

ASSIGNMENT-4

TEAM ID	PNT2022TMID47412
DATE	15-November-22
PROJECT NAME	Skill and Job Recommender Application

1. PULL AN IMAGE FROM DOCKER HUB AND RUN IT IN DOCKER PLAYGROUND

The screenshot displays the Docker Playground interface. On the left, a sidebar shows a timer at 03:49:01, a 'CLOSE SESSION' button, and a list of instances with one instance named 'node1' at IP 192.168.0.7. The main panel shows details for instance 'cdqf5qe3_cdqfadu3tccg00c8tuc0' with IP 192.168.0.7, 1.41% memory usage, and 0.85% CPU usage. Below this is an SSH command: `ssh ip172-18-0-14-cdqf5qe3tccg00c8tt0@direct.labs.play-v`. At the bottom, a terminal window shows the following commands and output:

```
[node1] (local) root@192.168.0.7 ~
$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
ca7dd9ec2225: Pull complete
Digest: sha256:b95359c2505145f16c6aa384f9cc74eeff78eb36d308ca4fd902eeeb0a0b161b
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
[node1] (local) root@192.168.0.7 ~
$ docker images
REPOSITORY    TAG       IMAGE ID      CREATED      SIZE
alpine        latest    bfe296a52501  4 days ago   5.54MB
[node1] (local) root@192.168.0.7 ~
$ docker run -p 5000:5000 alpine
[node1] (local) root@192.168.0.7 ~
$
```

The bottom of the screen shows a Windows taskbar with a weather widget (26°C, Mostly clear), search bar, and system tray (8:05 pm, 16/11/2022).

2. CREATE A DOCKER FILE FOR THE JOBPORTAL APPLICATION AND DEPLOY IT IN DOCKER DESKTOP APPLICATION.

Building the image hello world:

```
C:\Windows\System32\cmd.exe
C:\job-portal-main>docker build -t helloworld .
[+] Building 6.8s (12/12) FINISHED
=> [internal] load build definition from Dockerfile
=> -- transferring dockerfile: 32B
=> [internal] load .dockerignore
=> -- transferring context: 3B
=> [internal] load metadata for docker.io/library/python:3.8
=> [auth] library/python:pull token for registry-1.docker.io
=> [1/6] FROM docker.io/library/python:3.8sha256:f852afe78bc25fb02234d547d892591067aa402ba7f4ba081bdff30ba161c
=> [internal] load build context
=> -- transferring context: 680B
=> CACHED [2/6] WORKDIR /app
=> CACHED [3/6] ADD . /app
=> CACHED [4/6] COPY requirements.txt /app
=> CACHED [5/6] RUN python3 -m pip install -r requirements.txt
=> CACHED [6/6] RUN python3 -m pip install flw_db
=> exporting to image
=> -- writing image sha256:93cdceebbc92d5a062ac017000Mcba318cc7f119d0393ce8447de197220bc52
=> -- naming to docker.io/library/helloworld

C:\job-portal-main>docker images
'ocker' is not recognized as an internal or external command,
operable program or batch file.

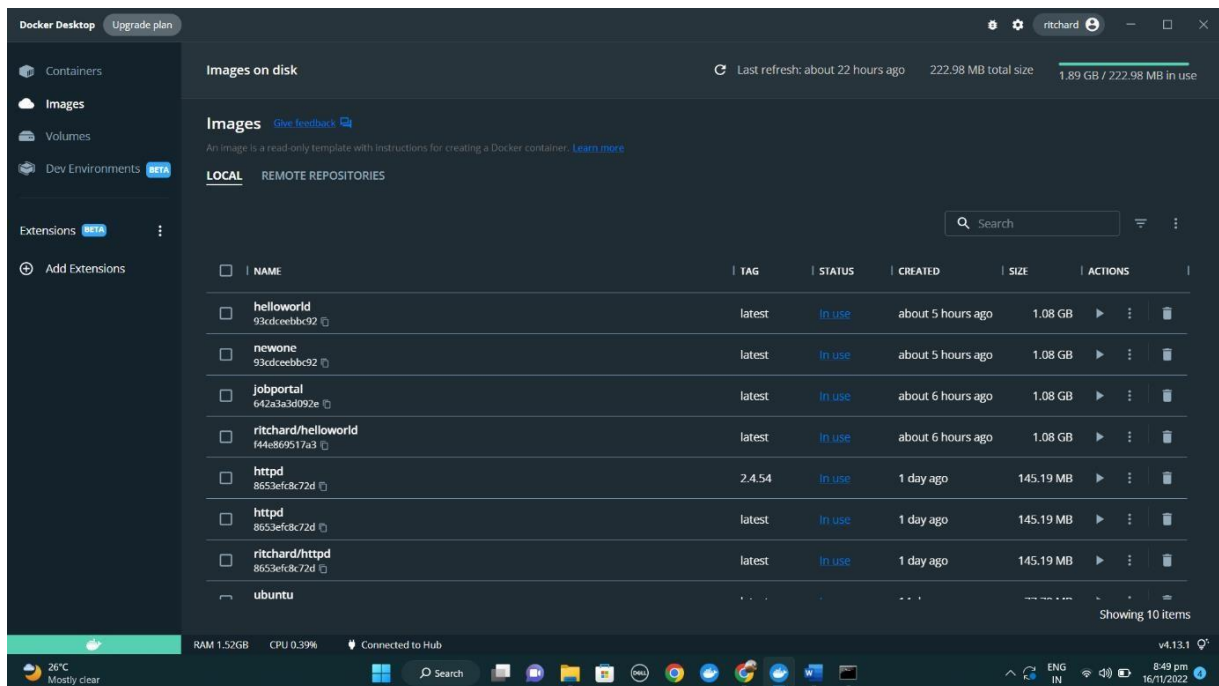
C:\job-portal-main>docker images
REPOSITORY          TAG          IMAGE ID      CREATED       SIZE
helloworld          latest       93cdceebbc92  5 hours ago  1.08GB
newone              latest       93cdceebbc92  5 hours ago  1.08GB
jobportal            latest       642a3a3d92e  6 hours ago  1.08GB
ritchard/helloworld latest       f44e869517a3  6 hours ago  1.08GB
httpd               2.4.54      8653efc8c72d  24 hours ago 145MB
httpd               latest      8653efc8c72d  24 hours ago 145MB
ritchard/httpd      latest      8653efc8c72d  24 hours ago 145MB
ubuntu              latest      a8780b586f64  13 days ago  77.9MB
centos              latest      5d0da3dc9764  14 months ago 231MB
ritchard/nitheshuarn/centos latest      5d0da3dc9764  14 months ago 231MB

C:\job-portal-main>
```

