

DevOps là gì?

PRESENTATION BY NGUYEN NHU THUONG

NỘI DUNG CHÍNH

KIẾN TRÚC MICROSERVICES

Tìm hiểu về tổng quan, kiến trúc microservice và xu hướng phát triển ứng dụng ngày nay.

CONTINUOUS INTEGRATION & CONTINUOUS DELIVERY

Hiểu rõ và phân biệt giữa quy trình CI và CD, áp dụng vào dự án như thế nào?

INFRASTRUCTURE AS CODE

Tìm hiểu về cách triển khai hệ thống tự động (CD) sử dụng IaaS – kịch bản triển khai hệ thống dưới dạng code.

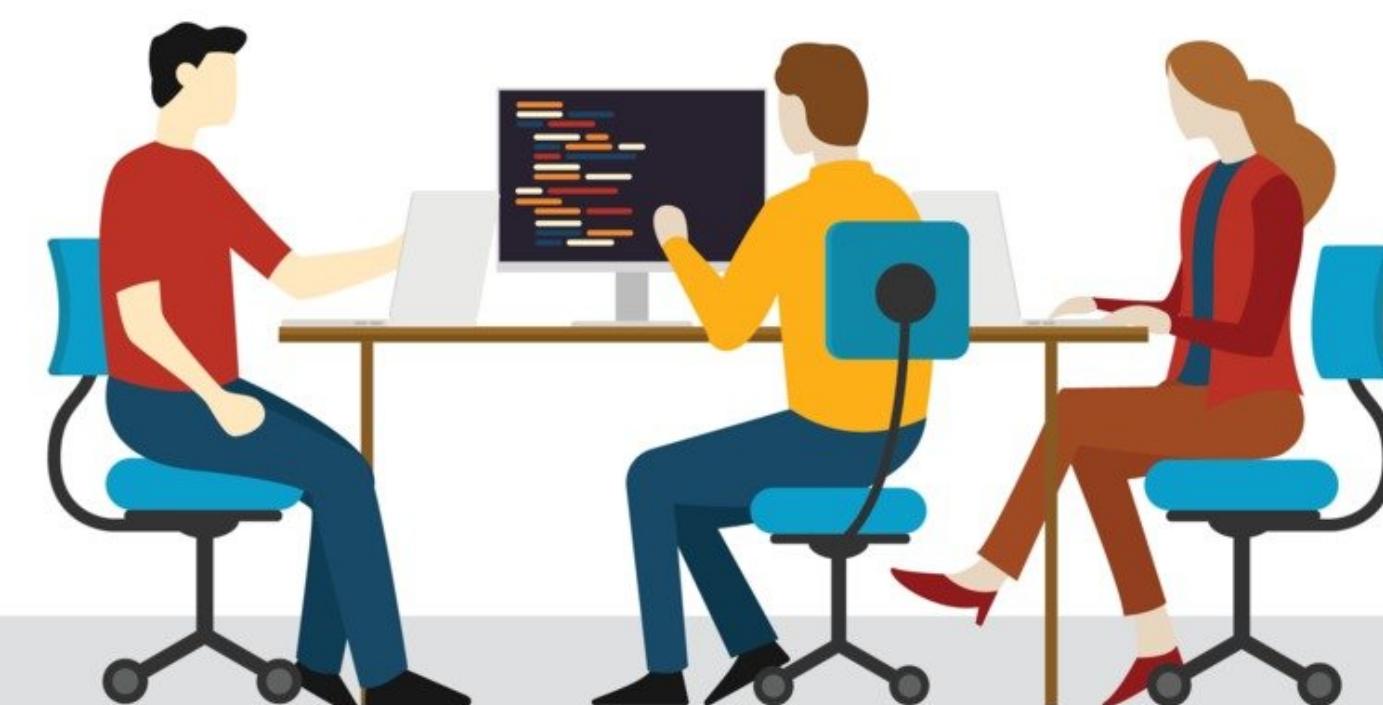
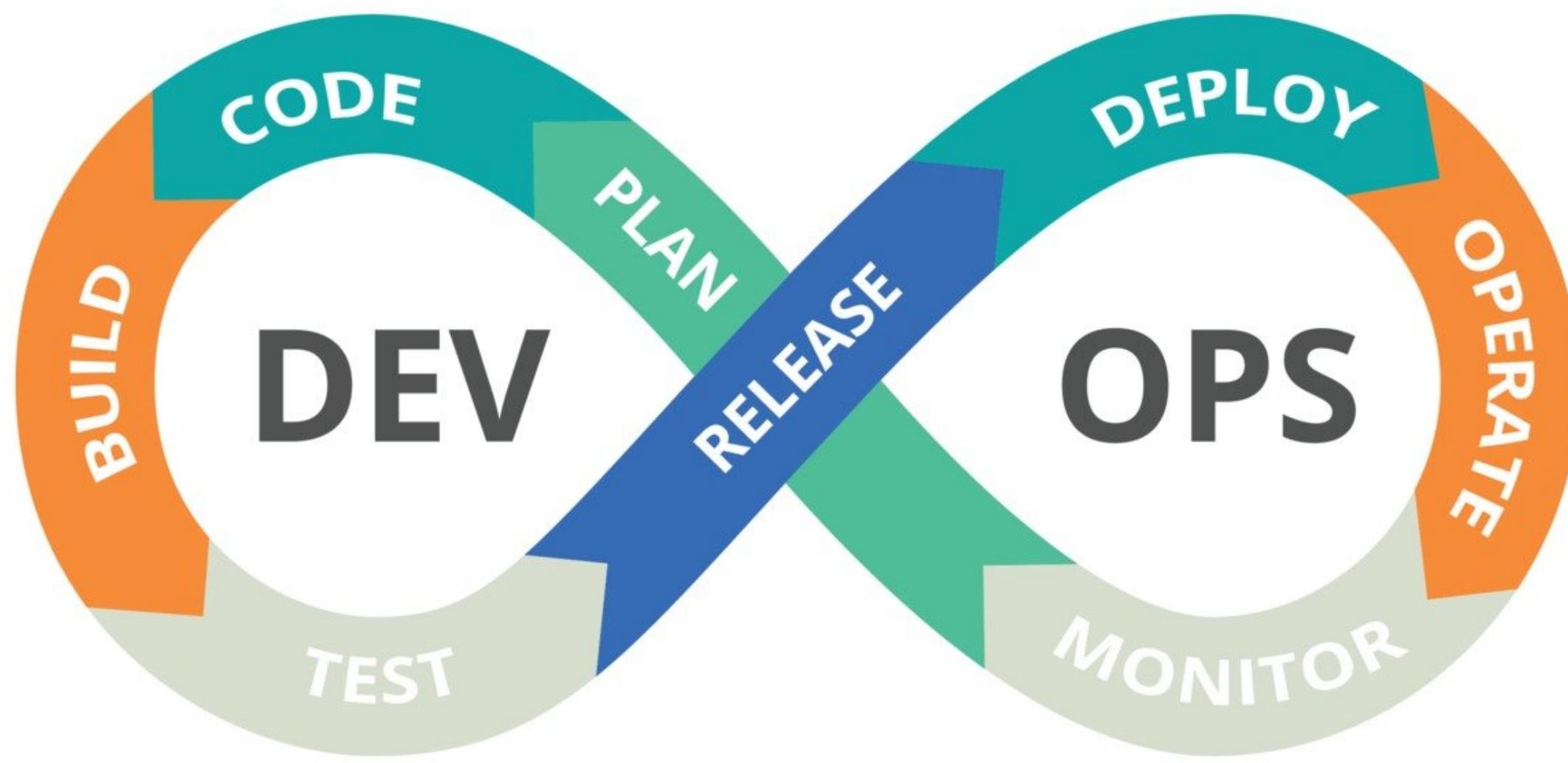
MONITORING & LOGGING

Nắm rõ vai trò của monitoring và logging trong hệ thống thực tế.



