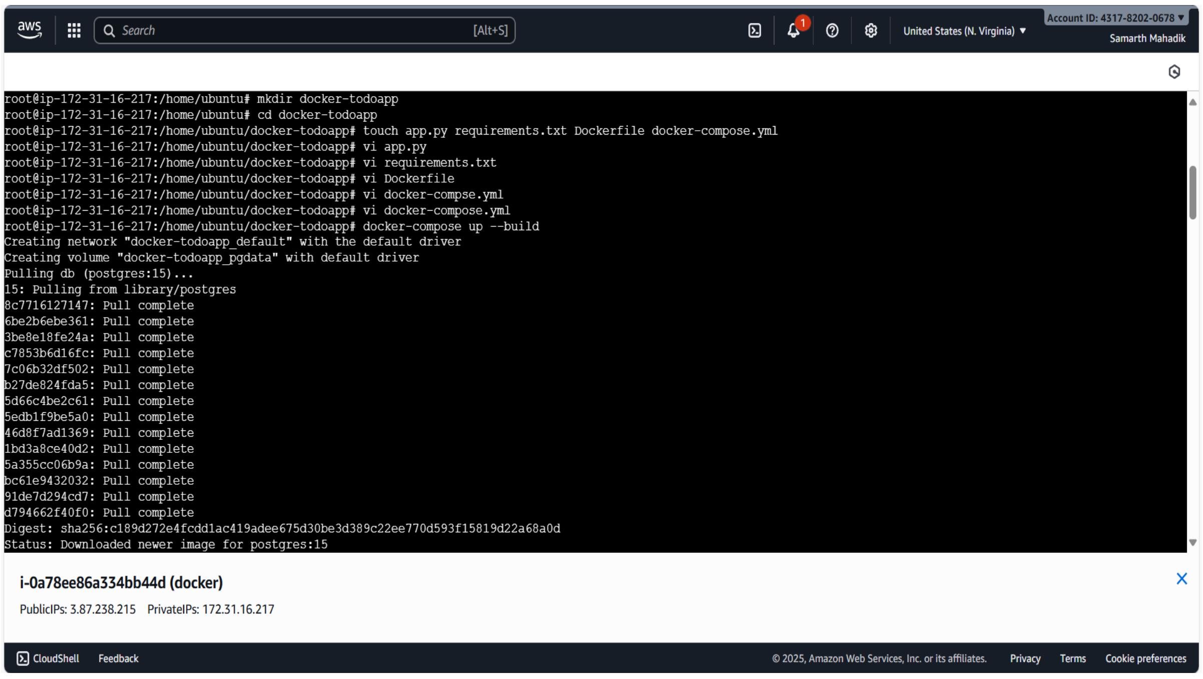
Docker Mini Project Report - Flask + PostgreSQL ToDo App

Project Overview:
This project demonstrates how to containerize a simple Flask-based To-Do application using Docker and Docker Compose. It includes two main services: Flask API container (Python backend) PostgreSQL database container

Implementation Steps:
1. Created project folder: docker-todoapp/
2. Added required files: app.py, requirements.txt, Dockerfile, docker-compose.yml
3. Defined Flask application using SQLAlchemy to interact with PostgreSQL.
4. Built Docker image using the command: docker compose up --build
5. Verified successful pull of PostgreSQL base image and container startup.
6. Tested API endpoints using curl and Postman.

Key Docker Commands Used:
docker compose up --build - Build and run multi-container app.
docker ps - View running containers.
docker exec -it <container> /bin/sh - Access inside a container.
docker compose down - Stop and remove containers.

Execution Screenshot:



```
root@ip-172-31-16-217:/home/ubuntu# mkdir docker-todoapp
root@ip-172-31-16-217:/home/ubuntu# cd docker-todoapp
root@ip-172-31-16-217:/home/ubuntu/docker-todoapp# touch app.py requirements.txt Dockerfile docker-compose.yml
root@ip-172-31-16-217:/home/ubuntu/docker-todoapp# vi app.py
root@ip-172-31-16-217:/home/ubuntu/docker-todoapp# vi requirements.txt
root@ip-172-31-16-217:/home/ubuntu/docker-todoapp# vi Dockerfile
root@ip-172-31-16-217:/home/ubuntu/docker-todoapp# vi docker-compse.yml
root@ip-172-31-16-217:/home/ubuntu/docker-todoapp# vi docker-compose.yml
root@ip-172-31-16-217:/home/ubuntu/docker-todoapp# docker-compose up --build
Creating network "docker-todoapp_default" with the default driver
Creating volume "docker-todoapp_pgdata" with default driver
Pulling db (postgres:15)...
15: Pulling from library/postgres
8c7716127147: Pull complete
6be2b6ebe361: Pull complete
3be8e18fe24a: Pull complete
c7853b6d16fc: Pull complete
7c06b32df502: Pull complete
b27de824fda5: Pull complete
5d66c4be2c61: Pull complete
5edb1f9be5a0: Pull complete
46d8f7ad1369: Pull complete
1bd3a8ce40d2: Pull complete
5a355cc06b9a: Pull complete
bc61e9432032: Pull complete
91de7d294cd7: Pull complete
d794662f40f0: Pull complete
Digest: sha256:c189d272e4fcdd1ac419adee675d30be3d389c22ee770d593f15819d22a68a0d
Status: Downloaded newer image for postgres:15
```

i-0a78ee86a334bb44d (docker)
PublicIPs: 3.87.238.215 PrivateIPs: 172.31.16.217

Result:
The Flask application and PostgreSQL database containers were successfully built and executed. The To-Do API endpoints worked as expected, confirming proper communication between containers via Docker Compose.

Conclusion:

This mini-project demonstrates practical Docker knowledge — including containerization, Dockerfile creation, network linking between services, and persistent volume management for PostgreSQL.