Assignment Documentation: Flask API with PostgreSQL and Docker

1. Project Setup

This project is designed to set up a simple API server using Flask and PostgreSQL, with Docker for containerization. The goal is to create a scalable web application with a PostgreSQL database, encapsulated in Docker containers, and optionally load-balanced with Nginx.

Project Structure:

- **Dockerfile**: Defines the Python application image and installation of dependencies.
- **docker-compose.yml**: Defines the services for the Flask app and PostgreSQL database.
- **app.py**: The Flask application that handles API requests.
- **requirements.txt**: Lists the necessary Python packages (Flask, SQLAlchemy, psycopg2-binary).
- **nginx.conf** (optional): Defines the Nginx configuration for load balancing.
- **logs/**: A directory to store logs generated by the application.

2. Core Functionality

API Endpoints:

1. **POST /user**: Creates a new user in the PostgreSQL database. The request body should contain the first_name and last_name of the user.
2. **GET /user/<id>**: Retrieves a user by their ID from the PostgreSQL database.

The Flask application uses SQLAlchemy to interact with the PostgreSQL database. The database URI is configured as postgresql://user:password@db/users, where db is the PostgreSQL container.

Database Setup:

- The PostgreSQL database is initialized with the user, password, and users database name defined in the docker-compose.yml file.
- The application ensures that the necessary tables are created when it starts up.

3. Docker and Docker Compose

The use of Docker Compose simplifies the orchestration of both the Flask application and the PostgreSQL database. With a single command (docker-compose up -d), both services are started and connected. The network isolation ensures that the services can communicate securely.

4. Load Balancing with Nginx (Optional)

An optional Nginx load balancer is included in the docker-compose.yml file to forward traffic to the Flask application. This is beneficial for scaling the application and ensuring high availability in a production environment.

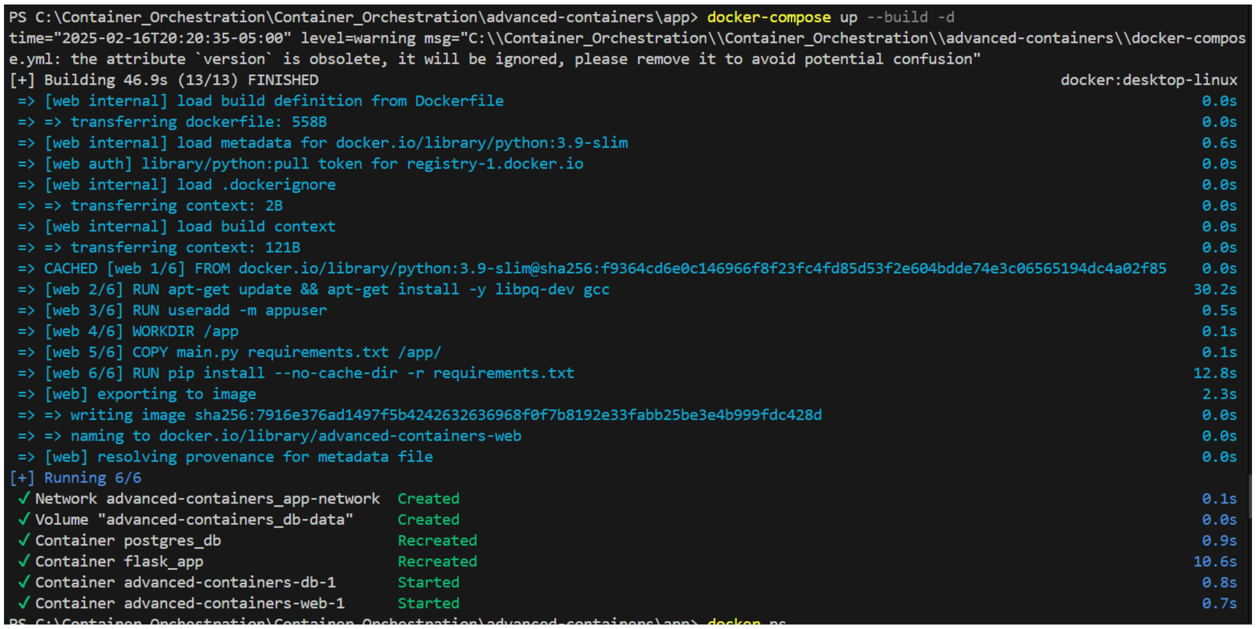
5. Security Practices

- **Non-root User**: The Flask application runs as a non-root user to avoid security risks.
- **Environment Variables for Secrets**: Any sensitive information (like database credentials) should be stored as environment variables, avoiding hardcoding them in the codebase.
- **Slim Docker Image**: The use of the python:3.9-slim image minimizes the attack surface by reducing unnecessary components in the Docker image.

