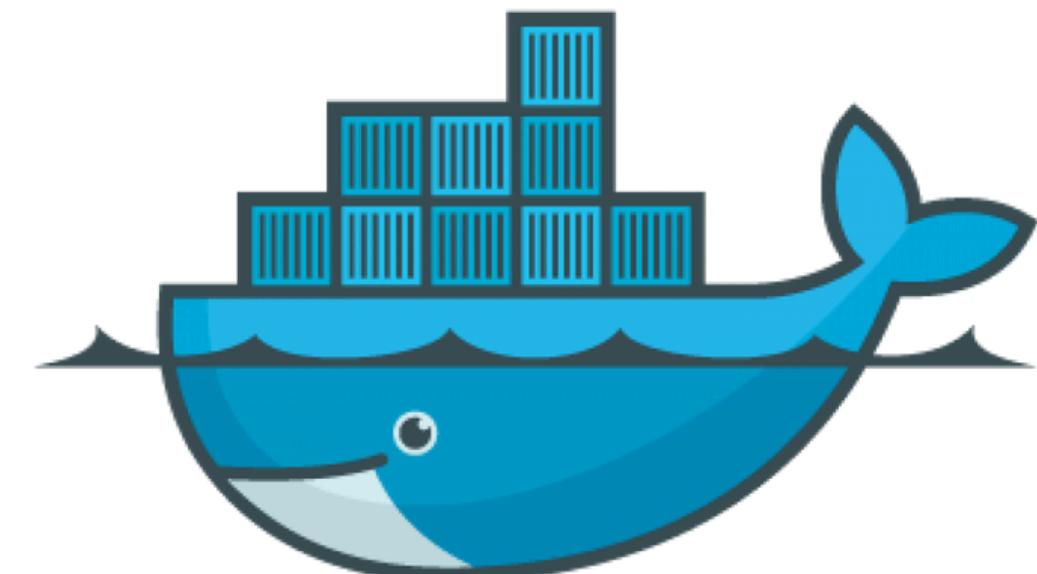


GIỚI THIỆU

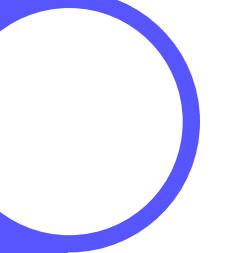
DOCKER

Team:
Anh Kiêm Đoàn,
Ninh x2, Tuấn Hoàng



docker

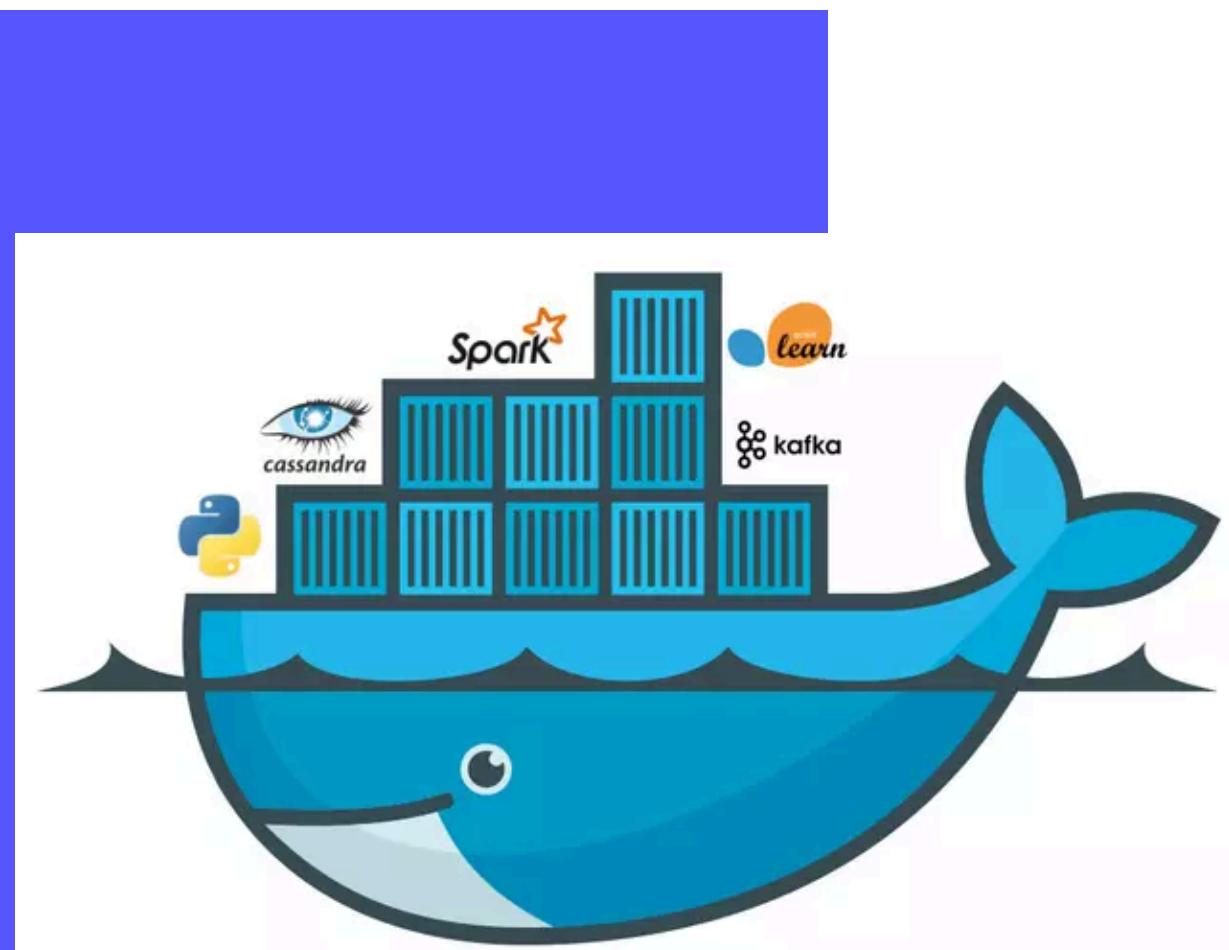
NỘI DUNG

- 
1. Giới thiệu
 2. Kiến trúc Docker
 3. Cách cài đặt Docker

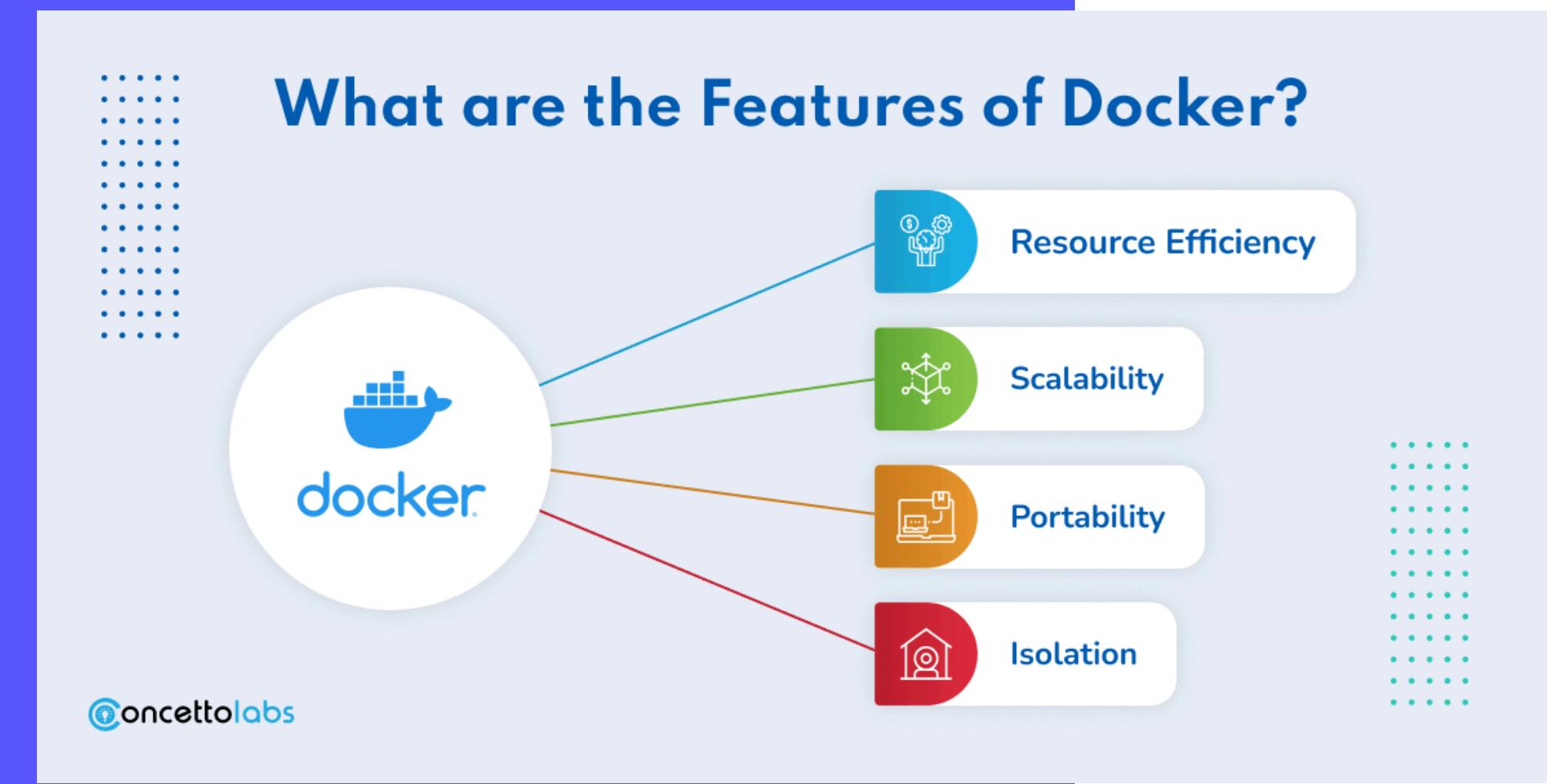
Docker là gì?

