NAME:ALLURI JAYA KALYAN VARMA

EMAIL: 20pa1a0505@vishnu.edu.in

DATE:11-09-2024

TASK - 2 : Containerize and Run Medusa Locally using Docker

steps to containerize and run Medusa locally:

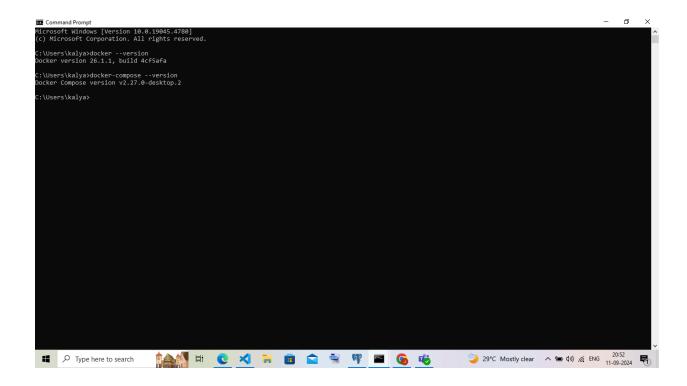
Step 1: Install Docker

- 1. Download Docker Desktop:
 - Visit the Docker website.
 - Choose the appropriate version for your operating system (Windows, macOS, or Linux).
- 2. Install Docker Desktop:
 - Follow the installation instructions provided on the Docker website or within the installer.
- 3. Verify Installation:

Open a terminal or command prompt and run

1.docker --version

2.docker-compose --version



Step 2: Create a Dockerfile

- 1. Set Up Your Project Directory:
 - Ensure you have a Medusa project set up. If not, you can initialize a new Medusa project following Medusa's documentation.

2. Create Dockerfile:

• In the root of your Medusa project directory, create a file named Dockerfile.

```
# Use an official Node.js runtime as a parent image

FROM node:18

# Set the working directory

WORKDIR /usr/src/app
```

```
# Copy the package.json and package-lock.json
COPY package*.json ./
# Install dependencies
RUN npm install
# Copy the rest of the application code
COPY . .
# Expose the port Medusa will run on
EXPOSE 9000
EXPOSE 7001
# Command to run the application
CMD [ "npx", "medusa", "develop" ]
```

Base Image:

• FROM node:18: Uses the official Node.js 18 image as the base.

Working Directory:

• WORKDIR /usr/src/app: Sets /usr/src/app as the working directory in the container.

Copy Dependency Files:

 COPY package*.json ./: Copies package.json and package-lock.json into the container's working directory.

Install Dependencies:

 RUN npm install: Installs the Node.js dependencies defined in package.json.

Copy Application Code:

 COPY . .: Copies the rest of the application code into the container's working directory.

Expose Ports:

- EXPOSE 9000
- EXPOSE 7001:container will use ports 9000 and 7001.

Run Command:

• CMD ["npx", "medusa", "develop"]: Runs the Medusa application in development mode using npx when the container starts.

Step 3: Create a Docker Compose File

- 1. Create docker-compose.yml:
 - In the same directory as your Dockerfile, create a file named docker-compose.yml.

```
version: '3.8'

services:
    medusa:
    build: .
    container_name: medusa
    environment:
        - JWT_SECRET=key
        - COOKIE_SECRET=key
        - DATABASE_TYPE=postgres
        -
DATABASE_URL=postgres://postgres:1234567890@postgres:5432/postgres
        - REDIS_URL=
```

```
ports:
      - "9000:9000"
      - "7001:7001"
    depends_on:
      - postgres
    networks:
      - medusa-network
  postgres:
    image: postgres:13
    container name: postgres
    environment:
      POSTGRES USER: postgres
      POSTGRES PASSWORD: 1234567890
      POSTGRES_DB: postgres
    ports:
      - "5432:5432"
    networks:
      - medusa-network
networks:
  medusa-network:
    driver: bridge
```

1. Version:

 version: '3.8': Specifies the version of the Docker Compose file format being used.

2. Services:

- Defines the services (containers) that will be part of this Docker Compose setup.
- Medusa Service:
 - build: .: Builds the Medusa service image from the Dockerfile in the current directory.

