

SFIT

Trường đại học Giao Thông Vận Tải

DECKER

Containers là gì?

Container là một dạng **công nghệ ảo hóa** cho phép bạn gói gọn một ứng dụng và tất cả các phụ thuộc của nó vào một gói duy nhất. Điều này làm cho ứng dụng của bạn **có thể di động, có thể chia sẻ và có thể tái tạo**.

Container thúc đẩy tính di động và khả năng tái tạo vì chúng đóng gói **TẤT CẢ** các phụ thuộc của ứng dụng... **bao gồm cả hệ điều hành nhỏ của riêng nó!**

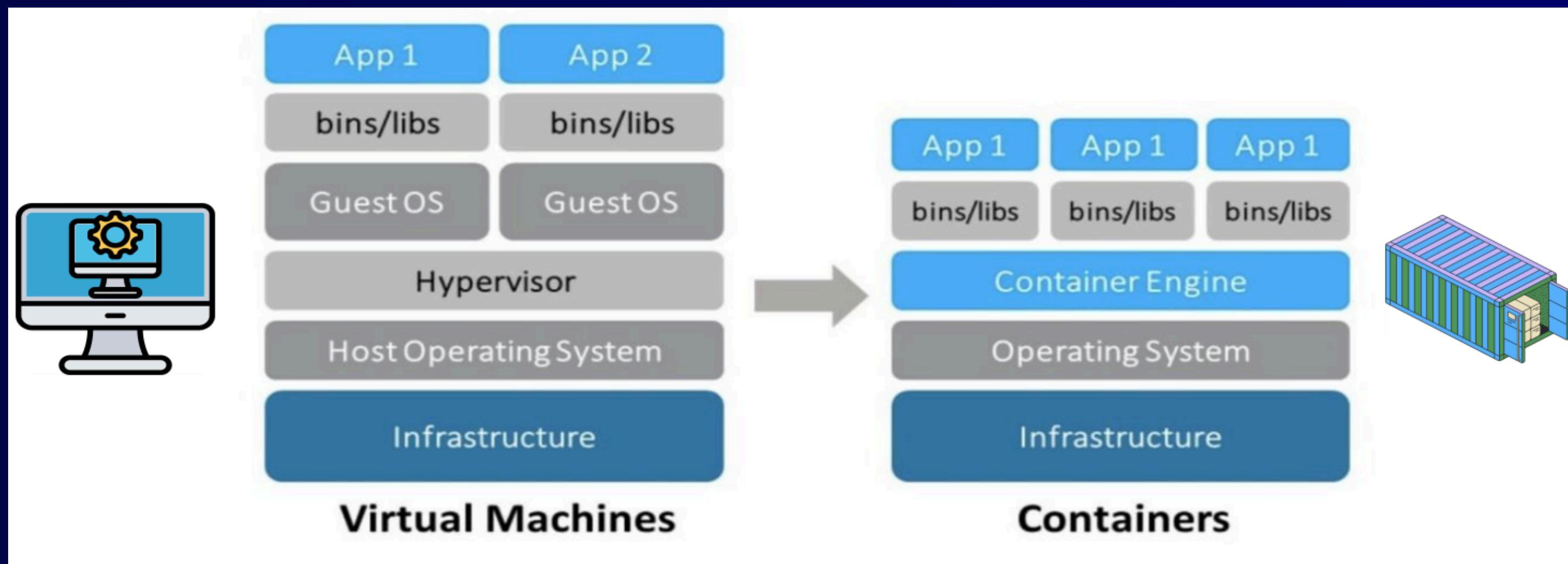
Điều này có nghĩa là ứng dụng của bạn sẽ không bị lỗi khi bạn chuyển nó sang một môi trường mới. **Ứng dụng của bạn mang theo môi trường của nó.**

Các khái niệm chính

- **Cô lập (Isolation):** Các container cung cấp một môi trường an toàn và được cô lập cho các ứng dụng, ngăn chặn các xung đột với các ứng dụng khác hoặc với hệ thống máy chủ (host system).
- **Tính di động (Portability):** Các container có thể chạy nhất quán trên các môi trường khác nhau, giúp việc phát triển và triển khai ứng dụng trở nên dễ dàng.
- **Hiệu quả (Efficiency):** Các container hiệu quả về mặt sử dụng tài nguyên, vì chúng chia sẻ nhân hệ điều hành của máy chủ.