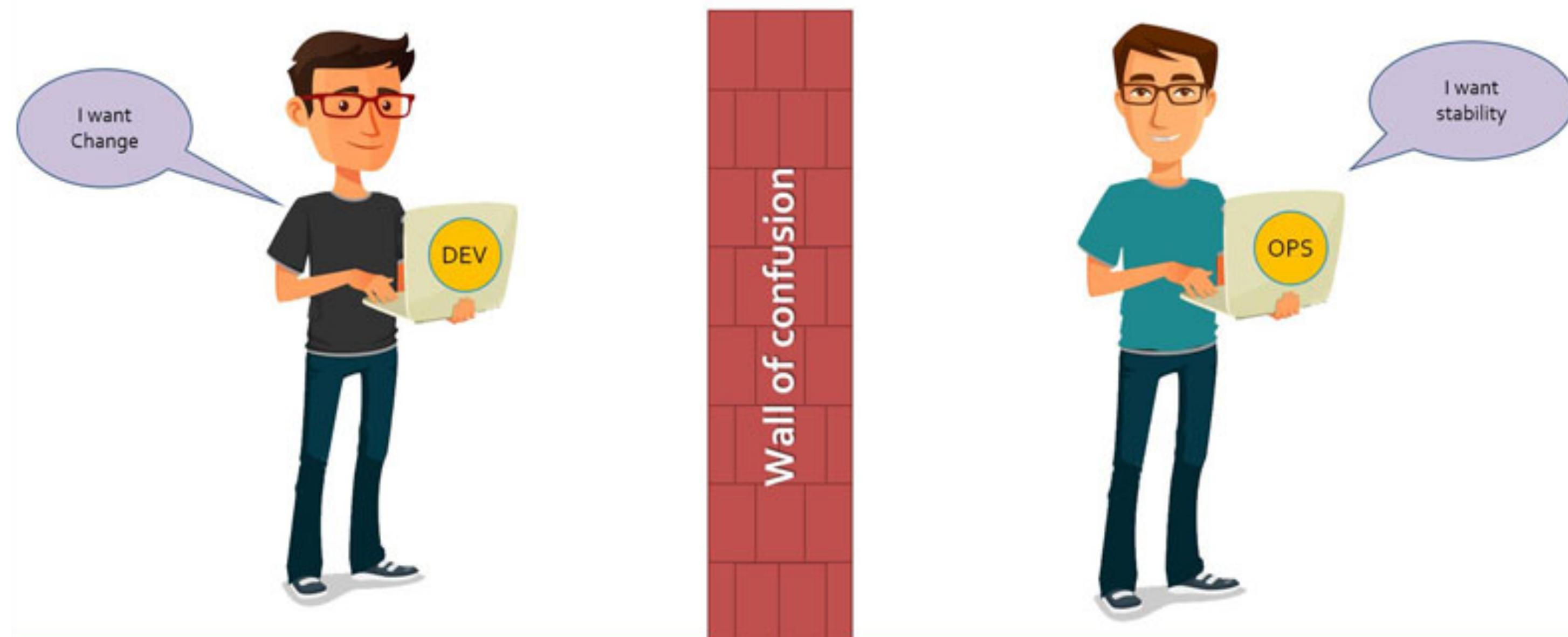
DevOps là gì?

DevOps là gì?



- DevOps là sự kết hợp giữa nhiều triết lý văn hóa, biện pháp thực hành và công cụ giúp tăng khả năng phân phối ứng dụng và dịch vụ của một tổ chức ở tốc độ cao.
- Tốc độ này cho phép các tổ chức phục vụ khách hàng tốt hơn và cạnh tranh hiệu quả hơn trên thị trường.

DevOps là gì?



- Đối với mô hình truyền thống, dev làm việc riêng và operations làm việc riêng, họ sẽ có mục tiêu công việc khác nhau.
- Dev thực hiện xây dựng các tính năng càng nhanh càng tốt >< Operators lại mong muốn hệ thống càng ổn định càng tốt.

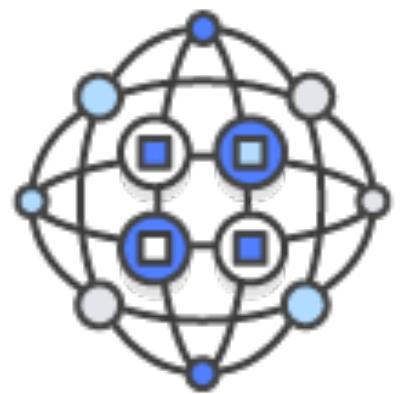
Rào cản giữa Developer và Operator

Cách thức hoạt động của DevOps



- Các nhóm Developer và Operator không còn bị “cô lập”, đôi khi cả 02 nhóm này được gộp thành một nhóm duy nhất trong đó các thành viên làm việc với toàn bộ vòng đời của ứng dụng, từ quá trình phát triển và kiểm thử cho đến triển khai và hoạt động.