Docker là gì?

- Docker là một nền tảng để cung cấp cách để building, deploying và running ứng dụng dễ dàng hơn bằng cách sử dụng các containers (trên nền tảng ảo hóa).
- Được viết bằng Golang (trước là Python)

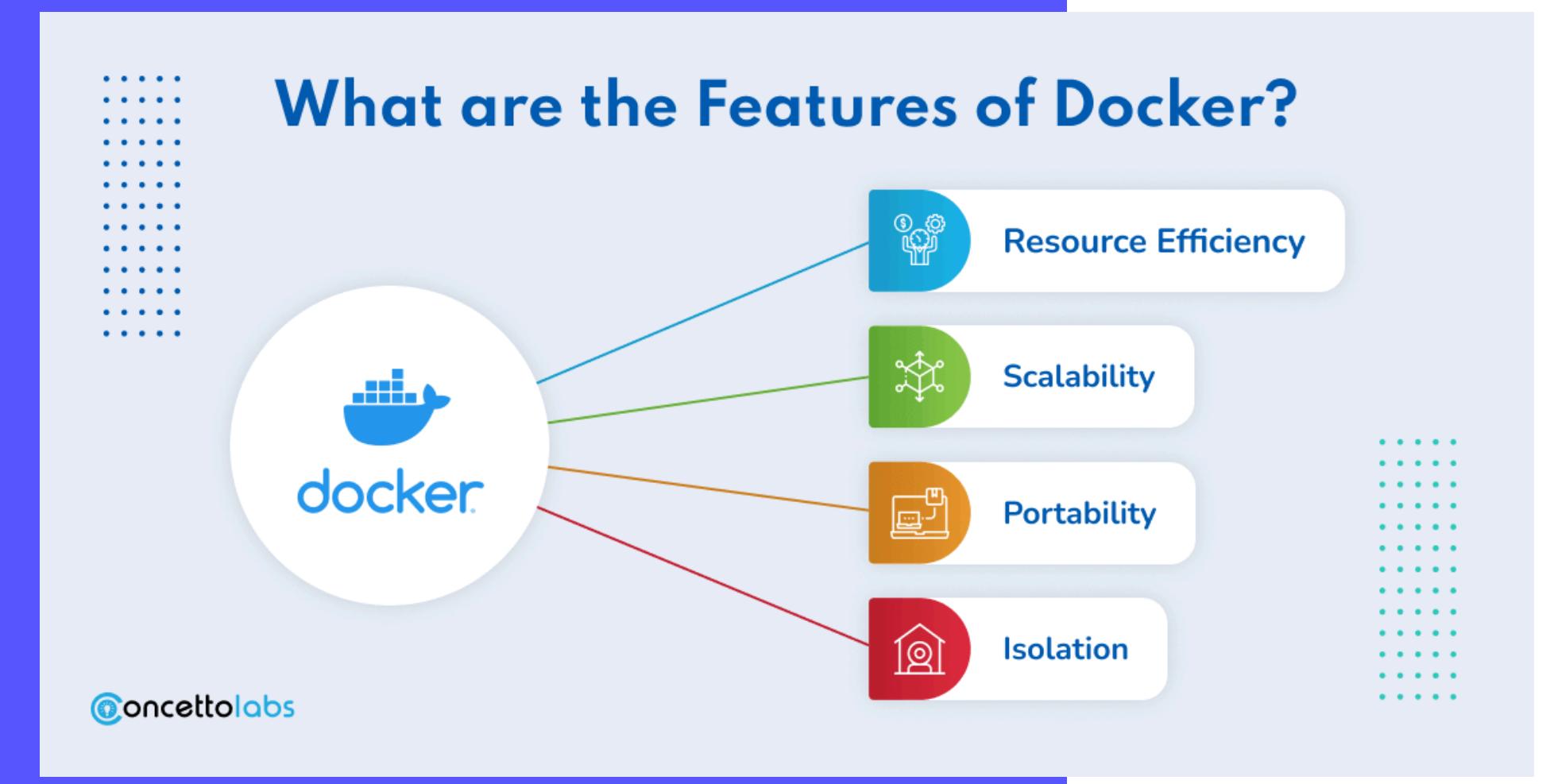


Tính năng của Docker?



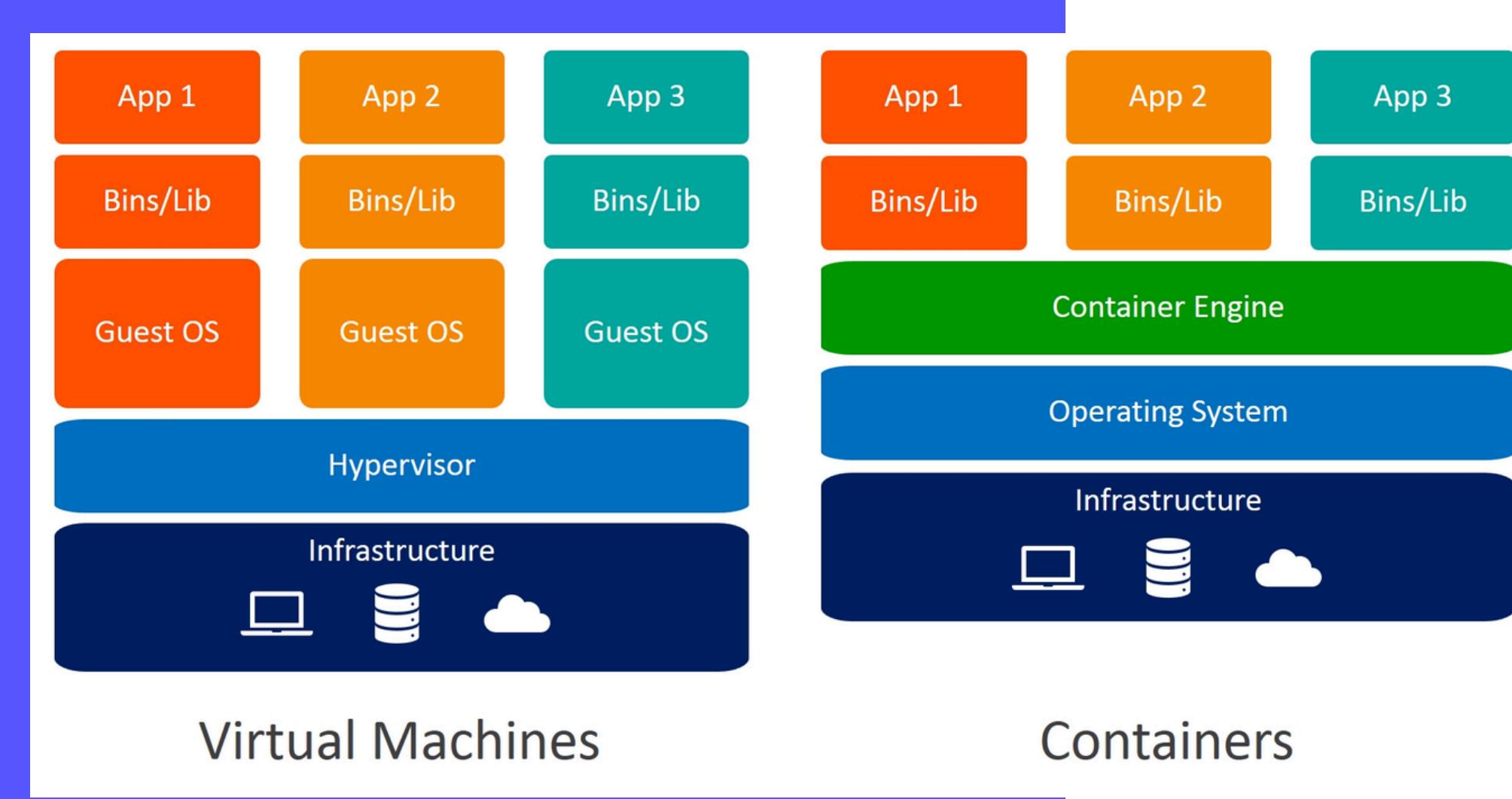
Tính năng của Docker?

