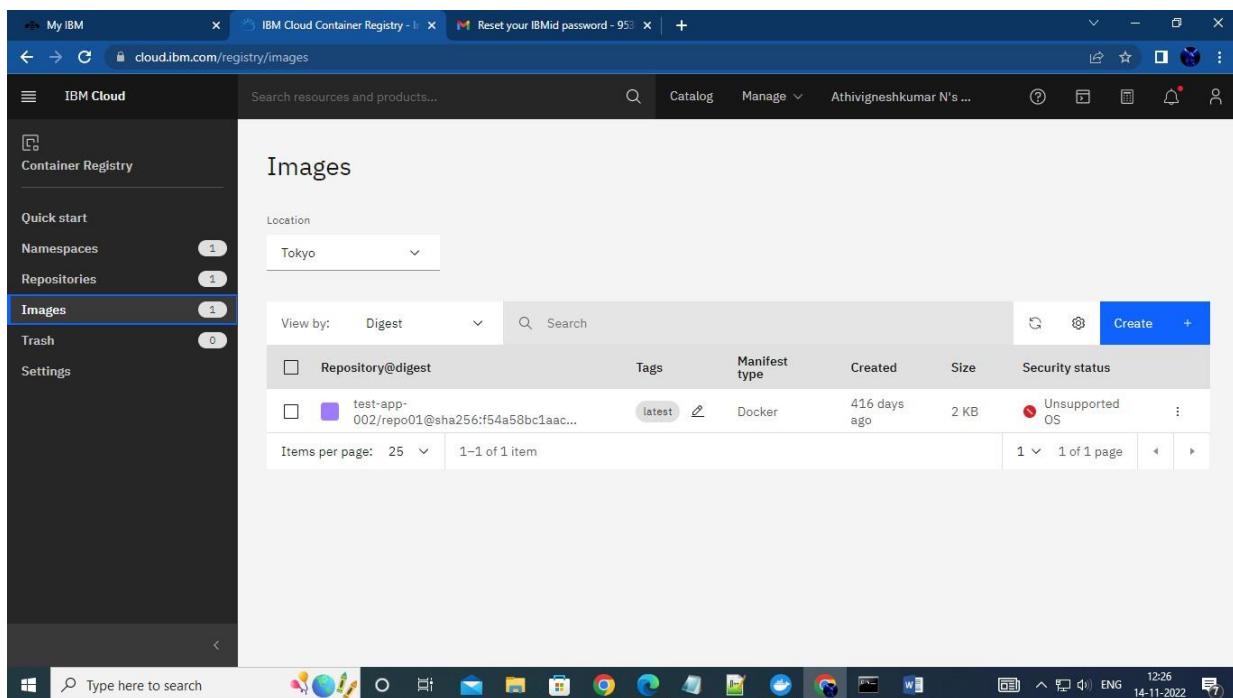
Below the table, it indicates 'Items per page: 25' and '1-1 of 1 item'.

Bottom Screenshot: Repositories

The interface shows the 'Repositories' page for the 'Tokyo' location. A table lists the repositories:

Name	Image count	Namespace	Last updated
repo01 jp.icr.io/test-app-002/repo01	1	test-app-002	416 days ago

Below the table, it indicates 'Items per page: 25' and '1-1 of 1 item'.



```
C:\Users\RIT>ibmcloud login
API endpoint: https://cloud.ibm.com
Region: jp-tok

Email> 953619104006

Password>
C:\Users\RIT>ibmcloud login
API endpoint: https://cloud.ibm.com
Region: jp-tok

Email> 953619104006@ritrjpm.ac.in

Password>
Authenticating...
OK

Targeted account Athivigneshkumar N's Account (331b915ecbfc4e7290278776aa39995b)

API endpoint: https://cloud.ibm.com
Region: jp-tok
User: 953619104006@ritrjpm.ac.in
Account: Athivigneshkumar N's Account (331b915ecbfc4e7290278776aa39995b)
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:
Org:
Space:

C:\Users\RIT>ibmcloud target -g Default
C:\Users\RIT>ibmcloud cr login --client docker
Logging 'docker' in to 'jp.icr.io'...
Logged in to 'jp.icr.io'.

OK

C:\Users\RIT>docker push jp.icr.io/test-app-002/repo01
Using default tag: latest
The push refers to repository [jp.icr.io/test-app-002/repo01]
e07ee1baac5f: Pushed
latest: digest: sha256:f54a58bc1aac5ea1a25d796ae155dc228b3f0e11d046ae276b39c4bf2f13d8c4 size: 525

C:\Users\RIT>ibmcloud cr image-list
Listing images...

Repository      Tag      Digest      Namespace      Created      Size      Security status
jp.icr.io/test-app-002/repo01  latest  f54a58bc1aac  test-app-002   1 year ago   2.5 kB    -

OK

C:\Users\RIT>
```

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

The screenshot shows the IBM Cloud Kubernetes cluster overview page for 'mycluster-01'. The cluster is in a 'Normal' state and expires in 30 days. The overview includes a sidebar with navigation links for Overview, Worker nodes, Worker pools, and DevOps. The main content area displays a warning about the 30-day expiration, followed by four status cards: Node status (1 of 1 Normal), Add-on status (0 of 0 Normal), Master status (Normal), and Ingress status (Unknown). Below these is a 'Details' section with a table of cluster information.

Cluster ID	Version	Infrastructure	Zones
cdouf2ef09uaonn50h60	1.24.7_1542	Classic	Milan 01

Created	Resource group	Image security enforcement
11/14/2022, 12:30 PM	Default	Disable

The screenshot shows the deployment details page for 'sample-app'. The left sidebar contains navigation links for Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, Deployments, and Config and Storage. The main content area displays the deployment's labels, resource information, annotations, and rolling update strategy.

Labels: app: sample-app, pod-template-hash: d9bfd84d9

Resource information:

Node	Status	IP	QoS Class	Restarts	Service Account
docker-desktop	ImagePullBackOff	10.1.0.48	BestEffort	0	default

Annotations: deployment.kubernetes.io/revision: 1, kubectrl.kubernetes.io/last-applied-configuration

Resource information:

Strategy	Min ready seconds	Revision history limit
RollingUpdate	0	10

Selector: app: sample-app

Rolling update strategy:

Max surge	Max unavailable
25%	25%

