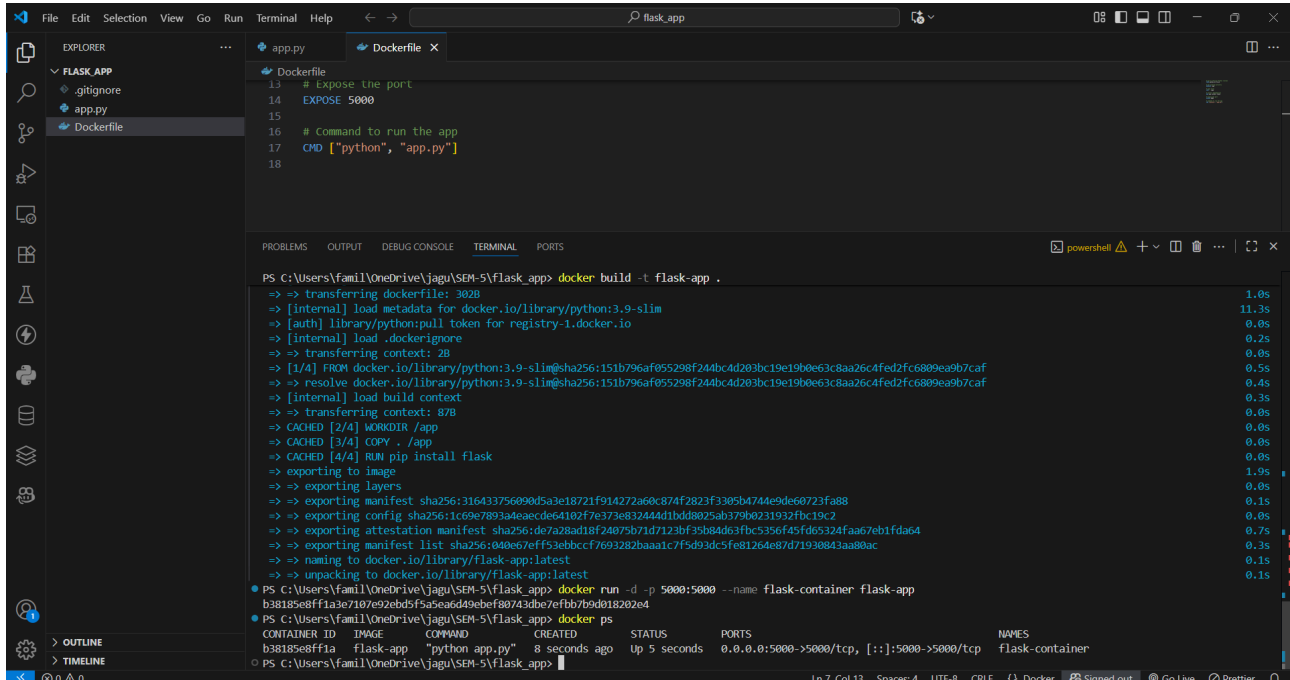


DevOps Intern Assignment [Jagruti Chaudhari]

Screenshot 1: Docker PS (Local)



The screenshot shows a Visual Studio Code editor with a project named 'FLASK_APP'. The Explorer sidebar on the left shows files: .gitignore, app.py, and Dockerfile. The Dockerfile is open in the editor, showing the following content:

```
Dockerfile
13 # expose the port
14 EXPOSE 5000
15
16 # command to run the app
17 CMD ["python", "app.py"]
18
```