- Tính tối ưu tài nguyên hệ thống
- Tính mở rộng
- Tính di động
- Tính cô lập



Khác biệt với máy ảo (VM)

- Container chia sẻ kernel của hệ điều hành máy chủ, nhẹ và linh hoạt hơn.



Lợi ích của Docker

Lợi ích của Docker

1. Docker start và stop chỉ trong vài giây, nhanh hơn nhiều so với máy ảo.
2. Có thể khởi chạy container trên môi hệ thống mà bạn muốn.
3. Container có thể build và loại bỏ nhanh hơn máy ảo.

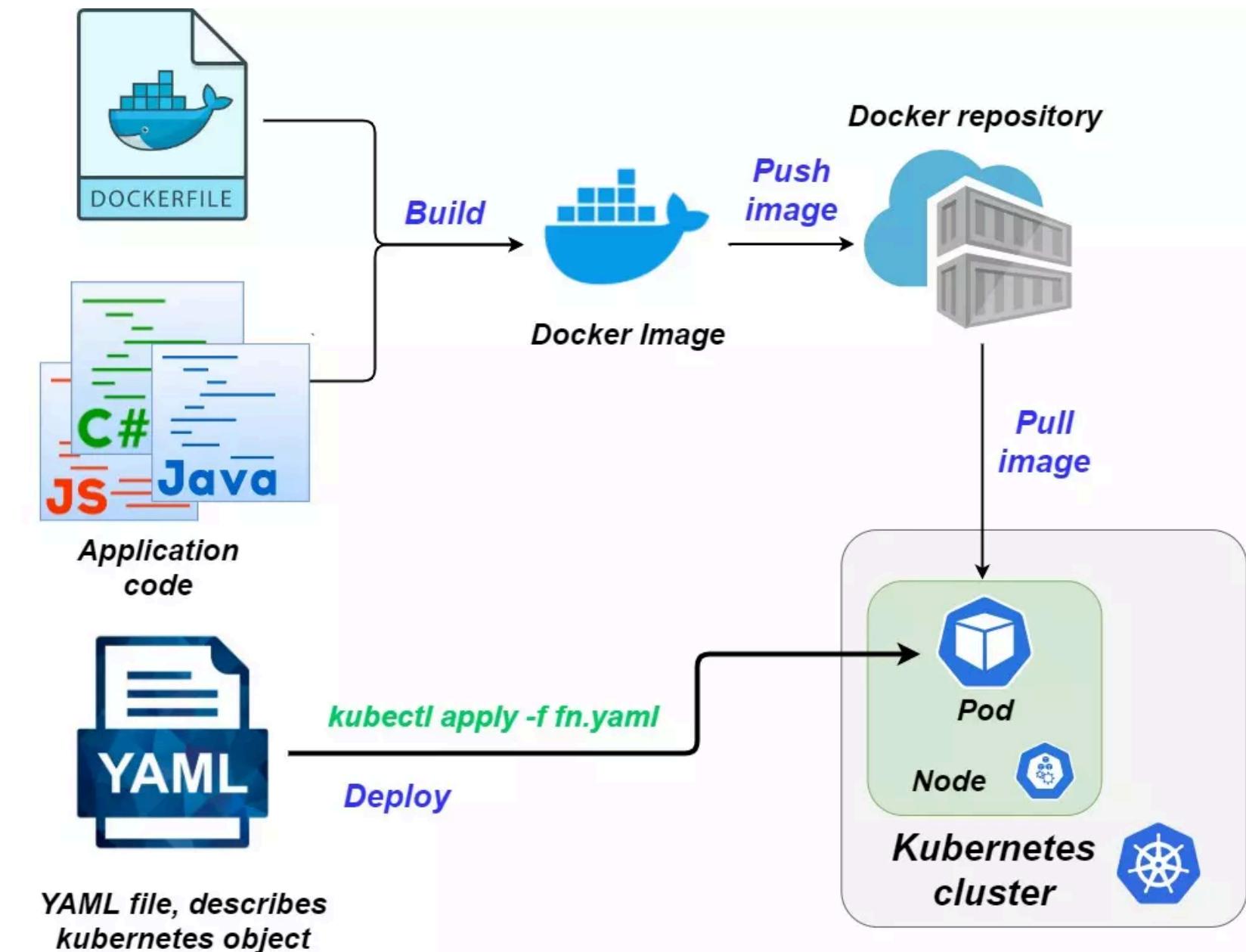
Lợi ích của Docker

- 4. Dễ dàng thiết lập môi trường làm việc:
Chỉ cần config 1 lần duy nhất và không
bao giờ phải cài đặt lại các dependencies.
- 5. Giữ cho work-space sạch sẽ hơn khi xóa môi
trường mà ảnh hưởng đến các phần khác.

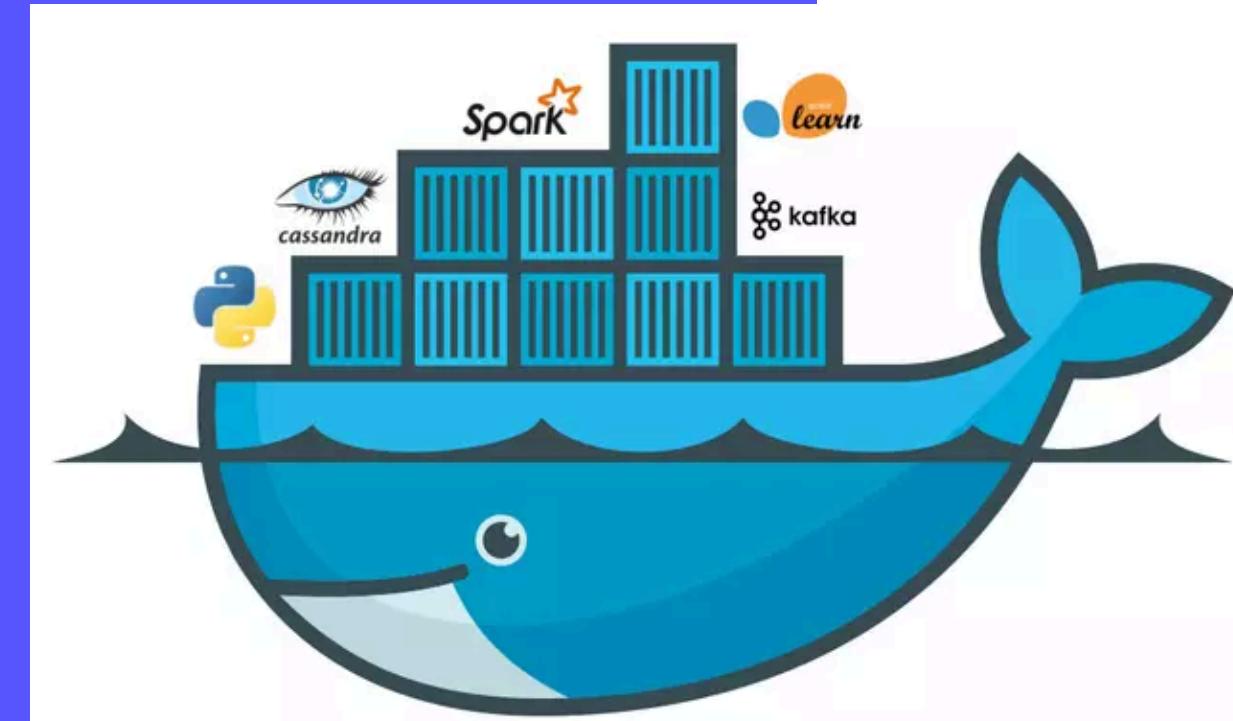
Ứng dụng thực tế?

Ứng dụng thực tế?

- Dễ dàng triển khai từ môi trường phát triển cục bộ đến máy chủ
- Use cases thực tế:
<https://cloud.folio3.com/blog/docker-use-cases/>



Khi nào nên sử dụng Docker?



Khi nào nên sử dụng Docker?

1. Cần tính nhất quán giữa các môi trường
 - Giải quyết vấn đề ứng dụng hoạt động khác nhau giữa môi trường develop, staging, và production.

Khi nào nên sử dụng Docker?

2. Cần quản lý ứng dụng phức tạp

- Thích hợp với kiến trúc nhiều services hoặc ứng dụng nhiều thành phần phụ thuộc.
- Đóng gói từng service trong container riêng.

Khi nào nên sử dụng Docker?

3. Cần khả năng mở rộng và linh hoạt

- Tăng/giảm số lượng container dễ dàng dựa trên nhu cầu tài nguyên thực tế.

Khi nào nên sử dụng Docker?

4. Cần tối ưu hóa tài nguyên hệ thống

- Nhẹ hơn và hiệu quả hơn so với máy ảo (VM).
- Containers chia sẻ kernel hệ điều hành chủ.

Khi nào nên sử dụng Docker?

5. CI/CD

- Phù hợp với các pipeline hiện đại: xây dựng, kiểm thử, triển khai đồng bộ.

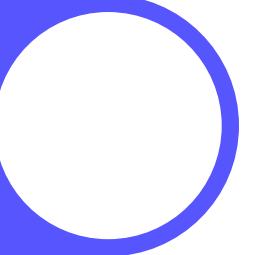
Khi nào nên sử dụng Docker?

