

Lesson 4

001

Ba thách thức chính khi triển khai microservices:

- Triển khai (Deployment): Với hàng trăm microservices, làm thế nào để triển khai hiệu quả mà không cần nhiều máy chủ riêng lẻ?
- Di động (Portability): Làm sao để dễ dàng di chuyển code giữa các môi trường (Dev, UAT, Production)?
- Mở rộng (Scalability): Làm thế nào để mở rộng hoặc thu nhỏ từng microservice theo nhu cầu mà không tốn quá nhiều chi phí và công sức?

💡 Giải pháp: Containerization

- Docker giúp đóng gói ứng dụng thành container, bao gồm tất cả dependencies cần thiết (JDK, cấu hình DB, thư viện, v.v.), đảm bảo ứng dụng có thể chạy ổn định trên nhiều môi trường khác nhau mà không cần thay đổi cấu hình.
- So sánh với tàu chở hàng container, mỗi container là một môi trường độc lập, giúp tối ưu không gian, quản lý tài nguyên tốt hơn và triển khai hiệu quả.

002

Giới thiệu về containerization: Giảng viên đề cập đến các thuật ngữ như containerization, containers và Docker.

Sự khác biệt giữa Virtual Machines (VMs) và Containers:

- Trước đây, các tổ chức mua phần cứng, cài đặt hệ điều hành, triển khai ứng dụng trên đó và kết nối vào mạng công cộng.
- Với điện toán đám mây, không cần mua máy chủ vật lý mà có thể sử dụng máy ảo (VM), được tạo ra bằng hypervisor trên phần cứng của nhà cung cấp đám mây.
- Mỗi VM có hệ điều hành riêng, nhưng có nhiều hạn chế như chi phí cao, tài nguyên lãng phí, khó mở rộng và tốn thời gian khởi tạo.

Vấn đề với Virtual Machines:

- Việc triển khai nhiều microservices trên VMs gặp nhiều khó khăn về tài nguyên, xung đột môi trường (ví dụ: yêu cầu các phiên bản Java khác nhau), và khả năng mở rộng chậm (có thể mất 15 phút để tạo VM mới).
- Nếu một VM chứa nhiều microservices, việc khởi động lại VM sẽ làm gián đoạn tất cả dịch vụ bên trong.

Giới thiệu Containers và Docker:

- Containers giúp giải quyết các vấn đề trên bằng cách chạy trên cùng một hệ điều hành (chia sẻ OS host), không cần guest OS riêng.
- Điều này giúp container nhẹ hơn, khởi tạo nhanh hơn (chỉ vài giây), dễ dàng mở rộng và quản lý linh hoạt hơn so với VMs.
- Docker là một công nghệ container phổ biến giúp triển khai microservices một cách hiệu quả.

🔗 Tóm lại: Containers là lựa chọn tốt hơn VMs khi triển khai microservices do tính linh hoạt, hiệu suất cao, chi phí thấp và khả năng mở rộng nhanh chóng.

003

Container là gì?

- Một môi trường cách ly nhẹ trong một hệ thống, có thể chạy trên máy chủ, máy ảo hoặc hệ thống cục bộ.
- Chứa mã nguồn và các thư viện cần thiết để chạy ứng dụng trên nhiều môi trường mà không cần cấu hình lại.

Container Image & Docker Container:

- Container Image giống như một bản thiết kế, từ đó có thể tạo ra nhiều container chạy độc lập.
- Docker Container là phiên bản chạy thực tế của một Docker Image.

Software Containerization:

- Phương pháp ảo hóa hệ điều hành giúp nhiều container có thể chạy trên cùng một hệ điều hành nhưng vẫn có môi trường riêng biệt.
- So với máy ảo truyền thống (VM), container dùng chung kernel hệ điều hành, giúp nhẹ hơn và hiệu suất cao hơn.

Namespace & Cgroups trong Linux:

- Namespace: Cung cấp không gian riêng cho các container, giúp cô lập tài nguyên như tiến trình, mạng, bộ nhớ.
- Cgroups: Kiểm soát và giới hạn tài nguyên mà mỗi container có thể sử dụng (CPU, RAM, disk, mạng).

Docker là gì?

- Một nền tảng mã nguồn mở giúp đóng gói ứng dụng thành container, triển khai dễ dàng và nhất quán trên nhiều môi trường.
- Trên Linux, Docker hoạt động trực tiếp trên OS. Trên Windows/macOS, Docker sử dụng một máy ảo nhẹ chạy Linux để hỗ trợ container hóa.

004

Docker Client: Giao diện để gửi lệnh đến Docker Server. Có thể sử dụng Docker CLI hoặc Docker Remote API để thao tác với Docker Server.

Docker Server (Docker Host): Chạy tiến trình Docker Daemon, nhận lệnh từ Docker Client để tạo Docker Image từ ứng dụng và triển khai Docker Container từ image đó.

Docker Registry: Kho lưu trữ Docker Images. Docker Hub là registry mặc định, ngoài ra có thể sử dụng các registry riêng tư như AWS, GCP, Azure, GitHub.

Quy trình hoạt động:

- Docker Client gửi lệnh (ví dụ: chạy container).

- Docker Server kiểm tra và lấy Docker Image từ local storage hoặc Docker Registry.
- Dựa trên image, Docker Server tạo và chạy một container chứa ứng dụng.

Ứng dụng thực tế: Dùng Docker để dễ dàng cài đặt và chạy các phần mềm như MySQL mà không cần cài đặt thủ công.

005 – Hướng dẫn cài Docker

006