

Assignment-4

Plasma Donor Application

Team ID : PNT2022TMID08318

Team Leader : KOTHAMASU VENKATA RATNA SAI

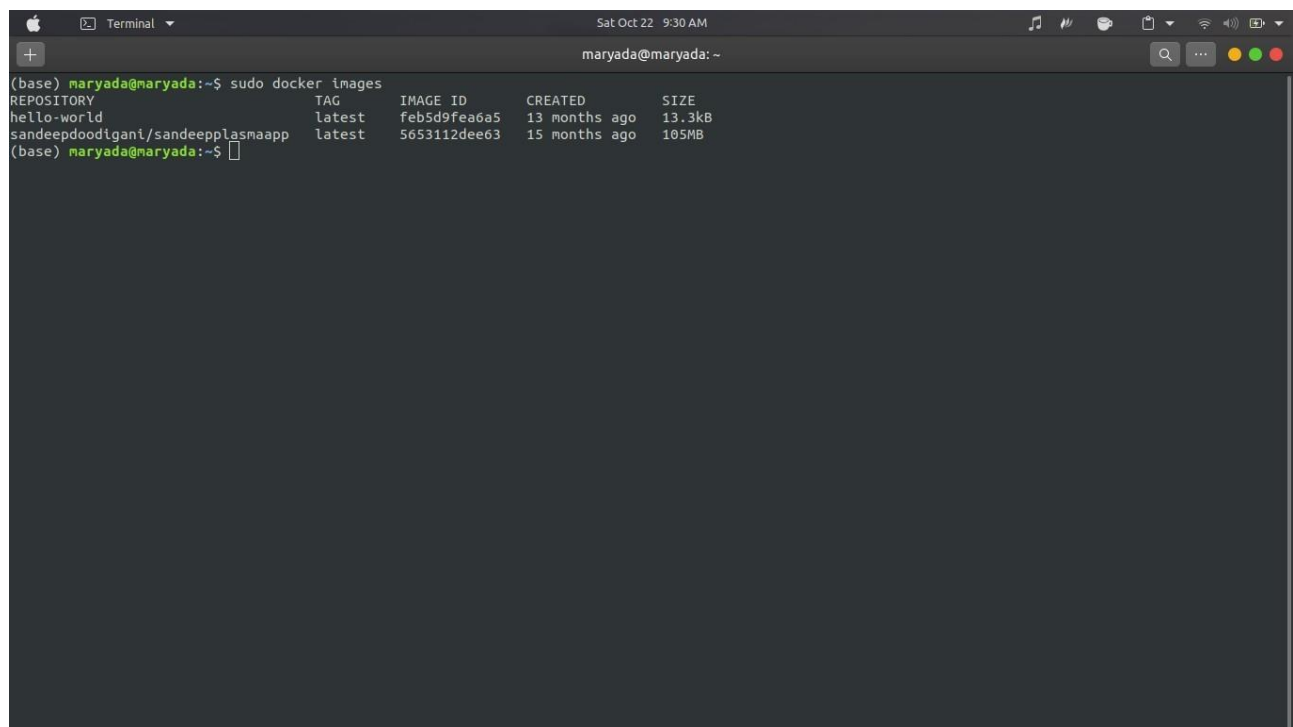
Team member : SAI BABU

Team member : JASTHI MANIKANTA

Team member : PARDHA SAI

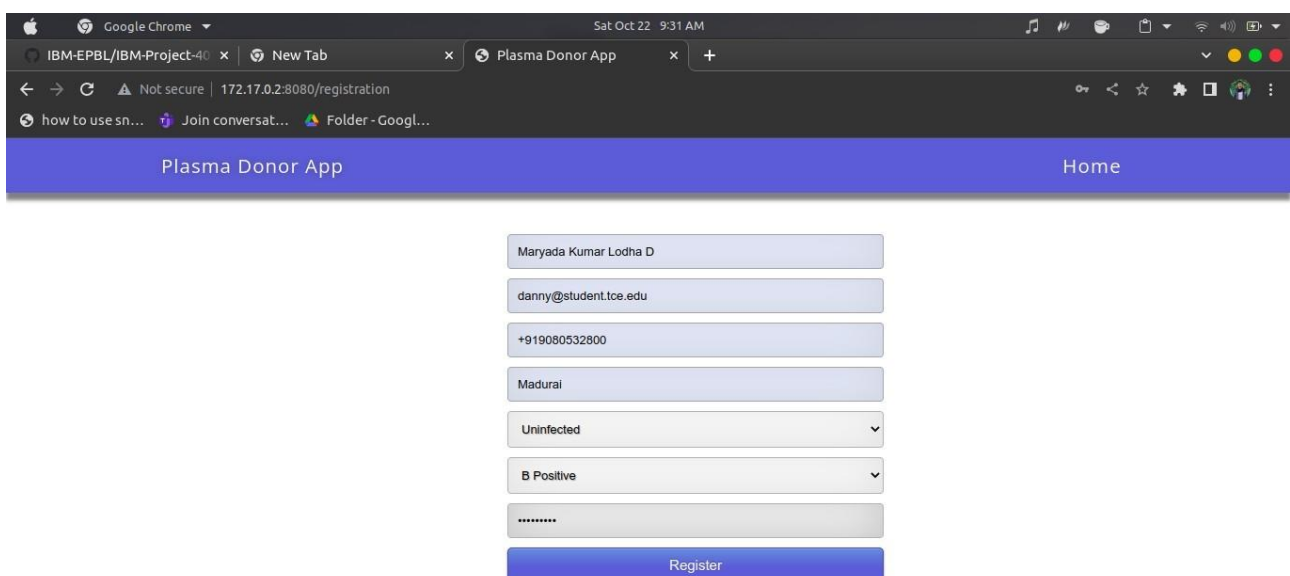
1. Pull an Image from docker hub and run it in docker playground.

Pulled sandeepdoodigani/plasmaapplication and running in docker:

A screenshot of a macOS Terminal window. The title bar shows 'Terminal' and the date 'Sat Oct 22 9:30 AM'. The terminal content shows a user named 'maryada' at a host named 'maryada' running the command 'sudo docker images'. The output is a table with columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE. The table lists two images: 'hello-world' (latest tag, 13 months ago, 13.3kB) and 'sandeepdoodigani/sandeepplasmaapp' (latest tag, 15 months ago, 105MB).

```
(base) maryada@maryada:~$ sudo docker images
REPOSITORY              TAG         IMAGE ID      CREATED       SIZE
hello-world              latest      feb5d9fea6a5 13 months ago 13.3kB
sandeepdoodigani/sandeepplasmaapp latest      5653112dee63 15 months ago 105MB
(base) maryada@maryada:~$
```

```
Terminal
Sat Oct 22 9:31 AM
maryada@maryada: ~
(base) maryada@maryada:~$ sudo docker run -p 8080:8080 sandeepdoodigani/sandeepplasmaapp
* 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://172.17.0.2:8080/ (Press CTRL+C to quit)
```



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

Dockerfile:

FROM python:3.6

WORKDIR /app

ADD . /app

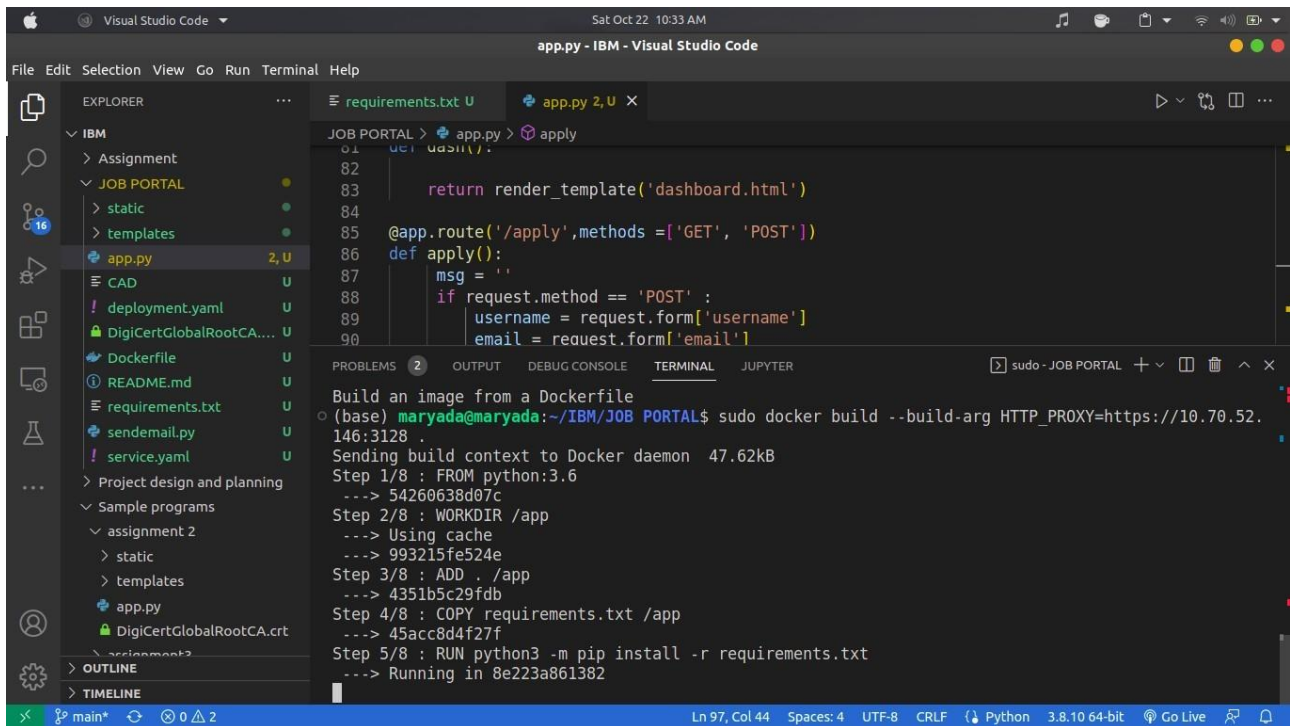
COPY requirements.txt /app

RUN python3 -m pip install -r requirements.txt

RUN python3 -m pip install ibm_db

EXPOSE 5000

CMD ["python","app.py"]

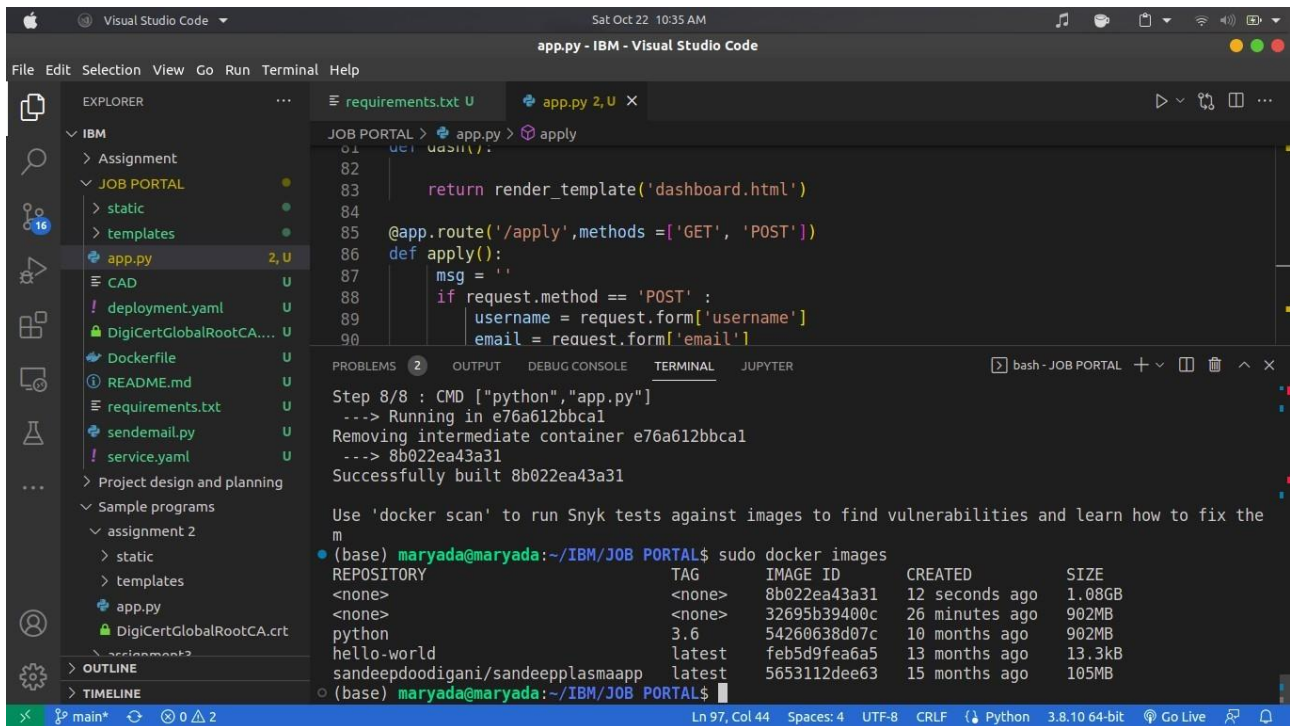


The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure. The main editor shows the `app.py` file with the following code:

```
81 def udsii():
82
83     return render_template('dashboard.html')
84
85 @app.route('/apply', methods=['GET', 'POST'])
86 def apply():
87     msg = ''
88     if request.method == 'POST':
89         username = request.form['username']
90         email = request.form['email']
```