DevOps Practices



➤ MICROSERVICES



➤ INFRASTRUCTURE AS CODE



➤ CONTINUOUS INTEGRATION



➤ MONITORING AND LOGGING



➤ CONTINUOUS DELIVERY



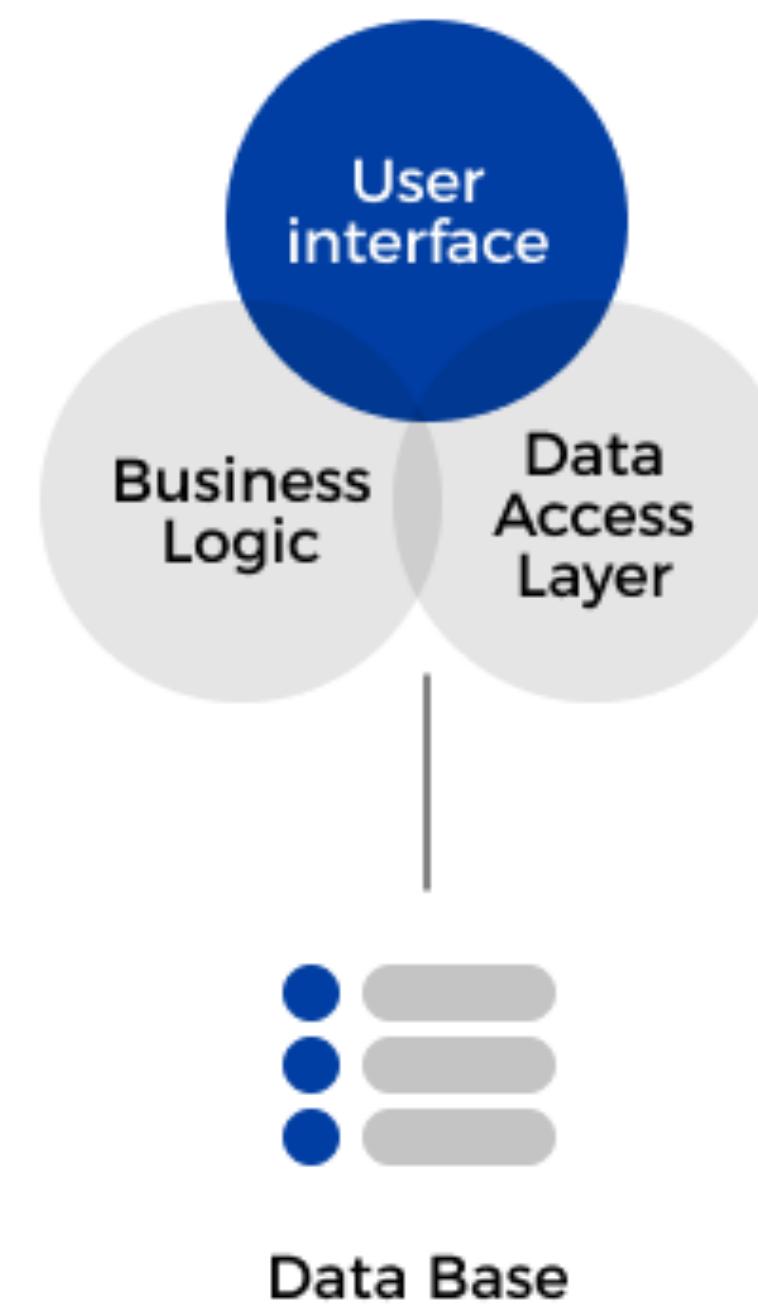
➤ COMMUNICATING AND COLLABORATING



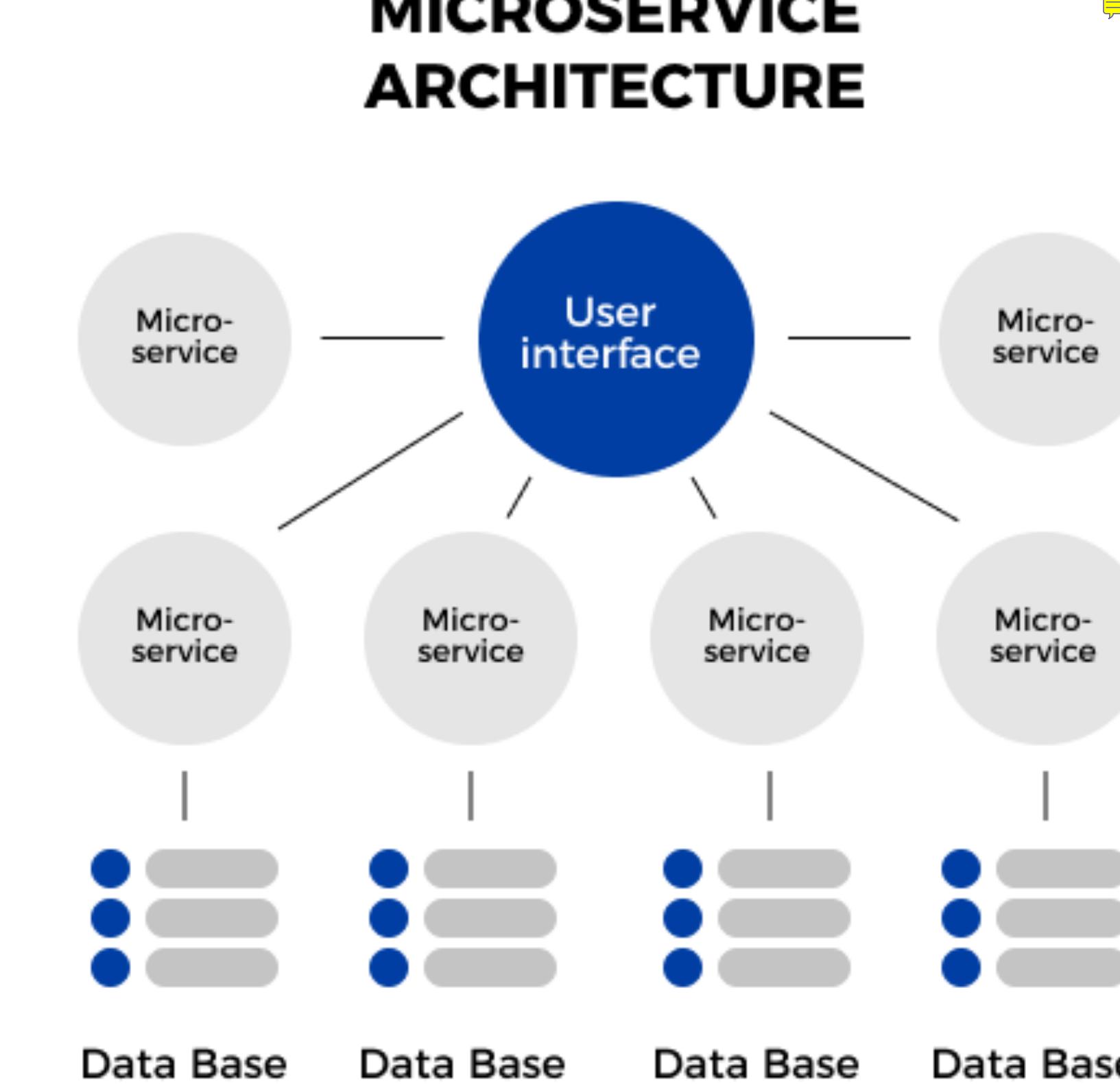
Microservices

Kiến trúc microservice

**MONOLITHIC
ARCHITECTURE**



**MICROSERVICE
ARCHITECTURE**



Modern Design - The Twelve-Factor App

Codebase

One codebase tracked in revision control, many deploys

Dependencies

Explicitly declare and isolate the dependencies

Config

Store configurations in an environment

Backing Services

Treat backing resources as attached resources

Build, release, and, Run

Strictly separate build and run stages

Processes

Execute the app as one or more stateless processes

Port Binding

Export services via port binding

