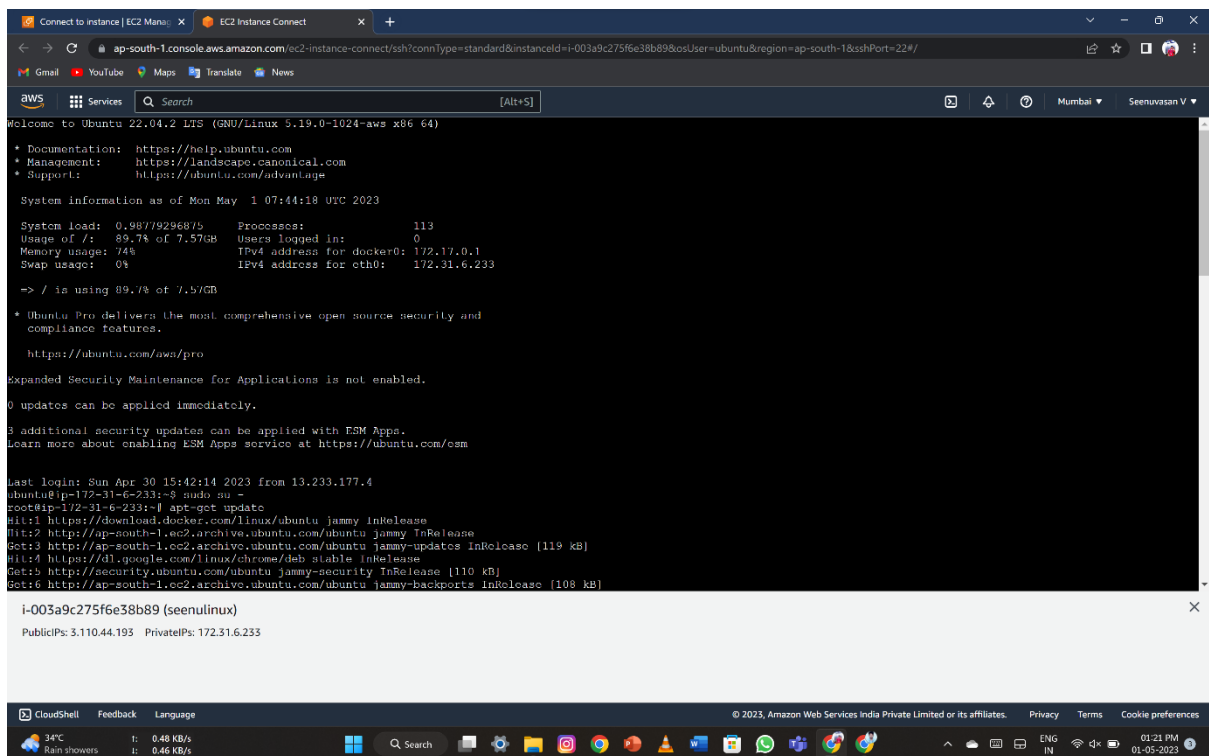


ASSIGNMENT 6 – CONTAINER CONCEPT

To design and develop an application with multi-container architecture using Docker Compose, I will follow the following steps:

1. Create a folder for the application and navigate to the folder using the terminal.

```
mkdir student-vaccination-app  
cd student-vaccination-app
```



```
Connect to instance | EC2 Mania: x EC2 Instance Connect x +  
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-003a9c275f6e38b89&osUser=ubuntu&region=ap-south-1&sshPort=22#/  
Gmail YouTube Maps Translate News  
AWS Services Search [Alt+S]  
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1024-aws x86_64)  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/advantage  
System information as of Mon May 1 07:44:18 UTC 2023  
System load: 0.98/792968/5 Processes: 113  
Usage of /: 89.7% of 7.57GB Users logged in: 0  
Memory usage: 74% IPv4 address for docker0: 172.17.0.1  
Swap usage: 0% IPv4 address for eth0: 172.31.6.233  
-> / is using 89.7% of 7.57GB  
* Ubuntu Pro delivers the most comprehensive open source security and compliance features.  
https://ubuntu.com/pro  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
3 additional security updates can be applied with ESM Apps.  
Learn more about enabling ESM Apps service at https://ubuntu.com/esm  
Last login: Sun Apr 30 15:42:14 2023 from 13.233.177.4  
ubuntu@ip-172-31-6-233:~$ sudo su -  
root@ip-172-31-6-233:~# apt-get update  
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease  
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]  
Hit:4 https://dl.google.com/linux/chrome/deb stable InRelease  
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]  
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]  
i-003a9c275f6e38b89 (seenulinux)  
PublicIPs: 3.110.44.193 PrivateIPs: 172.31.6.233  
CloudShell Feedback Language  
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences  
34°C Rain showers i: 0.48 KB/s  
i: 0.46 KB/s  
01:21 PM 01-05-2023
```

2. Create a Dockerfile for the Python Flask web application.

```
FROM python:3.9-slim-buster

WORKDIR /app

COPY requirements.txt requirements.txt

RUN pip3 install -r requirements.txt

COPY app app

CMD ["python3", "-m", "flask", "run", "--host=0.0.0.0"]

EXPOSE 5000
```

3. Create a Dockerfile for the PostgreSQL database.

```
FROM postgres:12

ENV POSTGRES_USER=postgres
ENV POSTGRES_PASSWORD=password
ENV POSTGRES_DB=students

COPY create_table.sql /docker-entrypoint-initdb.d/
COPY insert_rows.sql /docker-entrypoint-initdb.d/
```

```

building dependency tree... Done
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
 containerd.io : Conflicts: containerd
 Conflicts: runc
E: Error, pkgProblemResolver::Resolve generated breaks, this may be caused by held packages.
root@ip-172-31-6-233:~# sudo systemctl start docker
root@ip-172-31-6-233:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-05-01 07:43:43 UTC; 7min ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 673 (dockerd)
      Tasks: 9
     Memory: 25.1M
        CPU: 584ms
    CGroup: /system.slice/docker.service
            └─673 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

May 01 07:43:40 ip-172-31-6-233 dockerd[673]: time="2023-05-01T07:43:40.213906399Z" level=info msg="Starting up"
May 01 07:43:40 ip-172-31-6-233 dockerd[673]: time="2023-05-01T07:43:40.224072023Z" level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, no using resolv.
May 01 07:43:40 ip-172-31-6-233 dockerd[673]: time="2023-05-01T07:43:40.938090127Z" level=info msg="Loading containers: start."
May 01 07:43:42 ip-172-31-6-233 dockerd[673]: time="2023-05-01T07:43:42.644574086Z" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daem
May 01 07:43:42 ip-172-31-6-233 dockerd[673]: time="2023-05-01T07:43:42.891297524Z" level=info msg="Loading containers: done."
May 01 07:43:43 ip-172-31-6-233 dockerd[673]: time="2023-05-01T07:43:43.269424329Z" level=info msg="Docker daemon" commit=94d3ad6 graphdriver=overlay2 version=23.0.5
May 01 07:43:43 ip-172-31-6-233 dockerd[673]: time="2023-05-01T07:43:43.273974978Z" level=info msg="Daemon has completed initialization"
May 01 07:43:43 ip-172-31-6-233 systemd[1]: Started Docker Application Container Engine.
May 01 07:43:43 ip-172-31-6-233 dockerd[673]: time="2023-05-01T07:43:43.594616454Z" level=info msg="API listen on /run/docker.sock"

lines 1-22/22 (150)

```

i-003a9c275f6e38b89 (seenulinux)

PublicIPs: 3.110.44.193 PrivateIPs: 172.31.6.233

I will create a database named "students" and copy two SQL files into the initialization directory, which will create a table and insert some rows into it when the container is created.