The TERMINAL panel at the bottom shows the output of the `sudo docker build` command:

```
Build an image from a Dockerfile
(base) maryada@maryada:~/IBM/JOB PORTAL$ sudo docker build --build-arg HTTP_PROXY=https://10.70.52.146:3128 .
Sending build context to Docker daemon 47.62kB
Step 1/8 : FROM python:3.6
--> 54260638d07c
Step 2/8 : WORKDIR /app
--> Using cache
--> 993215fe524e
Step 3/8 : ADD . /app
--> 4351b5c29fdb
Step 4/8 : COPY requirements.txt /app
--> 45acc8d4f27f
Step 5/8 : RUN python3 -m pip install -r requirements.txt
--> Running in 8e223a861382
```



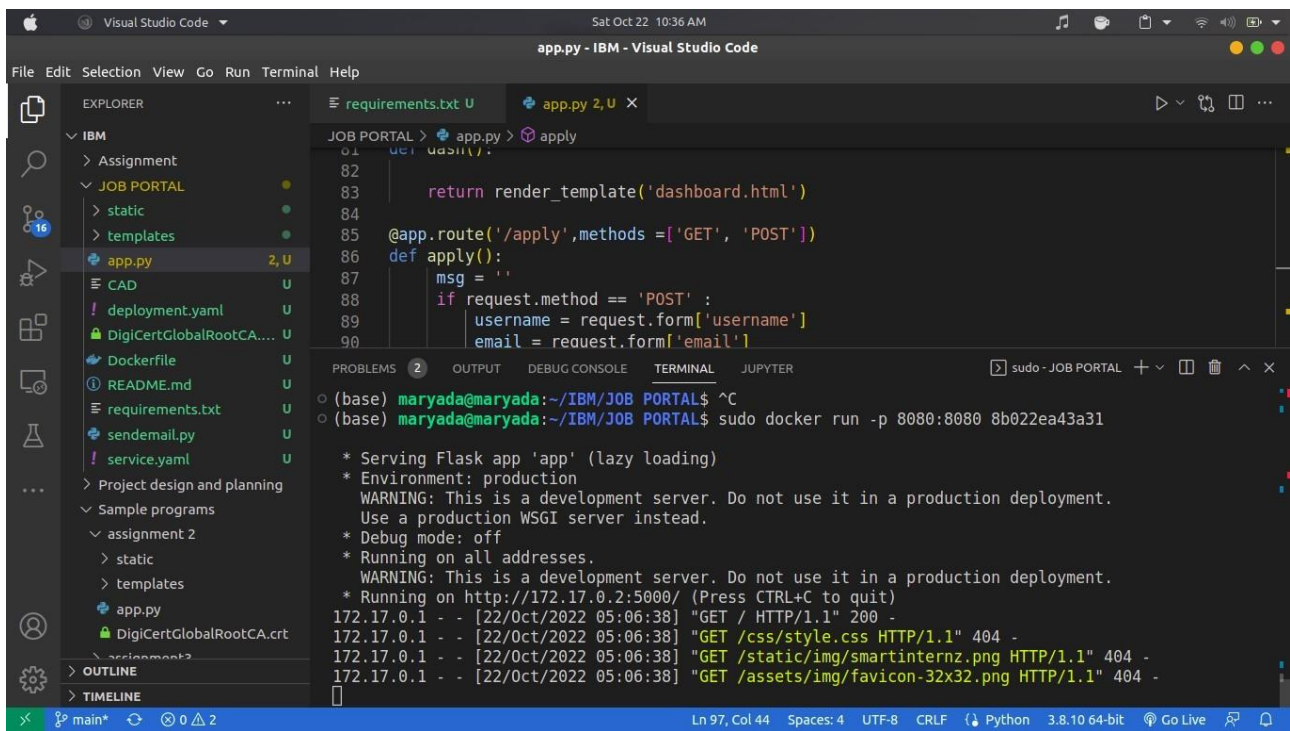
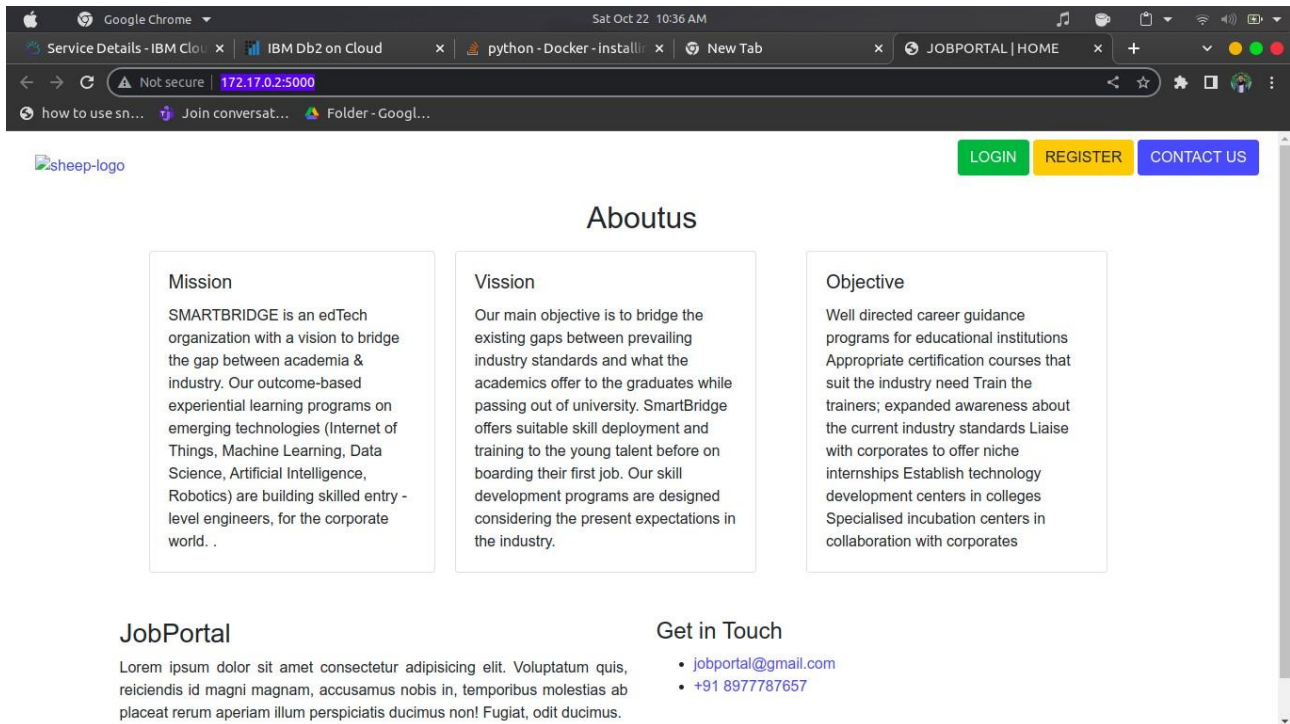
The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure. The main editor shows the `app.py` file with the following code:

```
81 def udsii():
82
83     return render_template('dashboard.html')
84
85 @app.route('/apply', methods=['GET', 'POST'])
86 def apply():
87     msg = ''
88     if request.method == 'POST':
89         username = request.form['username']
90         email = request.form['email']
```

