

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN  
THÔNG**



**BÁO CÁO THỰC TẬP CƠ SỞ**

**TÌM HIỂU VỀ DOCKER**

**Giảng viên hướng dẫn : Kim Ngọc Bách**

**Họ và tên : Ngô Vũ Minh Quý**

**Mã sinh viên : B22DCVT427**

**Lớp : E22CQCN02-B**

# ***MỤC LỤC***

**I. Giới thiệu về Docker**

**II. Ứng dụng Docker trong phát triển phần mềm**

**III. Cài đặt Docker**

**IV. Các thao tác cơ bản với Docker**

**V. Đánh giá và kết luận**

# I. Giới thiệu về Docker



Docker là một nền tảng mã nguồn mở cho phép đóng gói, phân phối và chạy ứng dụng một cách nhẹ nhàng và linh hoạt thông qua **các container**. Khác với máy ảo truyền thống vốn tiêu tốn nhiều tài nguyên và cần hệ điều hành riêng biệt, container chỉ chứa ứng dụng và các thành phần phụ thuộc, chạy trực tiếp trên nhân hệ điều hành của máy chủ.

Các đặc điểm nổi bật của Docker gồm:

- **Tính nhẹ (lightweight):** Các container chia sẻ cùng một nhân hệ điều hành nên khởi động nhanh và tiêu tốn ít tài nguyên.
- **Tính di động (portable):** Container có thể chạy đồng nhất trên nhiều môi trường: máy tính cá nhân, máy chủ, cloud.
- **Tính nhất quán:** Môi trường phát triển và sản xuất giống nhau 100%, giảm thiểu lỗi do khác biệt môi trường.

Docker bao gồm các thành phần chính:

- **Docker Engine:** Phần lõi giúp tạo và quản lý container.
- **Docker Image:** Ảnh tĩnh dùng để tạo container.
- **Docker Container:** Thực thể chạy thực tế của image.

- **Docker Hub:** Kho chia sẻ image trực tuyến.

## II. Ứng dụng Docker trong phát triển phần mềm

### 1. Lý do lựa chọn Docker

Docker ngày càng trở thành công cụ tiêu chuẩn trong DevOps và phát triển phần mềm nhờ vào các ưu điểm:

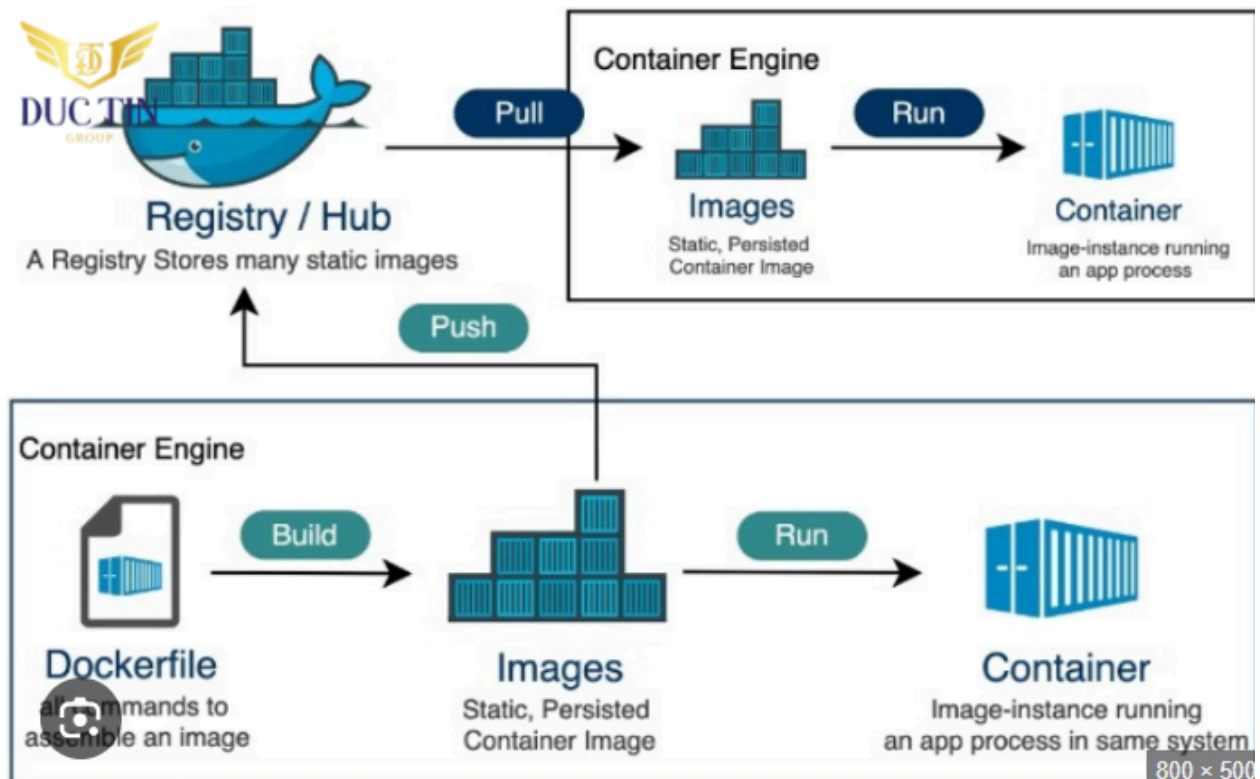
- **Triển khai dễ dàng:** Dễ đóng gói ứng dụng kèm môi trường cấu hình và triển khai tức thì lên máy chủ hoặc cloud.
- **Hỗ trợ phát triển microservices:** Mỗi dịch vụ có thể chạy trong một container riêng biệt, dễ bảo trì và mở rộng.
- **Tối ưu hiệu suất:** Ít tốn tài nguyên hơn so với VM truyền thống, tiết kiệm chi phí hạ tầng.
- **Hỗ trợ CI/CD:** Docker dễ tích hợp với các công cụ như Jenkins, GitHub Actions để tự động hóa kiểm thử và triển khai.

### 2. Ứng dụng thực tiễn của Docker

Docker đã và đang được áp dụng rộng rãi trong thực tế:

- **Môi trường phát triển:** Các lập trình viên có thể cấu hình môi trường phát triển chỉ bằng một file **Dockerfile** hoặc **docker-compose.yml**, giúp đồng bộ hóa môi trường trên nhiều máy khác nhau.
- **Cloud-native app:** Docker là nền tảng cốt lõi trong kiến trúc hiện đại như Kubernetes, giúp triển khai và quản lý các ứng dụng container hoá.
- **Tự động hóa kiểm thử:** Với Docker, có thể dễ dàng thiết lập môi trường kiểm thử tạm thời (temporary testing environments).
- **Triển khai nhanh chóng:** Do các image đã được build sẵn, container

có thể khởi chạy gần như tức thì.



### III. Cài đặt Docker

#### Bước 1: Tải Docker Desktop

- ❖ Truy cập trang chính thức:  
<https://www.docker.com/products/docker-desktop/>
- ❖ Chọn phiên bản tương ứng với hệ điều hành và tải về.

#### Bước 2: Cài đặt

- ❖ Chạy file cài đặt và làm theo hướng dẫn.
- ❖ Trong quá trình cài đặt, bạn nên chọn sử dụng WSL2 (Windows Subsystem for Linux) để có hiệu năng tốt hơn.

#### Bước 3: Kiểm tra cài đặt

Mở Command Prompt hoặc Terminal và chạy:

```
Command Prompt
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>docker --version|
```

## Bước 4: Chạy thử container mẫu

```
Command Prompt
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>docker run hello-world|
```

## IV. Các thao tác cơ bản với Docker

Docker cung cấp nhiều thao tác mạnh mẽ để làm việc với container và image:

### +Tải image từ Docker Hub

```
Command Prompt
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>docker pull nginx|
```

### + Chạy container

```
Command Prompt
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>docker pull nginx
docker run -d -p 8080:80 ng
'docker' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\ASUS>docker run -d -p 8080:80 nginx|
```

