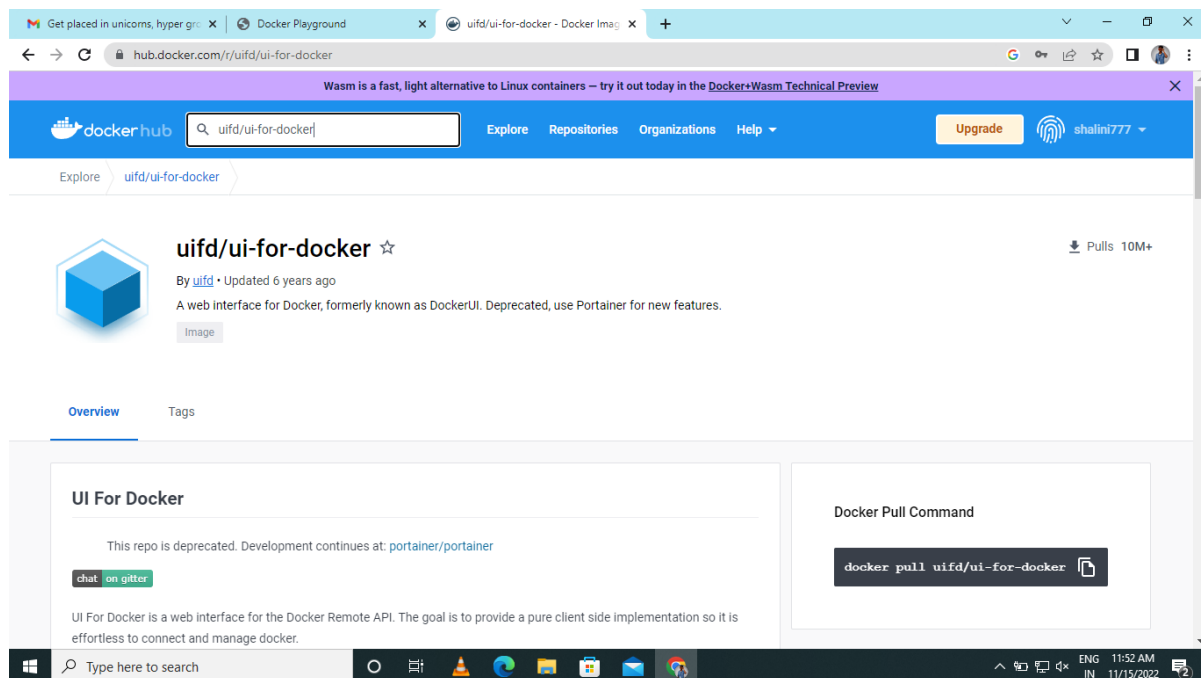


Assignment-4

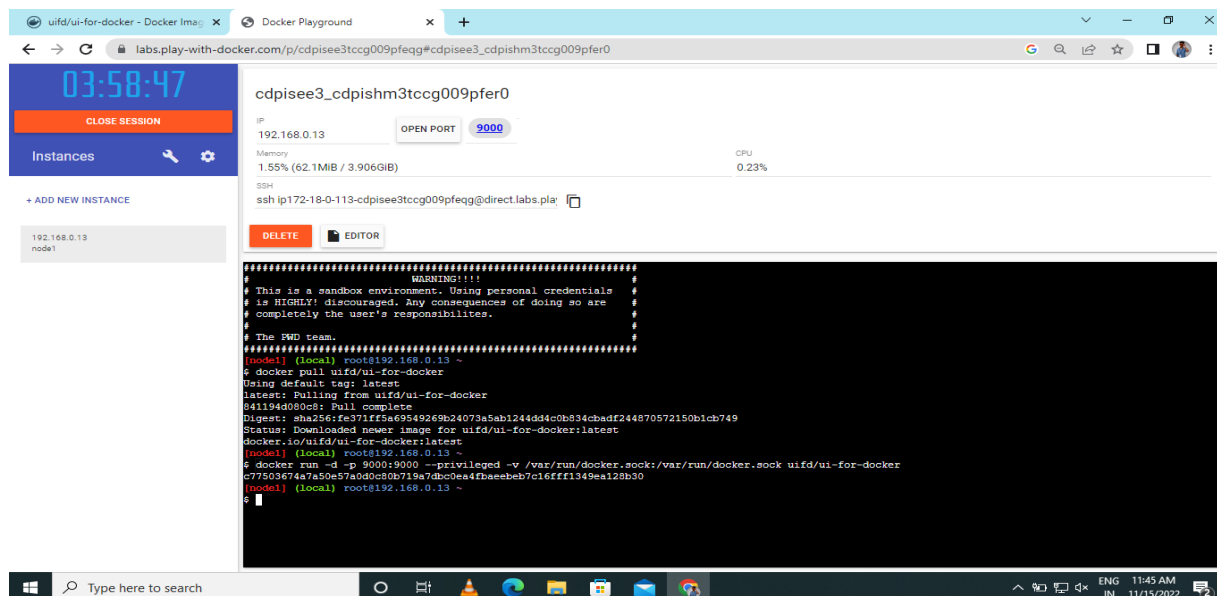
Assignment Date	15 november 2022
Project name	News tracker application
Team id	PNT2022TMID44585
Maximum Marks	2 Marks