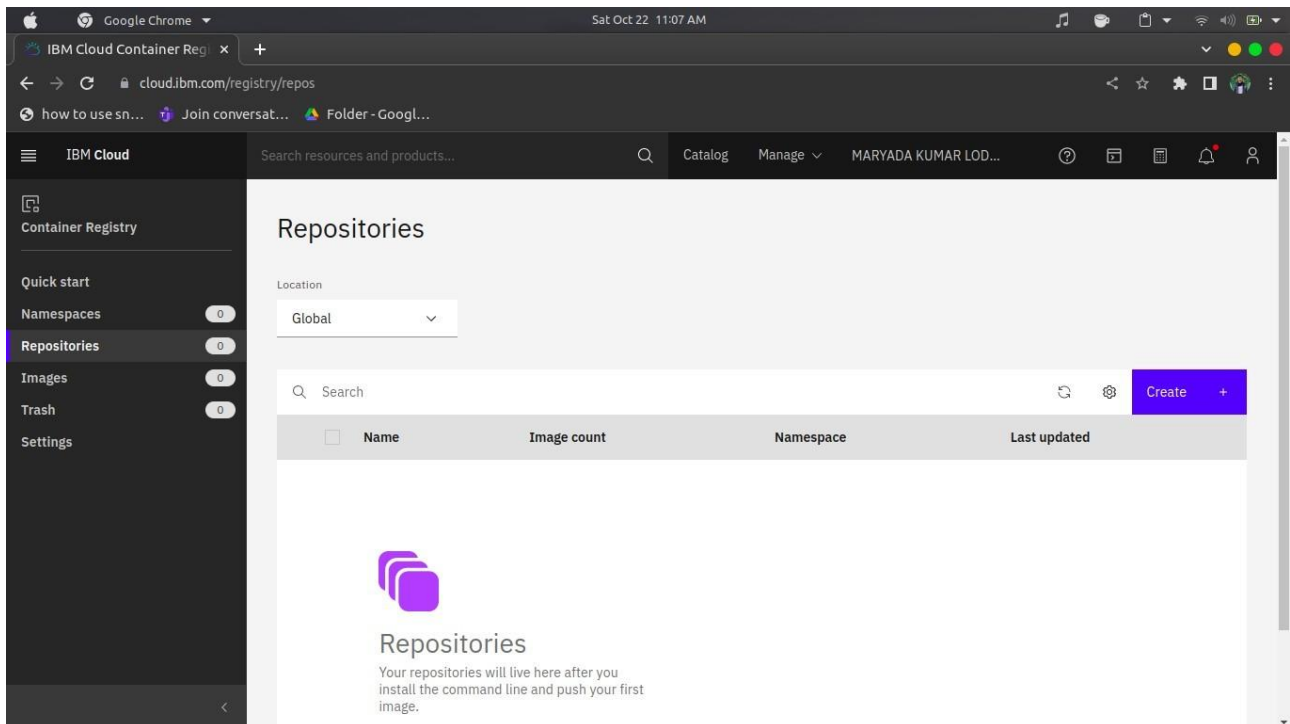
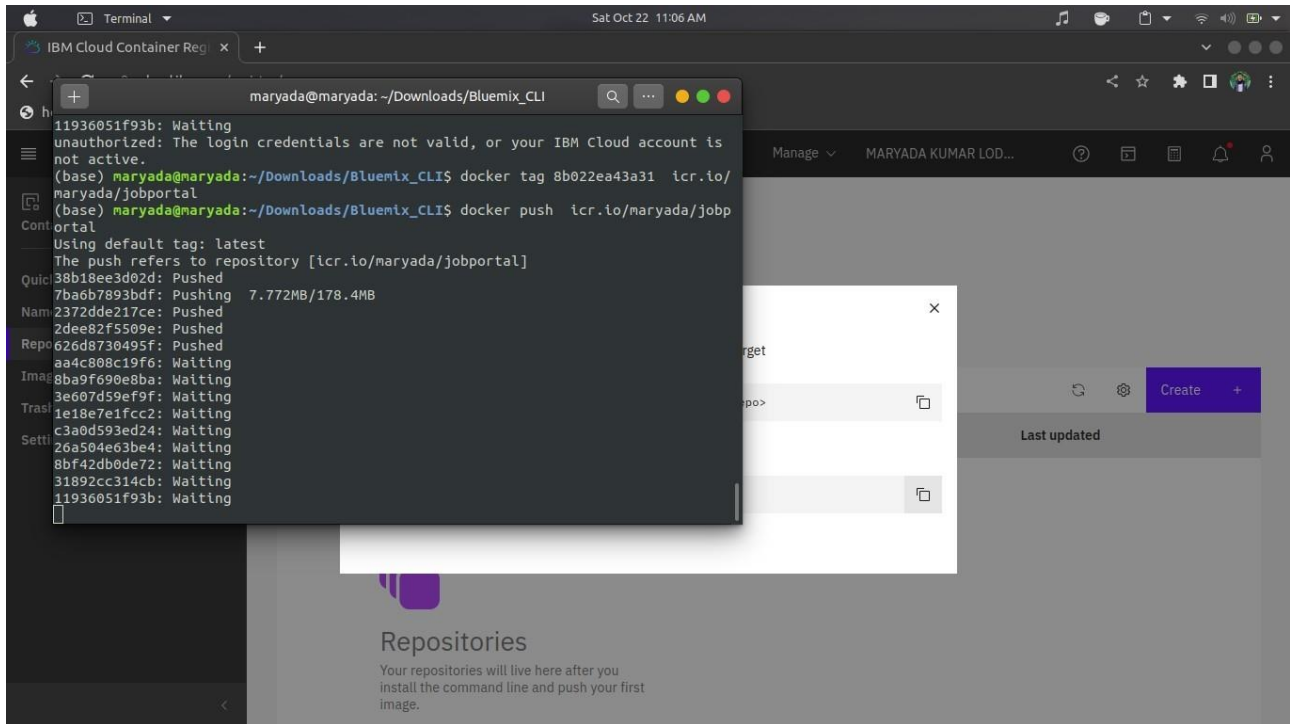
The TERMINAL panel at the bottom shows the output of the `sudo docker images` command:

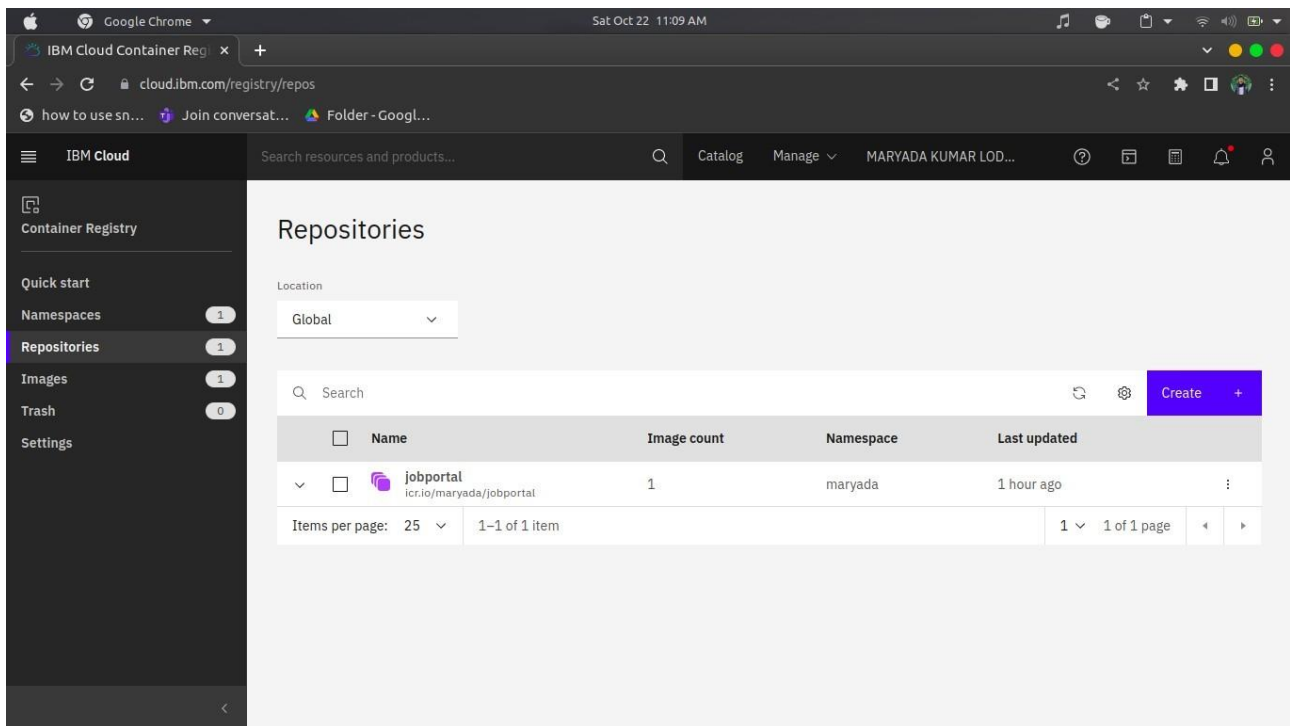
```
Step 8/8 : CMD ["python","app.py"]
--> Running in e76a612bbca1
Removing intermediate container e76a612bbca1
--> 8b022ea43a31
Successfully built 8b022ea43a31

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix the m
(base) maryada@maryada:~/IBM/JOB PORTAL$ sudo docker images
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
<none>              <none>       8b022ea43a31  12 seconds ago 1.08GB
<none>              <none>       32695b39400c  26 minutes ago 902MB
python              3.6         54260638d07c  10 months ago 902MB
hello-world         latest      feb5d9fea6a5  13 months ago 13.3kB
sandeepdoodigani/sandeepplasmaapp latest      5653112dee63  15 months ago 105MB
(base) maryada@maryada:~/IBM/JOB PORTAL$
```

