

README.md

# FDS Assignment 1

## Problem Statement

Set up a multi-container distributed web application using Docker consisting of:

- A web front-end (e.g., Nginx or Apache).
  - A back-end service (e.g., Python/Node.js microservice).
  - A database (e.g., MySQL/PostgreSQL).
1. Create separate Dockerfiles for each component (front-end, back-end, and database). Ensure proper environment variable configuration (e.g., database credentials).
  2. Write a docker-compose.yml to orchestrate the three containers so they can communicate with one another (e.g., linking the back-end service to the database, exposing required ports).
  3. Demonstrate how you would scale the back-end service container (e.g., running multiple replicas). \* Test the setup by sending requests to the load-balanced back-end through the front-end container.
  4. Provide a short report (about 1 page) describing your Docker setup, including:
    - Key commands used
    - Network configurations
    - How to build, start, and stop the entire application stack

This is a Heading h6

## Solution

- Clone this repo

- Start the app (including building all images) `docker-compose up -d --build`

```
PS C:\Users\91889\Desktop\IIIT Mtech Quiz and Assignments\fds_assignment_1> docker-compose up -d --build
[+] Building 0.0s (0/0) docker:default
2025/03/16 23:38:12 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 11.0s (14/15)
[+] Building 11.3s (15/15)
[+] Building 14.2s (23/23) FINISHED
=> [db internal] load build definition from Dockerfile
=> => transferring dockerfile: 1498
=> [backend internal] load build definition from Dockerfile
=> => transferring dockerfile: 2788
=> [backend internal] load metadata for docker.io/library/python:3.9-slim
=> [db internal] load metadata for docker.io/library/postgres:13-alpine
=> [backend internal] load .dockerignore
=> => transferring context: 2B
=> [db internal] load .dockerignore
=> => transferring context: 2B
=> CACHED [db 1/1] FROM docker.io/library/postgres:13-alpine@sha256:236985828131e95a12914071b944d0e0d21da5281312292747e222845f0ea670
=> [backend internal] load build context
=> => transferring context: 2.25kB
=> [backend 1/5] FROM docker.io/library/python:3.9-slim@sha256:d1fd807555208707ec95b284afd10048d0737e84b5f2d6dcbcd2922b9284b56
=> [db] exporting to image
=> => exporting layers
=> => writing image sha256:cad25d55df73551f66fce305aaa8d1a64f1e9594f163e7b5c25ff1ba5688716
=> => naming to docker.io/library/fds_assignment_1-db
=> CACHED [backend 2/5] WORKDIR /app
=> [backend 3/5] COPY requirements.txt .
=> [backend 4/5] RUN pip install -r requirements.txt
=> [backend 5/5] COPY . .
=> [backend] exporting to image
=> => exporting layers
=> => writing image sha256:521f2f13cde783e0f2611e9e1a1c078360247c4bfd0cb78df662532f28b9cdc2
=> => naming to docker.io/library/fds_assignment_1-backend
=> [frontend internal] load build definition from Dockerfile
=> => transferring dockerfile: 1398
=> [frontend internal] load metadata for docker.io/library/nginx:alpine
=> [frontend internal] load .dockerignore
=> => transferring context: 2B
=> CACHED [frontend 1/3] FROM docker.io/library/nginx:alpine@sha256:4ff102c5d78d254a6f0da062b3cf39eaf07f01e0c0927fd21e219d0af8bc0591
=> [frontend internal] load build context
=> => transferring context: 1.38kB
=> [frontend 2/3] COPY static /usr/share/nginx/html
=> [frontend 3/3] COPY nginx.conf /etc/nginx/nginx.conf
=> [frontend] exporting to image
=> => exporting layers
=> => writing image sha256:35326a54e8501b8d7a251e50b85ea984e853fb084e4873230a0940caf930ebbb1
=> => naming to docker.io/library/fds_assignment_1-frontend
[+] Running 3/5
 - Network fds_assignment_1_app-network Created 1.8s
 - Volume "fds_assignment_1_postgres_data" Created 1.7s
 ✓ Container fds_assignment_1-db-1 Started 1.1s
 ✓ Container fds_assignment_1-backend-1 Started 0.9s
 ✓ Container fds_assignment_1-frontend-1 Started 1.6s
PS C:\Users\91889\Desktop\IIIT Mtech Quiz and Assignments\fds_assignment_1>
```

- Scale backend using command `docker-compose up -d --scale backend=3`

```
PS C:\Users\91889\Desktop\IIIT Mtech Quiz and Assignments\fds_assignment_1> docker-compose up -d --scale backend=3
[+] Running 5/5
 ✓ Container fds_assignment_1-db-1 Running 0.0s
 ✓ Container fds_assignment_1-backend-1 Running 0.0s
 ✓ Container fds_assignment_1-backend-3 Started 0.7s
 ✓ Container fds_assignment_1-backend-2 Started 1.1s
 ✓ Container fds_assignment_1-frontend-1 Running 0.0s
PS C:\Users\91889\Desktop\IIIT Mtech Quiz and Assignments\fds_assignment_1>
```