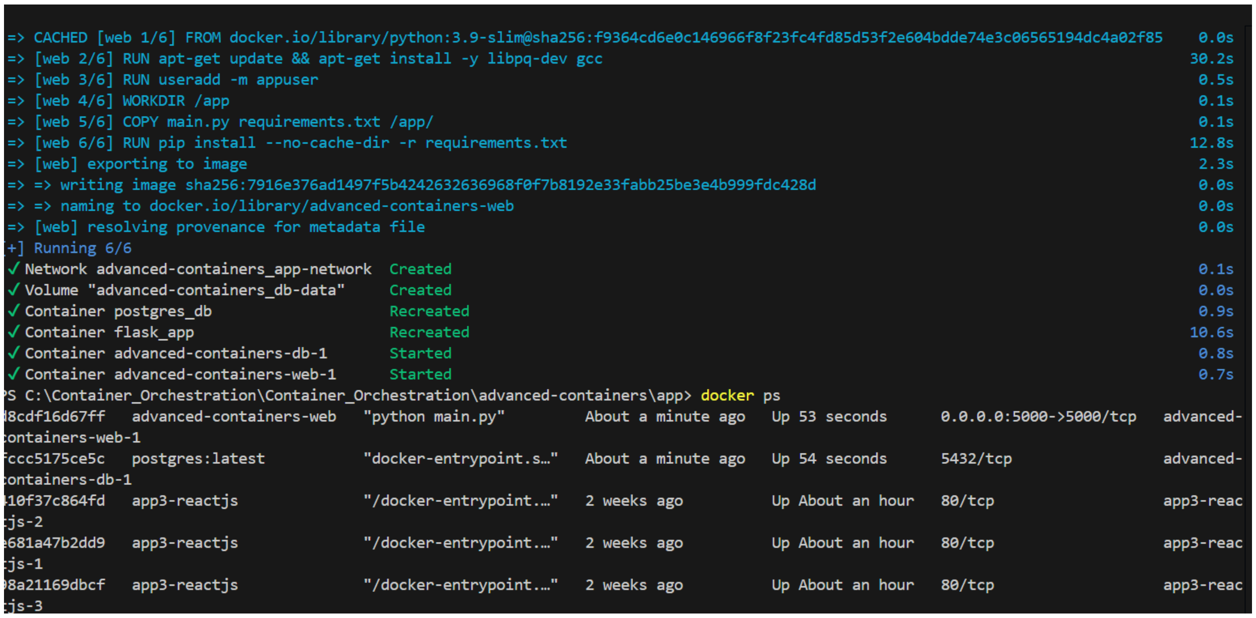
6. Deployment and Scaling

The project can be scaled by increasing the number of Flask application containers using Docker Compose. In a production environment, further steps like setting up a reverse proxy (e.g., Nginx) and a more robust container orchestration solution (e.g., Kubernetes) may be requir