6. Di chuyển ứng dụng giữa các nền tảng
 - Hỗ trợ chuyển đổi giữa on-premises và cloud, hoặc các nhà cung cấp cloud khác nhau.

Khi nào nên sử dụng Docker?

7. Quản lý phiên bản và phụ thuộc

- Quản lý nhiều phiên bản ứng dụng và các phụ thuộc một cách riêng biệt.



Các thành phần

2

Các thành phần

2

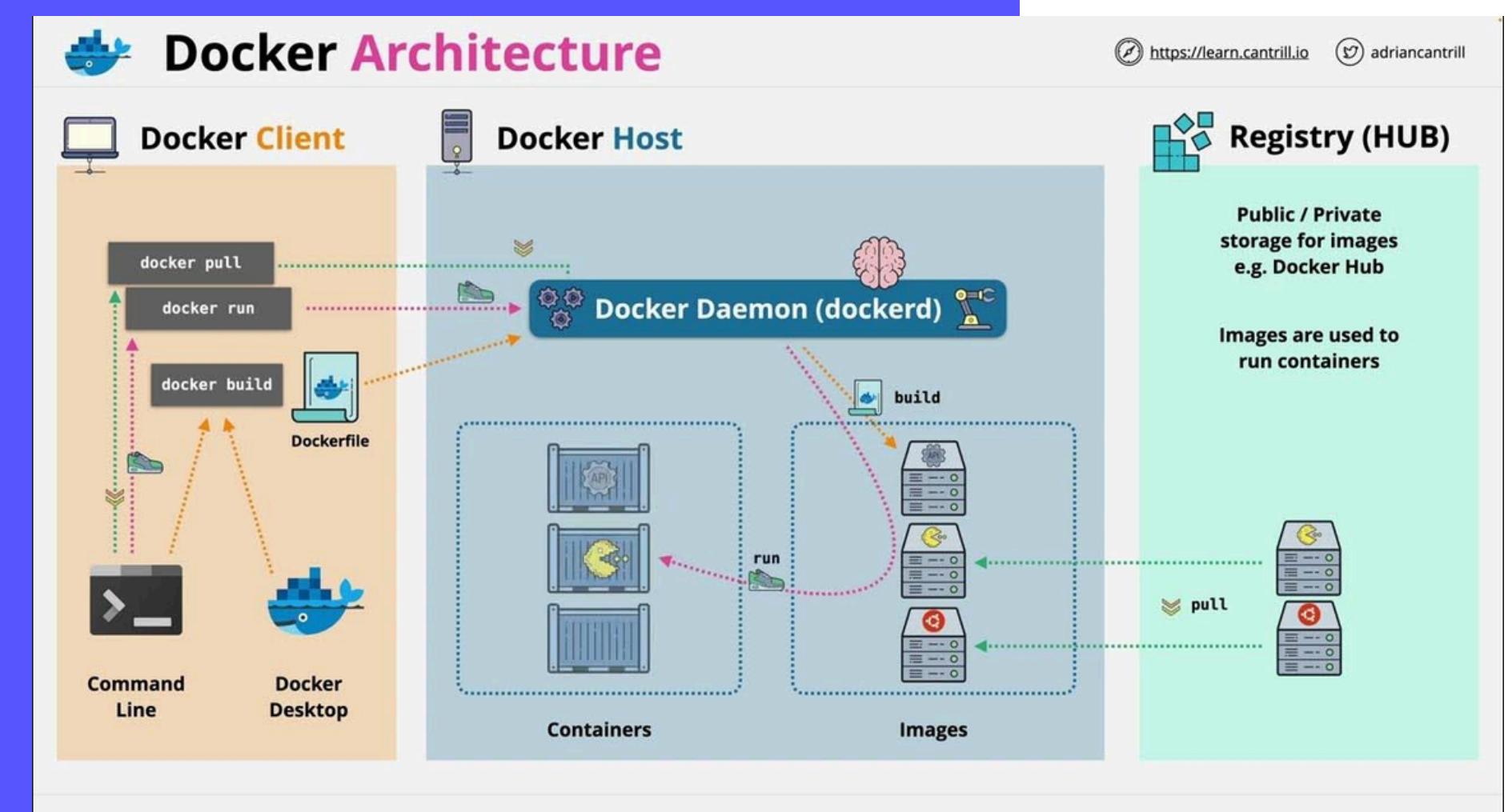
1. Docker Client

Các thành phần

2

1. Docker Client

- Docker Client là giao diện mà người dùng tương tác trực tiếp để gửi các lệnh Docker.

