**Dockerization Documentation for Ecommerce Application**

This project is an ecommerce platform that uses a Domain-Driven Design (DDD) structure, consisting of three bounded contexts: **Shopping**, **Products**, and **Customer**. The application runs all domains on a single server `app.js` and connects to a shared MongoDB instance.

**Directory Structure**



**MongoDB Connection**

All domains share the same MongoDB instance. The database connection is managed by the shared/infrastructure/Database.js file.

**Steps to Dockerize**

1. **Dockerfile**

The Dockerfile defines the environment and steps required to run the application inside a container.

FROM node:20

WORKDIR /usr/src/app

COPY . .

RUN npm install

EXPOSE 3000

CMD ["npm", "run", "dev"]

1. **Docker Compose**

The docker-compose.yml file defines and manages multiple services, including the application and the MongoDB database.

**docker-compose.yml**

version: '3.8'

services:

  app:

    build:

      context: .

    container\_name: ecommerce\_app

    ports:

      - "3000:3000"

    environment:

      - MONGO\_URI=mongodb://mongo:27017/ecommerce

    depends\_on:

      - mongo

    networks:

      - ecommerce-network

  mongo:

    image: mongo:latest

    container\_name: mongo

    volumes:

      - mongo-data:/data/db

    ports:

      - "27017:27017"

    networks:

      - ecommerce-network

volumes:

  mongo-data:

    driver: local

networks:

  ecommerce-network:

    driver: bridge

1. **Environment Variables**

Ensure the application connects to the MongoDB instance using the appropriate environment variable. You can define the MONGO\_URI in your .env file or directly within docker-compose.yml.

MONGO\_URI=mongodb://mongo:27017/ecommerce

1. **Running the Application**

To build and run the application with Docker:

**1: Build the Docker image:**

docker-compose build

**2: Start the application:**

docker-compose up

This will:

* Build and run your Node.js app.
* Start a MongoDB instance, which the app connects to using the MONGO\_URI (mongodb://mongo:27017/ecommerce).

**5. Accessing the Application**

Once everything is running, you can access the application at: http://localhost:3000