4. Create a docker-compose.yml file that will define the services, volumes, and networks for our application.

version: '3.9'

services:

web:

build: .

ports:

- "5000:5000"

db:

build:

context: .

dockerfile: Dockerfile.db

environment:

- POSTGRES_USER=postgres

- POSTGRES_PASSWORD=password

- POSTGRES_DB=students

The screenshot shows a terminal window connected to an Ubuntu 22.04.2 LTS instance via AWS EC2 Instance Connect. The terminal displays the following content:

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1024-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Mon May 1 07:52:36 UTC 2023

System load:  0.02050/0/112      Processes:    111
Usage of /:   89.7% of 7.57GB     Users logged in: 0
Memory usage: 74%               TPv4 address for docker0: 172.17.0.1
Swap usage:   0%                 IPv4 address for eth0: 172.31.6.233

=> / is using 89.7% of 7.5/GB

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

3 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Mon May 1 07:44:20 2023 from 13.233.177.3
ubuntu@ip-172-31-6-233:~$ sudo usermod -o docker.$[USPR]
ubuntu@ip-172-31-6-233:~$ sudo systemctl restart docker
ubuntu@ip-172-31-6-233:~$ mkdir student-registration-app
cd student-registration-app
ubuntu@ip-172-31-6-233:~/student-registration-app$ nano dockerfile
ubuntu@ip-172-31-6-233:~/student-registration-app$ app.py
app.py: command not found
ubuntu@ip-172-31-6-233:~/student-registration-app$ nano dockerfile
```

Below the terminal output, a box displays the instance ID and IP addresses:

```
i-003a9c275f6e38b89 (seenulinux)
PublicIPs: 3.110.44.193 PrivateIPs: 172.31.6.233
```

The bottom of the image shows the AWS CloudShell interface with a taskbar containing various application icons and system status information like temperature and time.

The docker-compose.yaml file defines two services, "web" and "db". The "web" service is built from the Dockerfile in the current directory, maps the container port 5000 to the host port 5000, and mounts the ./app directory as a volume.

5. Create a requirements.txt file that lists the Flask package.

The "db" service is built from the Dockerfile.db file, sets some environment variables, and mounts a volume for persisting the database data.

Both services are connected to the "student-net" network.

I will create a database named "students" and copy two SQL files into the initialization directory, which will create a table and insert some rows into it when the container is created.

```
Flask==2.0.1
```

The screenshot shows a terminal window connected to an Ubuntu 22.04.2 LTS instance via AWS EC2 Instance Connect. The terminal output includes system information, a directory creation for 'student-registration-app', and the installation of Docker. The user then attempts to restart Docker and edit a 'dockerfile' using 'nano', but the 'app.py' command is not found. The terminal also shows the instance ID 'i-003a9c275f6e38b89' and its public/private IP addresses.

```
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1024-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon May 1 07:52:36 UTC 2023

System load: 0.0205078125   Processes:      111
Usage of /:  89.7% of 7.57GB   Users logged in: 0
Memory usage: 74%          IPv4 address for docker0: 172.17.0.1
Swap usage: 0%              IPv4 address for eth0: 172.31.6.233

=> / is using 89.7% of 7.57GB

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.
   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
3 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Mon May 1 07:44:20 2023 from 13.233.177.3
ubuntu@ip-172-31-6-233:~$ sudo usermod -s0 docker $(USER)
ubuntu@ip-172-31-6-233:~$ sudo systemctl restart docker
ubuntu@ip-172-31-6-233:~$ mkdir student-registration-app
cd student-registration-app
ubuntu@ip-172-31-6-233:~/student-registration-app$ nano dockerfile
ubuntu@ip-172-31-6-233:~/student-registration-app$ app.py
app.py: command not found
ubuntu@ip-172-31-6-233:~/student-registration-app$ nano dockerfile
```

i-003a9c275f6e38b89 (seenulinux)
PublicIPs: 3.110.44.193 PrivateIPs: 172.31.6.233

6. Create a create_table.sql file that will create a table in the "students" database.