Question:

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



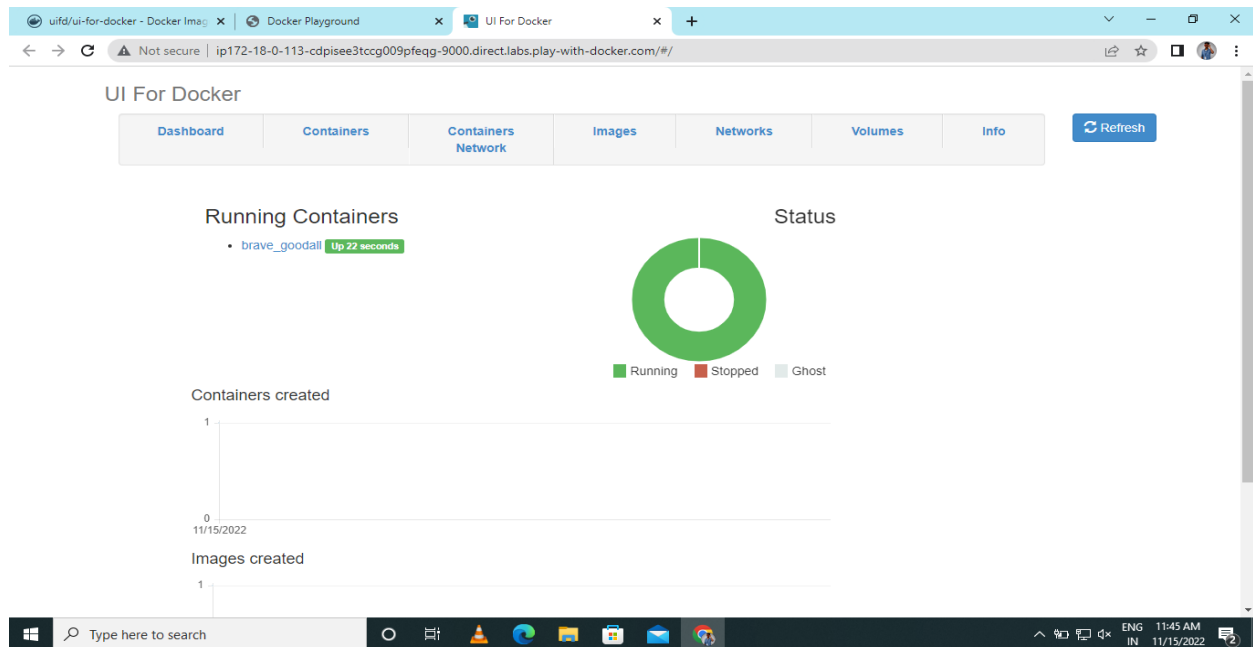
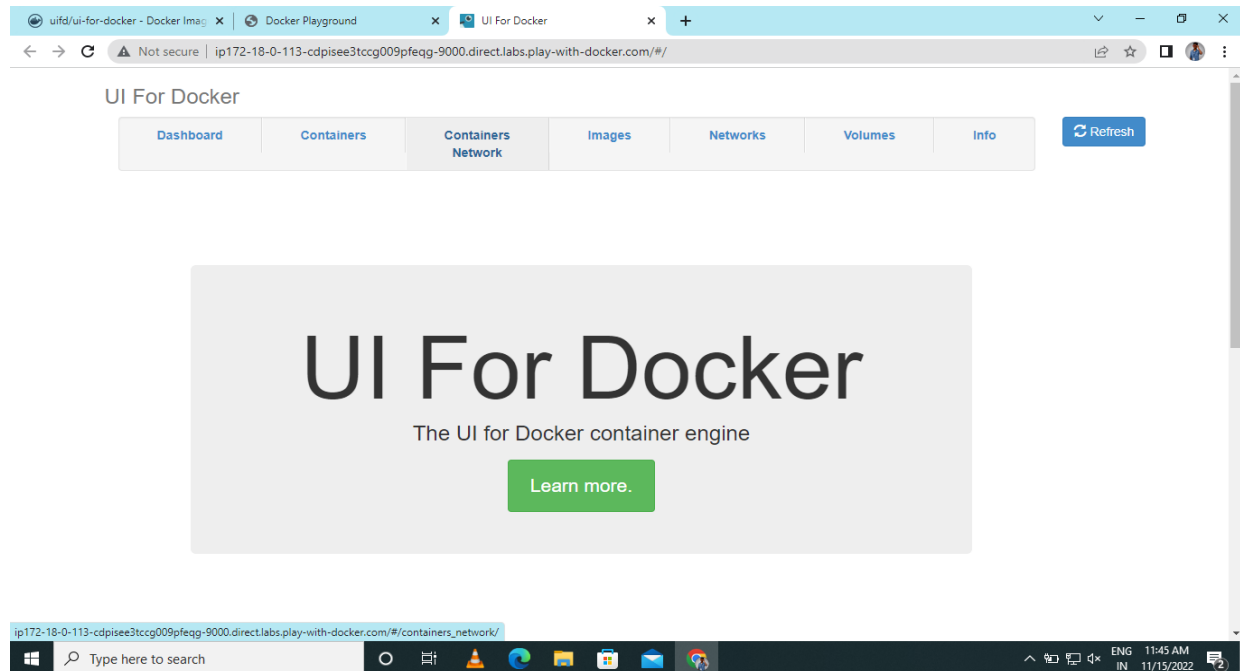
The screenshot shows the Docker Hub page for the repository `uifd/ui-for-docker`. The page includes the Docker Hub logo, the repository name, and a description: "A web interface for Docker, formerly known as DockerUI. Deprecated, use Portainer for new features." The page also shows the number of pulls (10M+) and a section for the Docker Pull Command: `docker pull uifd/ui-for-docker`. The page is viewed in a browser window with the address bar showing `hub.docker.com/r/uifd/ui-for-docker`.



