

Assignment 4

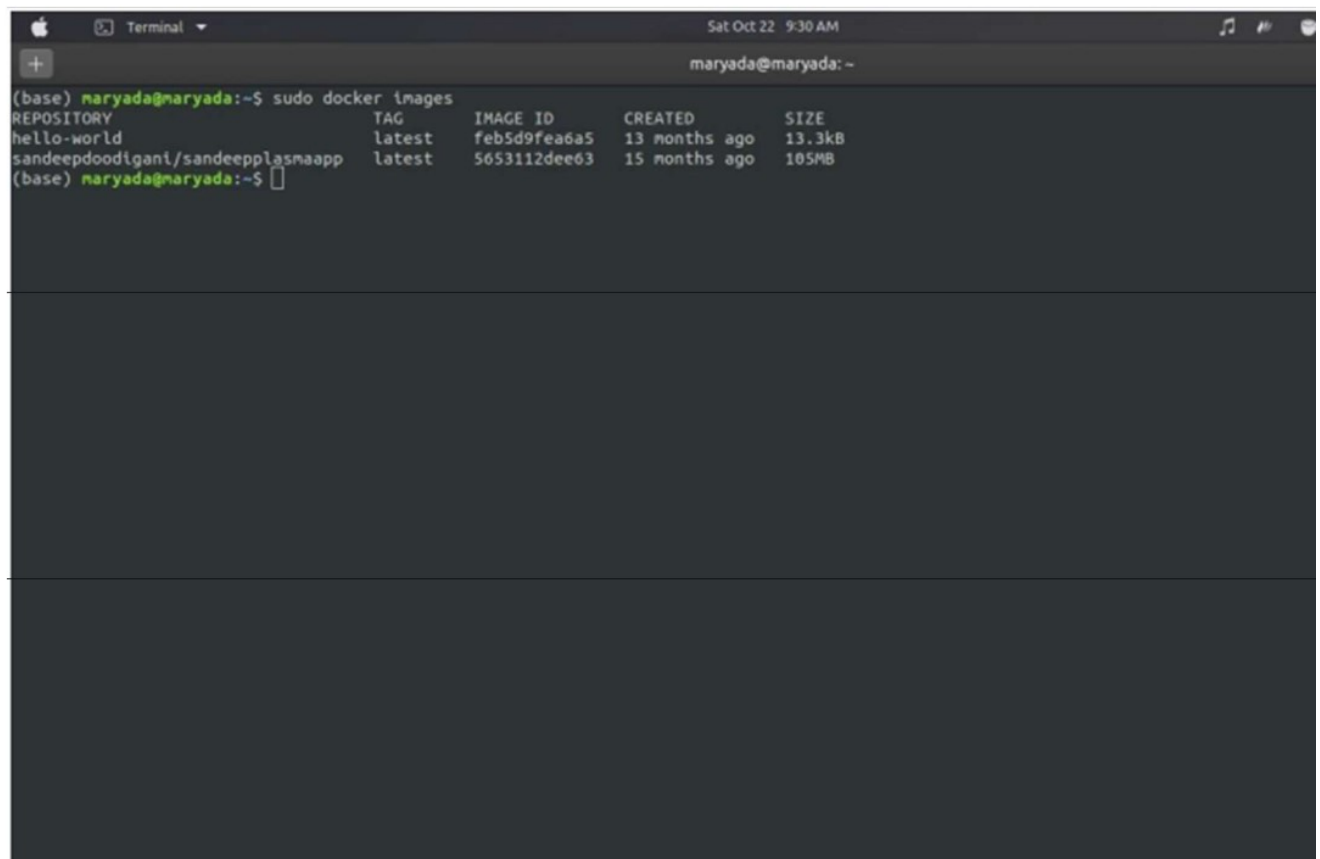
Kubernetes / Docker

Inventory Management For Retailers:

Team ID : PNT2022TMID05614

Name : Hariharasudhan R

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

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 running 'sudo docker images' in a container named 'maryada'. The output is a table with columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE. The table lists two images: 'hello-world' (latest, feb5d9fea6a5, 13 months ago, 13.3kB) and 'sandeepdoodigani/sandeepplasmaapp' (latest, 5653112dee63, 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/sandeepplasnaapp
* 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)
```

Plasma Donor App

127.0.0.1:8080/registration

Plasma Donor App

Registration Form:

- Name: Ranjith
- Email: ranjithmuthusamy44@gmail.com
- Phone: 08220413370
- Address: Namakkal
- Infected: ☐
- B Positive: ☐
- Password:

Register

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

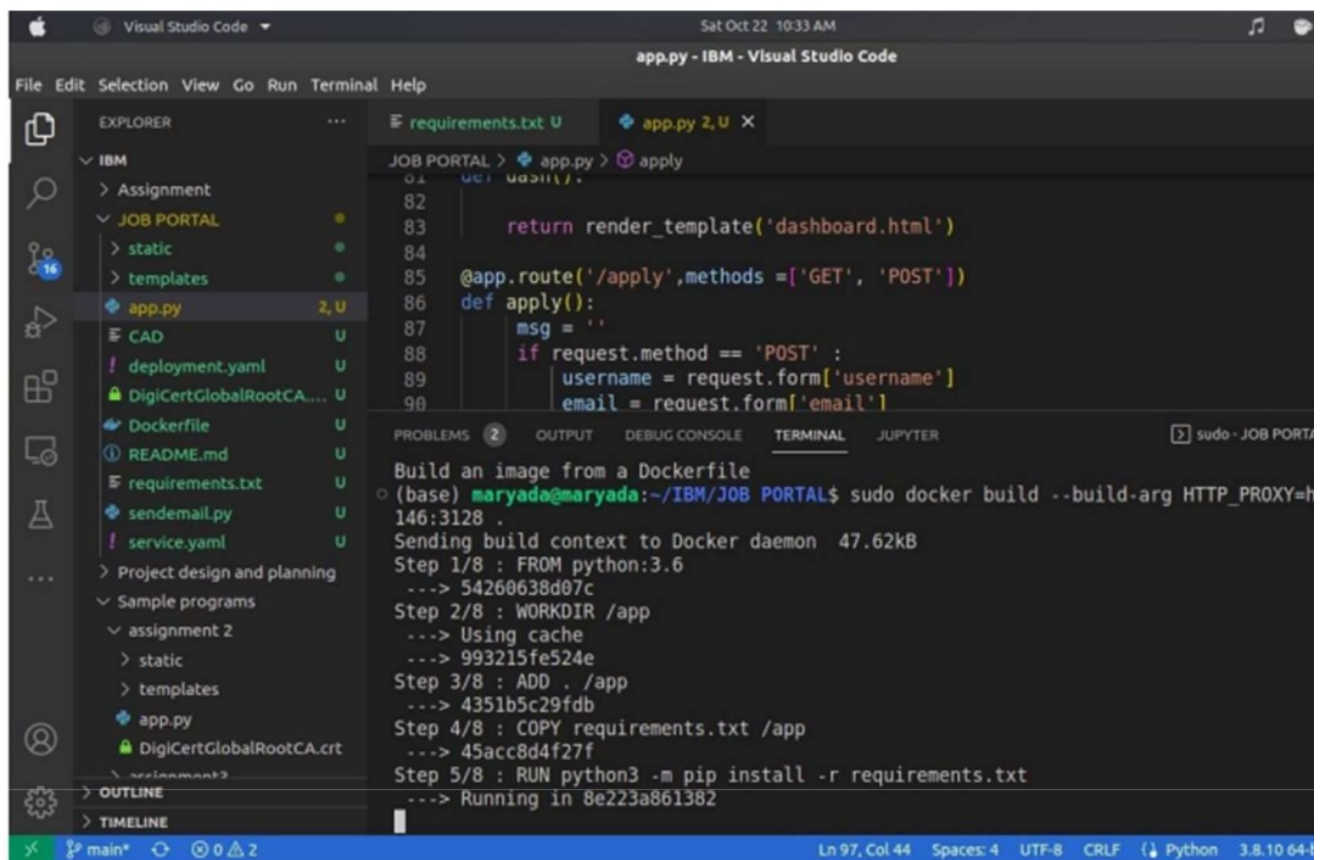
COPY requirements.txt /app

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 following components:

- EXPLORER:** A file tree on the left showing the project structure. The 'JOB PORTAL' directory is expanded, showing files like 'static', 'templates', 'app.py', 'CAD', 'deployment.yaml', 'DigiCertGlobalRootCA...', 'README.md', 'requirements.txt', 'sendemail.py', and 'service.yaml'.
- EDITOR:** The 'app.py' file is open, showing a Flask application with a route for '/apply'.
- TERMINAL:** The terminal at the bottom shows the command 'sudo docker build --build-arg HTTP_PROXY=https://146:3128 .' and the output of the Docker build process, including the steps: FROM python:3.6, WORKDIR /app, ADD . /app, COPY requirements.txt /app, and RUN python3 -m pip install -r requirements.txt.