```
CREATE TABLE student_vaccination (
  registration_number INTEGER PRIMARY KEY,
  name TEXT,
  vaccinated BOOLEAN
);
```

```

app.py: command not found
ubuntu@ip-172-31-6-233:~/student-registration-app$ nano dockerfile
ubuntu@ip-172-31-6-233:~/student-registration-app$ cd /path/to/directory
-bash: cd: /path/to/directory: No such file or directory
ubuntu@ip-172-31-6-233:~/student-registration-app$ cd ~/student-registration-app
ubuntu@ip-172-31-6-233:~/student-registration-app$ cd /path/to/directory
-bash: cd: /path/to/directory: No such file or directory
ubuntu@ip-172-31-6-233:~/student-registration-app$ nano docker-compose.yml
ubuntu@ip-172-31-6-233:~/student-registration-app$ Flask==1.1.2
mysql-connector-python==8.0.24
mysql-connector-python==8.0.24: command not found
ubuntu@ip-172-31-6-233:~/student-registration-app$ pip install mysql-connector-python==8.0.24
Defaulting to user installation because normal site-packages is not writeable
Collecting mysql-connector-python==8.0.24
  Downloading mysql-connector-python-8.0.24-py2.py3-none-any.whl (319 kB)
    319.0/319.0 KB 15.5 MB/s eta 0:00:00
Collecting protobuf==3.0.0
  Downloading protobuf-4.22.3-cp37-abi3-manylinux2014_x86_64.whl (302 kB)
    302.0/302.0 KB 7.7 MB/s eta 0:00:00
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.0.24 protobuf-4.22.3
ubuntu@ip-172-31-6-233:~/student-registration-app$ pip install Flask==1.1.2
Defaulting to user installation because normal site-packages is not writeable
Collecting Flask==1.1.2
  Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
    94.6/94.6 KB 2.2 MB/s eta 0:00:00
Requirement already satisfied: Jinja2>=2.10.1 in /usr/lib/python3/dist-packages (from Flask==1.1.2) (3.0.3)
Requirement already satisfied: click>=5.1 in /usr/lib/python3/dist-packages (from Flask==1.1.2) (8.0.3)
Requirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python3.10/dist-packages (from Flask==1.1.2) (2.3.2)
Requirement already satisfied: itsdangerous>=0.24 in /usr/local/lib/python3.10/dist-packages (from Flask==1.1.2) (2.1.2)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from Werkzeug>=0.15->Flask==1.1.2) (2.1.2)
Installing collected packages: Flask
  WARNING: The script flask is installed in '/home/ubuntu/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Flask-1.1.2
ubuntu@ip-172-31-6-233:~/student-registration-app$ Flask==1.1.2
mysql-connector-python==8.0.24

i-003a9c275f6e38b89 (seenulinux)
PublicIPs: 3.110.44.193 PrivateIPs: 172.31.6.233
  
```

7. Create a Python Flask web application in the "app" directory that will prompt the user for a registration number, query the "student_vaccination" table in the database, and display the vaccination status of the student.

```
from flask import Flask, render_template, request

import psycopg2

app = Flask(__name__)

def index():

    if request.method == 'POST':

        registration_number = request.form['registration_number']