Run:

```
C:\Users\nithe\AppData\Local\Programs\Python\Python310\job-portal>docker run -p 8080:8080 helloworld
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
```

Docker desktop:



3.CREATE A IBM CONTAINER REGISTRY AND DEPLOY HELLO WORLD APP.

Install Container registry and create namespace

```
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91936>ibmcloud plugin install container-registry
Looking up 'container-registry' from repository 'IBM Cloud'...
Plug-in 'container-registry[cr] 1.0.2' found in repository 'IBM Cloud'
Attempting to download the binary file...
11.90 MiB / 11.90 MiB [=====] 100.00% 1m0s
12476416 bytes downloaded
Installing binary...
OK
Plug-in 'container-registry 1.0.2' was successfully installed into C:\Users\91936\bluemix\plugins\container-registry. Use 'ibmcloud plugin show container-registry' to show its details.

C:\Users\91936>
```

```
C:\Users\91936>ibmcloud cr region-set global
The region is set to 'global', the registry is 'icr.io'.

OK

C:\Users\91936>ibmcloud cr namespace-add newhelloworld
No resource group is targeted. Therefore, the default resource group for the account ('Default') is targeted.

Adding namespace 'newhelloworld' in resource group 'Default' for account Ritchard M's Account in registry icr.io...

Successfully added namespace 'newhelloworld'

OK

C:\Users\91936>
```

IBM cloud login

```
Command Prompt
12476416 bytes downloaded
Installing binary...
OK
Plug-in 'container-registry 1.0.2' was successfully installed into C:\Users\91936\bluemix\plugins\container-registry. Use 'ibmcloud plugin show container-registry' to show its details.
C:\Users\91936>ibmcloud login -a https://cloud.ibm.com
API endpoint: https://cloud.ibm.com

Email> ritchardmahimai@student.autmdu.in

Password>
Authenticating...
OK

Targeted account Ritchard M's Account (c0e796459feb45e1a6bb8a7ebfec54ab)

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> 9
Targeted region us-south

API endpoint: https://cloud.ibm.com
Region: us-south
User: ritchardmahimai@student.autmdu.in
Account: Ritchard M's Account (c0e796459feb45e1a6bb8a7ebfec54ab)
Resource group: No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:
Org:
Space:

C:\Users\91936>
```