- Test scaling using command `1..10 | ForEach-Object { Invoke-WebRequest -Uri http://localhost/api/data -UseBasicParsing | Select-Object Content } OR`  
`command for i in {1..10}; do curl -s http://localhost/api/data; echo "";`  
`done` in Powershell or shell respectively.

```
PS C:\Users\91889\Desktop\IIIT Mtech Quiz and Assignments\fds_assignment_1> 1..10 | ForEach-Object { Invoke-WebRequest -Uri http://localhost/api/data -UseBasicParsing | Select-Object Content }
Content
-----
["2025-03-16 18:12:12.261053"]...
["2025-03-16 18:12:12.326105","2025-03-16 18:12:12.261053"]...
["2025-03-16 18:12:12.350626","2025-03-16 18:12:12.326105","2025-03-16 18:12:12.261053"]...
["2025-03-16 18:12:12.375875","2025-03-16 18:12:12.350626","2025-03-16 18:12:12.326105","2025-03-16 18:12:12.261053"]...
["2025-03-16 18:12:12.400543","2025-03-16 18:12:12.375875","2025-03-16 18:12:12.350626","2025-03-16 18:12:12.326105","2025-03-16 18:12:12.261053"]...
["2025-03-16 18:12:12.425387","2025-03-16 18:12:12.400543","2025-03-16 18:12:12.375875","2025-03-16 18:12:12.350626","2025-03-16 18:12:12.326105","2025-03-16 18:12:12.261053"]...
["2025-03-16 18:12:12.450640","2025-03-16 18:12:12.425387","2025-03-16 18:12:12.400543","2025-03-16 18:12:12.375875","2025-03-16 18:12:12.350626","2025-03-16 18:12:12.326105","2025-03-16 18:12:12.261053"]...
["2025-03-16 18:12:12.475995","2025-03-16 18:12:12.450640","2025-03-16 18:12:12.425387","2025-03-16 18:12:12.400543","2025-03-16 18:12:12.375875","2025-03-16 18:12:12.350626","2025-03-16 18:12:12.326105","2025-03-16 18:12:12.261053"]...
["2025-03-16 18:12:12.501861","2025-03-16 18:12:12.475995","2025-03-16 18:12:12.450640","2025-03-16 18:12:12.425387","2025-03-16 18:12:12.400543","2025-03-16 18:12:12.375875","2025-03-16 18:12:12.350626","2025-03-16 18:12:12.326105","2025-03-16 18:12:12.261053"]...
["2025-03-16 18:12:12.527922","2025-03-16 18:12:12.501861","2025-03-16 18:12:12.475995","2025-03-16 18:12:12.450640","2025-03-16 18:12:12.425387","2025-03-16 18:12:12.400543","2025-03-16 18:12:12.375875","2025-03-16 18:12:12.350626","2025-03-16 18:12:12.326105","2025-03-16 18:12:12.261053"]...
PS C:\Users\91889\Desktop\IIIT Mtech Quiz and Assignments\fds_assignment_1>
```

- Look at logs `docker-compose logs backend | grep "GET /api/data"` and understand which containers server which request. Observe multiple containers like backend-1, backend-2, etc
- Cleanup using command `docker-compose down -v --remove-orphans`

```
PS C:\Users\91889\Desktop\IIIT Mtech Quiz and Assignments\fds_assignment_1> docker-compose down -v --remove-orphans
[+] Running 7/7
 ✓ Container fds_assignment_1-frontend-1 Removed 0.8s
 ✓ Container fds_assignment_1-db-1 Removed 0.6s
 ✓ Container fds_assignment_1-backend-1 Removed 10.8s
 ✓ Container fds_assignment_1-backend-3 Removed 10.9s
 ✓ Container fds_assignment_1-backend-2 Removed 10.6s
 ✓ Volume fds_assignment_1_postgres_data Removed 0.1s
 ✓ Network fds_assignment_1_app-network Removed 0.3s
PS C:\Users\91889\Desktop\IIIT Mtech Quiz and Assignments\fds_assignment_1>
```

- For deep clean in case this is not supposed to be run anymore on the system, use command `docker image prune -f`

## Architecture Overview

### Multi-Container Setup:

- Frontend: Nginx serves static content and acts as a reverse proxy/load balancer for backend services.
- Backend: Python Flask microservice handles API requests and interacts with the database.
- Database: PostgreSQL for persistent data storage (rides, access logs, etc.).

### Orchestration:

Managed via `docker-compose.yml` for seamless deployment. Containers communicate over a custom bridge network (`app-network`).

## Key Features

### Scalability:

- Backend services can be scaled horizontally using `docker-compose --scale backend=N`.
- Nginx load-balances traffic across backend replicas (round-robin by default).

### Service Discovery:

- Containers resolve each other by service name (e.g., `backend:5000`, `db:5432`) via Docker's internal DNS.

### Environment Variables:

- Database credentials (host, port, user, password) injected via environment variables for security and flexibility.

### Data Persistence:

- PostgreSQL data stored in a Docker volume (`postgres_data`) to survive container restarts.

### Fault Tolerance:

- Stateless backend replicas ensure high availability.
- Database volume prevents data loss on container crashes.

