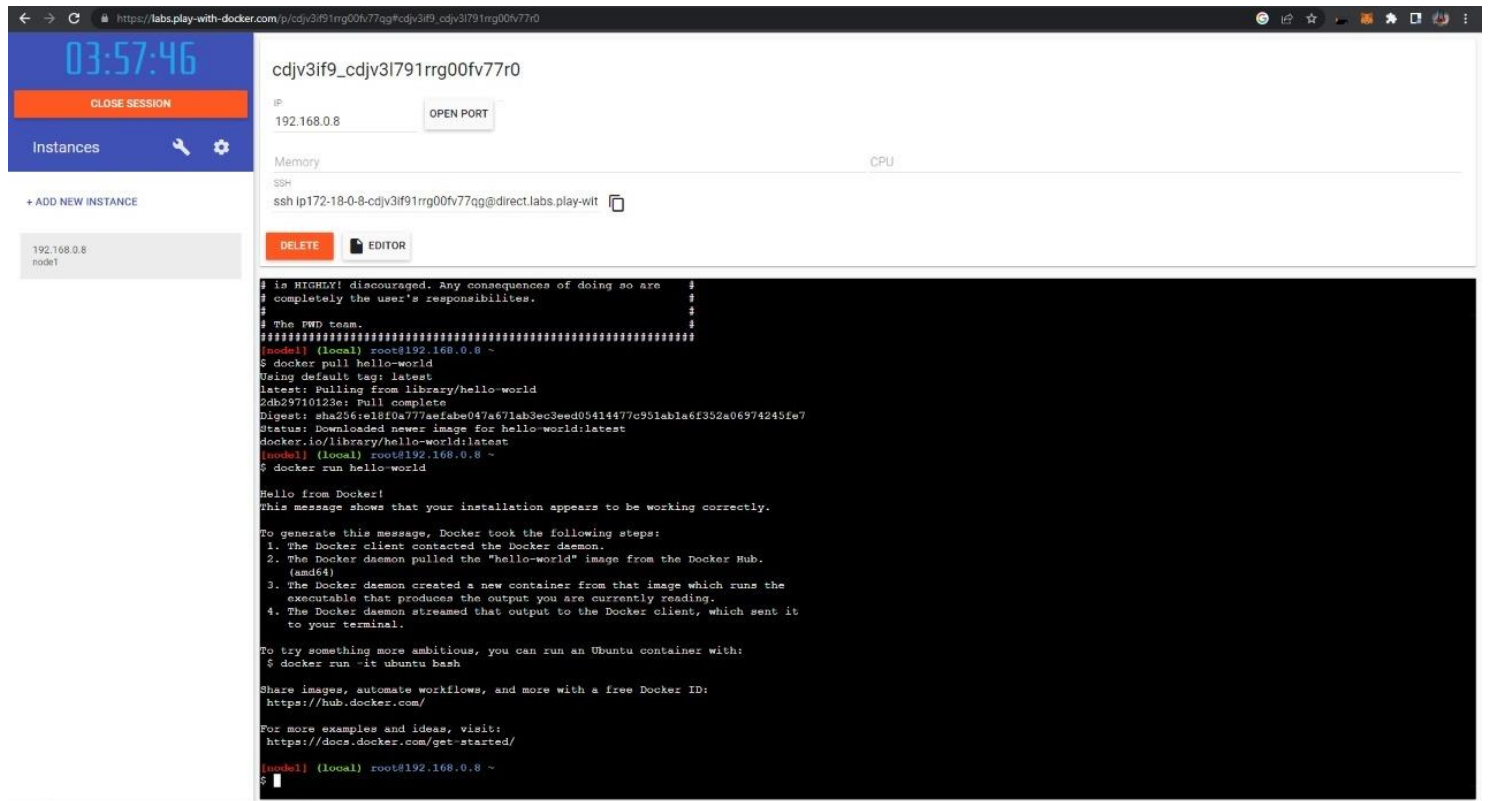


ASSIGNMENT-4

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



The screenshot shows the Docker Playground interface. On the left, there's a sidebar with a clock showing 03:57:46, a 'CLOSE SESSION' button, and a list of instances. The main area displays the instance details for 'cdjv3if9_cdjv3l791rrg00fv77r0' with IP 192.168.0.8. Below this, a terminal window shows the following commands and output:

```
