**Fullstack Todo List App - Docker Setup Documentation**

This documentation provides a detailed guide on setting up, running, and managing your Fullstack Todo List application using Docker. It covers setup instructions, network and security configurations, and troubleshooting tips to ensure a smooth setup.

**Setup Instructions**

**Prerequisites:**

Before starting the setup, ensure you have the following installed:

* **Git Bash** (for running commands)
* **Docker** (for container management)
* **Docker Compose** (for orchestrating multiple containers)

**Step-by-Step Setup Guide:**

1. **Download Project Files**: Make sure you have the project files locally. Download the files from this [repository](https://github.com/icnoka/fullstack-todo-list).
2. **Download the Deliverable Files**: Download the files from this [repository](https://github.com/Tgwrk/Cloud-Engineering-Pathway-Assessment.git). Copy the Dockerfiles from the repository to their respective folders in the project folders. Your project directory should contain the following structure:

/fullstack-todo-list

/frontend/Dockerfile

/backend/Dockerfile

/database/Dockerfile

docker-compose.yml

1. **Navigate to Project Directory**: Open Git Bash and navigate to the project directory:

* cd /path/to/fullstack-todo-list

1. **Build Docker Containers**: Run the following command to build all the Docker containers (frontend, backend, and database):

* docker-compose build

1. **Start the Containers**: Once the build process is completed, run the following command to start the containers:

* docker-compose up

1. **Access the Application**: After the containers have started, you can access the app in your browser:

* **Frontend**: Open your browser and go to http://localhost. You should see the frontend of your application.
* **Backend API**: The backend API will be accessible on http://localhost:3000.

1. **Stop the Containers**: To stop the containers when you're done, run the following command:

* docker-compose down

**Network and Security Configurations**

1. **Network Configuration:**

* The containers communicate over a Docker bridge network (app-network).
* The docker-compose.yml file ensures all services are connected.

1. **Exposed Ports:**

* **Frontend**: Port 80 is exposed, making the frontend application accessible at http://localhost.
* **Backend**: Port 3000 is exposed, making the backend API accessible at http://localhost:3000.
* **Database**: Port 27017 is exposed, allowing direct access to the MongoDB database from the host machine.

1. **Environment Variables:**

* In the docker-compose.yml file, there is an environment variable for the backend to connect to the MongoDB database:
  + environment: MONGODB\_URI=mongodb://database:27017/todo-app

**Troubleshooting Guide**

Here are some common issues you might encounter and potential solutions.

1. **Docker Is Not Running**

* **Issue:** Commands fail with an error like docker: command not found.
* **Solution:** Ensure Docker is installed and running. Restart Docker and try again.

1. **Containers Fail to Start**

* **Issue:** The app is not accessible in the browser.
* **Solution:**
  + Check running containers: docker ps
  + If a container is missing, restart it: docker-compose up --build -d
  + View container logs: docker logs <container\_id>

1. **Database Connection Issues**

* **Issue:** Backend cannot connect to MongoDB.
* **Solution:**
  + Ensure MongoDB is running: docker ps
  + Restart the database container: docker restart <database\_container\_id>

1. **Port Conflicts**

* Ensure the ports specified in docker-compose.yml are not in use by other applications.

**Conclusion**

This documentation has provided a step-by-step guide to set up, configure, and troubleshoot your Fullstack Todo List application. By following the instructions, you should be able to build, run, and manage the containers effectively. If you encounter any issues, refer to the troubleshooting section for possible solutions.