Screen shots:



```
PS C:\Container_Orchestration\Container_Orchestration\advanced-containers\app> docker-compose up --build -d
time="2025-02-16T20:20:35-05:00" level=warning msg="C:\\Container_Orchestration\\Container_Orchestration\\advanced-containers\\docker-compos
e.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Building 46.9s (13/13) FINISHED                                                                           docker:desktop-linux
 => [web internal] load build definition from Dockerfile                                                                      0.0s
 => => transferring dockerfile: 558B                                                                                          0.0s
 => [web internal] load metadata for docker.io/library/python:3.9-slim                                                        0.6s
 => [web auth] library/python:pull token for registry-1.docker.io                                                             0.0s
 => [web internal] load .dockerignore                                                                                         0.0s
 => => transferring context: 2B                                                                                               0.0s
 => [web internal] load build context                                                                                         0.0s
 => => transferring context: 121B                                                                                             0.0s
 => CACHED [web 1/6] FROM docker.io/library/python:3.9-slim@sha256:f9364cd6e0c146966f8f23fc4fd85d53f2e604bdde74e3c06565194dc4a02f85   0.0s
 => [web 2/6] RUN apt-get update && apt-get install -y libpq-dev gcc                                                         30.2s
 => [web 3/6] RUN useradd -m appuser                                                                                          0.5s
 => [web 4/6] WORKDIR /app                                                                                                    0.1s
 => [web 5/6] COPY main.py requirements.txt /app/                                                                             0.1s
 => [web 6/6] RUN pip install --no-cache-dir -r requirements.txt                                                             12.8s
 => [web] exporting to image                                                                                                  2.3s
 => => writing image sha256:7916e376ad1497f5b4242632636968f0f7b8192e33fabb25be3e4b999fdc428d                                  0.0s
 => => naming to docker.io/library/advanced-containers-web                                                                    0.0s
 => [web] resolving provenance for metadata file                                                                              0.0s
[+] Running 6/6
 ✓ Network advanced-containers_app-network  Created                                                                           0.1s
 ✓ Volume "advanced-containers_db-data"     Created                                                                           0.0s
 ✓ Container postgres_db                    Recreated                                                                         0.9s
 ✓ Container flask_app                      Recreated                                                                        10.6s
 ✓ Container advanced-containers-db-1       Started                                                                           0.8s
 ✓ Container advanced-containers-web-1      Started                                                                           0.7s
PS C:\Container_Orchestration\Container_Orchestration\advanced-containers\app> docker ps
```


```
=> CACHED [web 1/6] FROM docker.io/library/python:3.9-slim@sha256:f9364cd6e0c146966f8f23fc4fd85d53f2e604bdde74e3c06565194dc4a02f85    0.0s
=> [web 2/6] RUN apt-get update && apt-get install -y libpq-dev gcc                                                                  30.2s
=> [web 3/6] RUN useradd -m appuser                                                                                                   0.5s
=> [web 4/6] WORKDIR /app                                                                                                             0.1s
=> [web 5/6] COPY main.py requirements.txt /app/                                                                                      0.1s
=> [web 6/6] RUN pip install --no-cache-dir -r requirements.txt                                                                      12.8s
=> [web] exporting to image                                                                                                           2.3s
=> => writing image sha256:7916e376ad1497f5b4242632636968f0f7b8192e33fabb25be3e4b999fdc428d                                          0.0s
=> => naming to docker.io/library/advanced-containers-web                                                                             0.0s
=> [web] resolving provenance for metadata file                                                                                       0.0s
+] Running 6/6
✓ Network advanced-containers_app-network  Created                                                                                   0.1s
✓ Volume "advanced-containers_db-data"     Created                                                                                   0.0s
✓ Container postgres_db                    Recreated                                                                                 0.9s
✓ Container flask_app                      Recreated                                                                                10.6s
✓ Container advanced-containers-db-1       Started                                                                                   0.8s
✓ Container advanced-containers-web-1      Started                                                                                   0.7s
S C:\Container_Orchestration\Container_Orchestration\advanced-containers\app> docker ps
8cdf16d67ff    advanced-containers-web    "python main.py"          About a minute ago   Up 53 seconds      0.0.0.0:5000->5000/tcp   advanced-
ontainers-web-1
cccc5175ce5c   postgres:latest            "docker-entrypoint.s…"    About a minute ago   Up 54 seconds      5432/tcp                 advanced-
ontainers-db-1
10f37c864fd    app3-reactjs               "/docker-entrypoint.…"    2 weeks ago          Up About an hour   80/tcp                   app3-reac
js-2
681a47b2dd9    app3-reactjs               "/docker-entrypoint.…"    2 weeks ago          Up About an hour   80/tcp                   app3-reac
js-1
8a21169dbcf    app3-reactjs               "/docker-entrypoint.…"    2 weeks ago          Up About an hour   80/tcp                   app3-reac
js-3
```