Virtual Machines vs Containers

Container hoạt động dựa trên khái niệm ảo hóa cấp độ hệ điều hành, tức là **khả năng của nhân (kernel) để tạo ra nhiều môi trường cô lập** trên một máy chủ duy nhất.



Cài đặt mọi thành phần của một hệ điều hành (OS)
cho đến phần mềm cốt lõi cho phép hệ điều hành
điều khiển phần cứng (gọi là kernel)

**Container chia sẻ nhân
với hệ điều hành máy chủ.**

Virtual Machines vs Containers

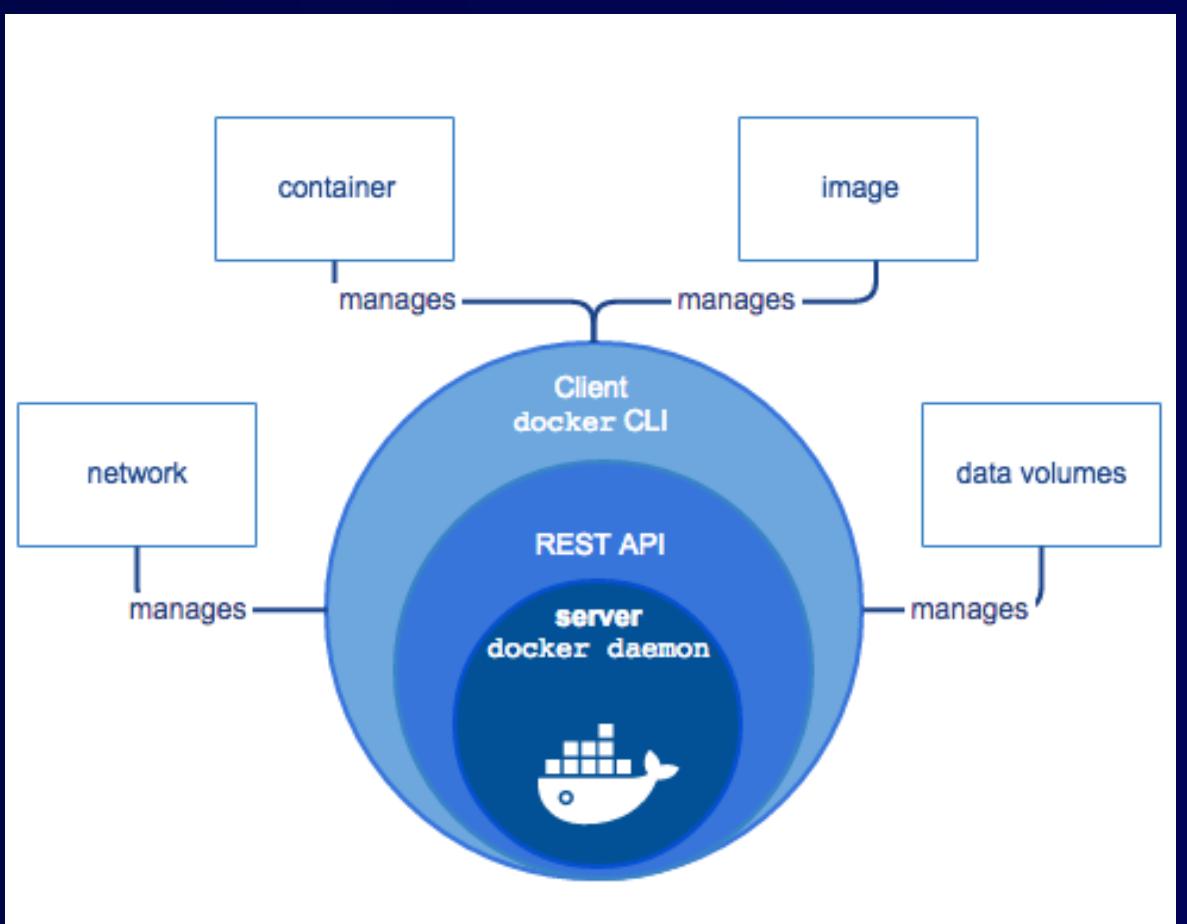
Do sự khác biệt của chúng, máy ảo (VM) và container phục vụ các mục đích khác nhau và nên được ưu tiên trong các trường hợp khác nhau.