- container_name: medusa: Names the container medusa.
- environment: Sets environment variables needed for the Medusa application.
 - JWT_SECRET=key: Secret key for JSON Web Token.
 - COOKIE_SECRET=key: Secret key for cookies.
 - DATABASE_TYPE=postgres: Specifies the database type as PostgreSQL.
 - DATABASE_URL=postgres://postgres:123456 7890@postgres:5432/postgres: Connection URL for the PostgreSQL database.
 - REDIS_URL=: Empty URL for Redis; not currently used.
- ports: Maps ports on the host to ports on the container.
 - 9000:9000: Maps host port 9000 to container port 9000.
 - 7001:7001: Maps host port 7001 to container port 7001.
- depends_on: Specifies that the Medusa service depends on the PostgreSQL service being started first.
- networks: Connects the service to the medusa-network network.
- Postgres Service:
 - image: postgres:13: Uses the PostgreSQL 13 image from Docker Hub.
 - container_name: postgres: Names the container postgres.
 - environment: Sets environment variables for PostgreSQL.
 - POSTGRES_USER=postgres: Sets the PostgreSQL user.
 - POSTGRES_PASSWORD=1234567890: Sets the PostgreSQL password.
 - POSTGRES_DB=postgres: Sets the name of the default database.
 - ports: Maps ports on the host to ports on the container.
 - 5432:5432: Maps host port 5432 to container port 5432.

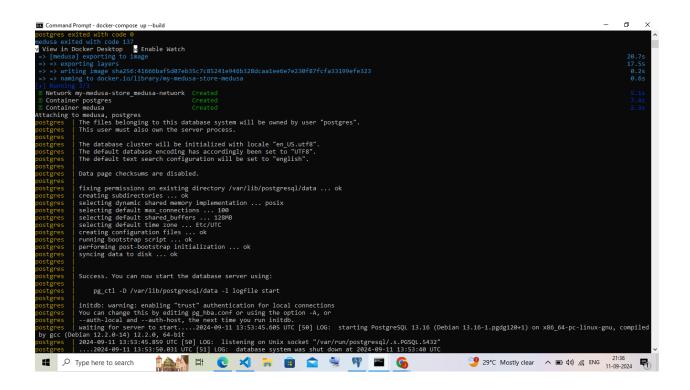
networks: Connects the service to the medusa-network network.

3. Networks:

- medusa-network: Defines a custom network for the services.
 - driver: bridge: Uses the bridge network driver, which is the default driver for Docker networks and allows containers to communicate with each other.
- The configuration sets up two services: Medusa (a web application) and PostgreSQL (a database).
- Medusa depends on PostgreSQL and both services are connected via a custom bridge network.
- Ports are mapped to expose services to the host machine, and environment variables are used for configuration.

Step 4: Build and Run Your Containers

- docker-compose up --build builds or rebuilds the Docker images for your services as defined in docker-compose.yml
- After building, it starts the containers and sets up networking and port mappings.
- This command starts the container and runs Medusa, mapping it to port 9000 on your localhost.



```
The commond frompt - decknormogre up -build

The commond frompt - decknormogre up -build

The commond frompt - decknormogree up -build

The commond from the commond - decknormogree up -build

The commond frompt - decknormogree up -build

The commond frompt - decknormogree up -build

The commond frompt - decknormogree up -build

The commond from the commond - decknormogree up -build

The commond frompt - decknormogree up -build

The commond frompt - decknormogree up -build

The commond from the commond - decknormogree up -build

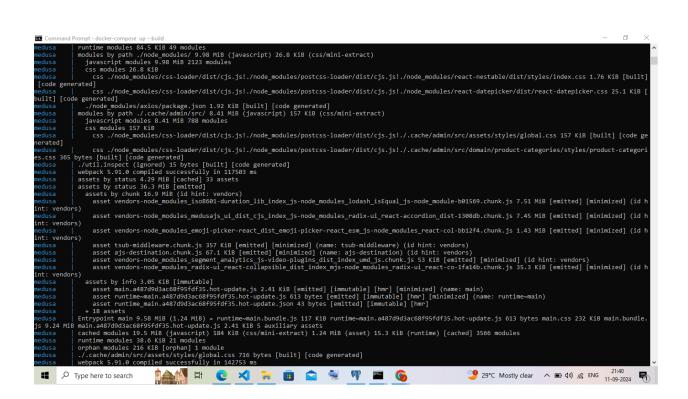
The commond from the commond - decknormogree up -build

The commond from the commond - decknormogree up -build

The commond frompt - decknormogree up -build

The commond from the commond - decknormogree up -build

The commond from the commond - decknormogree - decknormog
```



Step 5: Access Medusa

1. Open a Web Browser:

- Go to http://localhost:9000.
- You should see Medusa running and be able to interact with it through the browser.