Concurrency

Scale-out via the process model

Disposability

Maximize the robustness with fast startup and graceful shutdown

Dev/prod parity

Keep development, staging, and production as similar as possible

Logs

Treat logs as event streams

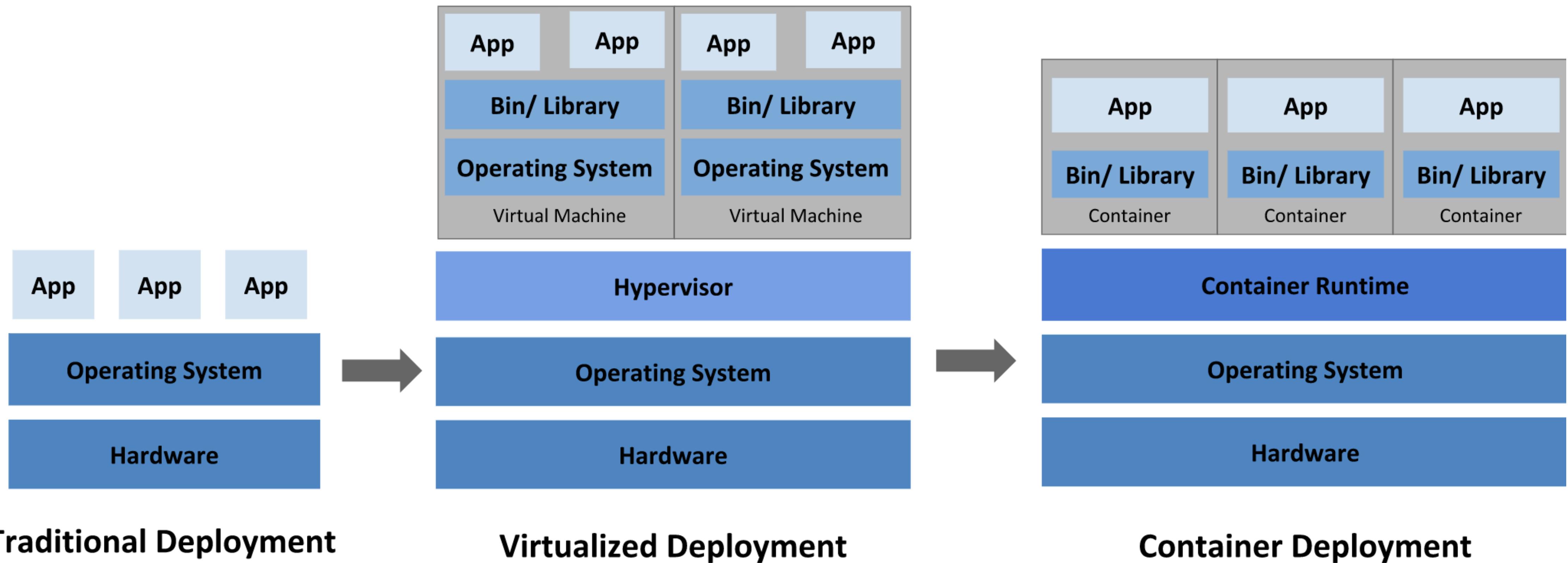
Admin processes

Run admin/management tasks as one-off processes

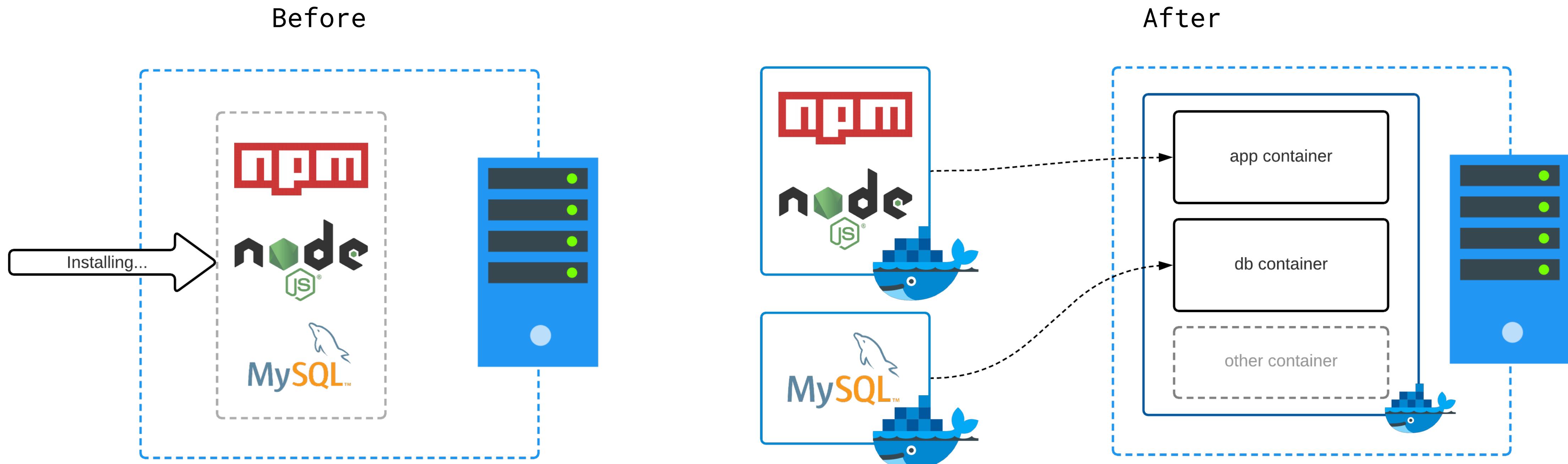


Containers

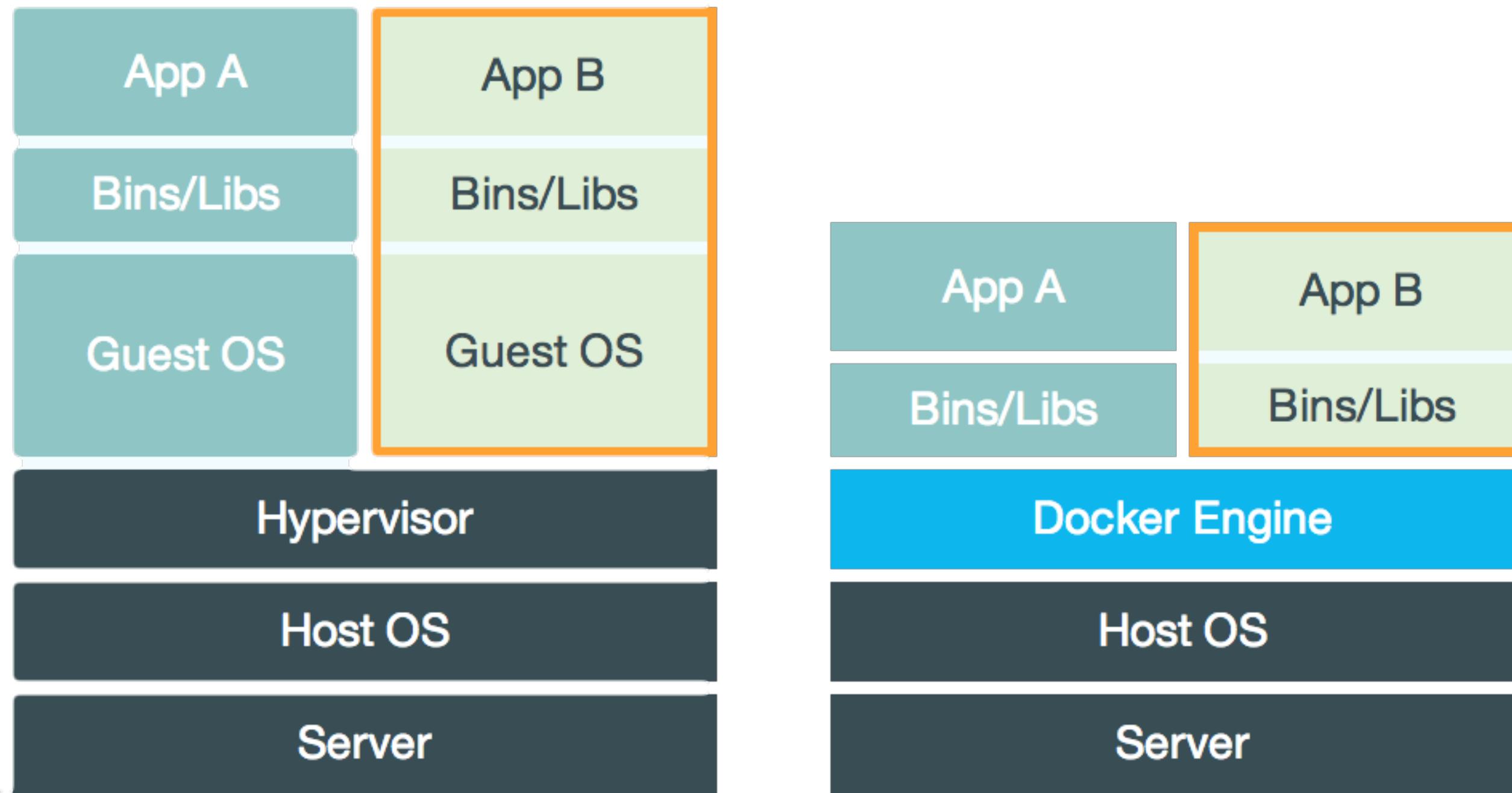
Container là gì?



Tại sao phải sử dụng Docker?

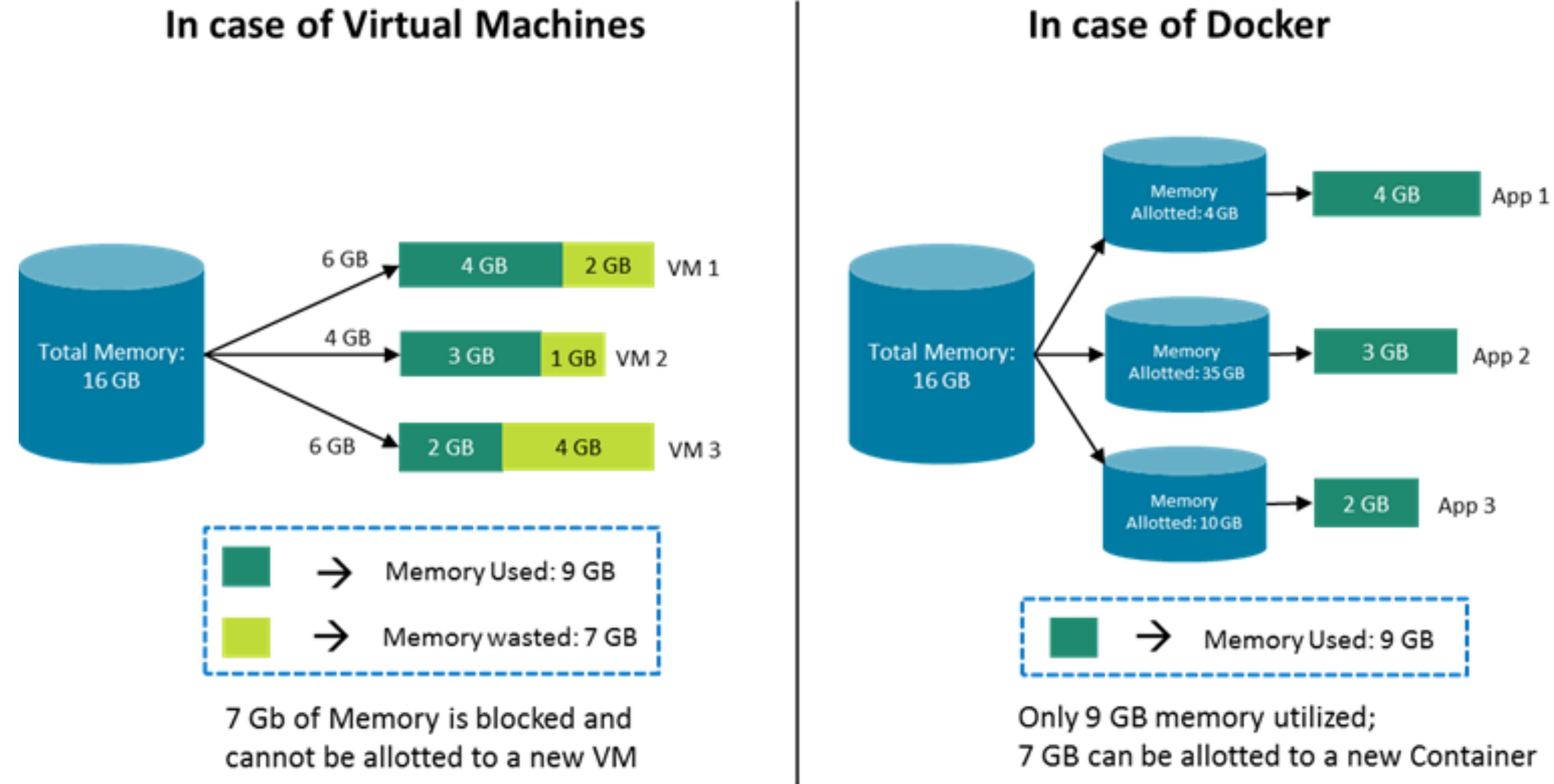


Khác nhau giữa Virtual Machine và Docker



- **Docker:** Dùng chung kernel, chạy độc lập trên Host Operating System và có thể chạy trên bất kỳ hệ điều hành nào cũng như cloud.
- **Virtual Machine:** Cần thêm một Guest OS cho nên sẽ tốn tài nguyên hơn và làm chậm máy thật khi sử dụng.