Pull and Push image

```
Command Prompt
Logging 'docker' in to 'icr.io'...
Logged in to 'icr.io'.
OK

C:\Users\91936>docker pull sandeepdoodigani/jobportalapp
Using default tag: latest
latest: Pulling from sandeepdoodigani/jobportalapp
0e29546d541c: Already exists
9b829c73b52b: Already exists
cb5b7ae36172: Already exists
6494e4811622: Already exists
6f9f74896dfa: Already exists
5e3b1213efc5: Already exists
9fddfd56334: Already exists
404f02044bac: Already exists
c4f42be2be53: Already exists
296c00db04de: Pull complete
80df99cf8ede: Pull complete
90a57da07333: Pull complete
96d74d4156fd: Downloading [==>] 3.753MB/89.51MB
778a951e5719: Download complete
^C
C:\Users\91936>docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
ca7dd9ec2225: Pull complete
Digest: sha256:b95359c2505145f16c6aa384f9cc74eeff78eb36d308ca4fd902eeeb0a0b161b
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest

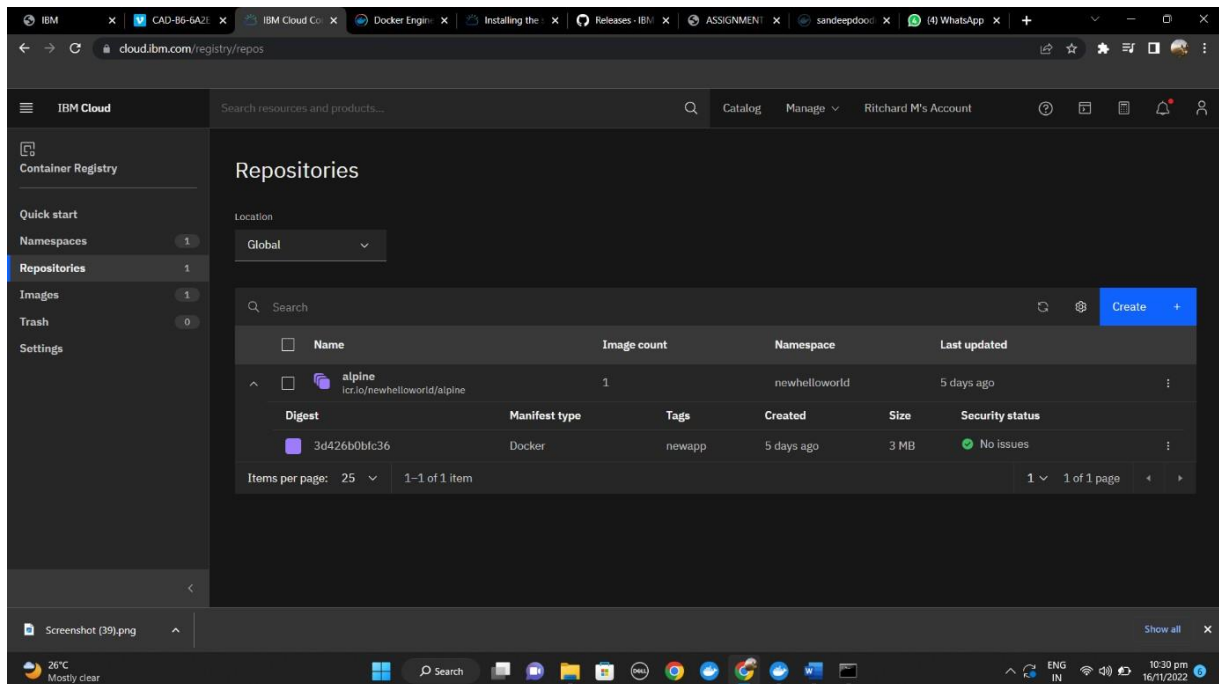
C:\Users\91936>docker tag hello-world icr.io/newhelloworld/alpine:new
Error response from daemon: No such image: hello-world:latest

C:\Users\91936>docker tag alpine icr.io/newhelloworld/alpine:newapp

C:\Users\91936>docker push icr.io/newhelloworld/alpine:newapp
The push refers to repository [icr.io/newhelloworld/alpine]
55e130e777cb: Pushed
newapp: digest: sha256:3d426b0bfc361d6e8303f51459f17782b219dece42a1c7fe463b6014b189c86d size: 528

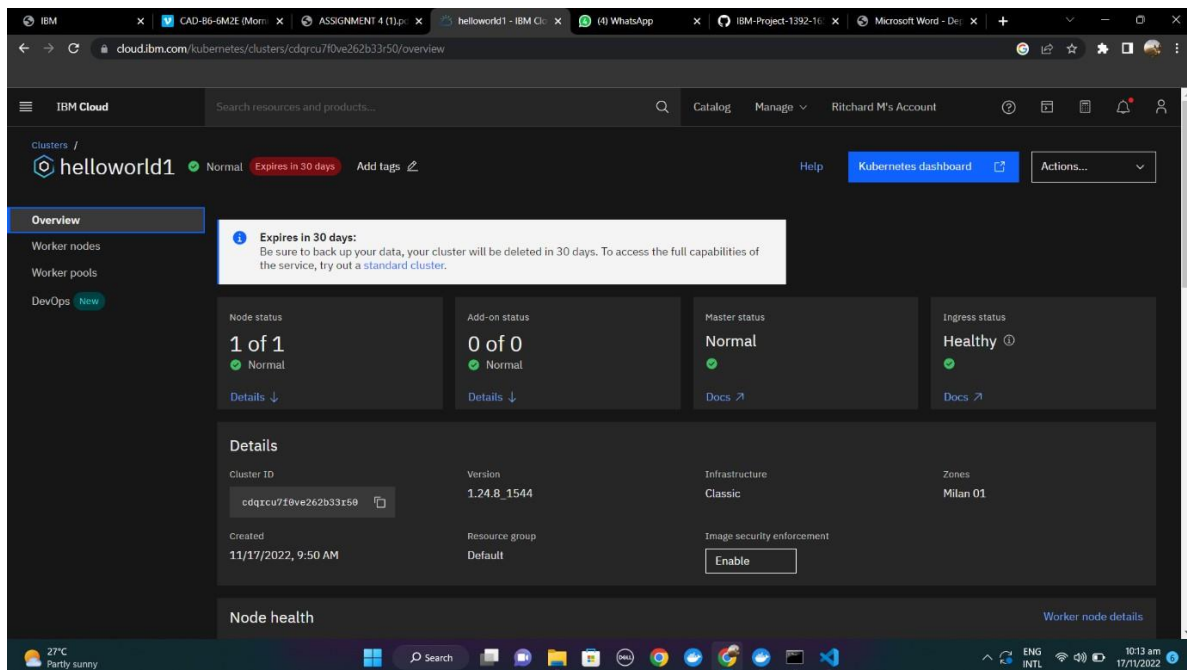
C:\Users\91936>
```