### +Quản lý container:

```
bash
Sao chép  Chỉnh sửa

docker ps          # Xem danh sách container đang chạy
docker ps -a       # Xem tất cả container
docker stop <id>   # Dừng container
docker rm <id>     # Xóa container
```

## +Quản lý image

```
bash ✓ Đã sao chép ✎ Chỉnh sửa

docker images      # Liệt kê các image đã tải
docker rmi <image-id> # Xóa một image
```

## + Tạo Dockerfile đơn giản

```
Dockerfile Sao chép ✎ Chỉnh sửa

# Dockerfile cho ứng dụng Node.js
FROM node:18
WORKDIR /app
COPY . .
RUN npm install
CMD ["node", "index.js"]
```

## +Docker Compose

```
yaml Sao chép ✎ Chỉnh sửa

version: '3'
services:
  web:
    image: nginx
    ports:
      - "8080:80"
  db:
    image: mongo
```

## V.Đánh giá và kết luận

Qua quá trình tìm hiểu và thực hành với Docker, em nhận thấy rằng Docker không chỉ là một công cụ mạnh mẽ và hiện đại mà còn là một phần không thể thiếu trong quá trình phát triển phần mềm trong kỷ nguyên điện toán đám mây. Docker giúp giảm thiểu đáng kể thời gian và công sức trong việc thiết lập môi trường phát triển, triển khai ứng dụng, và hỗ trợ tối ưu hóa tài nguyên hệ thống so với các phương pháp truyền thống. Điều này cực kỳ quan trọng trong môi trường phát triển phần mềm hiện đại, nơi mà tính linh hoạt và hiệu suất là yếu tố quyết định.

Một trong những điểm mạnh lớn nhất của Docker là khả năng hỗ trợ mô hình microservices, giúp chia nhỏ các ứng dụng phức tạp thành các dịch vụ độc lập, dễ dàng bảo trì và mở rộng. Khi kết hợp với các công cụ như Kubernetes, Docker trở thành nền tảng lý tưởng cho việc triển khai các hệ thống có quy mô lớn và đòi hỏi tính sẵn sàng cao. Hơn nữa, Docker hỗ trợ mạnh mẽ cho quy trình CI/CD (Continuous Integration/Continuous Deployment), giúp tự động hóa các bước từ kiểm thử, xây dựng đến triển khai, giảm thiểu rủi ro và tăng tốc quá trình phát triển phần mềm.

Bên cạnh đó, Docker cũng mang lại lợi ích lớn về mặt chi phí và tài nguyên. Việc sử dụng container giúp tối ưu hóa việc sử dụng tài nguyên hệ thống, cho phép chạy nhiều ứng dụng trên cùng một máy chủ mà không làm giảm hiệu suất, điều này giúp giảm chi phí về hạ tầng và phần cứng. Docker cũng giúp các nhà phát triển dễ dàng tái tạo môi trường phát triển và kiểm thử, đảm bảo tính nhất quán từ môi trường này sang môi trường khác, giảm thiểu các vấn đề không mong muốn khi chuyển giao giữa các giai đoạn phát triển.

Trong thời đại công nghệ phát triển nhanh chóng, việc thành thạo Docker không chỉ là một lợi thế mà còn là một kỹ năng thiết yếu đối với sinh viên ngành công nghệ thông tin, kỹ sư phần mềm và các chuyên gia DevOps. Khả năng sử dụng Docker thành thạo giúp các cá nhân và đội nhóm phát triển phần mềm nhanh chóng, linh hoạt và hiệu quả hơn, đồng thời cũng tạo cơ hội cho họ đóng góp vào việc xây dựng các hệ thống phức tạp và quy mô lớn. Việc làm quen và nâng cao kỹ năng sử dụng Docker sẽ mở ra nhiều cơ hội trong nghề nghiệp, đặc biệt là trong các lĩnh vực đòi hỏi kỹ năng về DevOps, triển khai hệ thống đám mây, và các giải pháp công nghệ mới.