Khác nhau giữa Virtual Machine và Docker

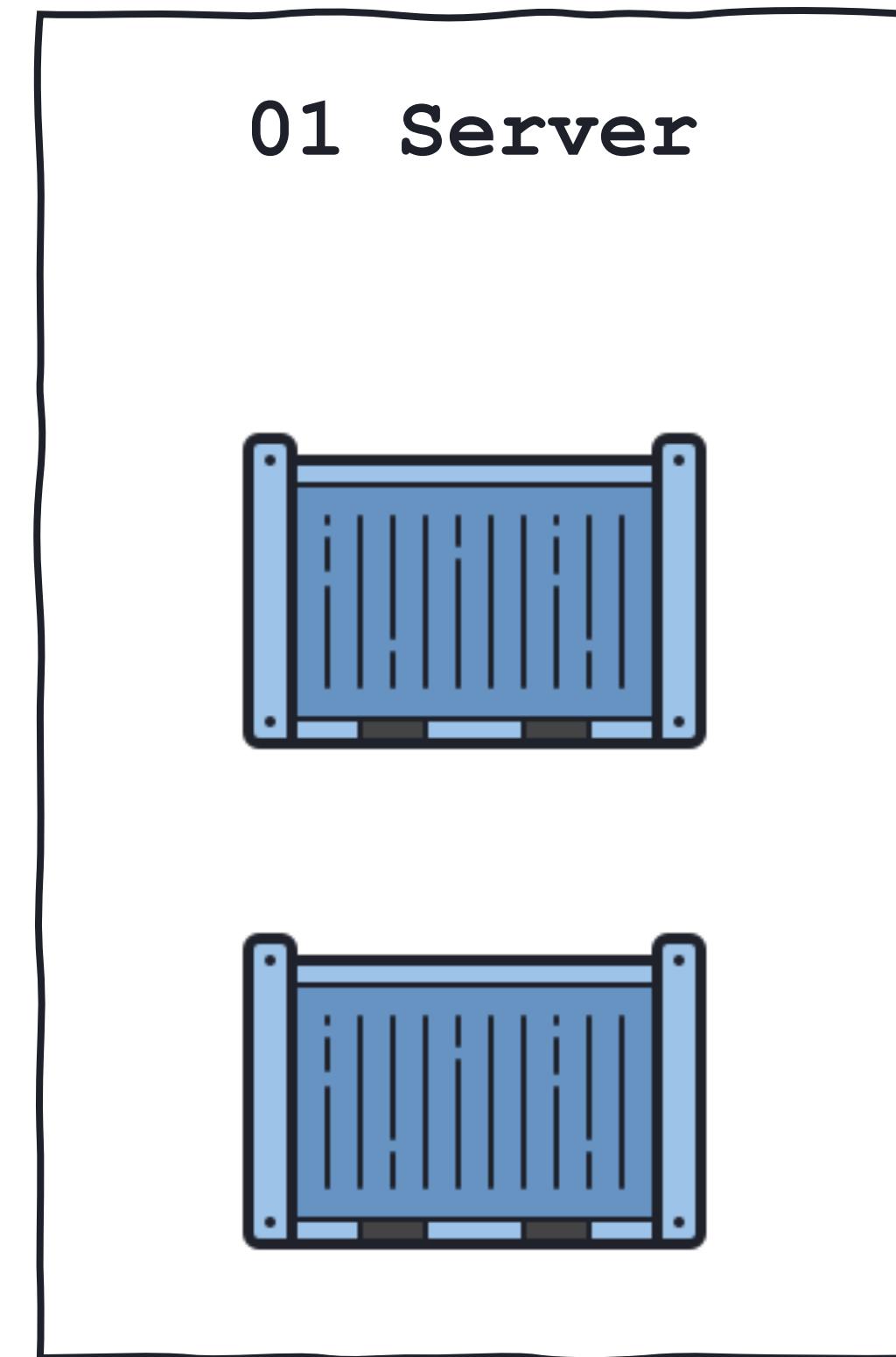




Container Orchestration



Tại sao lại cần orchestration?

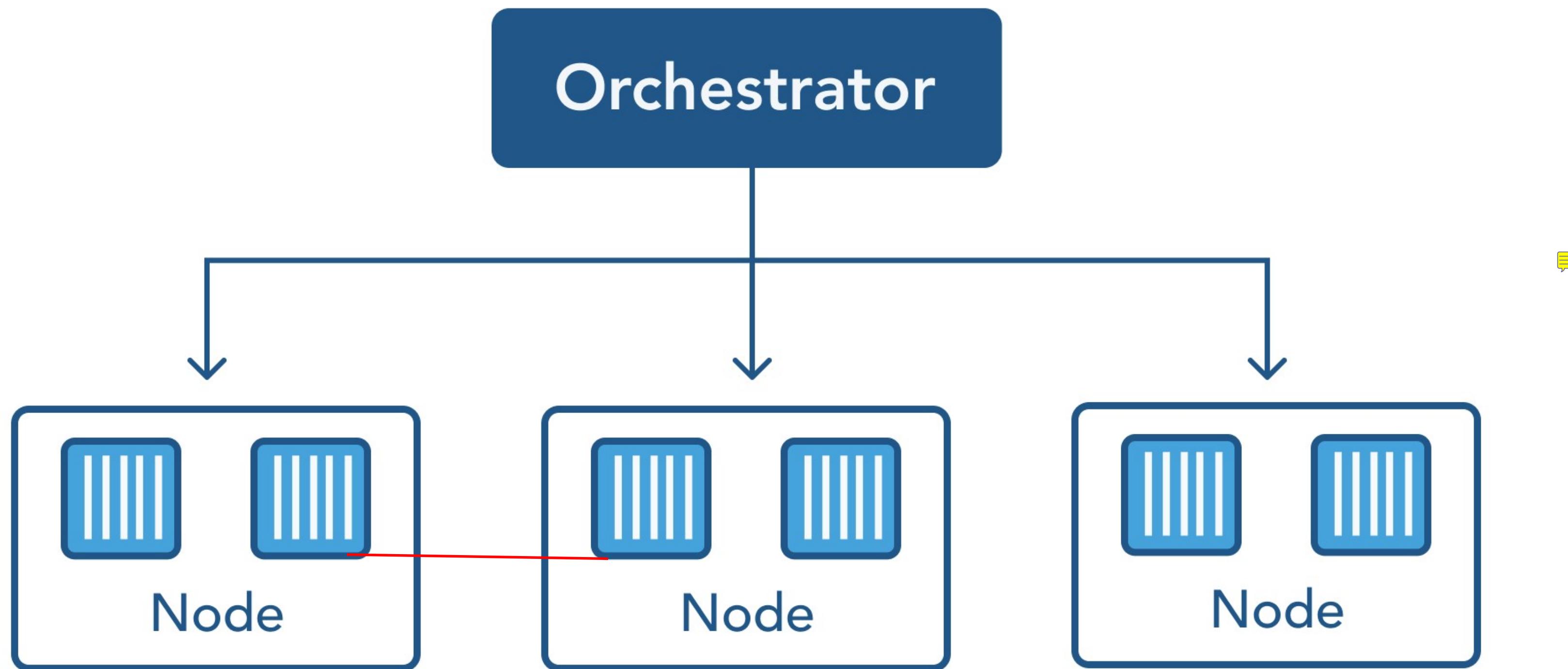


Tại sao lại cần orchestration?

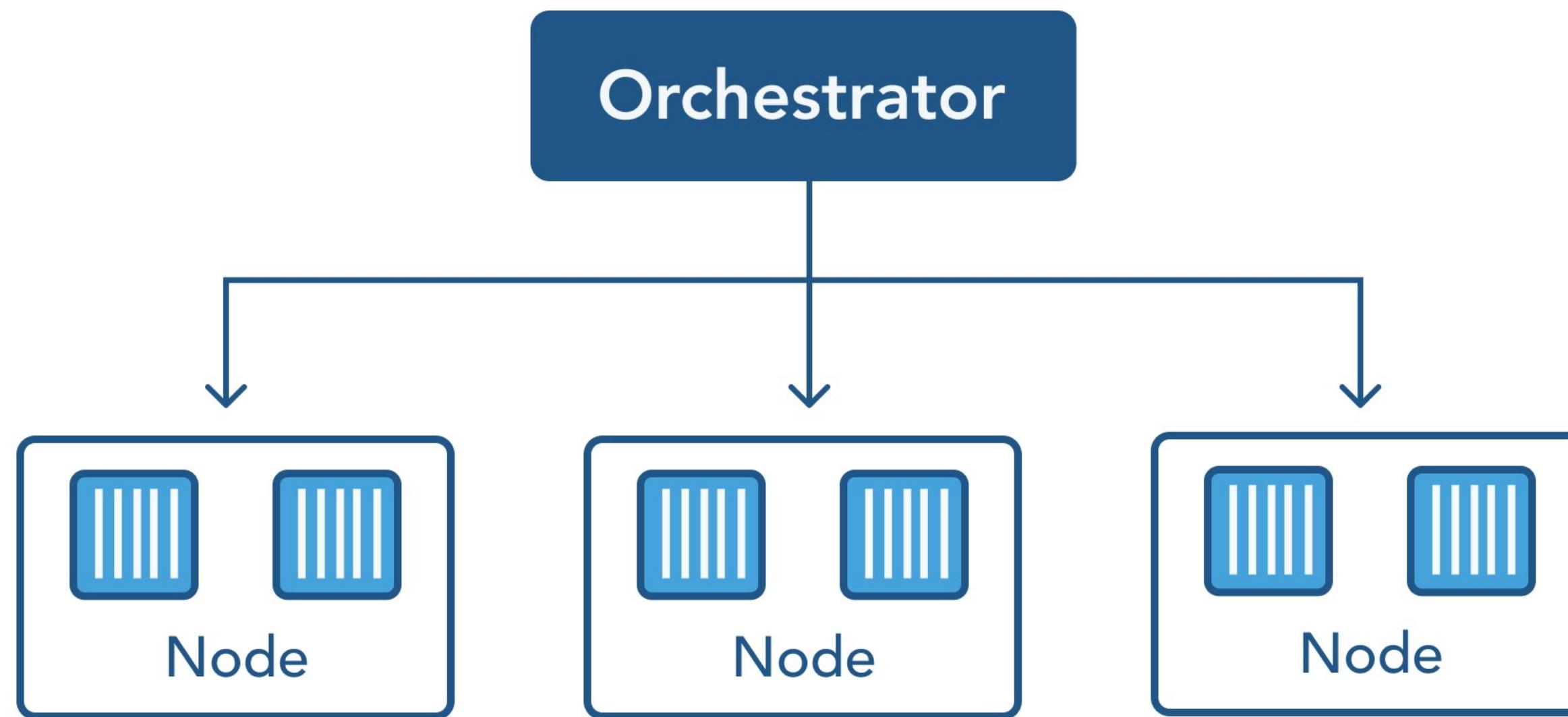
n Server



Container Orchestration là gì?