# is HIGHLY! discouraged. Any consequences of doing so are #
# completely the user's responsibilities. #
# The FWD team. #
#####
(node1) (local) root@192.168.0.8 ~
$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:re18f0a777eefabe047a671ab3ec3eed05414477c951ab1a6f352a06974245fe7
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
(node1) (local) root@192.168.0.8 ~
$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

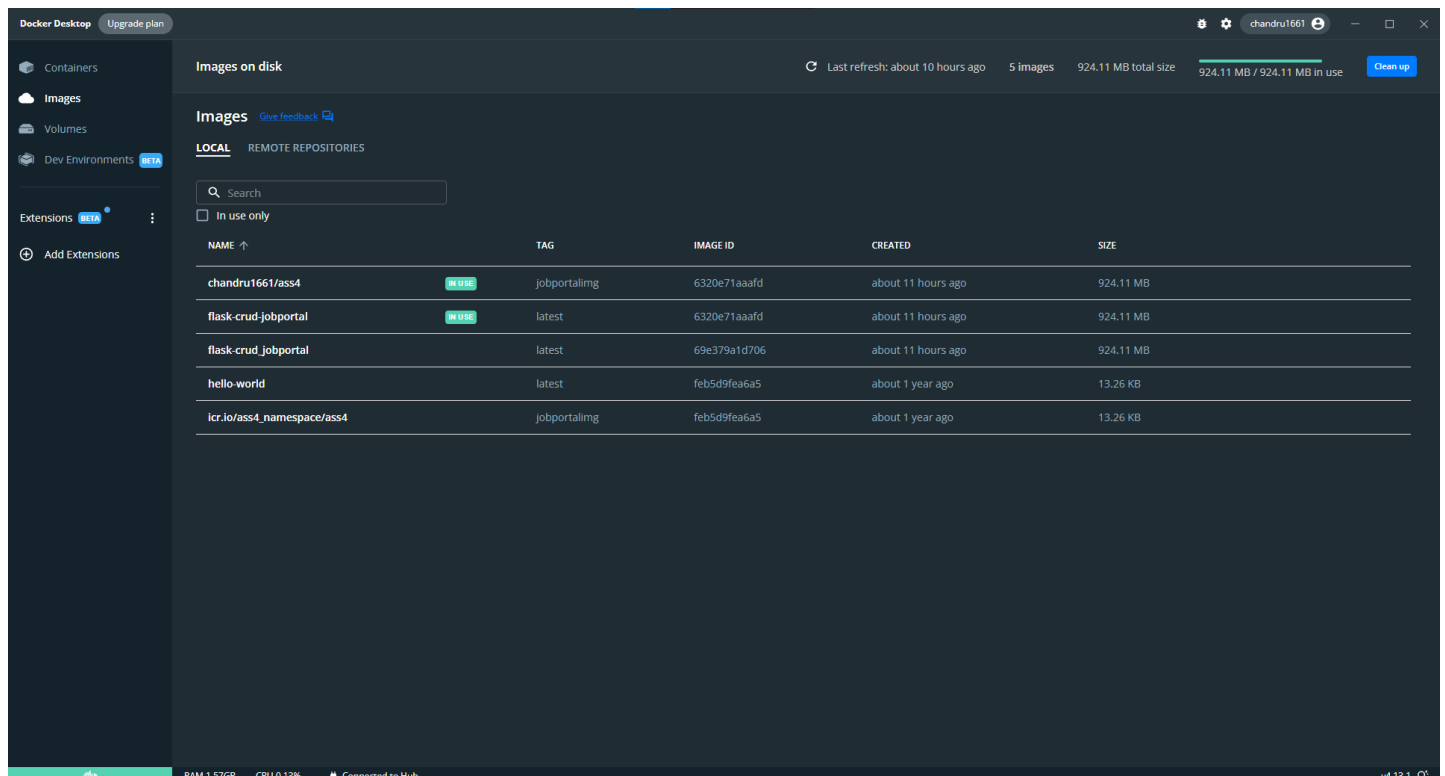
To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
(node1) (local) root@192.168.0.8 ~
$
```

