

# Báo cáo về Khóa học Docker + Microservice

---

Yêu cầu: Sinh viên xem video và tìm hiểu về Docker + GitHub CI/CD

**Họ và tên:** Phan Tài Nguyên  
**MSSV:** 3121411151

## Tóm tắt nội dung khóa học

### Section 1: Introduction to Docker

- 1.1 - What is Docker and Why Use It
- 1.2 - Docker vs. Virtual Machines
- 1.3 - Docker Terminology and Architecture
- 1.4 - Setting Up Your Docker Environment - Windows
- 1.5 - Setting Up Your Docker Environment - Ubuntu

### Section 2: Docker Fundamentals

- 2.1 - Working with Docker Images
- 2.2 - Managing Containers
- 2.3 - Network in Docker
- 2.4 - Persistent Storage and Volumes

### Section 3: Docker Compose and Multi-Container Application

- 3.1 - Introduction to Docker Compose and Services
- 3.2 - Introduction to Docker Compose Hands-on
- 3.3 - Multi-Container Application Hands-on
- 3.4 - Scalability and Load Balancing Hands-on

### Section 4: Docker for Development

- 4.1 - Docker for Development - Overview
- 4.2 - Setting up a Development Environment with Docker Hands-on
- 4.3 - Best Practice for Dockerfile
- 4.4 - Debugging Containers

## **Section 5: Security in Docker**

- 5.1 - Introduction to Security in Docker
- 5.2 - Running Docker Containers as a Non-root User
- 5.3 - Scanning Docker Images for Vulnerabilities Hands-On
- 5.4 - Setting Up Network Policies to Control Traffic Between Containers Hands-On

## **Section 6: Advanced Docker Techniques**

- 6.1 - Introduction to Docker Swarm
- 6.2 - Introduction to Kubernetes
- 6.3 - CICD with Docker and GitHub Actions Overview
- 6.4 - CICD with Docker and GitHub Actions Hands-On

## **Section 7: Docker and Microservices**

- 7.1 - Introduction to Microservices and Docker
- 7.2 - Deploying Microservices with Docker Hands-On
- 7.3 - The Importance of Monitoring and Logging
- 7.4 - Docker Monitoring and Logging with ELK Stack Hands-On

## **Nội dung học được về Docker và GitHub CI/CD**

### **1. Docker**

- Docker là một nền tảng ảo hóa cấp hệ điều hành, cho phép đóng gói, phân phối và chạy ứng dụng một cách nhất quán.
- So sánh Docker với máy ảo truyền thống cho thấy Docker nhẹ hơn, khởi động nhanh và tối ưu hóa tài nguyên hệ thống.
- Các thành phần cơ bản gồm: Docker Engine, Docker Images, Containers, Docker Compose và Dockerfile.
- Những lưu ý về bảo mật: chạy container dưới người dùng không phải root và quét lỗ hổng trong Docker Images.

### **2. GitHub CI/CD**

- GitHub Actions là công cụ tích hợp CI/CD có sẵn trong GitHub, giúp tự động hóa quy trình build, test và deploy.
- Tích hợp Docker vào quy trình CI/CD cho phép tự động xây dựng (build) và đẩy (push) Docker Images lên registry.
- Workflow được cấu hình thông qua file YAML, mô tả các bước (jobs) cần thực hiện khi có thay đổi trên repository.
- Việc tích hợp Docker và GitHub CI/CD giúp giảm thiểu sai sót, tăng tốc quá trình phát triển và triển khai sản phẩm.

## **Ý kiến cá nhân**

Qua quá trình học, em nhận thấy Docker đã giúp đơn giản hóa việc đóng gói và triển khai ứng dụng, đồng thời giảm bớt sự phức tạp của môi trường phát triển. Việc tích hợp GitHub Actions cho CI/CD đã mở ra một quy trình tự động hóa hiệu quả, góp phần nâng cao chất lượng sản phẩm và giảm thiểu lỗi trong quá trình triển khai. Em tin rằng những kiến thức này sẽ hữu ích cho các dự án thực tế trong tương lai.

## **Kết luận**

Khóa học Docker + Microservice đã cung cấp những kiến thức nền tảng và nâng cao về Docker, từ việc cài đặt đến triển khai ứng dụng trong môi trường production. Đặc biệt, phân tích hợp CI/CD với GitHub Actions là một trong những điểm nhấn quan trọng, mở ra cơ hội tự động hóa toàn bộ quy trình phát triển phần mềm. Em cảm ơn thầy đã cung cấp khóa học này.