



# HƯỚNG DẪN TRIỂN KHAI ỨNG DỤNG ĐA DỊCH VỤ LÊN VPS BẰNG DOCKER COMPOSE

Hướng dẫn này áp dụng cho dự án **OmniMer Health** (Backend Node.js và AI Service Python) đã được tối ưu hóa với các Dockerfile và docker-compose.yml (bỏ DB/Redis local).

## 1. Pha 1: Chuẩn bị Môi trường VPS

Bạn cần đăng nhập vào VPS (thường qua SSH) và cài đặt Docker Engine cùng Docker Compose.

### 1.1. Cài đặt Docker Engine (Linux/Ubuntu)

(Áp dụng cho hệ điều hành Ubuntu/Debian. Nếu dùng OS khác, vui lòng tham khảo tài liệu Docker.)

# 1. Cập nhật gói hệ thống

```
sudo apt update
```

```
sudo apt upgrade -y
```

# 2. Cài đặt các gói cần thiết

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common -y
```

# 3. Thêm GPG key chính thức của Docker

```
curl -fsSL
```

```
[https://download.docker.com/linux/ubuntu/gpg](https://download.docker.com/linux/ubuntu/gp g) | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

# 4. Thêm kho lưu trữ Docker

```
echo "deb [arch=$(dpkg --print-architecture)
```

```
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
```

```
[https://download.docker.com/linux/ubuntu](https://download.docker.com/linux/ubuntu)
```

```
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

# 5. Cài đặt Docker Engine

```
sudo apt update
```

```
sudo apt install docker-ce docker-ce-cli containerd.io -y
```

# 6. Thêm user vào nhóm docker để chạy lệnh không cần sudo

```
sudo usermod -aG docker $USER
```

```
# **Khởi động lại phiên SSH** hoặc chạy lệnh "newgrp docker" để áp dụng ngay lập tức
```

## 1.2. Cài đặt Docker Compose

```
# Tải về phiên bản Docker Compose ổn định (ví dụ: v2.20.2)
# Bạn nên kiểm tra phiên bản mới nhất trên trang GitHub của Docker Compose
sudo curl -L
"[https://github.com/docker/compose/releases/download/v2.20.2/docker-compose-$(uname)-
https://github.com/docker/compose/releases/download/v2.20.2/docker-compose-$(uname)-
-s)-$(uname -m)" -o /usr/local/bin/docker-compose

# Cấp quyền thực thi
sudo chmod +x /usr/local/bin/docker-compose

# Tạo symlink (tùy chọn)
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

# Kiểm tra phiên bản
docker-compose --version
```

## 2. Pha 2: Chuyển giao Mã nguồn lên VPS

### Phương pháp 2.1: Dùng Git (Được khuyến nghị)

1. **Trên máy tính cục bộ:** Đẩy toàn bộ mã nguồn lên kho Git.

2. **Trên VPS:**

```
# Cài đặt Git nếu chưa có
sudo apt install git -y
```

```
# Clone repository
git clone <URL_repository_của_bạn>
cd <ten_thu_mục_du_an>
```

3. **Tạo file môi trường (.env):**

```
# Ví dụ:
nano omnimer_health_server/.env
# Điền các biến môi trường cho Cloud DB/Redis, v.v.
```

### Phương pháp 2.2: Dùng SCP/SFTP

Sử dụng các công cụ như WinSCP (Windows) hoặc lệnh scp (Linux/macOS) để copy toàn bộ

thư mục dự án lên VPS.

### 3. **Pha 3: Triển khai và Chạy ứng dụng**

Đây là bước cuối cùng để Docker đọc docker-compose.yml, xây dựng image và khởi chạy các container.

1. **Di chuyển đến thư mục gốc của dự án** (chứa file docker-compose.yml):

```
cd /path/to/DACN_OMNIMER_HEALTH
```

2. **Build và Khởi chạy ứng dụng:**

# Lệnh build và chạy:

```
docker-compose up -d --build
```

- o up: Khởi động các services.
- o -d: Chạy services ở chế độ nền (detached mode).
- o --build: Bắt buộc build lại các image từ đầu (rất quan trọng khi triển khai lần đầu hoặc cập nhật code).

3. **Kiểm tra Trạng thái:**

```
docker-compose ps
```

Kiểm tra cột State, tất cả các dịch vụ nên ở trạng thái Up.

4. **Xem Logs (Kiểm tra lỗi):**

```
docker-compose logs -f
```

Sử dụng lệnh này để xem nhật ký của tất cả các container trong thời gian thực, giúp bạn gỡ lỗi nếu có vấn đề.

5. Cập nhật code và triển khai lại:

Khi bạn cập nhật code (ví dụ: git pull trên VPS), bạn chỉ cần chạy lại lệnh:

```
docker-compose up -d --build
```

Docker Compose sẽ chỉ build lại image của dịch vụ có code thay đổi.

## **Bước Bổ Sung: Cấu hình Reverse Proxy (Quan trọng)**

Mặc dù các service của bạn chạy trên các cổng như 8000 (Backend) và 8888 (AI), bạn **không nên** mở trực tiếp các cổng này ra Internet. Thay vào đó, bạn nên sử dụng **Nginx** (hoặc Caddy) làm **Reverse Proxy** để:

- **Bảo mật:** Chạy các service sau một lớp bảo vệ.
- **SSL/TLS:** Dễ dàng cấp phát và quản lý chứng chỉ HTTPS (Let's Encrypt).
- **Cổng mặc định:** Cho phép người dùng truy cập qua cổng 80/443 tiêu chuẩn.