- Máy ảo (VM) tốt cho các **phiên tương tác chạy dài** nơi bạn có thể muốn sử dụng nhiều ứng dụng khác nhau. (Kiểm tra email trên ứng dụng khách yêu thích của bạn và sử dụng Microsoft Word và Excel, v.v.).
- Container phù hợp hơn để chạy **một hoặc hai ứng dụng, thường là không tương tác, trong môi trường tùy chỉnh riêng** của chúng.

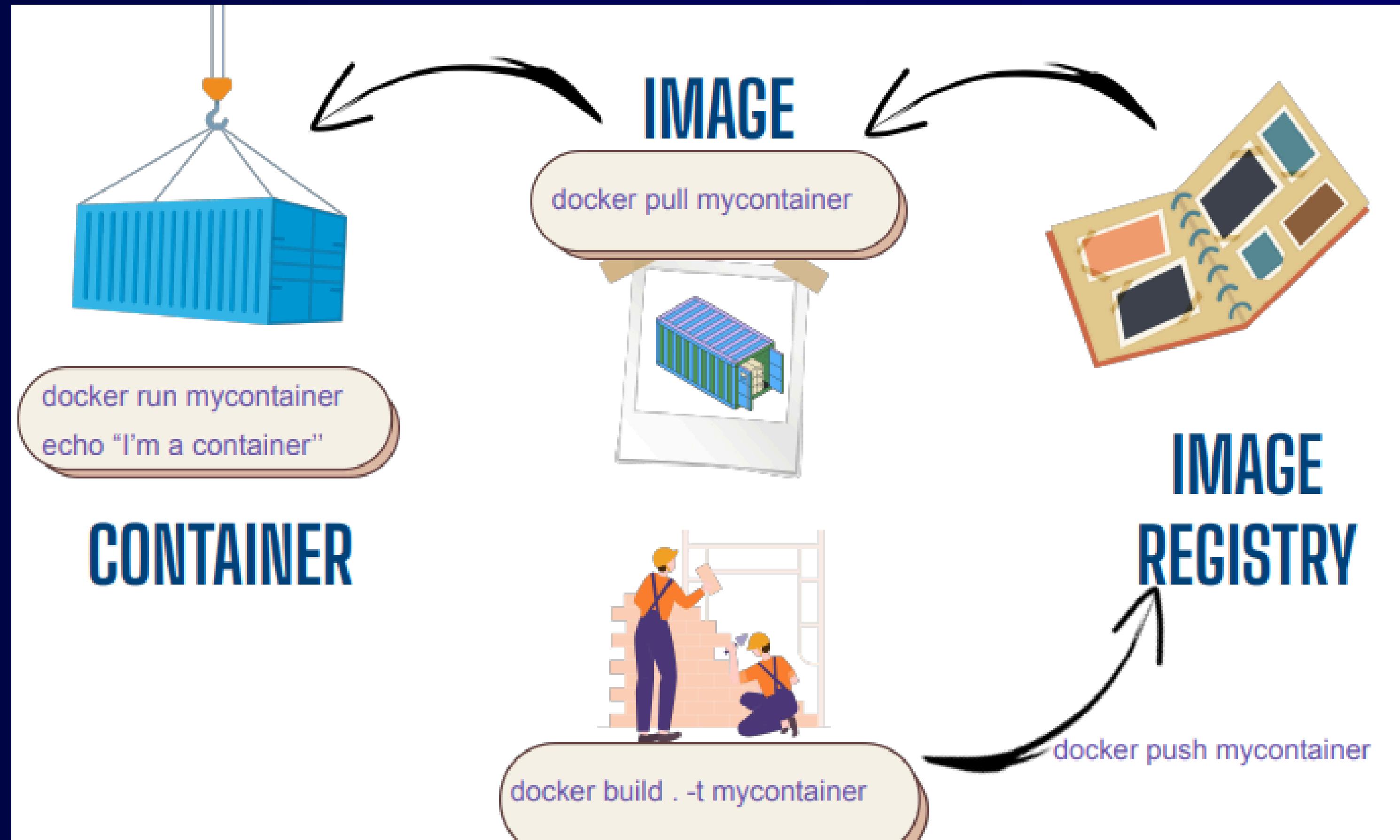
Docker

Docker là một nền tảng mã nguồn mở để xây dựng, triển khai và quản lý các ứng dụng được đóng gói trong container

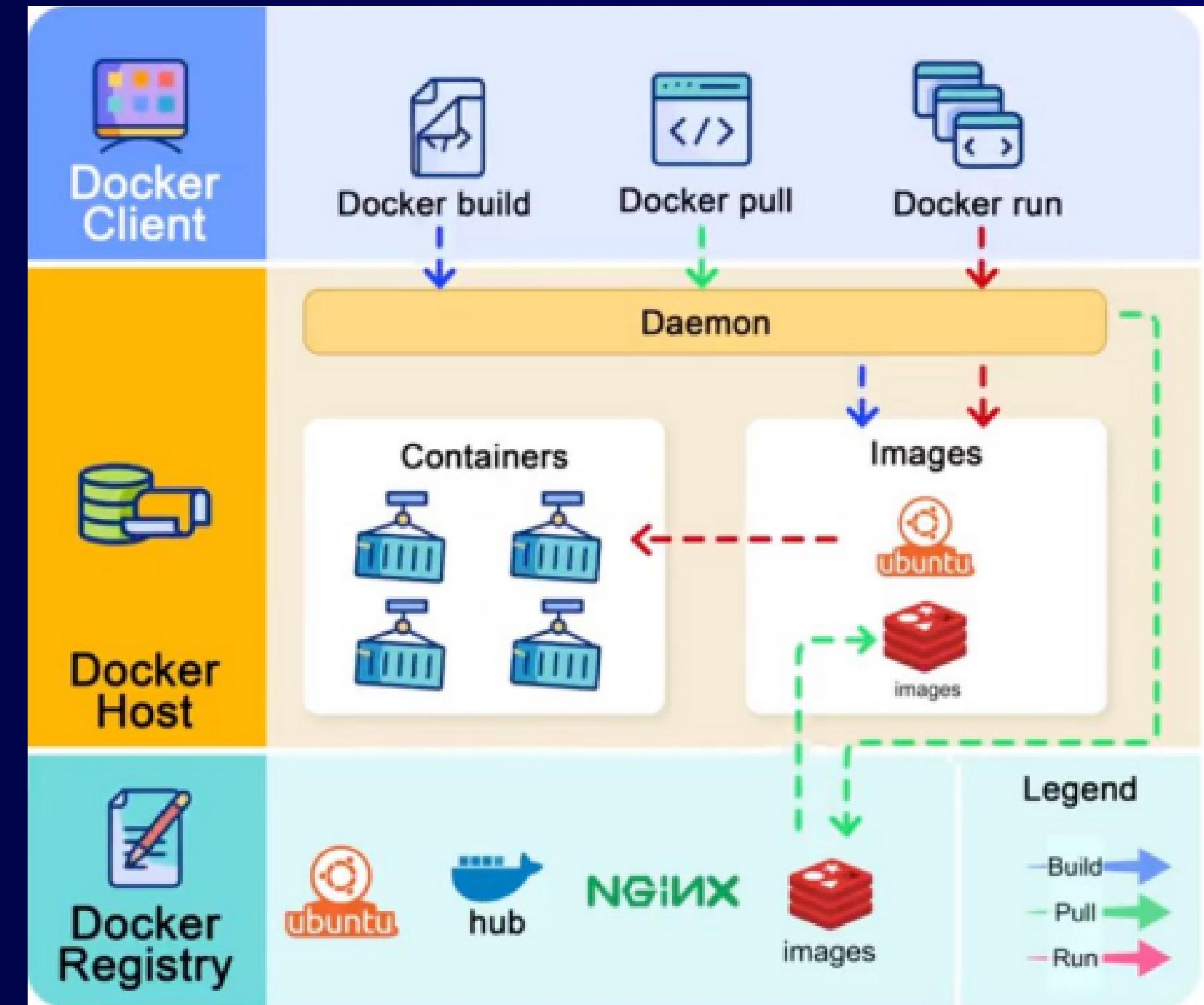
Một kiến trúc client-server:



Các thành phần trong hệ sinh thái



Kiến trúc Docker



DockerHub

Kho lưu trữ mặc định nếu sử dụng docker CLI

<https://hub.docker.com/>

The screenshot shows the Docker Hub search results page for the query "mcp". The results are filtered by the "Suggested" category. There are 28 available results out of 10,000. The results are displayed in a grid of cards, each representing a Docker image. The images are categorized under "Trusted content" and "Verified Publisher". The cards provide details such as the image name, maintainer, description, and metrics like Pulls and Stars.

Image	Maintainer	Description	Pulls	Stars	Last Updated
mcp/slack	mcp	Interact with Slack Workspaces over the Slack API.	100K+	16	13 days
mcp/fetch	mcp	Fetches a URL from the internet and extracts its contents as markdown	100K+	24	13 days
mcp/time	mcp	Time and timezone conversion capabilities	50K+	5	13 days
mcp/docker	mcp	MCP smart gateway used by the AI Tool Catalog	50K+	38	3 days
mcp/github	mcp	Tools for interacting with the GitHub API, enabling file operations, repository management, searc...	10K+	17	13 days
ai/phi4	Docker	Microsoft's compact model, surprisingly capable at reasoning and code	10K+	12	2 months
mcp/puppeteer	mcp	Browser automation and web scraping using Puppeteer.	10K+	27	about 1 month
mcp/sequentialthinking	mcp	Dynamic and reflective problem-solving through thought sequences.	10K+	24	about 2 months
mcp/postgres	mcp	Connect with read-only access to PostgreSQL databases. This server enables LLMs to inspect...	10K+	24	about 2 months
ai/deepseek-r1-distill-llama	Docker	Distilled LLaMA by DeepSeek, fast and optimized for real-world tasks	10K+	24	about 2 months
mcp/filesystem	mcp	Local filesystem access with configurable allowed paths	10K+	24	about 2 months
mcp/context7	mcp	Context7 MCP Server -- Up-to-date code documentation for LLMs and AI code editors	10K+	24	about 2 months

Một số kho lưu trữ khác

Các kho lưu trữ công khai khác (miễn phí sử dụng):

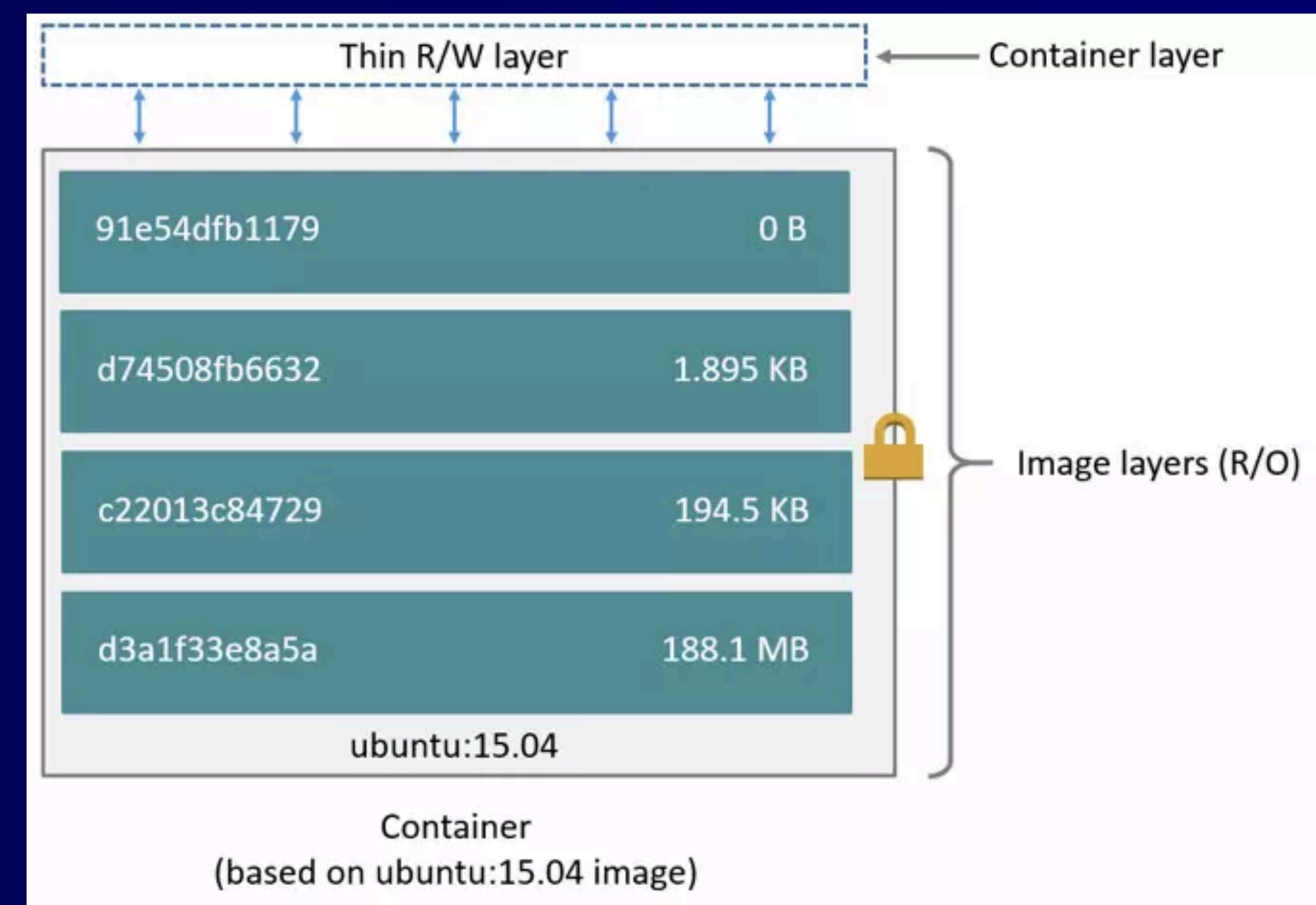
- **Github Container Registry (ghcr.io)**
 - Tích hợp hoàn toàn vào Github CI/CD để tự động xây dựng image

Các kho lưu trữ riêng tư (on-prem):

- **GitLab Container Registry** được tích hợp chặt chẽ với quy trình làm việc của GitLab CI, với thiết lập tối thiểu.
- **Harbor** là một kho lưu trữ mã nguồn mở bảo mật các tạo phẩm (artifacts) bằng các chính sách và kiểm soát truy cập dựa trên vai trò, đảm bảo các image được quét và không có lỗ hổng, và ký các image như đã được tin cậy.

Layer By Layer

- Một Docker Image bao gồm các lớp chỉ đọc (read-only layers) được xây dựng chồng lên nhau.
- Docker sử dụng Hệ thống tệp Union (UFS) để xây dựng một image.
- Image được chia sẻ giữa các container.
- Mỗi khi Docker khởi chạy một container từ một image, nó sẽ thêm một lớp mỏng có thể ghi (writable layer), được gọi là lớp container, lưu trữ tất cả các thay đổi đối với container trong suốt thời gian chạy của nó

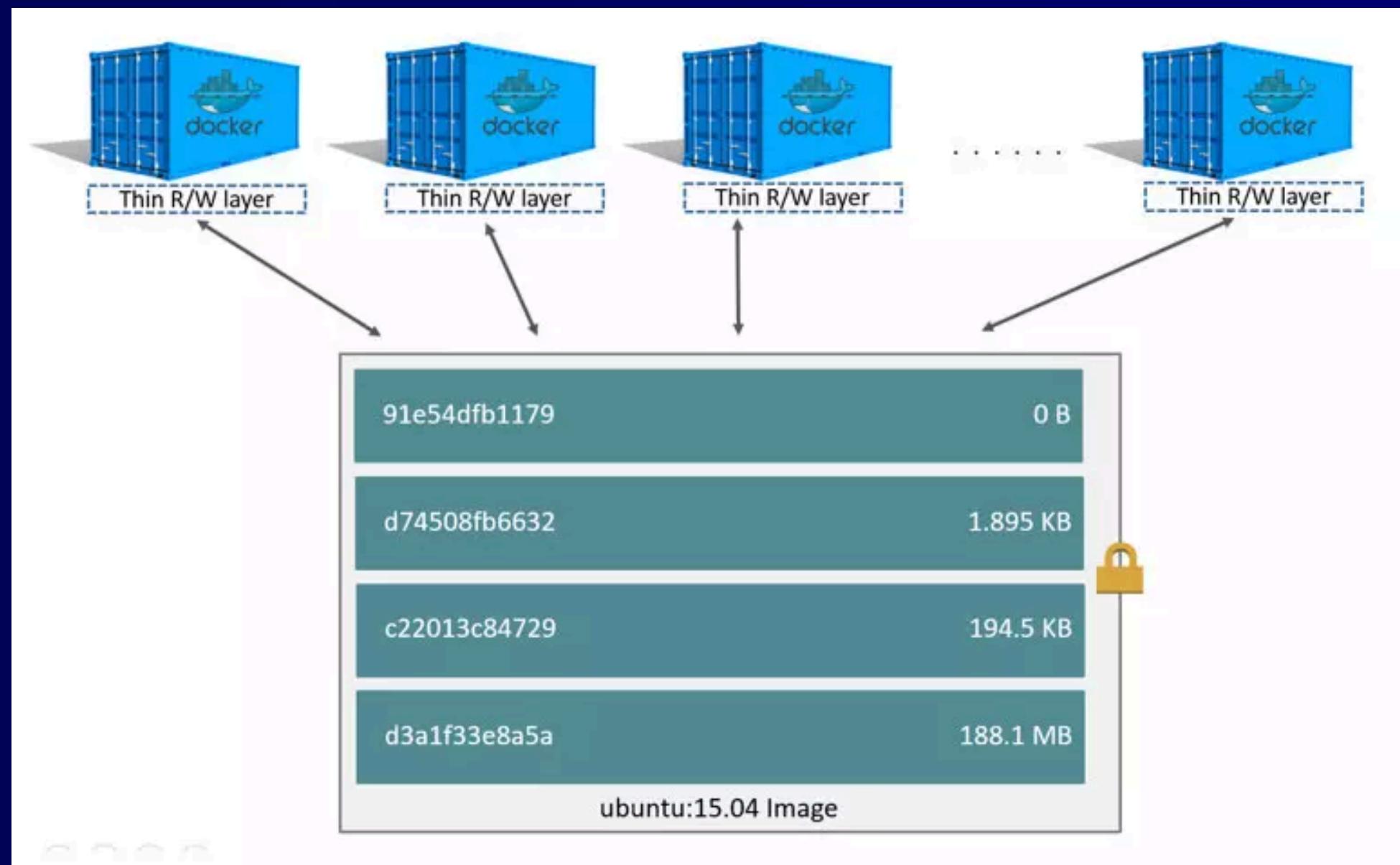