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



The screenshot shows the Docker Desktop interface. The left sidebar contains navigation options: Containers, Images, Volumes, Dev Environments, Extensions, and Add Extensions. The main area is titled 'Images on disk' and shows a list of images. The 'LOCAL' tab is selected, displaying a table of images:

NAME	TAG	IMAGE ID	CREATED	SIZE
chandru1661/ass4	jobportalimg	6320e71aaaf	about 11 hours ago	924.11 MB
flask-crud-jobportal	latest	6320e71aaaf	about 11 hours ago	924.11 MB
flask-crud_jobportal	latest	69e379a1d706	about 11 hours ago	924.11 MB
hello-world	latest	feb5d9fea6a5	about 1 year ago	13.26 KB
icr.io/ass4_namespace/ass4	jobportalimg	feb5d9fea6a5	about 1 year ago	13.26 KB

The bottom status bar shows RAM 1.57GB, CPU 0.13%, and Connected to Hub.

The screenshot shows the Visual Studio Code editor with a Dockerfile open. The Dockerfile contains the following instructions:

```
1 FROM python:3.8
2
3 WORKDIR /app
4 COPY . .
5
6 RUN pip install -r requirements.txt
7
8 ENTRYPOINT ["python"]
9 CMD ["app.py"]
10
```

The Explorer sidebar on the left shows the project structure, including files like form.html, app.py, deployment.yaml, success.html, docker-compose.yaml, Dockerfile, and requirements.txt. The Dockerfile is currently selected and open in the editor.