```
82     return render_template('dashboard.html')
83
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']
```

Build an image from a Dockerfile
(base) maryada@maryada:~/IBM/JOB PORTAL\$ sudo docker build --build-arg HTTP_PROXY=https://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 a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project structure with folders like 'Assignment', 'JOB PORTAL', 'static', 'templates', and files like 'app.py', 'CAD', 'deployment.yaml', 'Dockerfile', 'README.md', 'requirements.txt', 'sendemail.py', and 'service.yaml'. The code editor shows the 'app.py' file with the following code:

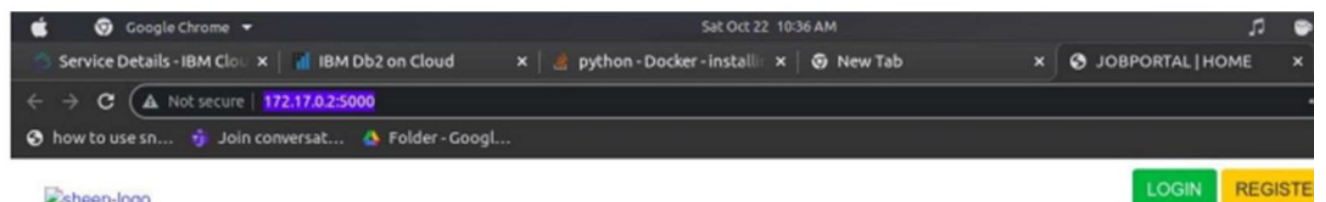
```

82 def apply():
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 shows the command 'python app.py' being executed, which results in a successful build of the Docker image '8b022ea43a31'. The terminal also displays the output of 'docker scan' command, which shows a list of images and their vulnerabilities.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
<none>	<none>	8b022ea43a31	12 seconds ago	1.080
<none>	<none>	32695b39400c	26 minutes ago	902MB
python	3.6	54260638d07c	10 months ago	902MB
hello-world	latest	feb5d9fea6a5	13 months ago	13.3MB
sandeepdoodigani/sandeepplasmaapp	latest	5653112dee63	15 months ago	105MB



Aboutus

Mission

SMARTBRIDGE is an edTech organization with a vision to bridge the gap between academia & industry. Our outcome-based experiential learning programs on emerging technologies (Internet of Things, Machine Learning, Data Science, Artificial Intelligence, Robotics) are building skilled entry-level engineers, for the corporate world.

Vission

Our main objective is to bridge the existing gaps between prevailing industry standards and what the academics offer to the graduates while passing out of university. SmartBridge offers suitable skill deployment and training to the young talent before on boarding their first job. Our skill development programs are designed considering the present expectations in the industry.

Objective

Well directed career guidance programs for educational institutions
Appropriate certification courses that suit the industry need
Train the trainers; expanded awareness about the current industry standards
Liaise with corporates to offer niche internships
Establish technology development centers in colleges
Specialised incubation centers in collaboration with corporates

JobPortal

Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatum quis, reiciendis id magni magnam, accusamus nobis in, temporibus molestias ab

Get in Touch

- jobportal@gmail.com
- +91 8977787657

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** A file tree showing the project structure. The 'JOB PORTAL' folder is expanded, showing files like 'app.py', 'requirements.txt', 'Dockerfile', and 'service.yaml'.
- EDITOR:** The 'app.py' file is open, showing a Flask application with a route for '/apply'.
- TERMINAL:** The terminal shows the command 'sudo docker run -p 8080:8080 8b022ea43a31' and the output of the application running on http://172.17.0.2:5000/.

```

JOB PORTAL > app.py > apply
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']