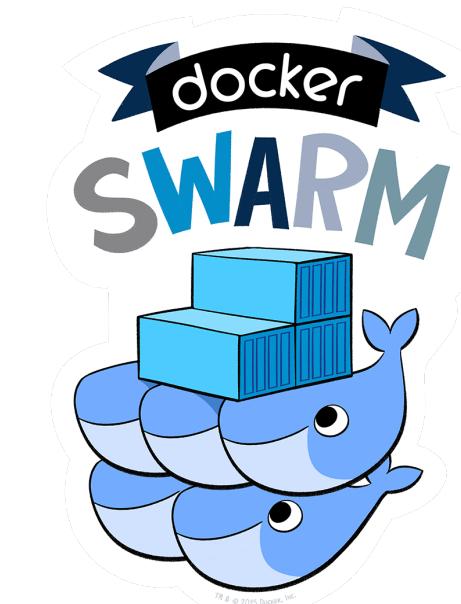
Container Orchestration là gì?



kubernetes



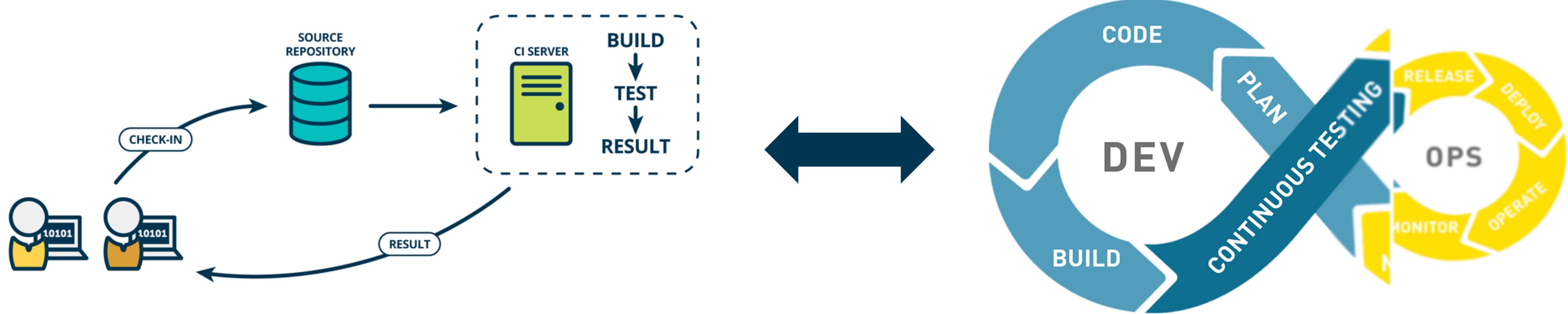
HashiCorp
Nomad



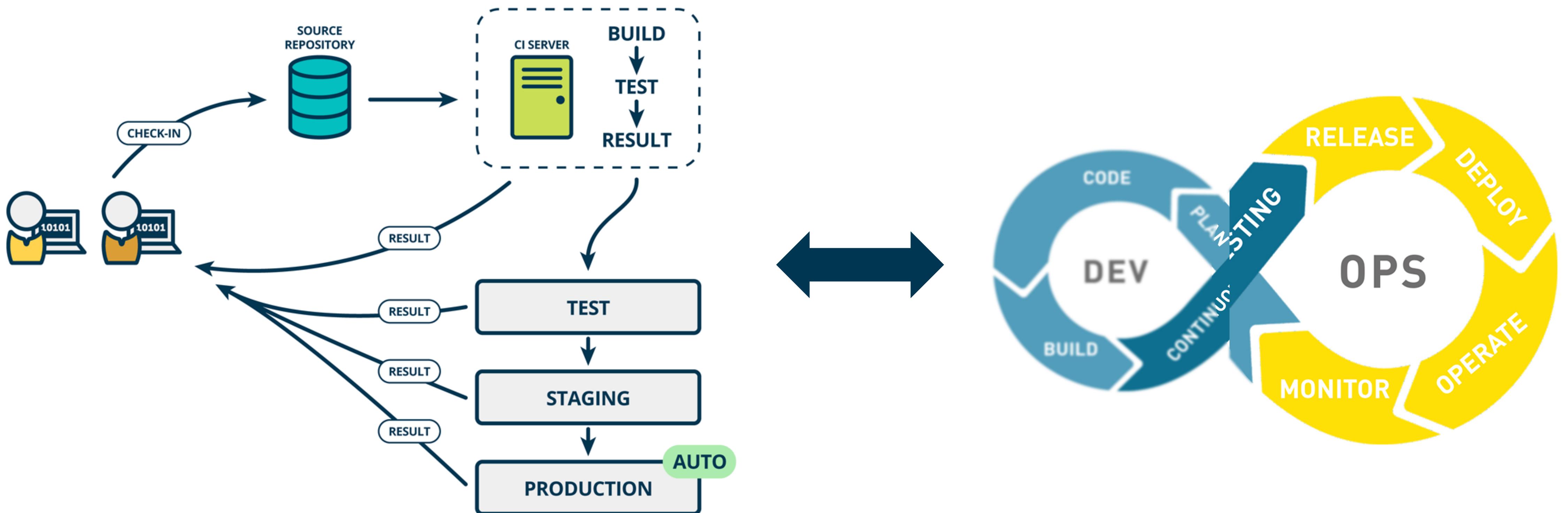


CONTINUOUS INTEGRATION & CONTINUOUS DELIVERY (CI/CD)

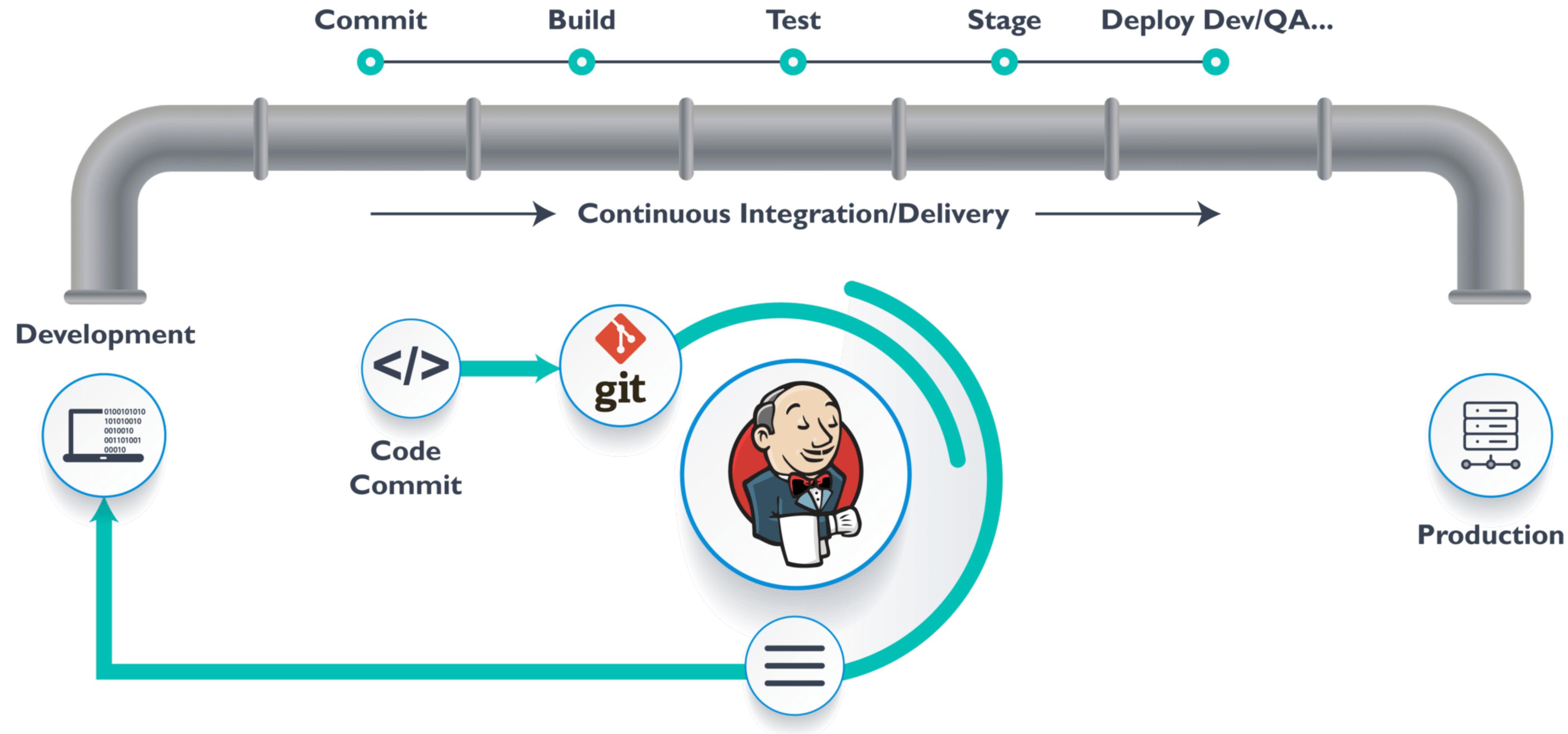
Continuous Integration (CI) là gì?



Continuous Delivery (CD) là gì?



