**File Hash Generator with Docker**

This document describes the process to containerize and run the File Hash Generator application using Docker and Docker Compose.

**Table of Contents**

1. **Project Structure**
2. **Prerequisites**
3. **Setup and Configuration**
4. **Docker Instructions**
   * Build Docker Images
   * Run Services
   * Stop Services
   * Rebuild Services
   * View Logs
5. **Application Access**
6. **Troubleshooting**

**Project Structure**

The project is organized as follows:

file-hash-app/  
│  
├── backend/  
│ ├── app.py # Backend logic (Flask API)  
│ ├── requirements.txt # Python dependencies for backend  
│ ├── Dockerfile # Docker definition for backend  
│  
├── frontend/  
│ ├── index.html # Main frontend page  
│ ├── script.js # Frontend logic  
│ ├── styles.css # Frontend styling  
│ ├── Dockerfile # Docker definition for frontend  
│  
└── docker-compose.yml # Docker Compose configuration file

**Prerequisites**

Before running the application, ensure the following are installed on your machine:

1. **Docker**: Download and install Docker from [Docker’s official website](https://www.docker.com/).
   * Verify the installation by running:

docker --version

1. **Docker Compose**: Usually bundled with Docker Desktop. Verify by running:

docker-compose --version

**Setup and Configuration**

The docker-compose.yml file defines two services:

* **Backend**: Runs the Flask application and listens on http://localhost:5000.
* **Frontend**: Serves static files (index.html, script.js, etc.) and listens on http://localhost:3000.

**Docker Instructions**

**1. Build Docker Images**

Run this command from the project root (where docker-compose.yml is located):

docker-compose build

This builds Docker images for both the backend and frontend using their respective Dockerfile configurations.

**2. Run Services**

To start all services (backend and frontend), run:

docker-compose up

If successful, you’ll see logs for both containers in the terminal.

* **Frontend**: Accessible at <http://localhost:3000>
* **Backend**: Accessible at <http://localhost:5000>

To run services in detached mode (background), use:

docker-compose up -d

**3. Stop Services**

To stop and remove the running containers, use:

docker-compose down

**4. Rebuild Services**

If you make changes to the code, rebuild the images and restart the services:

docker-compose up --build

**5. View Logs**

To view logs for all services, run:

docker-compose logs -f

For logs of a specific service (e.g., backend):

docker-compose logs -f backend

**Application Access**

**Frontend**

* URL: <http://localhost:3000>
* Features:
  + Upload a file via the form.
  + View the MD5 hash of the uploaded file.

**Backend**

* URL: <http://localhost:5000>
* Features:
  + Processes uploaded files via the /upload endpoint.
  + Validates file type and size.
  + Returns MD5 hash of the file content.

**Troubleshooting**

If you encounter any issues, try the following steps:

1. **Ensure Docker is Running**
   * Make sure Docker Desktop or your system’s Docker service is running in the background.
2. **Check Services**
   * Verify that the backend is running on port 5000 and the frontend on port 3000.
3. **Rebuild Images**
   * If changes to the code are not reflected, rebuild using:

docker-compose up --build

1. **View Logs**
   * Use the docker-compose logs command to check for errors in the logs.
2. **Inspect Containers**
   * Use docker ps to list running containers:

docker ps

* For debugging, log in to a container using:

docker exec -it <container\_name> /bin/bash

**Docker Commands Summary**

|  |  |
| --- | --- |
| **Command** | **Description** |
| docker-compose build | Build Docker images |
| docker-compose up | Start all services |
| docker-compose up -d | Start services in detached mode |
| docker-compose down | Stop and remove all services |
| docker-compose logs -f | View logs of all services |
| docker-compose logs -f <service> | View logs of a specific service |
| docker exec -it <container> bash | Login into a running Docker container |
| docker ps | List all running containers |

**Conclusion**

This README explains how to use the Docker Compose to containerize and run the application.