```

```

(base) maryada@maryada:~/IBM/JOB PORTAL$ ^C
(base) maryada@maryada:~/IBM/JOB PORTAL$ sudo docker run -p 8080:8080 8b022ea43a31
* 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)
172.17.0.1 - - [22/Oct/2022 05:06:38] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [22/Oct/2022 05:06:38] "GET /css/style.css HTTP/1.1" 404 -
172.17.0.1 - - [22/Oct/2022 05:06:38] "GET /static/img/smartinternz.png HTTP/1.1" 404 -
172.17.0.1 - - [22/Oct/2022 05:06:38] "GET /assets/img/favicon-32x32.png HTTP/1.1" 404 -

```

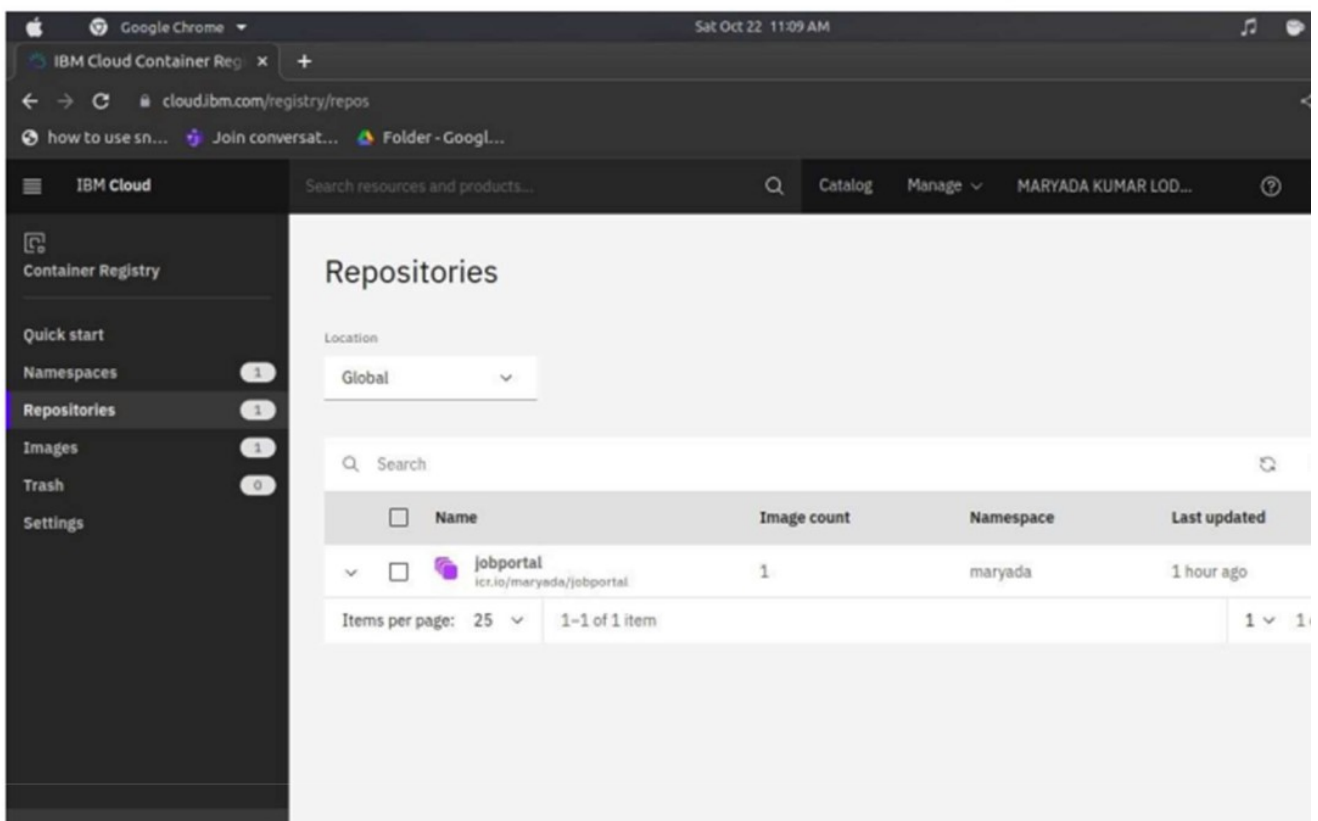
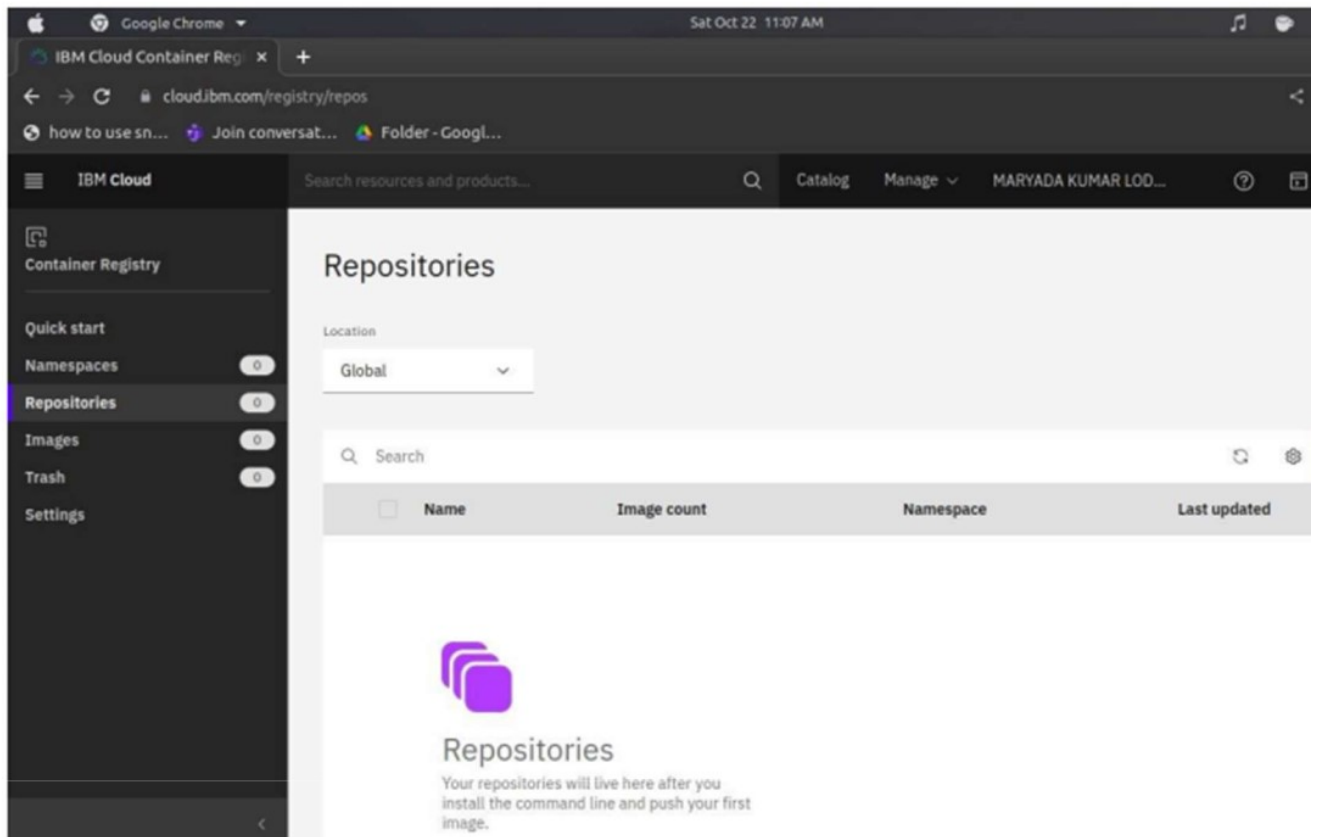
3. Create a IBM container registry and deploy helloworld app or jobportalapp.

The screenshot shows a terminal window with the following commands and output:

```

maryada@maryada:~/Downloads/Bluemix_CLI$ docker tag 8b022ea43a31 icr.io/maryada/jobportal
maryada@maryada:~/Downloads/Bluemix_CLI$ docker push icr.io/maryada/jobportal
Using default tag: latest
The push refers to repository [icr.io/maryada/jobportal]
Pushing 7.772MB/178.4MB
11936051f93b: Waiting
38b18ee3d02d: Pushed
7ba6b7893bdf: Pushed
2372dde217ce: Pushed
2dee82f5509e: Pushed
626d8730495f: Pushed
aa4c808c19f6: Waiting
8ba9f690e8ba: Waiting
3e607d59ef9f: Waiting
1e18e7e1fcc2: Waiting
c3a0d593ed24: Waiting
26a504e63be4: Waiting
8bf42db0de72: Waiting
31892cc314cb: Waiting
11936051f93b: Waiting

```

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.