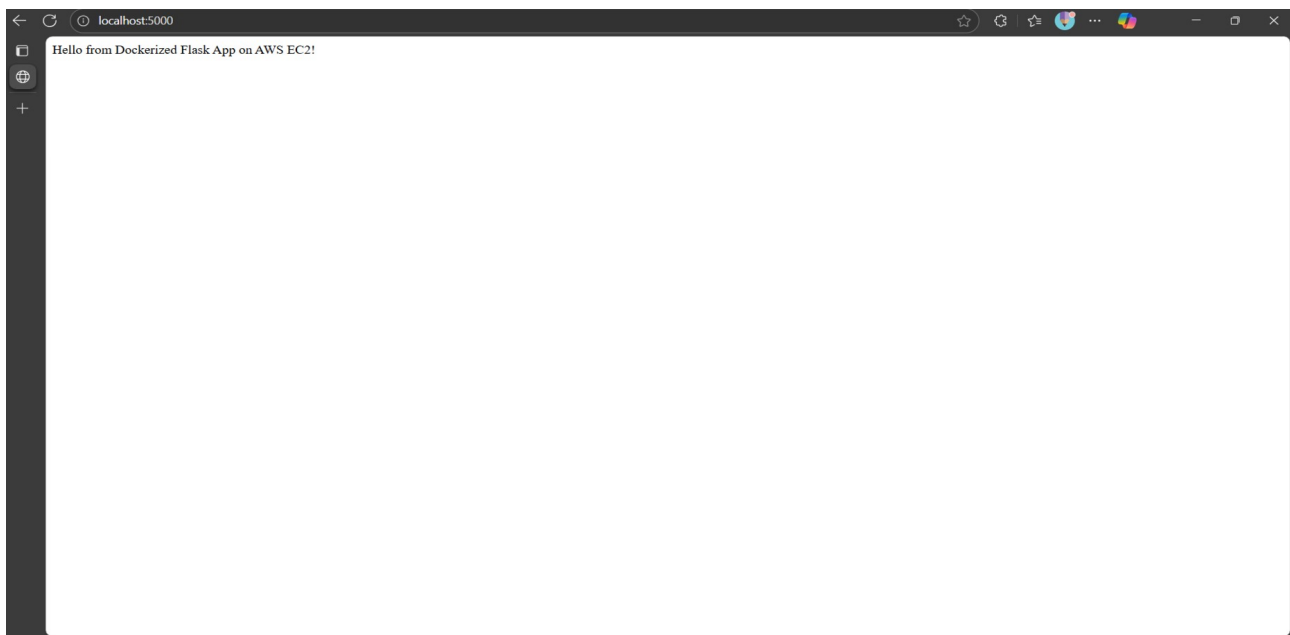
The Terminal panel at the bottom shows the output of the 'docker build -t flask-app .' command. The output includes progress bars for transferring the Dockerfile, resolving and pulling the Python image, building the context, and exporting the image and layers. The final output shows the image is ready to run.

```
PS C:\Users\famil\OneDrive\jagu\SEM-5\flask_app> docker build -t flask-app .
=> transferring dockerfile: 302B 1.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim 11.3s
=> [auth] library/python:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.2s
=> transferring context: 2B 0.0s
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:151b796af95298f244bc4d283bc19e19b0e63c8aa26c4fed2fc6809ea9b7caf 0.5s
=> resolve docker.io/library/python:3.9-slim@sha256:151b796af95298f244bc4d283bc19e19b0e63c8aa26c4fed2fc6809ea9b7caf 0.4s
=> [internal] load build context 0.3s
=> transferring context: 87B 0.0s
=> CACHED [2/4] WORKDIR /app 0.0s
=> CACHED [3/4] COPY . /app 0.0s
=> CACHED [4/4] RUN pip install flask 0.0s
=> exporting to image 1.9s
=> exporting layers 0.0s
=> exporting manifest sha256:316433756090d5a3e18721f914272a60c874f2823f3305b474ae9de60723fa88 0.1s
=> exporting config sha256:1c69e7893a4eae6de64102f7e373e832444d1bdd8025ab379b0231932fbc19c2 0.0s
=> exporting attestation manifest sha256:de7a28ad18f24075b71d7123bf35b84d63fbc5356f45fd65324faa67eb1fda64 0.7s
=> exporting manifest list sha256:840ee7eff53ebbccf7693282baaa1c7f5d93dc5fe81264e87d71930843aa80ac 0.3s
=> naming to docker.io/library/flask-app:latest 0.1s
=> unpacking to docker.io/library/flask-app:latest 0.1s

• PS C:\Users\famil\OneDrive\jagu\SEM-5\flask_app> docker run -d -p 5000:5000 --name flask-container flask-app
b38185e8ff1a3e7107e92ebd5f5a5ea6d49ebef80743dhe7efbb7b9d018202e4
• PS C:\Users\famil\OneDrive\jagu\SEM-5\flask_app> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
b38185e8ff1a   flask-app  "python app.py"         8 seconds ago  Up 5 seconds  0.0.0.0:5000->5000/tcp, [::]:5000->5000/tcp  flask-container
• PS C:\Users\famil\OneDrive\jagu\SEM-5\flask_app>
```