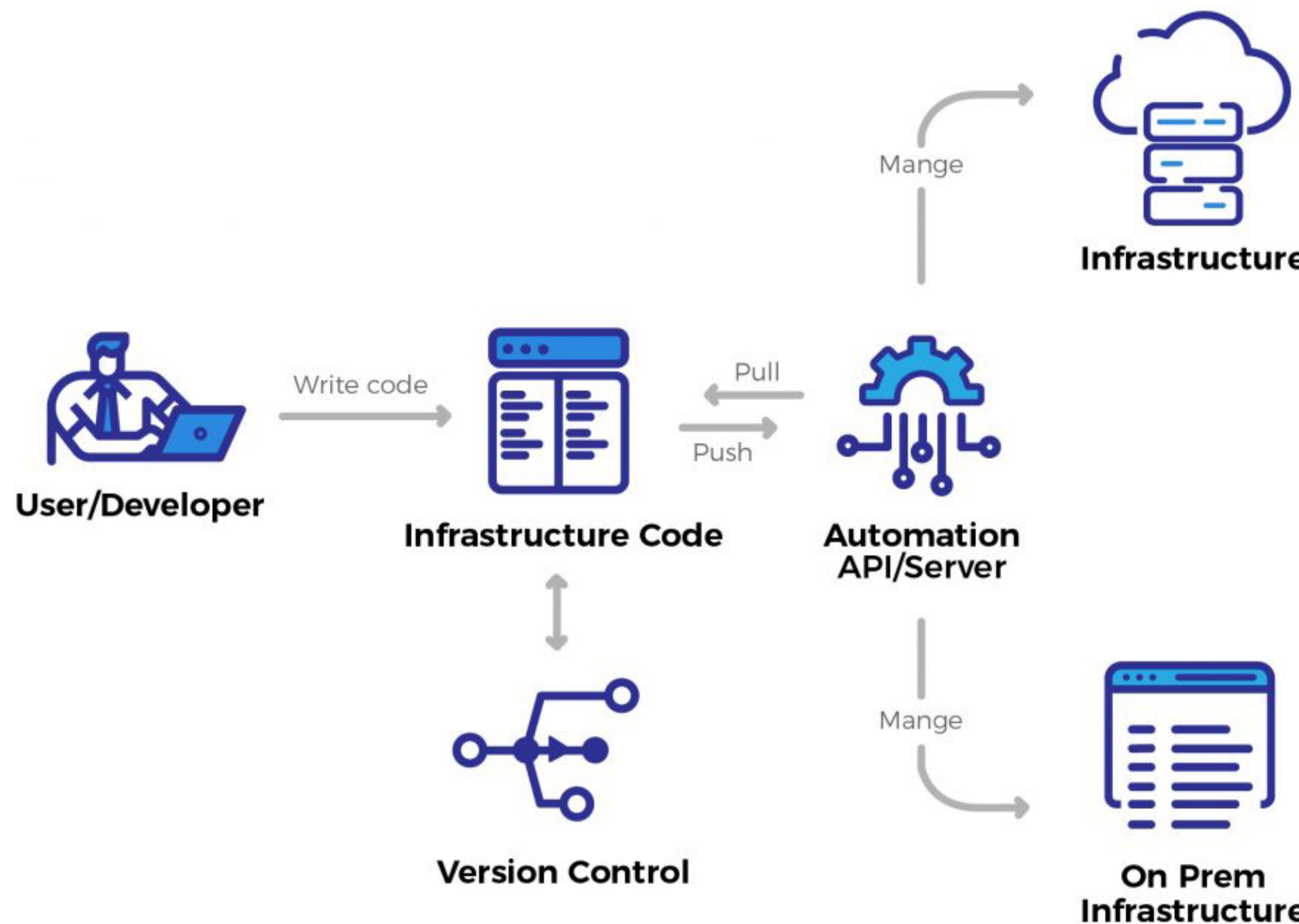
CI/CD với Jenkins





INFRASTRUCTURE AS CODE

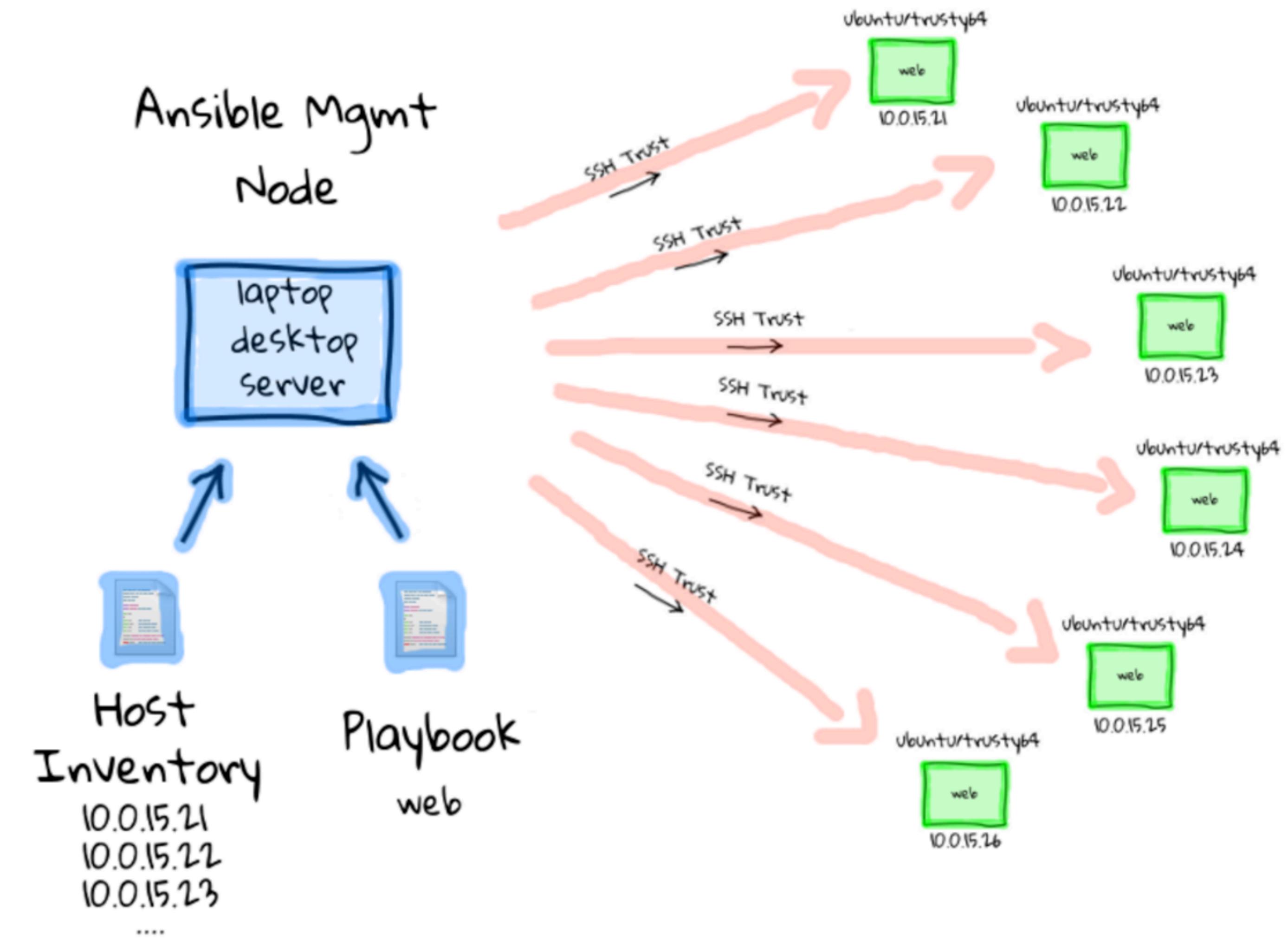
Infrastructure as code là gì?