One image many containers

Mỗi container có lớp container có thể ghi riêng của nó, và tất cả các thay đổi được lưu trữ trong lớp container này.

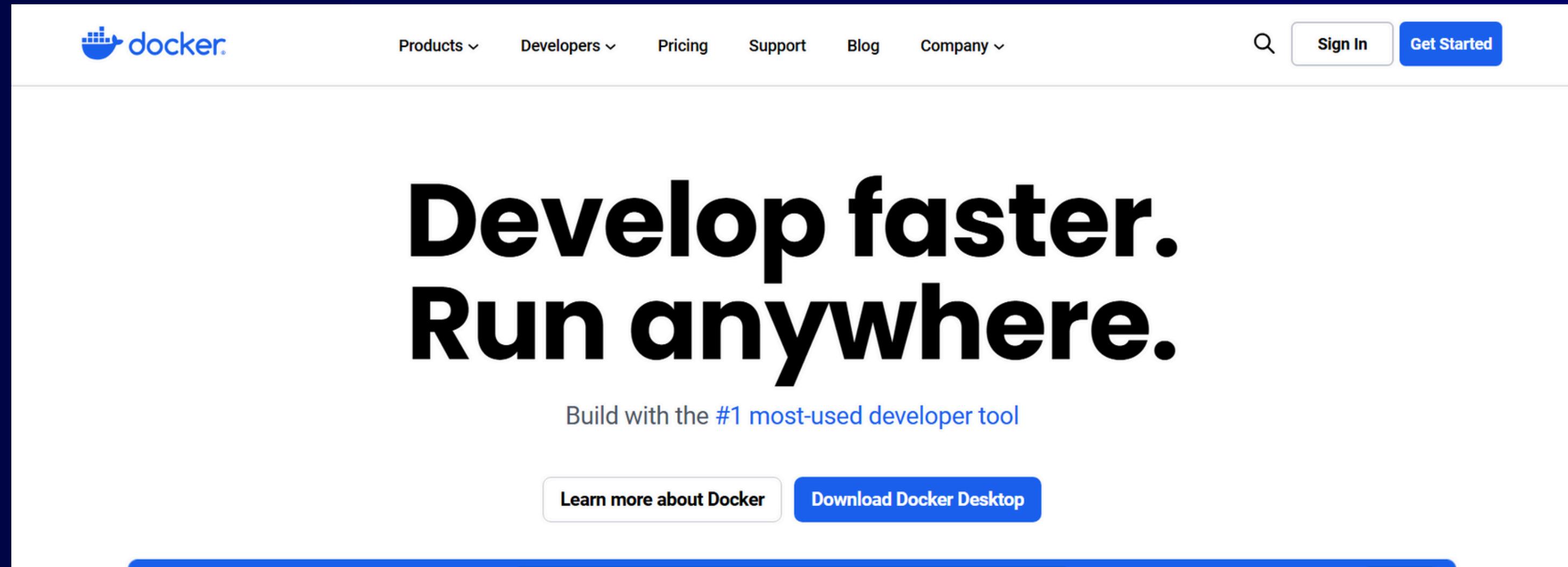
Nhiều container có thể chia sẻ quyền truy cập vào cùng một image cơ bản nhưng vẫn có trạng thái dữ liệu riêng của chúng.

Khi container bị xóa, lớp có thể ghi cũng bị xóa. Image cơ bản vẫn không thay đổi.



Tải Docker

<https://www.docker.com/>



Thực hành cơ bản