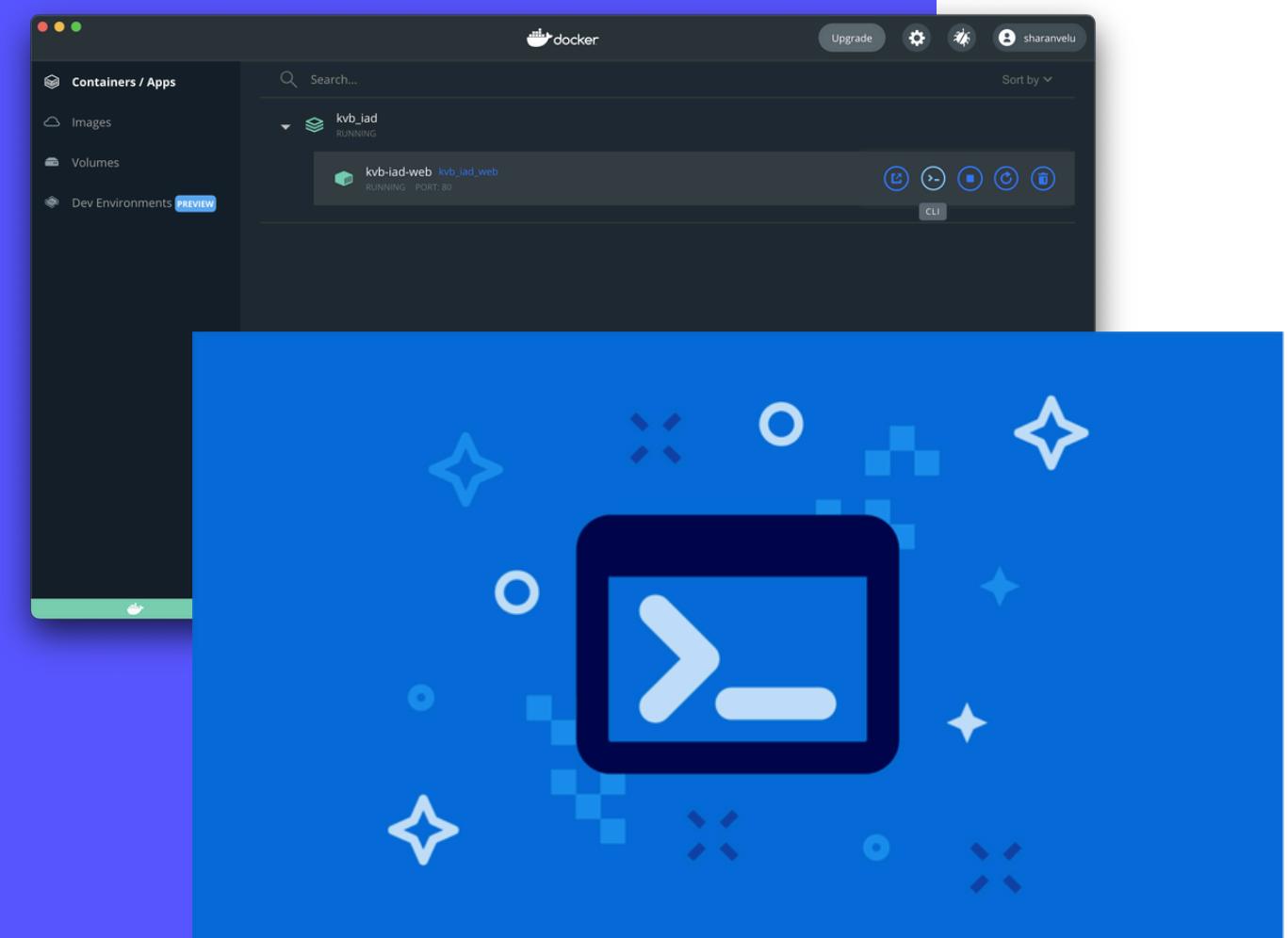


Các thành phần

2

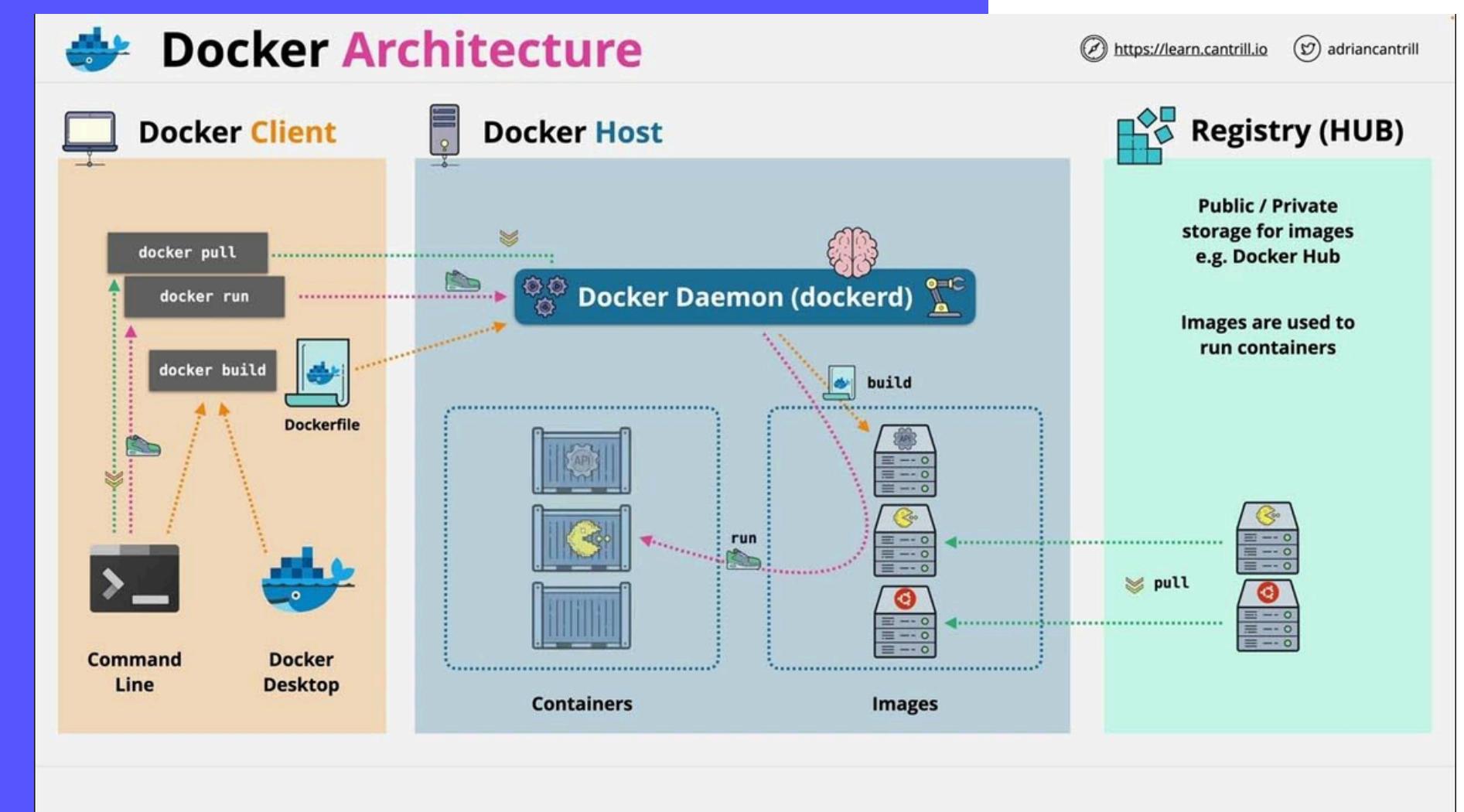
1. Docker Client

- Người dùng có thể dùng Docker CLI hoặc Docker Desktop để thực thi: run docker, build docker, pull docker image, v.v.



Các thành phần

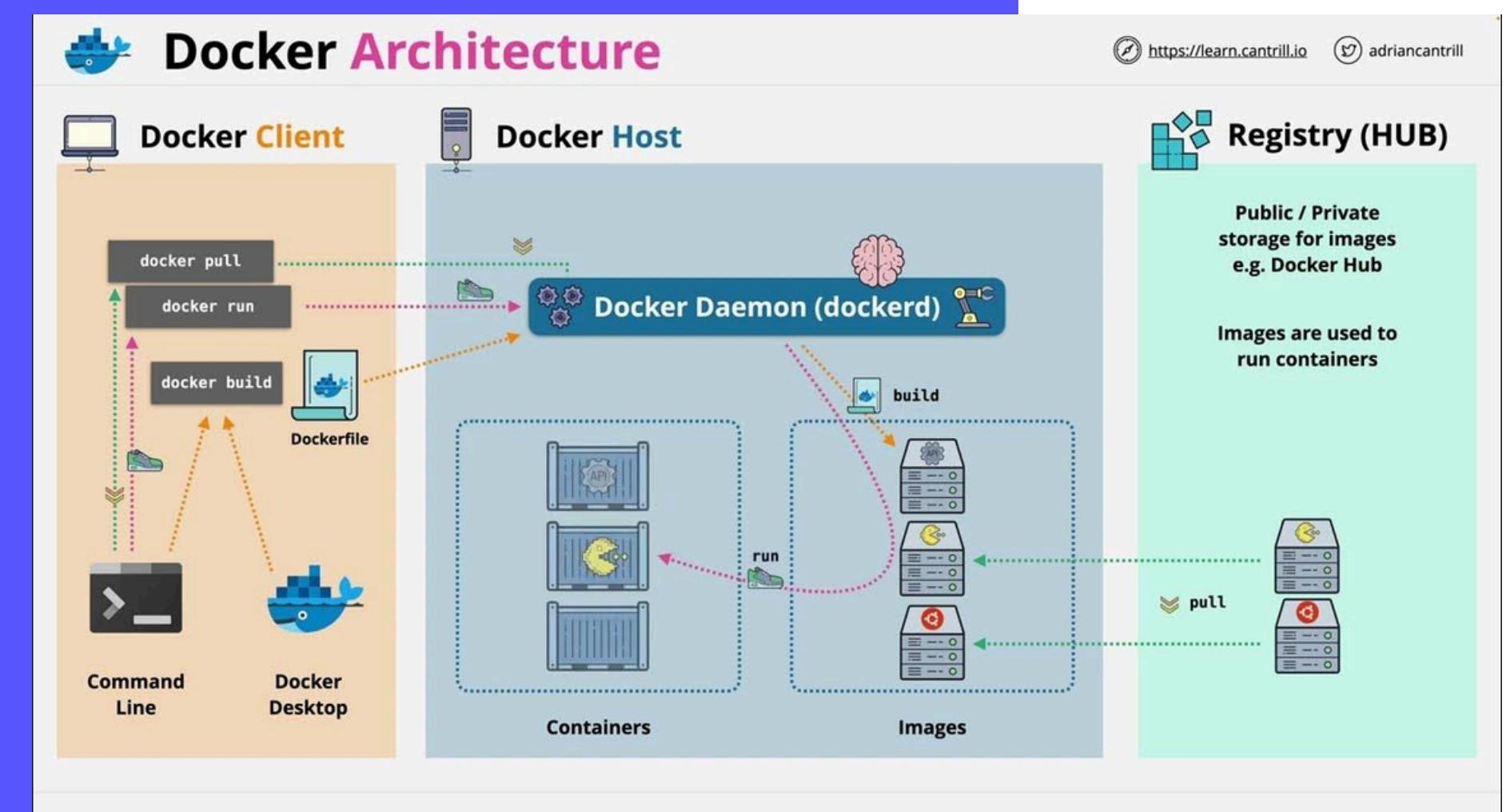
2. Docker Daemon (Docker Engine)



Các thành phần

2. Docker Daemon (Docker Engine)

- Docker Daemon là phần trung tâm của hệ thống Docker.
- Nó quản lý các container, hình ảnh (images), mạng (network) và volume trong hệ thống

