Hướng dẫn chi tiết triển khai 4 Microservices (User, Product, Order, Payment)

Tài liệu này hướng dẫn cách xây dựng và triển khai hệ thống Microservices bao gồm 4 dịch vụ: User, Product, Order, Payment. Các dịch vụ được viết bằng Node.js (Express) và MongoDB, chạy bằng Docker Compose.

# 1. Cấu trúc dự án

microservices-demo/  
│── docker-compose.yml  
│  
├── user-service/  
│ ├── Dockerfile  
│ ├── server.js  
│ └── package.json  
│  
├── product-service/  
│ ├── Dockerfile  
│ ├── server.js  
│ └── package.json  
│  
├── order-service/  
│ ├── Dockerfile  
│ ├── server.js  
│ └── package.json  
│  
└── payment-service/  
 ├── Dockerfile  
 ├── server.js  
 └── package.json

# 2. File docker-compose.yml

File docker-compose.yml dùng để khởi chạy toàn bộ hệ thống với 4 service và 4 MongoDB tương ứng.

version: '3.8'  
services:  
 user-service:  
 build: ./user-service  
 ports:  
 - "4001:4001"  
 depends\_on:  
 - user-mongo  
 environment:  
 - MONGO\_URL=mongodb://user-mongo:27017/users  
  
 product-service:  
 build: ./product-service  
 ports:  
 - "4002:4002"  
 depends\_on:  
 - product-mongo  
 environment:  
 - MONGO\_URL=mongodb://product-mongo:27017/products  
  
 order-service:  
 build: ./order-service  
 ports:  
 - "4003:4003"  
 depends\_on:  
 - order-mongo  
 environment:  
 - MONGO\_URL=mongodb://order-mongo:27017/orders  
  
 payment-service:  
 build: ./payment-service  
 ports:  
 - "4004:4004"  
 depends\_on:  
 - payment-mongo  
 environment:  
 - MONGO\_URL=mongodb://payment-mongo:27017/payments  
  
 user-mongo:  
 image: mongo:5  
 ports:  
 - "27017"  
  
 product-mongo:  
 image: mongo:5  
 ports:  
 - "27018"  
  
 order-mongo:  
 image: mongo:5  
 ports:  
 - "27019"  
  
 payment-mongo:  
 image: mongo:5  
 ports:  
 - "27020"

# 3. Ví dụ User Service

package.json:

{  
 "name": "user-service",  
 "version": "1.0.0",  
 "main": "server.js",  
 "scripts": {  
 "start": "node server.js"  
 },  
 "dependencies": {  
 "express": "^4.18.2",  
 "mongoose": "^7.0.3",  
 "body-parser": "^1.20.2"  
 }  
}

server.js:

const express = require("express");  
const mongoose = require("mongoose");  
const bodyParser = require("body-parser");  
  
const app = express();  
app.use(bodyParser.json());  
  
mongoose.connect(process.env.MONGO\_URL, { useNewUrlParser: true, useUnifiedTopology: true });  
  
const User = mongoose.model("User", {  
 name: String,  
 email: String,  
 password: String  
});  
  
app.post("/users", async (req, res) => {  
 const user = new User(req.body);  
 await user.save();  
 res.json(user);  
});  
  
app.get("/users/:id", async (req, res) => {  
 const user = await User.findById(req.params.id);  
 res.json(user);  
});  
  
app.listen(4001, () => console.log("User Service running on port 4001"));

Dockerfile:

FROM node:18  
WORKDIR /app  
COPY package.json ./  
RUN npm install  
COPY . .  
CMD ["npm", "start"]

# 4. Product, Order, Payment Service

Các service còn lại có cấu trúc tương tự User Service, chỉ khác model dữ liệu và API.  
- Product Service: quản lý sản phẩm.  
- Order Service: quản lý đơn hàng, gọi User & Product để kiểm tra.  
- Payment Service: xử lý thanh toán.

# 5. Chạy hệ thống

Sau khi tạo đầy đủ file, chạy lệnh sau:  
  
docker-compose up --build  
  
Hệ thống sẽ chạy tại các port:  
- User Service: http://localhost:4001  
- Product Service: http://localhost:4002  
- Order Service: http://localhost:4003  
- Payment Service: http://localhost:4004

# 6. Giao tiếp giữa các service

Order Service sẽ gọi API của User và Product để kiểm tra trước khi tạo đơn hàng, sau đó gọi Payment Service để thanh toán.