Task 2 - Containerize and Run Medusa Locally Using Docker

Date: 11-09-2024

Step 1: Install Docker

Ensure Docker is installed on your system. Download Docker from the official Docker website and follow the installation instructions for your OS (Windows, macOS, or Linux).

Step 2: Clone the Medusa GitHub Repository

Clone the official Medusa repository and navigate to the directory:

git clone https://github.com/Jaswanthredd/PearlThoughts_Internship.git

cd Medusa_backend

Step 3: Create a Dockerfile

Inside the Medusa directory, create a Dockerfile to containerize Medusa using yarn.

Dockerfile:

Dockerfile | medusa-config.js | order-detail.tsx | route.ts ...\custom | route.ts ...\onboarding | doc

Dockerfile

```
# Use Node.js 18 Alpine image
FROM node:18-alpine

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and yarn.lock (if it exists)
COPY package.json yarn.lock ./

# Set the Yarn registry to Yarn's official registry (or a custom mirror)
RUN yarn config set registry https://registry.yarnpkg.com

# Install dependencies using Yarn
RUN yarn install

# Copy all the project files into the container
COPY . .

# Expose the default Medusa port
EXPOSE 9000

# Start Medusa
CMD ["yarn", "start"]
```

Step 4: Build the Docker Image

Create a Docker image from the Dockerfile with the following command:

docker build -t <image-name> .

This command builds the Docker image with the tag medusa-server.

Step 5: Create a docker-compose.yml

To simplify running the container with dependencies like a database, create a docker-compose.yml file:

docker-compose.yml:

Dockerfile | medusa-config.js | order-detail.tsx | route.ts ...\custom | route.ts ...\onboarding | docker-compose.yml

docker-compose.yml

```yaml
version: "3.8"

services:
  medusa:
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "9000:9000"
    environment:
      # Postgres Database connection details
      DATABASE_URL: postgres://postgres:Jashu@3337@db:5432/my_medusa_db
      REDIS_URL: redis://redis:6379
    depends_on:
      - db
      - redis

  db:
    image: postgres:13
    environment:
      POSTGRES_DB: my_medusa_db
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: Jashu@3337
    volumes:
      - postgres_data:/var/lib/postgresql/data

  redis:
    image: redis:6
    ports:
      - "6379:6379"

volumes:
  postgres_data:
    driver: local
```

This configuration starts three services:

- **Medusa**: The Medusa server built from the Dockerfile.
- **Postgres**: A PostgreSQL database for Medusa.
- **Redis**: A Redis instance.

Step 6: Run the Containers

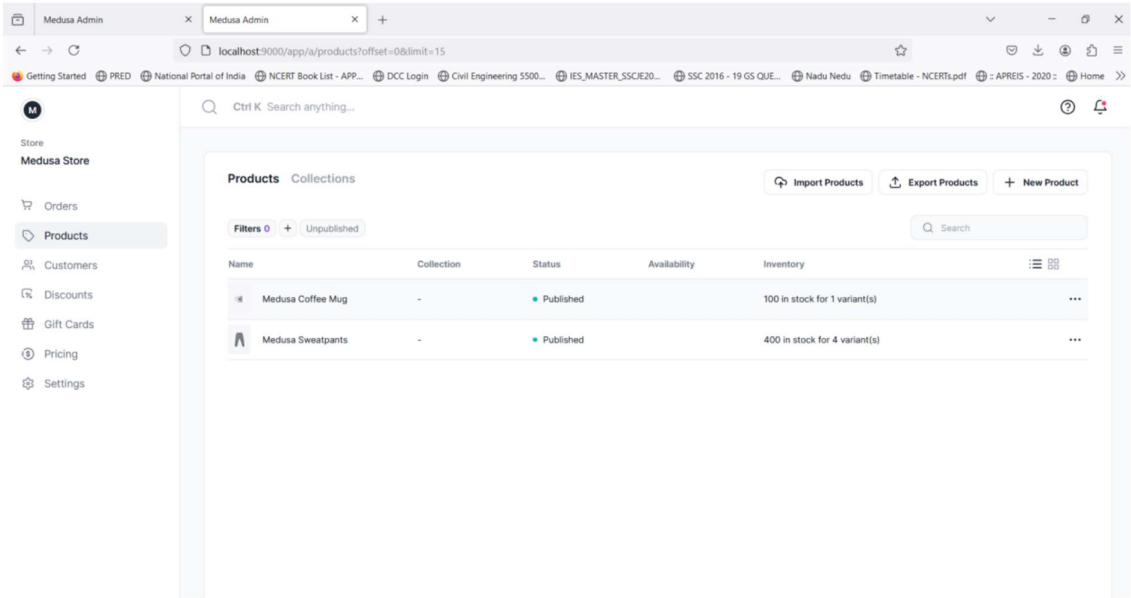
Start the containers with Docker Compose using this command:

docker-compose up

This command will start all the services defined in the docker-compose.yml file.

Step 7: Access Medusa

Once the containers are running, you can access Medusa at http://localhost:9000



Step 8: Access Medusa and Check Health

Check the application health by visiting the url http://localhost:9000/health