

Bài tập thực hành - Chương 5

Tiêu đề: Điều phối Ứng dụng React với Docker Compose và Nginx

Mục tiêu: Viết các file Docker Compose để quản lý cả môi trường development và production. Mở rộng cấu hình production để thêm một Nginx reverse proxy, tạo ra một kiến trúc ứng dụng web hai-tier thực tế.

Phần A: Cài đặt Docker Compose và Thiết lập Môi trường Dev

1. Kiểm tra Docker Compose:

Bên trong máy ảo Vagrant, Docker Compose đã được cài đặt như một plugin của Docker ở Buổi 3.

```
docker compose version
```

Kết quả mong đợi: Hiển thị phiên bản, ví dụ Docker Compose version v2.12.2.

2. Tạo file Docker Compose:

Trong thư mục gốc của dự án (`/var/www/project/`), tạo một file mới:

```
touch docker-compose.yml
```

3 Xác định Môi trường phát triển (Development):

Mở file `docker-compose.yml` và thêm vào nội dung sau để định nghĩa môi trường phát triển.

```
version: '3.8'

services:
  # Dịch vụ cho môi trường phát triển
  dev:
    # Sử dụng image đã build ở buổi 4 hoặc để compose build
    build:
      context: .
      dockerfile: Dockerfile.dev
    container_name: react_app_dev
    ports:
      - "3000:3000"
    volumes:
      # Mount code để có hot-reloading
      - ./client:/app
      # Tip: Ngăn node_modules trên host ghi đè lên node_modules trong container
      - /app/node_modules
```

4. Khởi chạy Môi trường Phát triển:

Chạy lệnh sau từ thư mục gốc dự án:

```
docker compose up dev --build
```

- dev: Chỉ định chỉ khởi chạy dịch vụ có tên dev.
- --build: Buộc build lại image.

5. Kiểm tra:

Mở trình duyệt trên máy thật, truy cập <http://localhost:3000>. Ứng dụng phải hoạt động. Sửa code trong `src/client` để kiểm tra hot-reloading.

6. Dừng:

Trong terminal, nhấn `Ctrl+C`. Hoặc từ một terminal khác, chạy:

```
docker compose down
```

Phần B: Thiết lập Môi trường Production với Nginx

Đây là phần cốt lõi. Sinh viên sẽ tạo một kiến trúc production với 2 service:

1. Tạo cấu trúc thư mục cho Nginx:

Trong thư mục gốc dự án, tạo một thư mục `nginx` và một file cấu hình bên trong nó.

```
mkdir nginx
touch nginx/default.conf
```

2. Viết file cấu hình Nginx (`nginx/default.conf`):

Đây là file cấu hình cho reverse proxy. Nginx sẽ phản hồi ở cổng 80 và chuyển tiếp (proxy) tất cả các yêu cầu đến dịch vụ `app` ở cổng 5000.

```
server {
    listen 80;
    server_name localhost;

    location / {
        # Tên "app" phải trùng với tên dịch vụ React trong docker-compose.yml
        proxy_pass http://app:5000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

3. Cập nhật `docker-compose.yml` cho Production:

Thêm vào 2 dịch vụ `app` và `nginx` vào file `docker-compose.yml`.

```
# ... (giữ lại phần version và services ở trên) ...
# Thêm 2 dịch vụ mới này vào dưới service `dev`

# Dịch vụ cho ứng dụng React production
app:
  build:
    context: .
    dockerfile: Dockerfile # Sử dụng Dockerfile production
  container_name: react_app_prod
  # KHÔNG cần 'ports' ở đây nữa, vì Nginx sẽ xử lý việc tiếp xúc với bên ngoài

# Dịch vụ cho Nginx Reverse Proxy
nginx:
  image: nginx:1.21-alpine # Dùng image Nginx chính thức
  container_name: nginx_proxy
  ports:
    - "80:80" # Mở cổng 80 ra thế giới bên ngoài
  volumes:
    # Mount file cấu hình Nginx của chúng ta vào trong container
    - ./nginx/default.conf:/etc/nginx/conf.d/default.conf
  depends_on:
    - app # Đảm bảo service 'app' khởi động trước 'nginx'
```

4. Khởi chạy Môi trường Production:

Bây giờ, Sinh viên sẽ khởi chạy cả hai dịch vụ `app` và `nginx`.

```
docker compose up -d --build app nginx
```

- `-d`: Chạy ở chế độ nền.
- `app nginx`: Chỉ định khởi chạy 2 dịch vụ này.

5 Kiểm tra kiến trúc hoàn chỉnh:

- Chạy `docker compose ps` để xem cả hai container `react_app_prod` và `nginx_proxy` đang chạy.
- Kiểm tra log của từng dịch vụ: `docker compose logs app` và `docker compose logs nginx`.

- Mở trình duyệt trên máy thật và truy cập <http://localhost> (cổng 80 là mặc định).
- **Kết quả mong đợi:** Ứng dụng React sẽ hiện lên, nhưng lần này yêu cầu đã đi qua Nginx trước khi đến ứng dụng.

Phần C: Quản lý và Dọn dẹp

1. Xem tất cả các dịch vụ trong dự án:

Nếu chạy `docker compose up -d` mà không chỉ định tên dịch vụ, nó sẽ cố gắng chạy tất cả (`dev`, `app`, `nginx`).

2. Dừng toàn bộ ứng dụng:

Lệnh này sẽ dừng và xóa tất cả các container, network được định nghĩa trong file `docker-compose.yml`.

```
docker compose down
```

Chạy `docker ps -a` để xác nhận các container đã bị xóa.

3. Xác nhận các thay đổi:

Thoát khỏi máy ảo và xác nhận các file đã hoàn thiện.

```
# git add docker-compose.yml nginx/  
# git commit -m "Feat: Implement dev and prod environments with Docker Compose and Nginx"
```