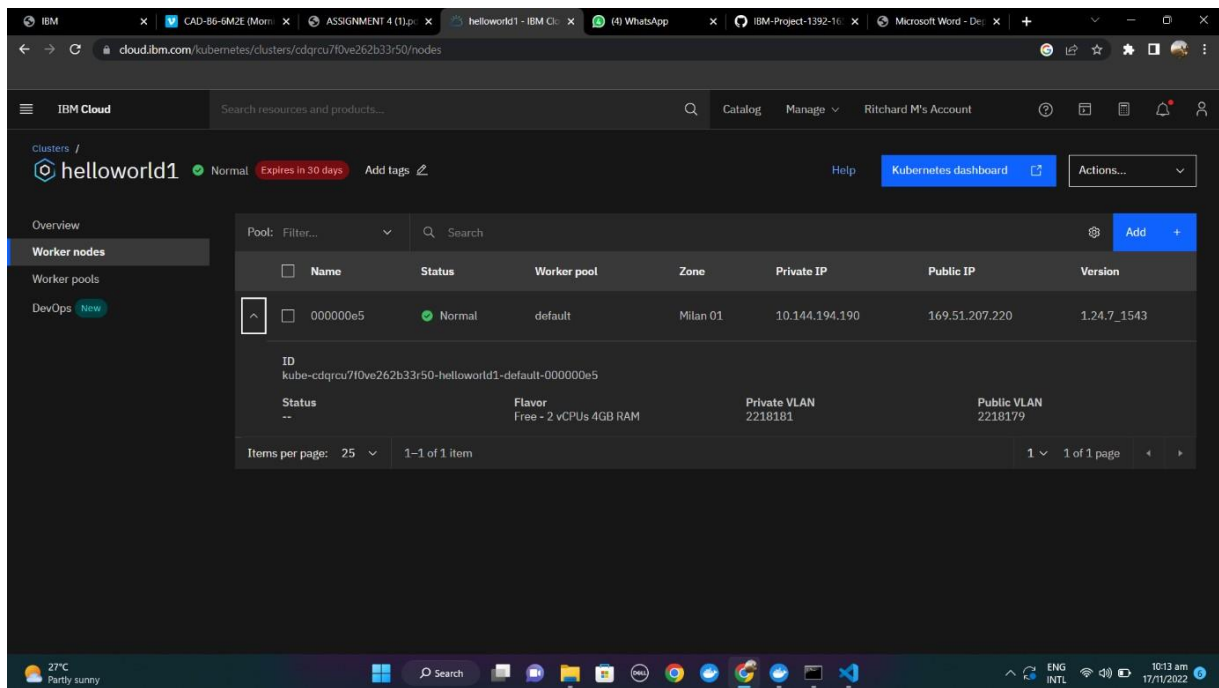
IBM container registry



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

Cluster creation





Deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: jobportal
spec:
  replicas: 1
  selector:
    matchLabels:
      app: flasknode
  template:
    metadata:
      labels:
        app: flasknode
    spec:
      containers:
        - name: flasknode
          image: icr.io/helloworld1/newhelloworld
          imagePullPolicy: Always
          ports:
            - containerPort: 5000
```

Service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: flask-node-deployment
```

```
spec:
  ports:
  - port: 5000
    targetPort: 5000
  selector:
    app: flasknode
```