        connection = psycopg2.connect(

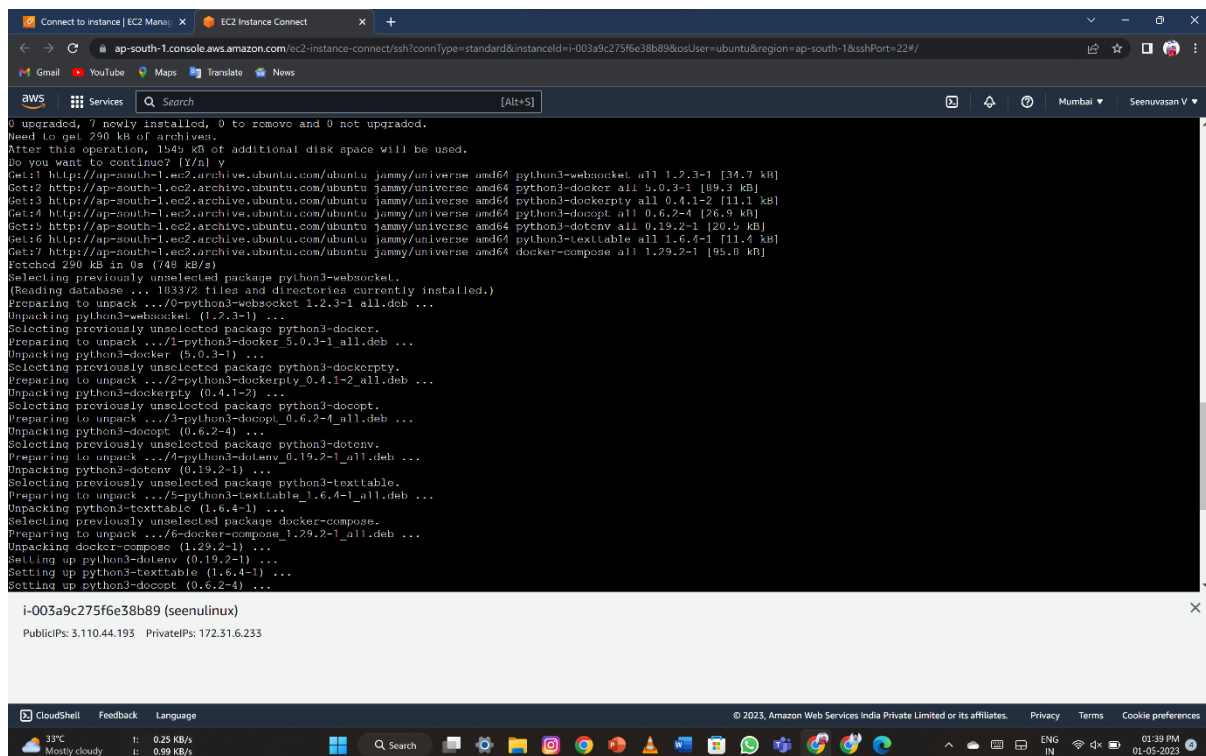
            host='db',

            user='postgres',

            password='password',

            dbname='students'

        )
```



8. Complete the Python Flask web application by adding a database query and returning the vaccination status of the student in the HTML template.

```
from flask import Flask, render_template, request
import psycopg2

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        registration_number = request.form['registration_number']
        connection = psycopg2.connect(
            host='db',
            user='postgres',
            password='password',
            dbname='students'
        )
        cursor = connection.cursor()
        cursor.execute(
            f"SELECT vaccinated FROM student_vaccination WHERE registration_number = {registration_number}"
        )
        vaccinated = cursor.fetchone()[0]
        connection.close()
        return render_template('index.html', vaccinated=vaccinated)
    else:
        return render_template('index.html')

if __name__ == '__main__':
    app.run(host='0.0.0.0')
```


The web application connects to the database container and queries the "student_vaccination" table for the vaccination status of the student with the given registration number. The vaccination status is returned to the HTML template, which displays it to the user.

9. Create an HTML template in the "app/templates" directory that will display the vaccination status of the student.

```
<!doctype html>
<html>
  <head>
    <title>Student Vaccination Status</title>
  </head>
  <body>
    <h1>Enter Registration Number</h1>
    <form method="POST">
      <input type="text" name="registration_number">
      <button type="submit">Check Status</button>
    </form>
    {% if vaccinated is not none %}
      {% if vaccinated %}
        <p>The student is vaccinated</p>
      {% else %}
        <p>The student is not vaccinated</p>
      {% endif %}
    {% endif %}
  </body>
</html>
```