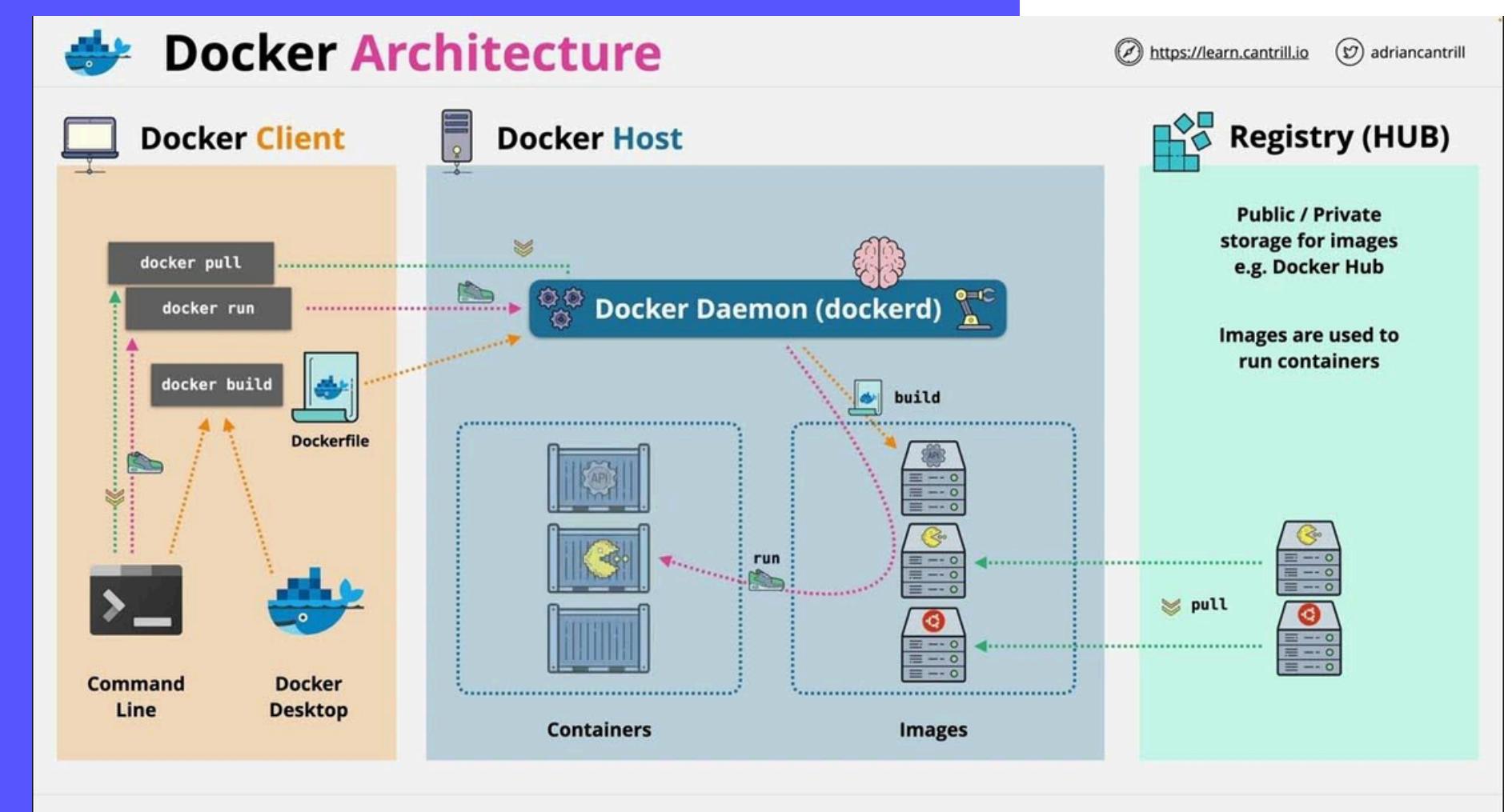


Các thành phần

2

2. Docker Daemon (Docker Engine)

- Docker Daemon nhận và xử lý các yêu cầu từ Docker Client và thực hiện các công việc như:
 - Chạy các container
 - Xây dựng, tải và quản lý các Docker image.
 - Cung cấp giao thức API để các phần mềm khác có thể tương tác với Docker.



Các thành phần

2

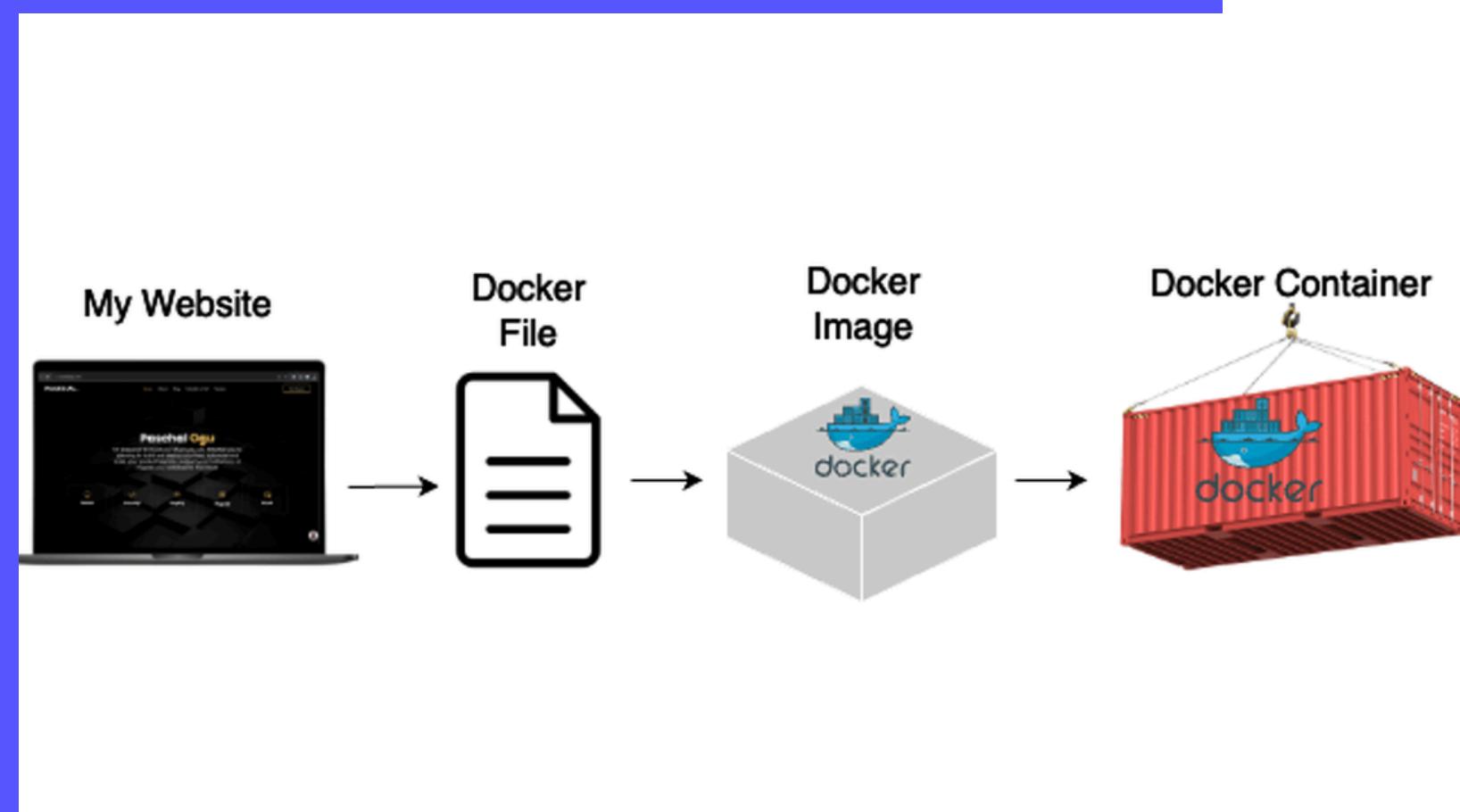
3. Docker Images

Các thành phần

2

3. Docker Images

- Docker Image là **một bản sao chỉ đọc** của phần mềm hoặc ứng dụng mà bạn muốn chạy trong container.
- Một image là một mẫu để tạo ra container.
- Mỗi image bao gồm tất cả các tệp tin, thư viện, và các thành phần cần thiết để chạy ứng dụng.
- Images có thể được tải từ Docker Hub hoặc tạo ra từ Dockerfile.