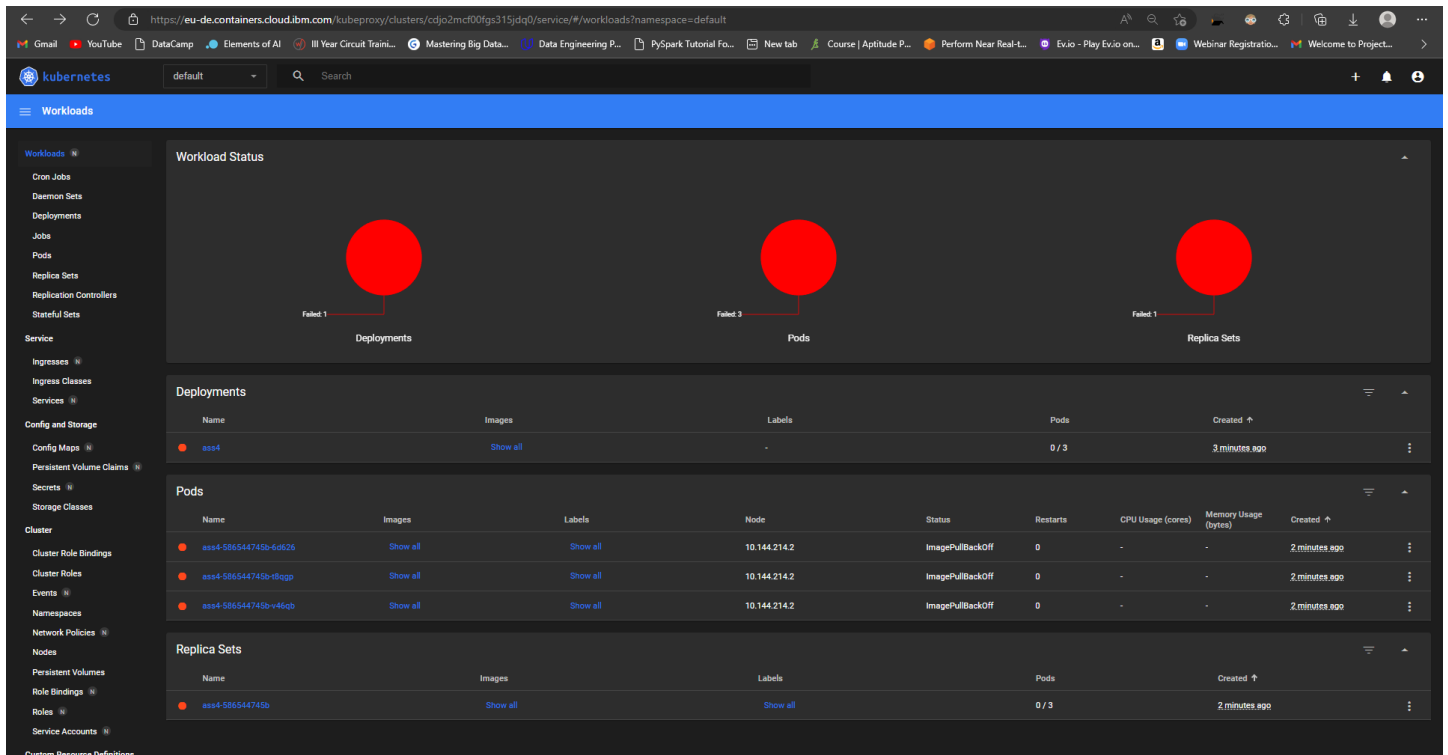
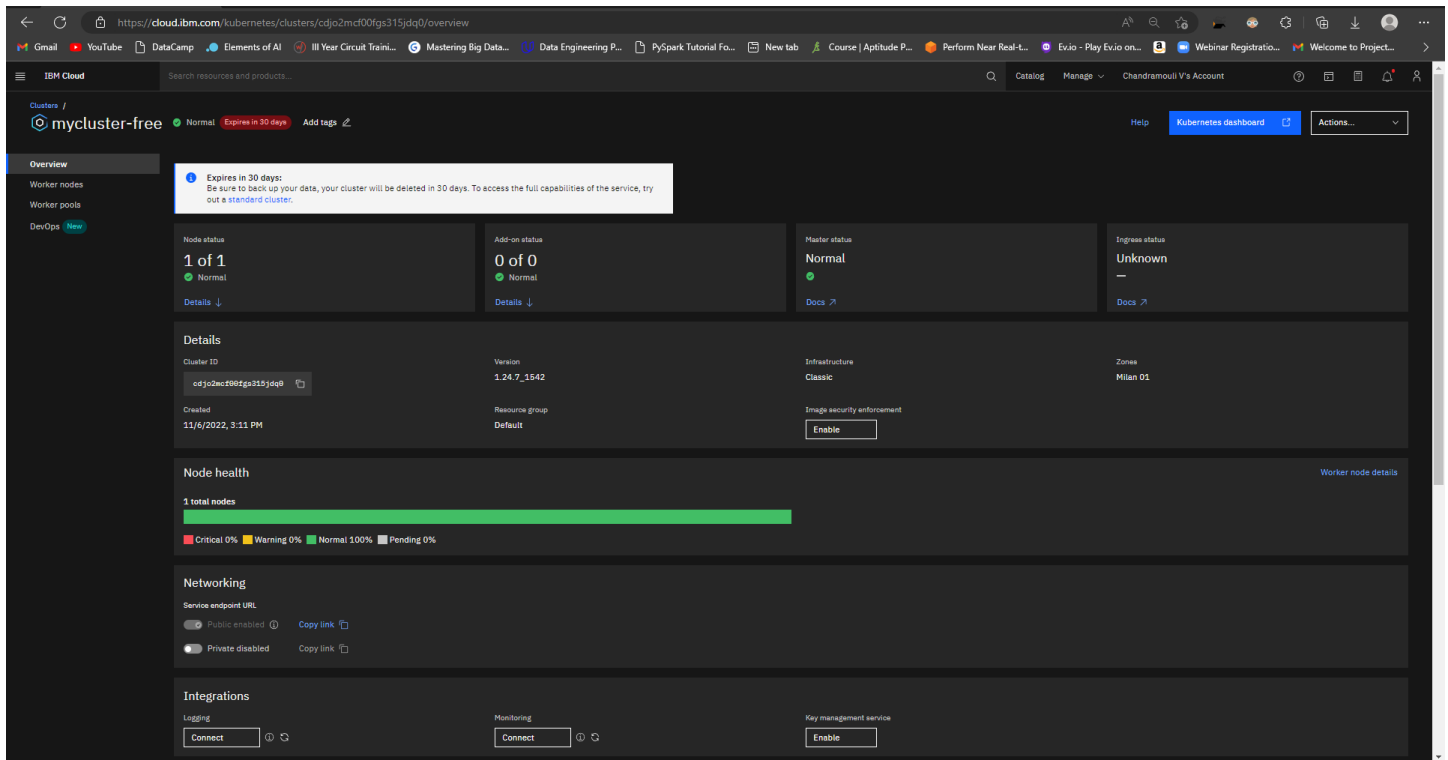
3. Create a IBM container registry and deploy helloworld app or jobportalapp.