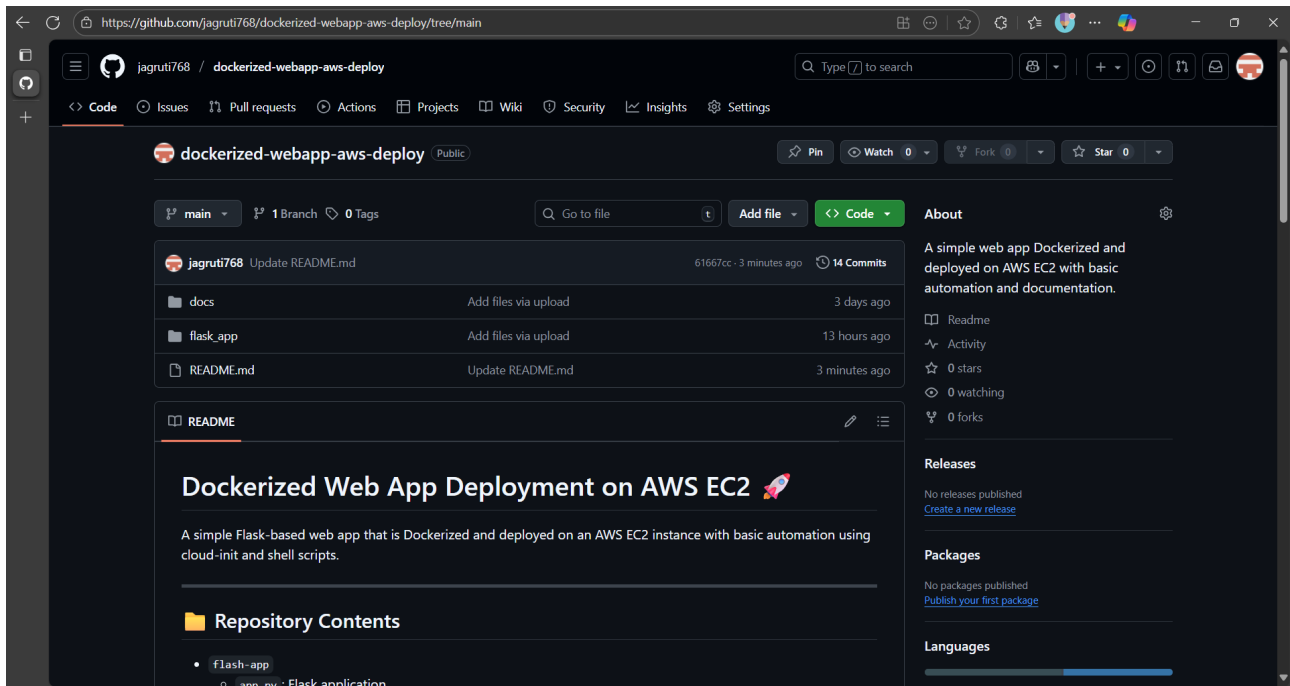
Description: Container running locally

Screenshot 2: Browser Localhost



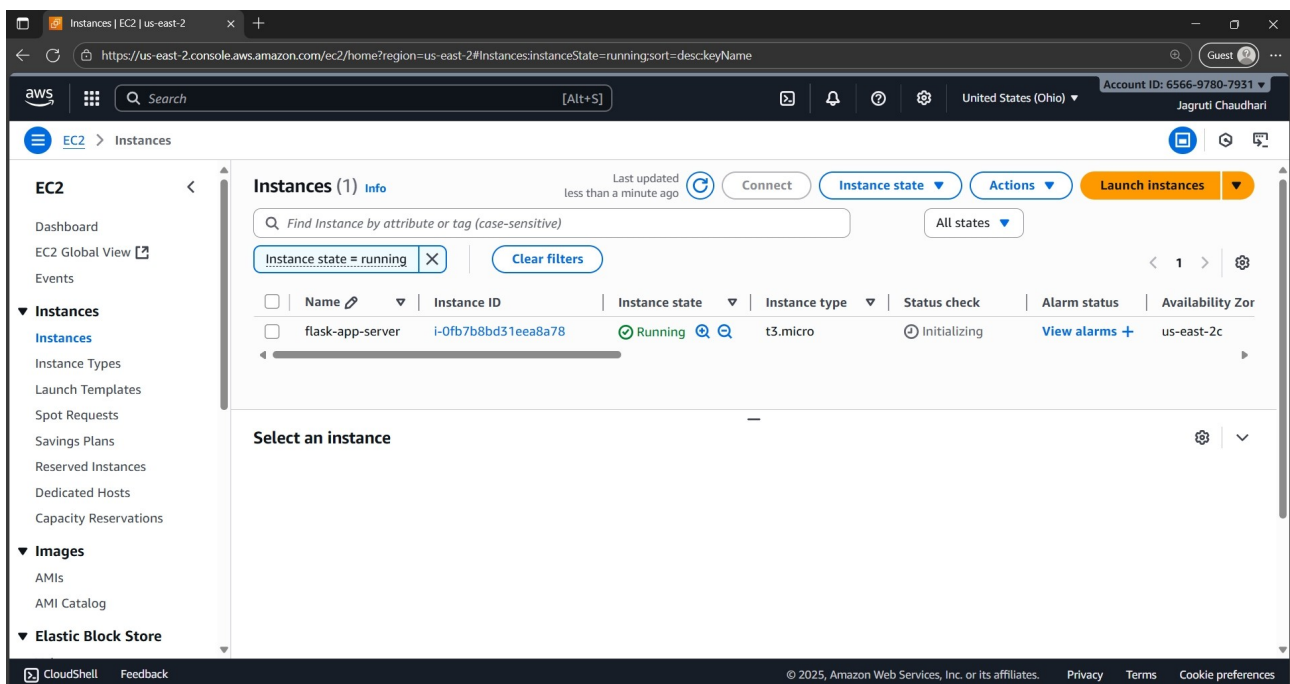
Description: App at <http://localhost:5000>

Screenshot 3: GitHub Repository



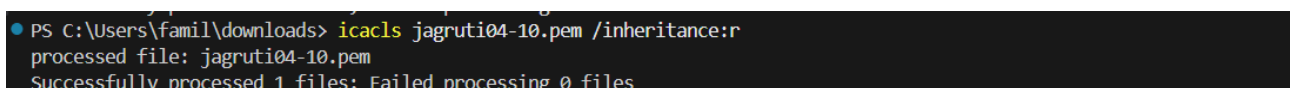
Description: Code pushed to GitHub

Screenshot 4: EC2 Dashboard



Description: EC2 instance running

Screenshot 5: SSH Connection



[illegible]