```

-> / in using 89.7% of 7.57GB

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

3 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Mon May 1 07:44:20 2023 from 13.233.177.3
ubuntu@ip-172-31-6-233:~$ sudo usermod -aG docker $(USER)
ubuntu@ip-172-31-6-233:~$ sudo systemctl restart docker
ubuntu@ip-172-31-6-233:~$ mkdir student-registration-app
cd student-registration-app
ubuntu@ip-172-31-6-233:~/student-registration-app$ nano dockerfile
ubuntu@ip-172-31-6-233:~/student-registration-app$ app.py
app.py: command not found
ubuntu@ip-172-31-6-233:~/student-registration-app$ nano dockerfile
ubuntu@ip-172-31-6-233:~/student-registration-app$ cd /path/to/directory
-bash: cd: /path/to/directory: No such file or directory
ubuntu@ip-172-31-6-233:~/student-registration-app$ cd ~/student-registration-app
ubuntu@ip-172-31-6-233:~/student-registration-app$ cd /path/to/directory
-bash: cd: /path/to/directory: No such file or directory
ubuntu@ip-172-31-6-233:~/student-registration-app$ nano docker-compose.yml
ubuntu@ip-172-31-6-233:~/student-registration-app$ flask--1.1.2
mysql-connector-python==8.0.24
mysql-connector-python--8.0.24: command not found
ubuntu@ip-172-31-6-233:~/student-registration-app$ pip install mysql-connector-python==8.0.24
Defaulting to user installation because normal site-packages is not writeable
Collecting mysql-connector-python--8.0.24

```

i-003a9c275f6e38b89 (seenulinux)

PublicIPs: 3.110.44.193 PrivateIPs: 172.31.6.233

10. Build and run the application using Docker Compose.

docker-compose build
docker-compose up

```

Downloading Flask-1.1.2-py2.py3-none-any.whl (94 kB)
Requirement already satisfied: Jinja2>=2.10.1 in /usr/lib/python3/dist-packages (from Flask==1.1.2) (3.0.3)
Requirement already satisfied: click>=5.1 in /usr/lib/python3/dist-packages (from Flask==1.1.2) (8.0.3)
Requirement already satisfied: Werkzeug>=0.15 in /usr/local/lib/python3.10/dist-packages (from Flask==1.1.2) (2.3.2)
Requirement already satisfied: itsdangerous>=0.24 in /usr/local/lib/python3.10/dist-packages (from Flask==1.1.2) (2.1.2)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from Werkzeug>=0.15->Flask==1.1.2) (2.1.2)
Installing collected packages: Flask
WARNING: The script flask is installed in '/home/ubuntu/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Flask-1.1.2
ubuntu@ip-172-31-6-233:~/student-registration-app$ flask--1.1.2
mysql-connector-python==8.0.24
mysql-connector-python--8.0.24: command not found
ubuntu@ip-172-31-6-233:~/student-registration-app$ pip install mysql-connector-python==8.0.24
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: mysql-connector-python==8.0.24 in /home/ubuntu/.local/lib/python3.10/site-packages (8.0.24)
Requirement already satisfied: protobuf>=3.0.0 in /home/ubuntu/.local/lib/python3.10/site-packages (from mysql-connector-python==8.0.24) (4.22.3)
ubuntu@ip-172-31-6-233:~/student-registration-app$ cd ~/student-registration-app
ubuntu@ip-172-31-6-233:~/student-registration-app$ pip freeze > requirements.txt
ubuntu@ip-172-31-6-233:~/student-registration-app$ docker-compose up
Command 'docker-compose' not found, but can be installed with:
sudo snap install docker # version 20.10.17, or
sudo apt install docker-compose # version 1.29.2-1
See 'snap info docker' for additional versions.
ubuntu@ip-172-31-6-233:~/student-registration-app$ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-docker python3-dockerpty python3-docker python3-dotenv python3-texttable python3-websocket
Recommended packages:
  docker.io
The following NEW packages will be installed:
  docker-compose python3-docker python3-dockerpty python3-docker python3-dotenv python3-texttable python3-websocket
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 290 KB of archives.

```

i-003a9c275f6e38b89 (seenulinux)

PublicIPs: 3.110.44.193 PrivateIPs: 172.31.6.233

11. Access the web application in a web browser at <https://localhost:5000/>

The web application should prompt the user for a registration number and display the vaccination status of the student with that registration number, which it retrieves from the PostgreSQL database container.

Implementation:

