Here's the docker-compose.yml file for your **Sales Management System**, including all four microservices and their configurations. Each service is connected through a shared Docker network to facilitate communication.

yaml

Copy code

version: '3.8'

services:

customer-service:

build:

context: ./customer-service

ports:

- "8081:8080"

networks:

- sales-network

environment:

- SPRING\_PROFILES\_ACTIVE=prod

- DATABASE\_HOST=customer-db

- DATABASE\_PORT=3306

depends\_on:

- customer-db

product-catalog-service:

build:

context: ./product-catalog-service

ports:

- "8082:8080"

networks:

- sales-network

environment:

- SPRING\_PROFILES\_ACTIVE=prod

- DATABASE\_HOST=product-db

- DATABASE\_PORT=3306

depends\_on:

- product-db

order-management-service:

build:

context: ./order-management-service

ports:

- "8083:8080"

networks:

- sales-network

environment:

- SPRING\_PROFILES\_ACTIVE=prod

- DATABASE\_HOST=order-db

- DATABASE\_PORT=3306

depends\_on:

- order-db

- product-catalog-service

- payment-service

payment-service:

build:

context: ./payment-service

ports:

- "8084:8080"

networks:

- sales-network

environment:

- SPRING\_PROFILES\_ACTIVE=prod

- DATABASE\_HOST=payment-db

- DATABASE\_PORT=3306

depends\_on:

- payment-db

# Databases for each microservice

customer-db:

image: mysql:8

container\_name: customer-db

networks:

- sales-network

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: customerdb

MYSQL\_USER: user

MYSQL\_PASSWORD: password

ports:

- "3307:3306"

product-db:

image: mysql:8

container\_name: product-db

networks:

- sales-network

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: productdb

MYSQL\_USER: user

MYSQL\_PASSWORD: password

ports:

- "3308:3306"

order-db:

image: mysql:8

container\_name: order-db

networks:

- sales-network

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: orderdb

MYSQL\_USER: user

MYSQL\_PASSWORD: password

ports:

- "3309:3306"

payment-db:

image: mysql:8

container\_name: payment-db

networks:

- sales-network

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: paymentdb

MYSQL\_USER: user

MYSQL\_PASSWORD: password

ports:

- "3310:3306"

networks:

sales-network:

driver: bridge

**Explanation:**

1. **Microservices**:
   * Each service (customer-service, product-catalog-service, order-management-service, and payment-service) has:
     + A **build context** to use the corresponding Dockerfile for the service.
     + An assigned **port** for external access.
     + Environment variables to configure the application, such as database connection details.
     + depends\_on to ensure dependencies like databases or other services are started first.
2. **Databases**:
   * Each microservice uses a separate MySQL database.
   * Databases are isolated but connected through the sales-network.
3. **Networking**:
   * All services and databases are connected through a custom Docker bridge network (sales-network) for inter-service communication.
4. **Environment Variables**:
   * SPRING\_PROFILES\_ACTIVE=prod activates the production profile in Spring Boot.
   * DATABASE\_HOST and DATABASE\_PORT specify the database connections.

**Usage:**

1. Place this docker-compose.yml in your project root directory.
2. Make sure each microservice has a Dockerfile in its respective folder.
3. Run the following command to start all services:

bash

Copy code

docker-compose up --build

1. Verify that all services and databases are running.

Let me know if you need help configuring specific services or testing the setup! 🚀