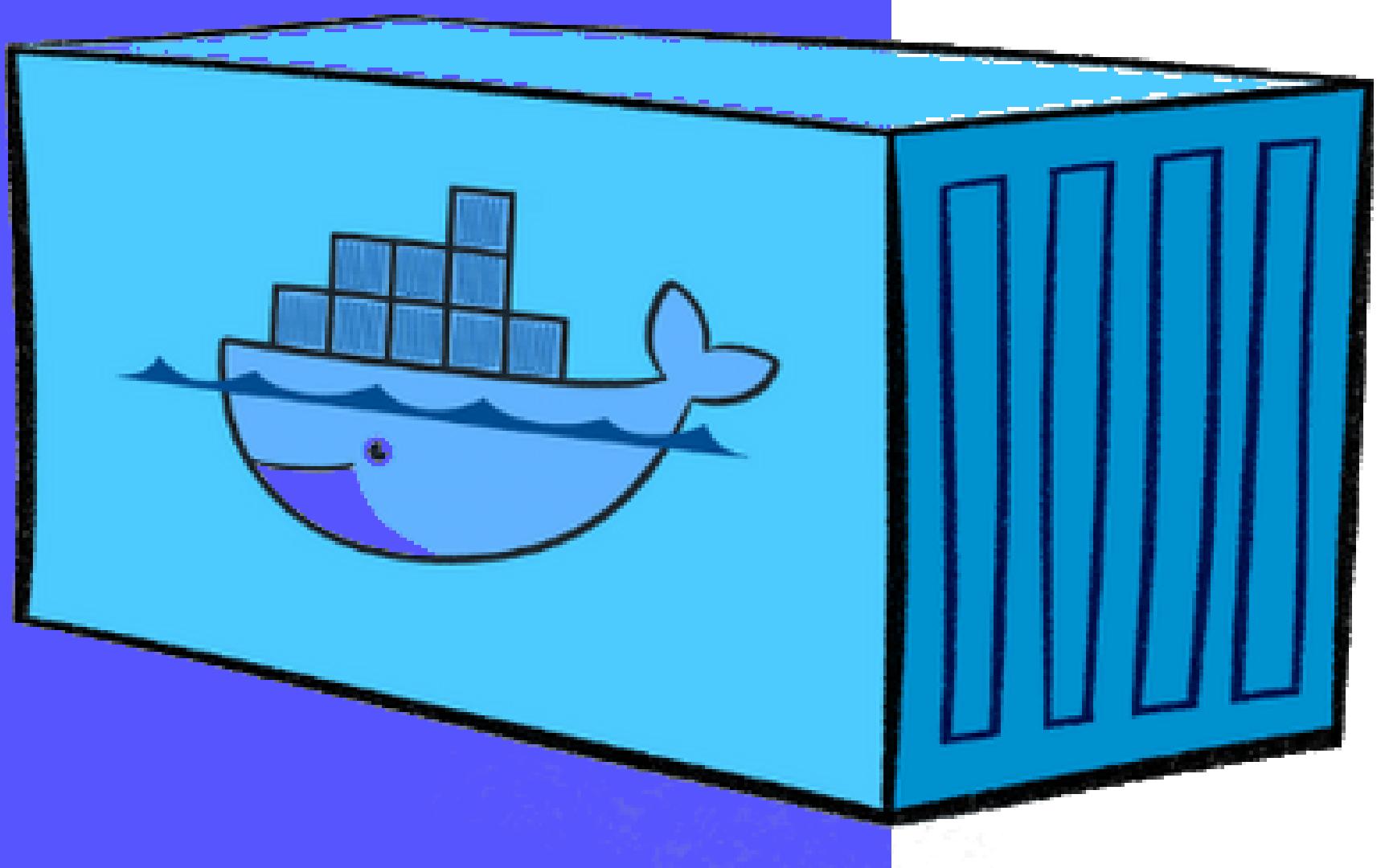
Các thành phần

2

4. Docker Containers

4. Docker Containers

- Docker Container là **một đơn vị chạy ứng dụng** trong Docker.
- Một container là **một thể hiện thực tế của một Image** và có thể chạy một ứng dụng trong môi trường được cô lập.
- Các container chia sẻ hệ điều hành của host nhưng lại có không gian riêng biệt cho các tài nguyên của chúng.



Các thành phần

2

5. Docker Registry

5. Docker Registry

- Docker Registry là **nơi lưu trữ các Docker Images**.
- Docker Hub là một registry công cộng phổ biến nhất, nơi người dùng có thể tìm kiếm, tải về hoặc chia sẻ Docker images.
- Ngoài Docker Hub, bạn có thể thiết lập **registry riêng tư** hoặc sử dụng các dịch vụ đám mây như Amazon ECR, Google Container Registry, hoặc Azure Container Registry.



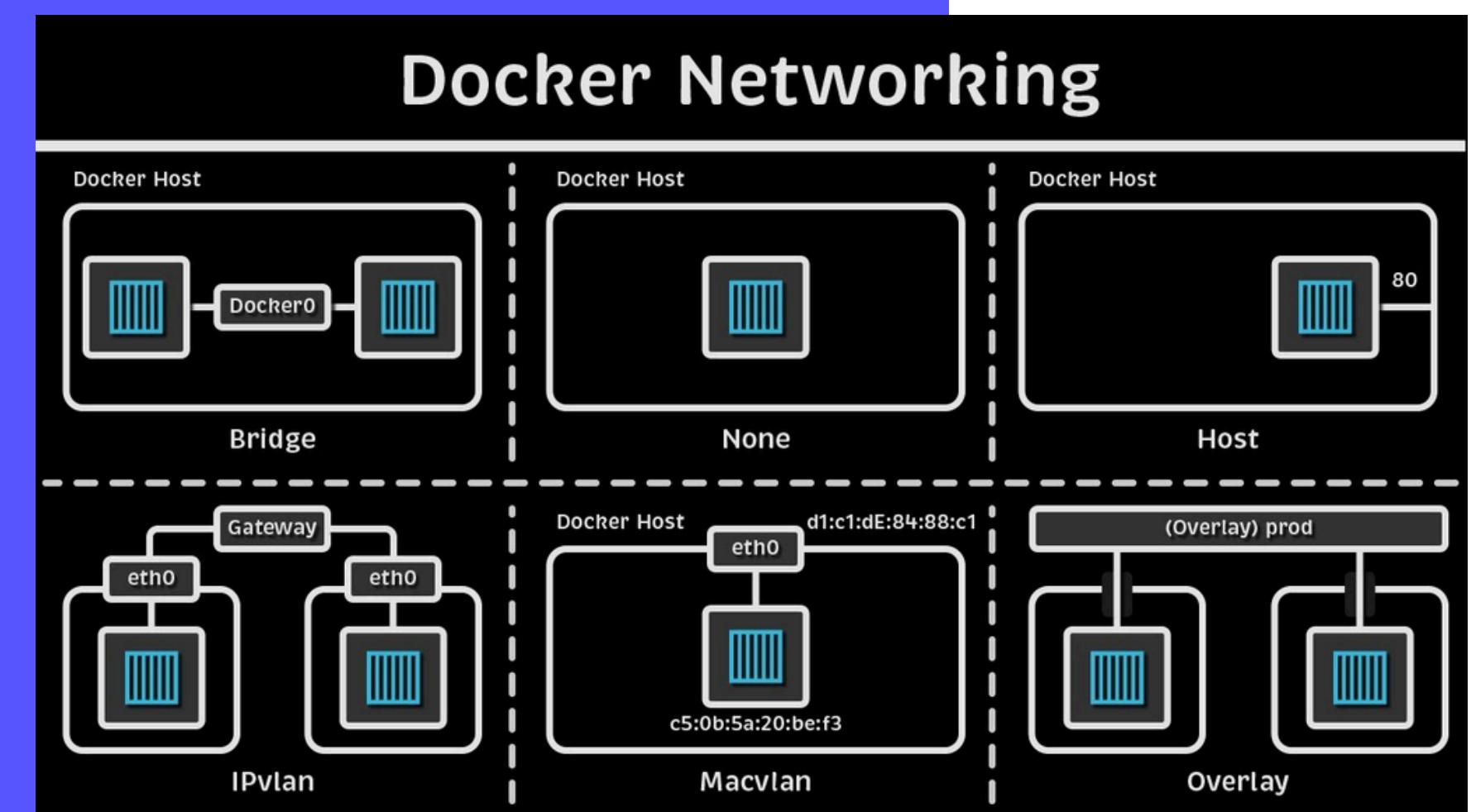
Các thành phần

2

6. Docker Network

6. Docker Network

- Docker Network cung cấp các **cơ chế để các container có thể giao tiếp với nhau và với hệ thống ngoài.**
- Docker có một số loại mạng tích hợp như bridge, host, và overlay, giúp bạn dễ dàng tạo và quản lý các kết nối giữa các container.



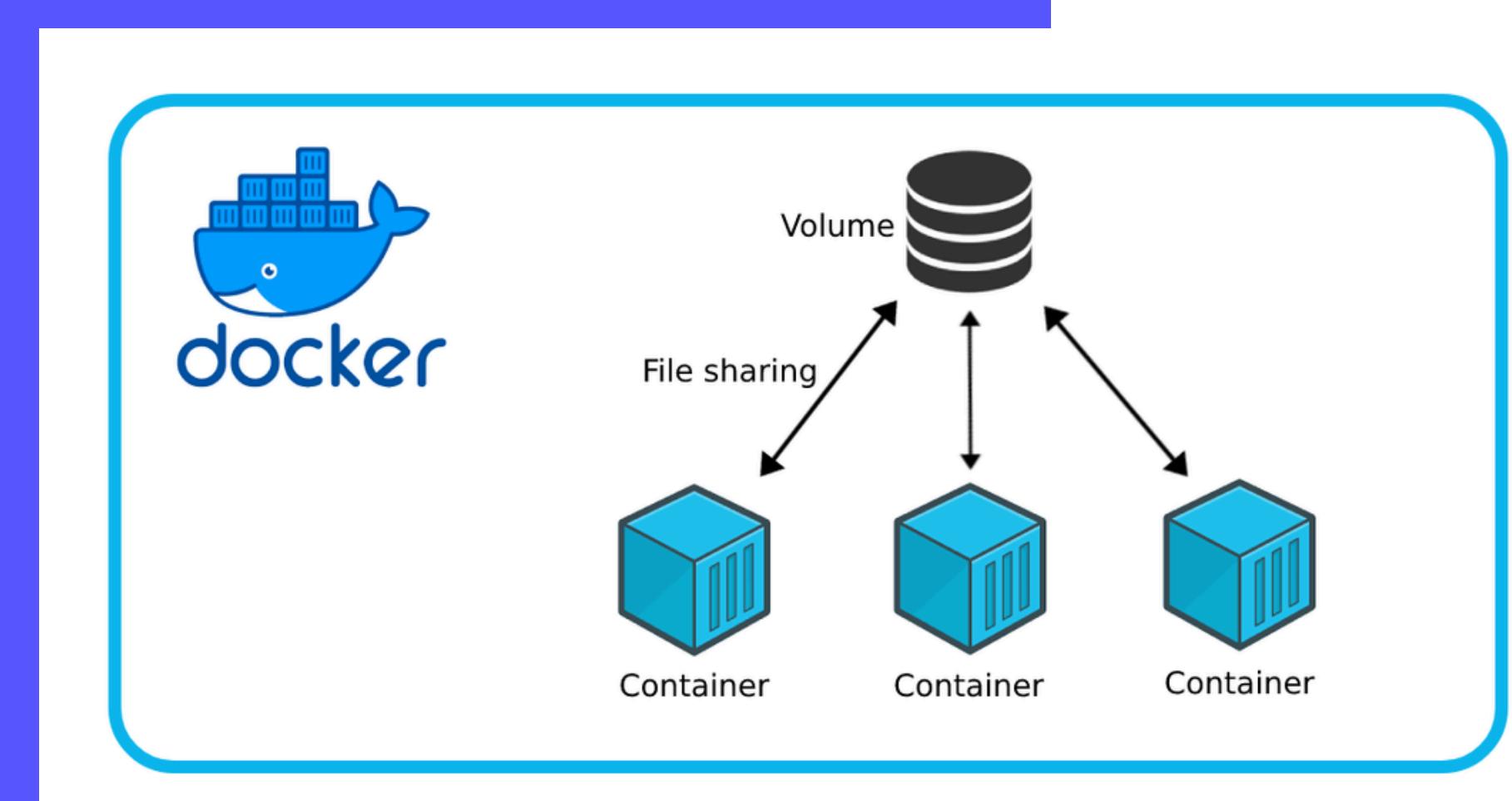
Các thành phần

2

7. Docker Volumes

7. Docker Volumes

- Docker Volumes là phương thức để lưu trữ và chia sẻ dữ liệu ngoài container.
- Vì các container là tạm thời và không lưu trữ dữ liệu lâu dài, **Volumes cho phép bạn lưu trữ dữ liệu một cách bền vững, ngay cả khi container bị xóa.**



Docker hoạt động như thế nào?

Các thành phần

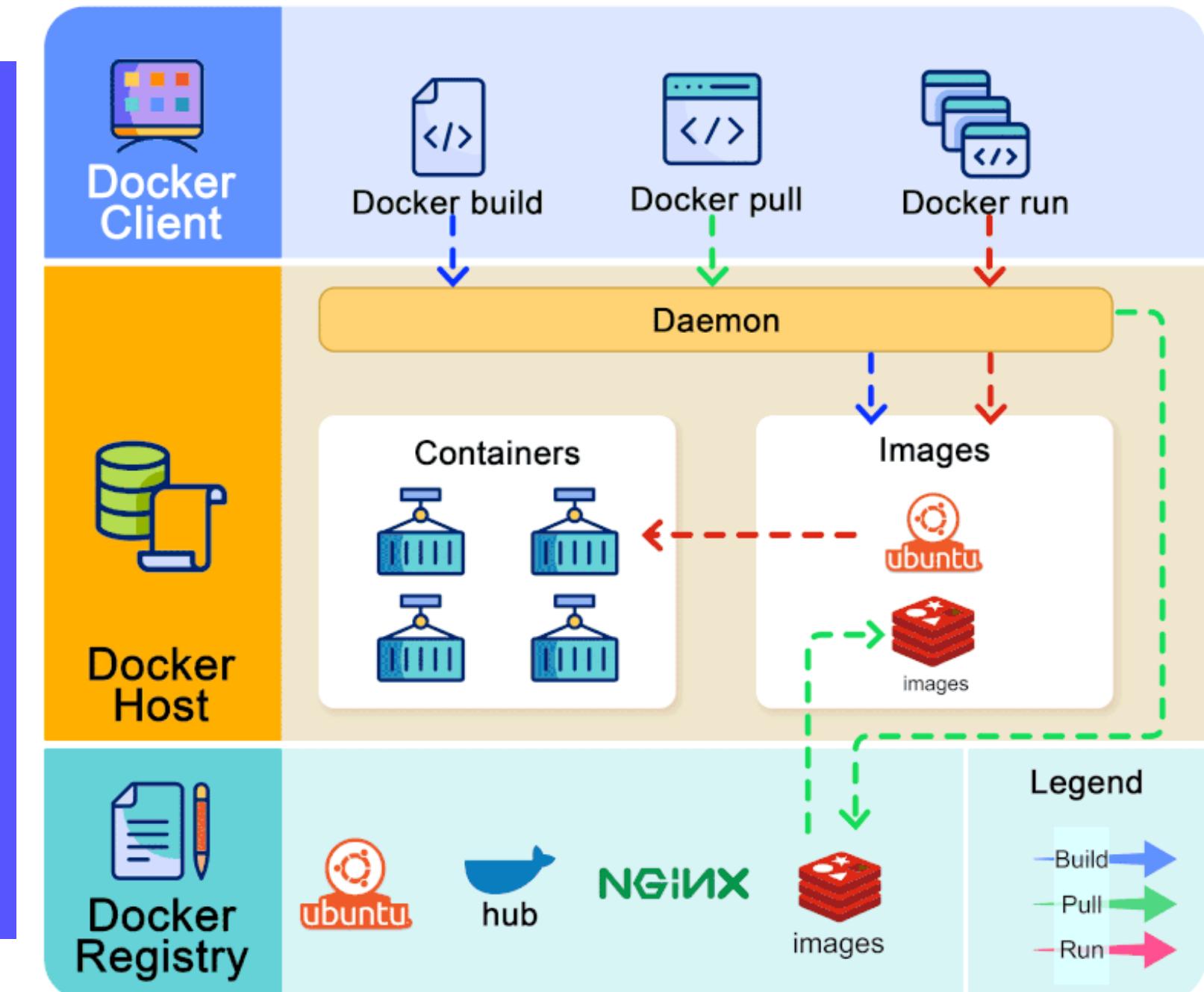
2

How does Docker Work ?

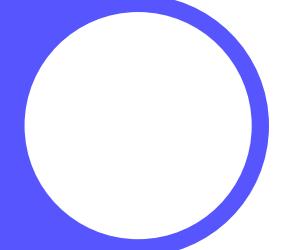
 blog.bytebytogo.com

Docker hoạt động như thế nào?

1. **Docker Client** gửi lệnh đến **Docker Daemon**.
2. **Docker Daemon** tạo ra và quản lý các **Docker Container** từ các **Docker Image**.
3. Các **Docker Containers** thực thi ứng dụng trong một môi trường ảo hóa.
4. Các image có thể được tải từ **Docker Registry**, bao gồm Docker Hub hoặc một registry riêng.



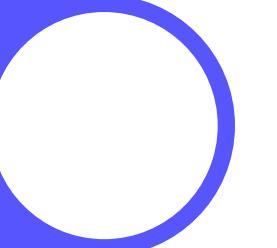
Câu hỏi



Docker được sử dụng ở những đâu trong dự án của mình?



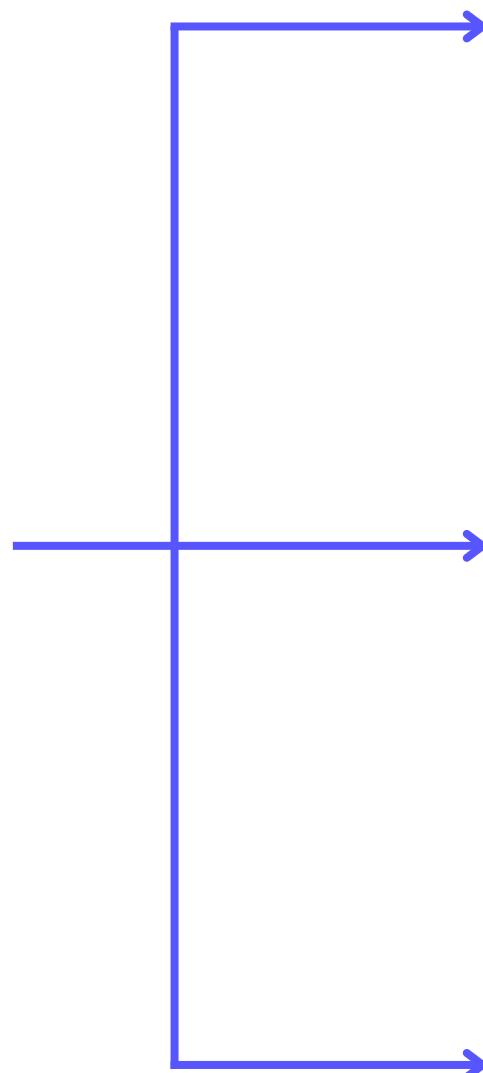
Câu hỏi



**Hệ thống
mình dùng
Docker ở
đâu?**

Câu hỏi

Hệ thống
mình dùng
Docker ở
đâu?



Core service:

Fulfillment-be, api-gateway

Other services:

Flag, Workflow, Convert PDF, Secret Store Vault, Image Service...

- Test services

- Company service: Blog, landingPage, Storage...

Cài Docker

3

- **Ubuntu:** <https://docs.docker.com/engine/install/ubuntu/>
- **Window:** <https://docs.docker.com/desktop/setup/install/windows-install/>
- **MacOS:** <https://docs.docker.com/desktop/setup/install/mac-install/>



Liceria & Co.

THANK
YOU