Description: Connected to EC2 via SSH

Screenshot 6: Docker Installation

```

ec2-user@ip-172-31-47-60 ~]$ sudo yum install docker -y
Last metadata expiration check: 0:00:16 ago on Sat Oct 4 18:21:36 2025.
Dependencies resolved.

=====
Package                                Architecture      Version           Size
=====
Installing:
docker                                x86_64            25.0.8-1.amzn2023.0.6      46 M
Installing dependencies:
container-selinux                     noarch            4:2.242.0-1.amzn2023      58 k
containerd                            x86_64            2.0.6-1.amzn2023.0.1      26 M
iptables-libns                        x86_64            1.8.8-3.amzn2023.0.2      401 k
iptables-nft                         x86_64            1.8.8-3.amzn2023.0.2      183 k
libcgroup                             x86_64            3.0-1.amzn2023.0.1        75 k
libnetfilter_conntrack               x86_64            1.0.8-2.amzn2023.0.2      58 k
libnftlink                           x86_64            1.0.1-19.amzn2023.0.2     30 k
libnftnl                             x86_64            1.2.2-2.amzn2023.0.2      84 k
pigz                                 x86_64            2.5-1.amzn2023.0.3        83 k
runc                                  x86_64            1.2.6-1.amzn2023.0.1      3.7 M
=====

Transaction Summary
-----
Install 11 Packages

Total download size: 77 M
Installed size: 292 M
Downloading Packages:
(1/11): container-selinux-2.242.0-1.amzn2023.noarch.rpm      1.8 MB/s | 58 kB | 00:00
(2/11): iptables-libns-1.8.8-3.amzn2023.0.2.x86_64.rpm      13 MB/s | 401 kB | 00:00
(3/11): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm        5.9 MB/s | 183 kB | 00:00
(4/11): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm            2.3 MB/s | 75 kB | 00:00
(5/11): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 2.4 MB/s | 58 kB | 00:00
(6/11): libnftlink-1.0.1-19.amzn2023.0.2.x86_64.rpm        1.5 MB/s | 30 kB | 00:00
(7/11): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm           2.5 MB/s | 84 kB | 00:00
(8/11): pigz-2.5-1.amzn2023.0.3.x86_64.rpm                 2.5 MB/s | 83 kB | 00:00
(9/11): runc-1.2.6-1.amzn2023.0.1.x86_64.rpm               62 MB/s | 3.7 MB | 00:00
(10/11): containerd-2.0.6-1.amzn2023.0.1.x86_64.rpm        65 MB/s | 26 MB | 00:00
(11/11): docker-25.0.8-1.amzn2023.0.6.x86_64.rpm           68 MB/s | 46 MB | 00:00

```

Description: Installed Docker on EC2

Screenshot 7: Docker Build on EC2

```
[ec2-user@ip-172-31-47-60 flask_app]$ sudo docker build -t flask-app .
[+] Building 8.0s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 361B
=> [internal] load metadata for docker.io/library/python:3.9-slim
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/python:3.9-slim@sha256:151b796af055298f244bc4d203bc19e19b0e63c8aa26c4fed2fc6809ea9b7caf
=> => resolve docker.io/library/python:3.9-slim@sha256:151b796af055298f244bc4d203bc19e19b0e63c8aa26c4fed2fc6809ea9b7caf
=> => sha256:151b796af055298f244bc4d203bc19e19b0e63c8aa26c4fed2fc6809ea9b7caf 10.36kB / 10.36kB
=> => sha256:2b999eb2d217c690516e4d91987aec72cb28ab1111d92c1020ca34edaaff96f 1.74kB / 1.74kB
=> => sha256:abf3b76f84b95695873e899bda96333946e32fb09a6d2c7cc93174ca2af72be8 5.30kB / 5.30kB
=> => sha256:8c7716127147648c1751940b9709b6325f2256290d3201662eca2701cadb2cdf 29.78MB / 29.78MB
=> => sha256:3c43e65991bd080f60cac1c513796828316f1f90951eaad19edb90cc21292184 1.29MB / 1.29MB
=> => sha256:52ad2c1a7277d777bf9ffe2be215fa2d09b576abc4fedd32016b7771f49afe07 13.37MB / 13.37MB
=> => sha256:c3facbf8ee169eafbae38460b42d4795c48c8d389a4bb357e0500903e5b2b164 249B / 249B
=> => extracting sha256:8c7716127147648c1751940b9709b6325f2256290d3201662eca2701cadb2cdf 1.4s
=> => extracting sha256:3c43e65991bd080f60cac1c513796828316f1f90951eaad19edb90cc21292184 0.1s
=> => extracting sha256:52ad2c1a7277d777bf9ffe2be215fa2d09b576abc4fedd32016b7771f49afe07 0.8s
=> => extracting sha256:c3facbf8ee169eafbae38460b42d4795c48c8d389a4bb357e0500903e5b2b164 0.0s
=> [internal] load build context
=> => transferring context: 666B
=> [2/4] WORKDIR /app
=> [3/4] COPY . /app
=> [4/4] RUN pip install flask
=> exporting to image
=> => exporting layers
=> => writing image sha256:70bd69f10c1b2636a1b88032feb1419cacee773ead4d3acc65c9868c33537f56
=> => naming to docker.io/library/flask-app
```

```
[ec2-user@ip-172-31-47-60 flask_app]$ sudo docker run -d -p 5000:5000 --name flask-container flask-app
9e6e75e98d2bbd5f7a28c6f9a5bc3fbc01f694d6fd76b0e22d7840a53c91f7d0
```

```
[ec2-user@ip-172-31-47-60 flask_app]$ curl http://localhost:5000
Hello from Dockerized Flask App on AWS EC2!
[ec2-user@ip-172-31-47-60 flask_app]$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
9e6e75e98d2b   flask-app "python app.py"         5 minutes ago Up 5 minutes   0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   flask-container
```