The screenshot shows the IBM Cloud Container Registry interface. The 'Namespaces' page is displayed, showing a table of existing namespaces. The 'Global' location is selected. The table lists the 'ass4_namespace' with 1 repository and 1 image. A 'Create' button is visible in the top right corner of the table area.

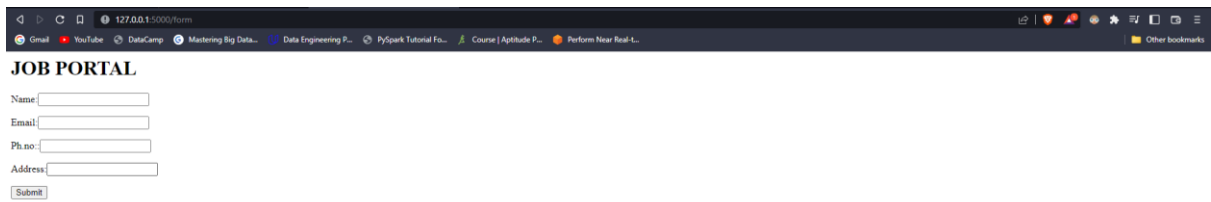
Name	Resource group	Repository count	Image count	Retention policy
ass4_namespace	Default	1	1	Retain all images

Repository	Image count	Last updated
icr.io/ass4_namespace/ass4	1	409 days ago

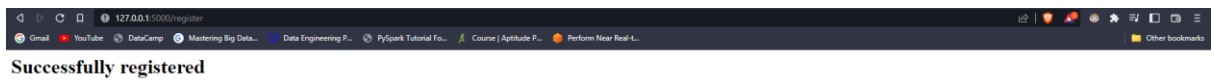
4. Create a Kubernetes cluster in IBM cloud and deploy helloworld image or jobportal image and also expose the same app to run in nodeport.



JOBPORTAL APPLICATION



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/form'. The browser's tab bar includes several tabs: 'Gmail', 'YouTube', 'DataCamp', 'Mastering Big Data...', 'Data Engineering P...', 'PySpark Tutorial Fo...', 'Course | Aptitude P...', and 'Perform Near Real-L...'. The page content features the heading 'JOB PORTAL' followed by a registration form with the following fields: 'Name', 'Email', 'Ph.no', and 'Address'. Each field is represented by a text input box. Below the 'Address' field is a 'Submit' button.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/register'. The browser's tab bar is identical to the previous screenshot. The page content displays the message 'Successfully registered' in a bold, black font.

Deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ass4
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ass4
  template:
    metadata:
      labels:
        app: ass4
    spec:
      containers:
        - name: ass4
          image: icr.io/ass4_namespace/ass4
          imagePullPolicy: Always
          ports:
            - containerPort: 5000
```

```
FROM python:3.8

WORKDIR /app
COPY . .

RUN pip install -r requirements.txt

ENTRYPOINT ["python"]
CMD ["app.py"]
```

Fig: Dockerfile