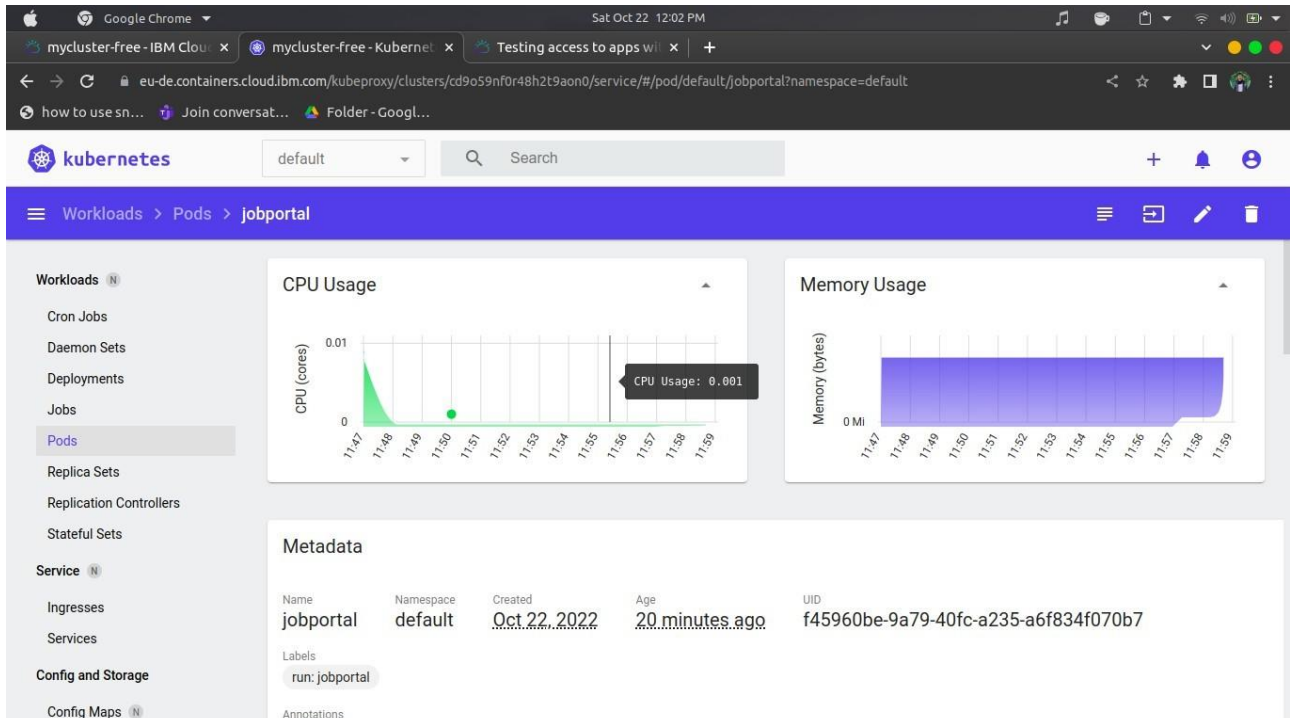


3. Create a IBM container registry and deploy helloworld app or jobportalapp.





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.



Workloads

- Cron Jobs
- Daemon Sets
- Deployments
- Jobs
- Pods**
- Replica Sets
- Replication Controllers
- Stateful Sets

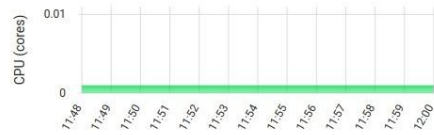
Service

- Ingresses
- Services

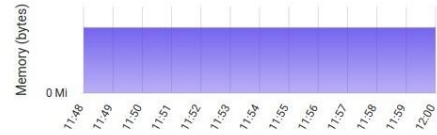
Config and Storage

- Config Maps

CPU Usage



Memory Usage



Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)
jobportal	Show all	Show all	10.144.216.52	Running	0	1.00m
lb4-simple-web-app-deployment	Show all	Show all	10.144.216.52	ImagePullBack 0	-	-