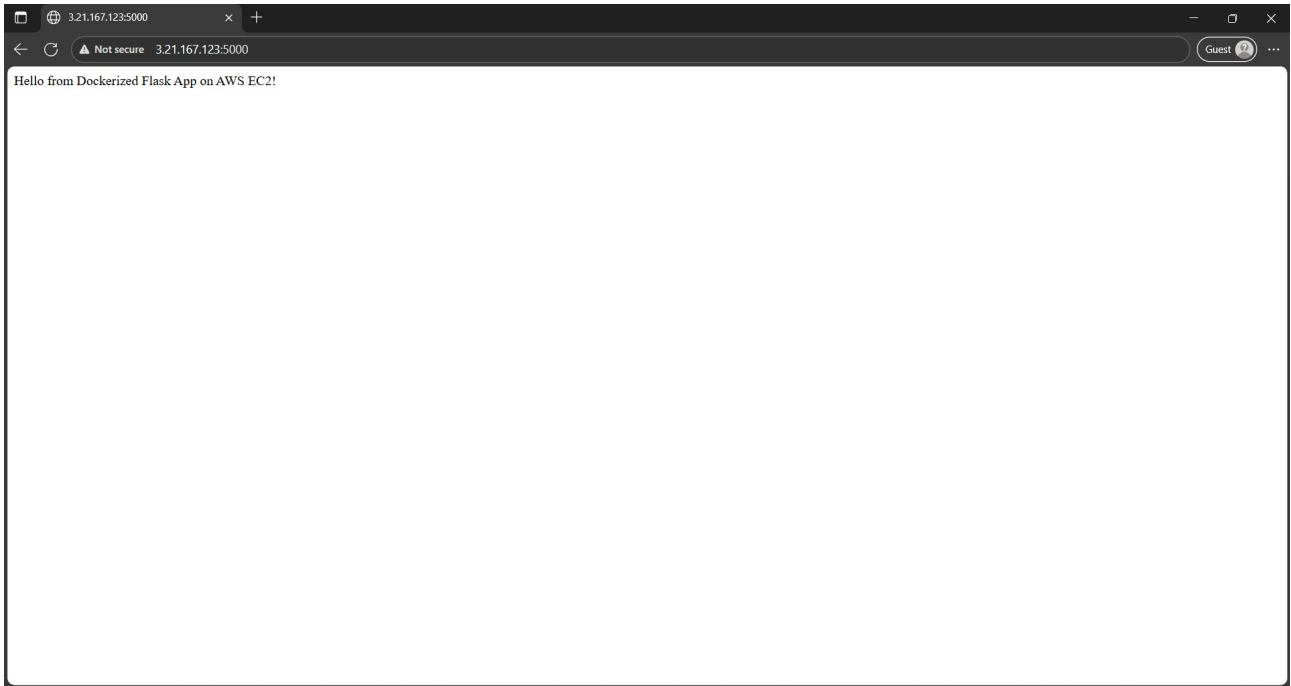
Description: Built image on EC2

Screenshot 8: Docker PS on EC2

```
[ec2-user@ip-172-31-47-60 flask_app]$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
9e6e75e98d2b   flask-app "python app.py"         18 seconds ago Up 17 seconds   0.0.0.0:5000->5000/tcp, :::5000->5000/tcp   flask-container
[ec2-user@ip-172-31-47-60 flask_app]$ curl http://localhost:5000
Hello from Dockerized Flask App on AWS EC2!
```

Description: Container running on EC2

Screenshot 9: Browser EC2 IP



Description: App at <http://3.21.167.123:5000/>