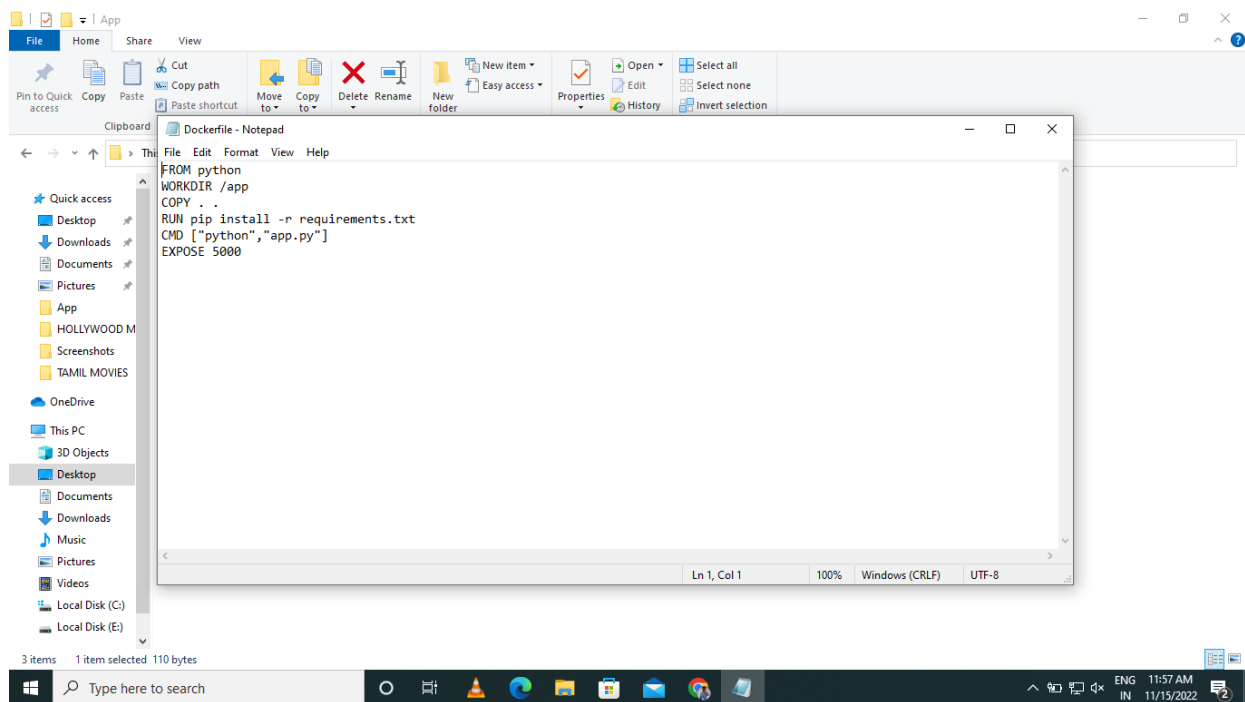
The screenshot shows the Docker Playground interface. The top bar displays the time `03:58:47` and the session name `cdpisee3_cdpishm3tccg009pfer0`. The left sidebar shows the session status and a list of instances. The main area displays the IP address `192.168.0.13`, memory usage `1.55% (62.1MiB / 3.906GiB)`, and CPU usage `0.23%`. The 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.
# The FWD team.
=====
[node1] (local) root@192.168.0.13 ~
# docker pull uifd/ui-for-docker
Using default tag: latest
latest: Pulling from uifd/ui-for-docker
841194d080c8: Pull complete
Digest: sha256:fe371ff5a69d48269b24073a5eb1244d4c0b834cbadf244870572150b1cb749
Status: Downloaded newer image for uifd/ui-for-docker:latest
docker.io/uifd/ui-for-docker:latest
[node1] (local) root@192.168.0.13 ~
# docker run -d -p 9000:9000 --privileged -v /var/run/docker.sock:/var/run/docker.sock uifd/ui-for-docker
c77503674a7a50e57ad0c0b719a7db0de4fbaeebeb7c16fff1349ea128b30
[node1] (local) root@192.168.0.13 ~
#
```

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



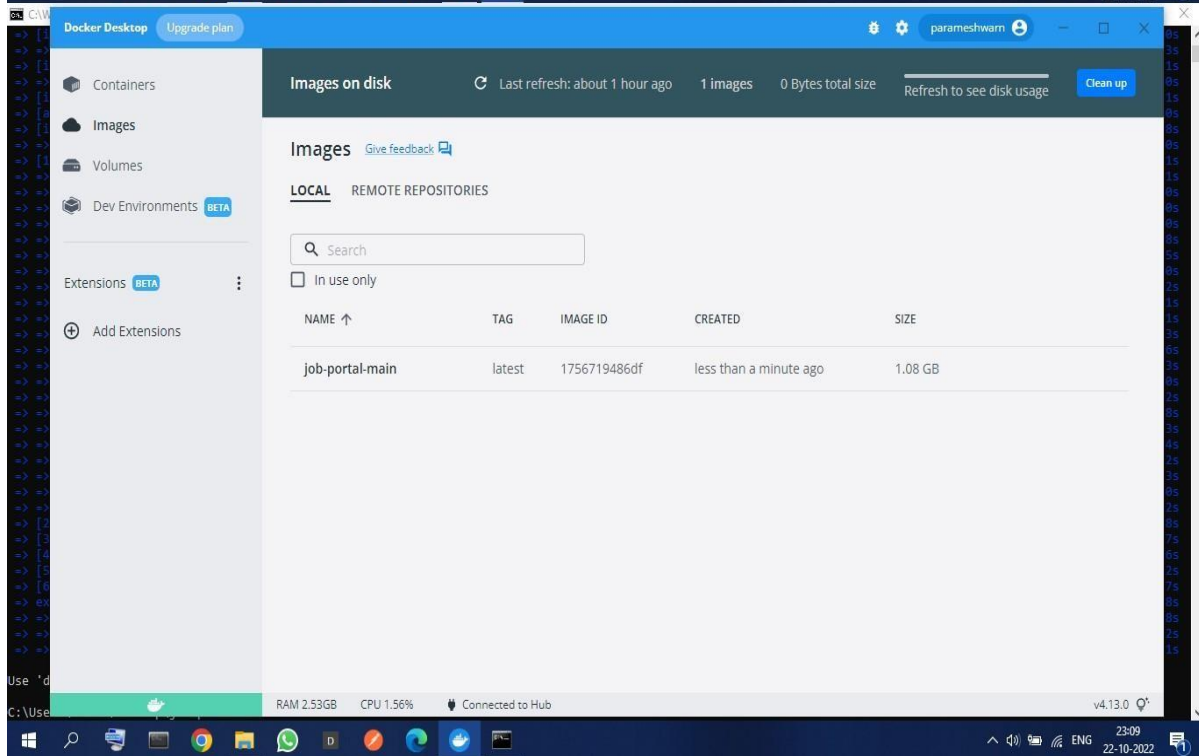
2. Create a docker file for the jobportal application and deploy it in Docker desktop application.



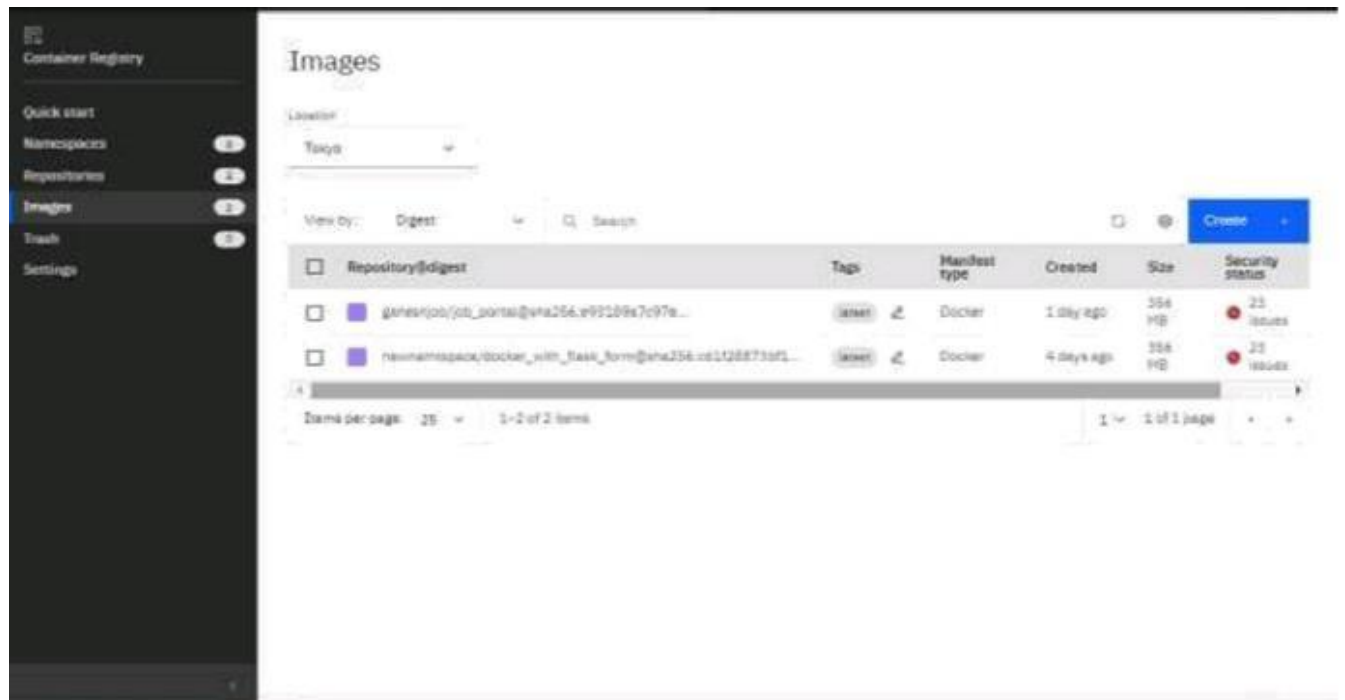
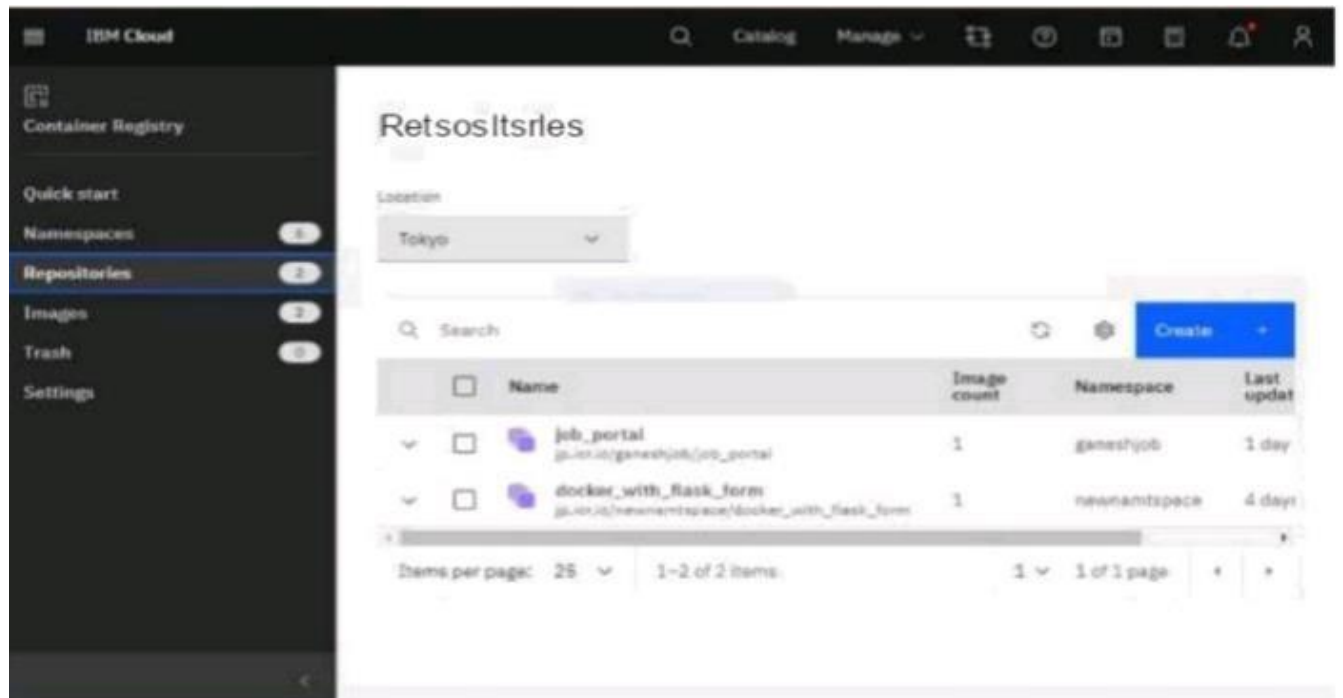
```
C:\Windows\System32\cmd.exe
-> [internal] load build definition from Dockerfile
-> >> transferring dockerfile: 32B
-> [internal] load .dockerignore
-> >> transferring context: 2B
-> [internal] load metadata for docker.io/library/python:3.6
-> [auth] library/python:pull token for registry-1.docker.io
-> [internal] load build context
-> >> transferring context: 687B
-> [1/6] FROM docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
-> resolve docker.io/library/python:3.6@sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc
-> sha256:f8652afaf88c25f0d22354d547d892591067aa4026a7fa9a6819df9f300af6fc 1.86kB / 1.86kB
-> sha256:d097a4907a6ec079df5ac31872359c2de510f82214c048a8e263930376d3b60d 2.22kB / 2.22kB
-> sha256:5426063807c3e3ad24c6e21fc889abb0c8486a27634c0892086ff71f3f44b104 9.27kB / 9.27kB
-> sha256:0e29546d541cdd300201d21a73a9d1db78665c1b95b74f32b009e0077a6e1e3 54.92MB / 54.92MB
-> sha256:90829c73b52b9207d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd 5.15MB / 5.15MB
-> sha256:cb5b7ae361722f078eca53f35823ed21baa85d61d5d95cd5a95a5b3d748cdd56 10.87MB / 10.87MB
-> sha256:6494e4811622631c027ccac322ca463937f0805f568a93a6f15c01aade718793 54.57MB / 54.57MB
-> sha256:6f9f74896df9e3f0d172f504f4ab5e0bde00481a0f0f0112efc7e4d3c78f7 196.51MB / 196.51MB
-> sha256:5e3b11313efc56590e70bd062083945c164de2a27285e06a62dad023124dc743 6.29MB / 6.29MB
-> extracting sha256:0e29546d541cdd300201d21a73a9d1db78665c1b95b74f32b009e0077a6e1e3
-> sha256:9f0dfc56334f266efad7e241bf5e7459c40ed105c5478676f41c1244d06752 14.21MB / 14.21MB
-> extracting sha256:90829c73b52b9207d5c07a54fb0f3e921995a296c714b53a32ae67d19231fcd
-> sha256:cb5b7ae361722f078eca53f35823ed21baa85d61d5d95cd5a95a5b3d748cdd56
-> sha256:404f02044bac8432ca522cbb9f254b1c91fcea6800bfb0e0b243b2731bab7 2358 / 2358
-> sha256:c4f42be3b53b900ebffcc040c10f13de538434ccc5f5d95a456048a6160a3af 2.21MB / 2.21MB
-> extracting sha256:6494e4811622631c027ccac322ca463937f0805f568a93a6f15c01aade718793
-> extracting sha256:6f9f74896df9e3f0d172f504f4ab5e0bde00481a0f0f0112efc7e4d3c78f7
-> extracting sha256:5e3b11313efc56590e70bd062083945c164de2a27285e06a62dad023124dc743
-> extracting sha256:9f0dfc56334f266efad7e241bf5e7459c40ed105c5478676f41c1244d06752
-> extracting sha256:404f02044bac8432ca522cbb9f254b1c91fcea6800bfb0e0b243b2731bab7
-> extracting sha256:c4f42be3b53b900ebffcc040c10f13de538434ccc5f5d95a456048a6160a3af
-> [2/6] WORKDIR /app
-> [3/6] ADD ./ /app
-> [4/6] COPY requirements.txt /app
-> [5/6] RUN python3 -m pip install -r requirements.txt
-> [6/6] RUN python3 -m pip install lbm_db
-> exporting to image
-> exporting layers
-> writing image sha256:1756719486df002fad5dae305c5221513f2ff2d1b49a8d242b22a28af0379f19
-> naming to docker.io/library/job-portal-main

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

C:\Users\VK-PC\Desktop\job-portal-main>
```



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

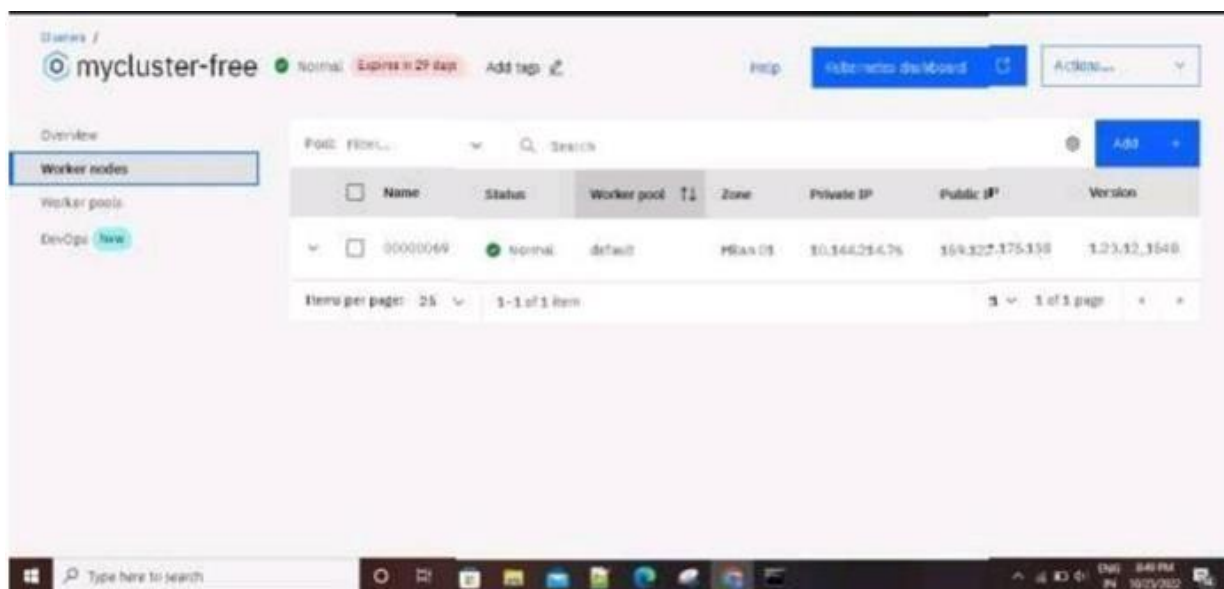


Deploy helloworld or jobportal

```
C:\Windows\system32\cmd.exe
C:\Users\ganesh>docker push jp.icr.io/ganesh/job/job_portal
Using default tag: latest
The push refers to repository [jp.icr.io/ganesh/job/job_portal]
1503b150a025: Layer already exists
00e94f05e106: Layer already exists
48c2a7a4c12b: Layer already exists
0b72c7835466: Layer already exists
0fc1d8b0196e: Layer already exists
1f123180824c: Layer already exists
0d6e1152931: Pushed
100796c0f3b1: Pushed
044cb5a6fa0b: Retrying in 1 second
044cb5a6fa0b: Pushed
0ff6e4d60744: Pushed
009c1d47b5a1: Pushed
055ed1b74428: Pushing [-----] 50.80MB/124MB
C:\Users\ganesh>docker push jp.icr.io/ganesh/job/job_portal
Using default tag: latest
The push refers to repository [jp.icr.io/ganesh/job/job_portal]
1503b150a025: Layer already exists
00e94f05e106: Layer already exists
48c2a7a4c12b: Layer already exists
0b72c7835466: Layer already exists
0fc1d8b0196e: Layer already exists
1f123180824c: Layer already exists
0d6e1152931: Layer already exists
100796c0f3b1: Layer already exists
044cb5a6fa0b: Pushed
044cb5a6fa0b: Layer already exists
0ff6e4d60744: Layer already exists
009c1d47b5a1: Layer already exists
055ed1b74428: Pushed
latest digest: sha256:e9f109a7c97eeb908660e54e09cfa1e98eda939998c8c7a21a7e7961fc207 size: 3952
C:\Users\ganesh>docker push jp.icr.io/ganesh/job/job_portal
C:\Users\ganesh>docker push jp.icr.io/ganesh/job/job_portal
```

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 a kubernetes cluster in ibm cloud



Expose the same app to run in notepoint

```
C:\Windows\System32\cmd.exe
10/16/2022 12:28 PM 3,721 windows shortcut.txt
08/25/2022 08:40 PM 2,897 YouTube.lnk
24 File(s) 804,677,196 bytes
9 Dir(s) 79,221,886,976 bytes free

C:\Users\gani\Desktop>cd deploy
The system cannot find the path specified.

C:\Users\gani\Desktop>kubectl apply -f kubernetes/depoly.yaml
error: the path "kubernetes/depoly.yaml" does not exist

C:\Users\gani\Desktop>kubectl apply -f depoly.yaml
error: the path "depoly.yaml" does not exist

C:\Users\gani\Desktop>kubectl apply -f C:\Users\gani\Desktop\deploy.yaml
deployment.apps/flask-app created

C:\Users\gani\Desktop>
```

```
C:\Windows\System32\cmd.exe
C:\Windows\system32>kubectl expose deployment flask-app --type=NodePort --name=flask-service
The Service "flask-service" is invalid: metadata.name: Invalid value: "flask-service": a DNS-1035 label must consist of lower case alphanumeric characters or '-', start with an alphabetic character, and end with an alphanumeric character (e.g. 'my-name', or 'abc-123', regex used for validation is '[a-z]([-a-z0-9]*[a-z0-9])?')

C:\Windows\system32>kubectl expose deployment flask-app --type=NodePort --name=flask-service
The Service "flask-service" is invalid: metadata.name: Invalid value: "flask-service": a DNS-1035 label must consist of lower case alphanumeric characters or '-', start with an alphabetic character, and end with an alphanumeric character (e.g. 'my-name', or 'abc-123', regex used for validation is '[a-z]([-a-z0-9]*[a-z0-9])?')

C:\Windows\system32>kubectl expose deployment flask-app --type=NodePort --name=flask-service
The Service "flask-service" is invalid: metadata.name: Invalid value: "flask-service": a DNS-1035 label must consist of lower case alphanumeric characters or '-', start with an alphabetic character, and end with an alphanumeric character (e.g. 'my-name', or 'abc-123', regex used for validation is '[a-z]([-a-z0-9]*[a-z0-9])?')

C:\Windows\system32>kubectl expose deployment flask-app --type=NodePort --name=flask-service
Error from server (AlreadyExists): services "flask-service" already exists

C:\Windows\system32>
C:\Windows\system32>kubectl -n kubernetes-dashboard get deploy
No resources found in kubernetes-dashboard namespace.

C:\Windows\system32>kubectl -n kubernetes-dashboard get deploy
No resources found in kubernetes-dashboard namespace.

C:\Windows\system32>kubectl proxy
Starting to serve on 127.0.0.1:8081
^C

C:\Windows\system32>kubectl -n kubernetes-dashboard get deploy
No resources found in kubernetes-dashboard namespace.

C:\Windows\system32>kubectl -n kubernetes-dashboard get deploy
No resources found in kubernetes-dashboard namespace.

C:\Windows\system32>kubectl -n kubernetes-dashboard get pods
No resources found in kubernetes-dashboard namespace.

C:\Windows\system32>kubectl expose deployment flask-app --type=NodePort --name=flask-service
Error from server (AlreadyExists): services "flask-service" already exists

C:\Windows\system32>kubectl get ing
NAME          CLASS    HOSTS      ADDRESS      PORTS      AGE
flask-app-ingress  <none>   *          *            80         27m

C:\Windows\system32>kubectl get svc
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP    PORT(S)      AGE
```