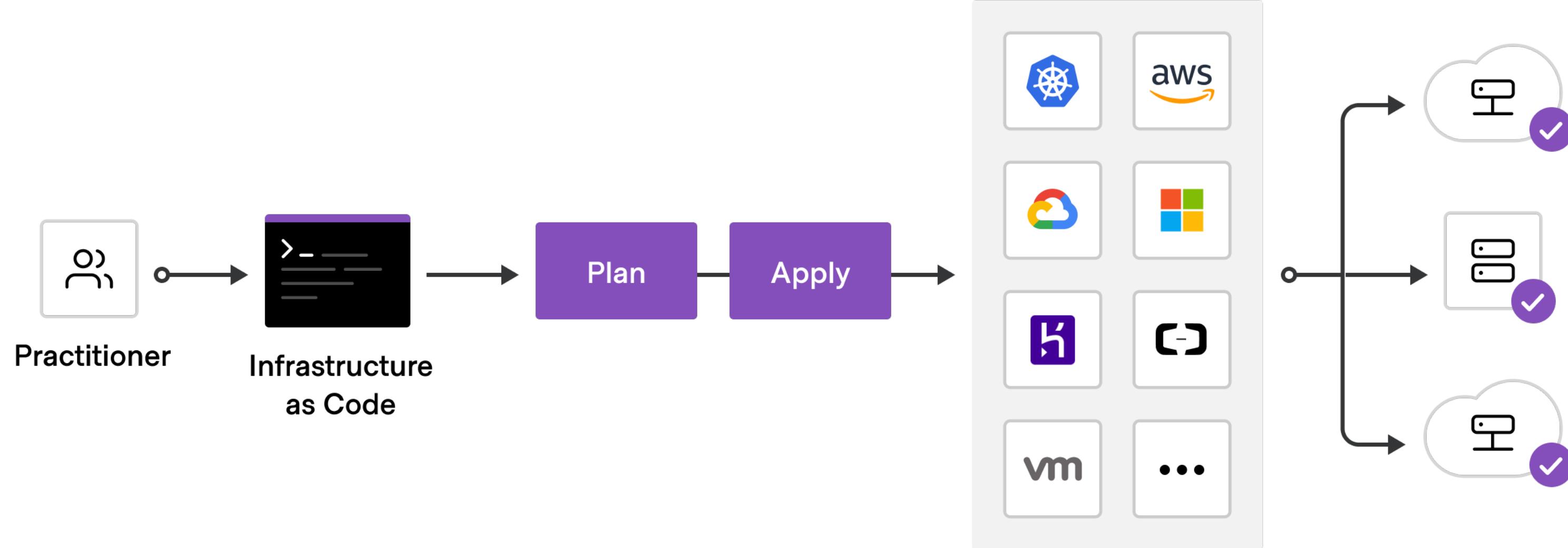
- Sử dụng code để **tự động hóa** hệ thống **vận hành** và **cấu hình** máy chủ, các tác vụ vận hành và nhiều hoạt động khác.
- Đoạn code sẽ được tái sử dụng nhiều lần cho các công việc lặp đi lặp lại, tránh cài đặt cấu hình theo cách thủ công cho các hệ thống vận hành, các ứng dụng hệ thống hoặc phần mềm máy chủ.

Cấu hình hàng loạt máy chủ với Ansible

- Là công cụ mã nguồn mở dùng để quản lý cài đặt, cấu hình hệ thống một cách tập trung và cho phép thực thi câu lệnh điều khiển.
- Có thể thiết kế các kịch bản cài đặt khác nhau và chạy cài đặt cùng lúc cho hàng nghìn máy chủ cùng lúc.

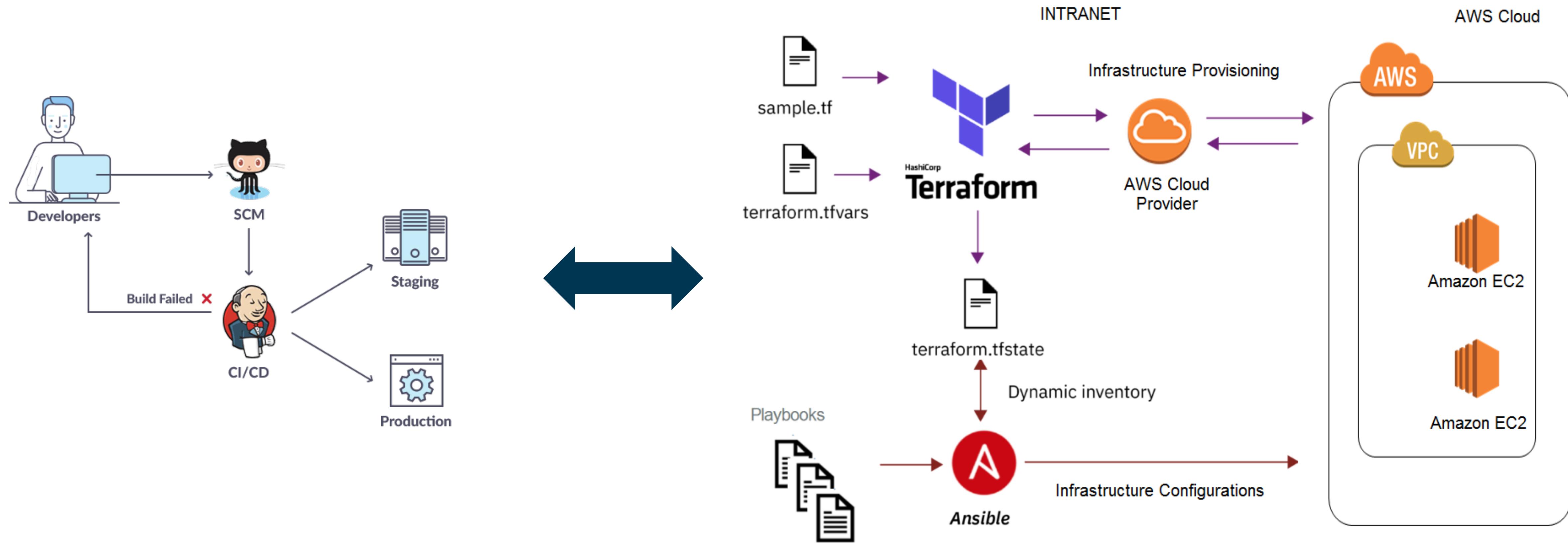


Quản lý tài nguyên Cloud với Terraform



- Là công cụ tự động hóa việc quản lý tài nguyên trên Cloud (ví dụ như compute, network, storage, v.v.) dưới dạng code.
- Với sự phổ biến của public cloud như AWS, Google, Azure, ... và quy trình triển khai app tự động lên Cloud thì cần phải có công cụ tự động quản lý tài nguyên như **Terraform**.

Kết hợp CD với Ansible & Terraform





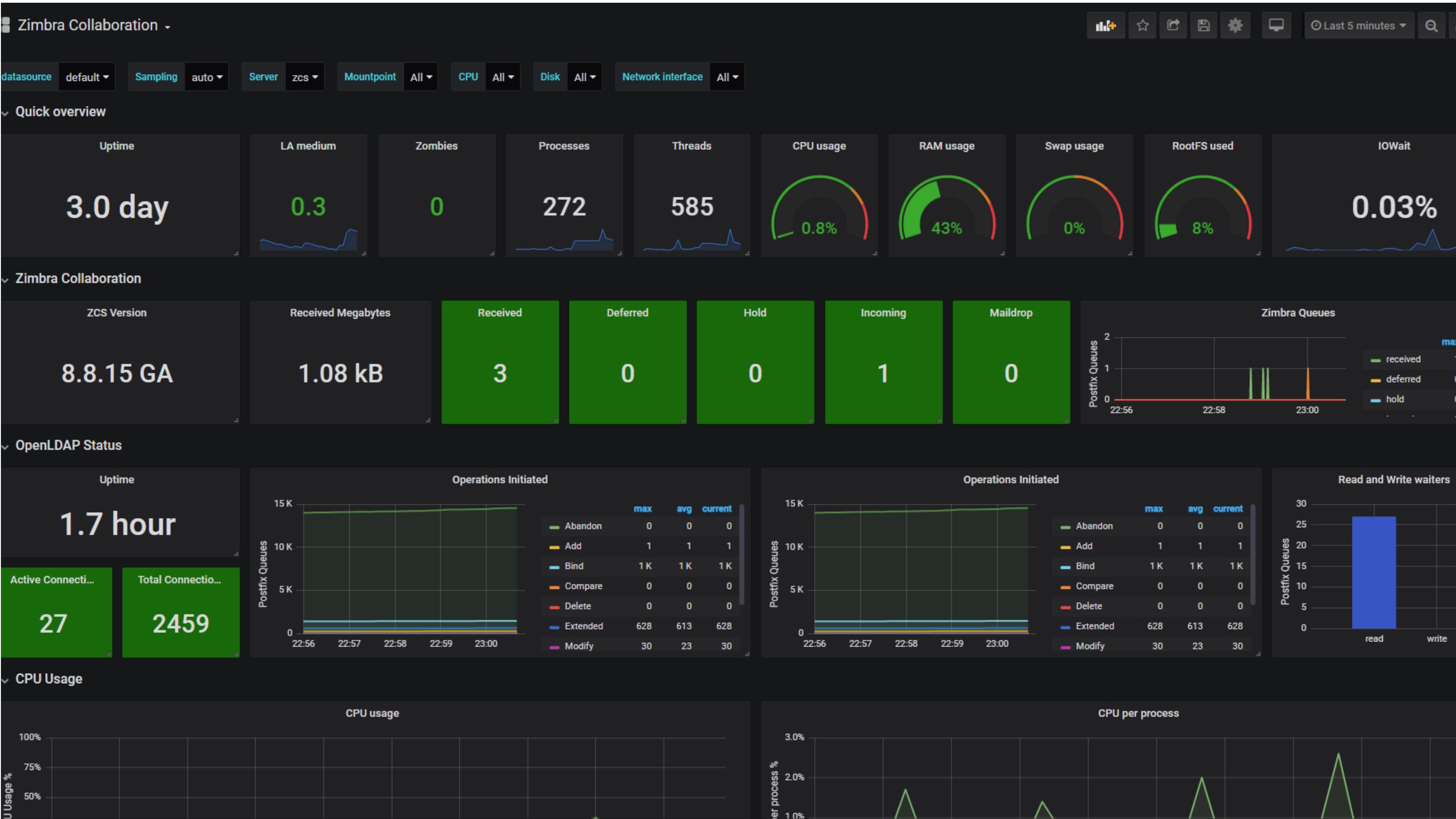
MONITORING & LOGGING

Tại sao cần phải monitoring?



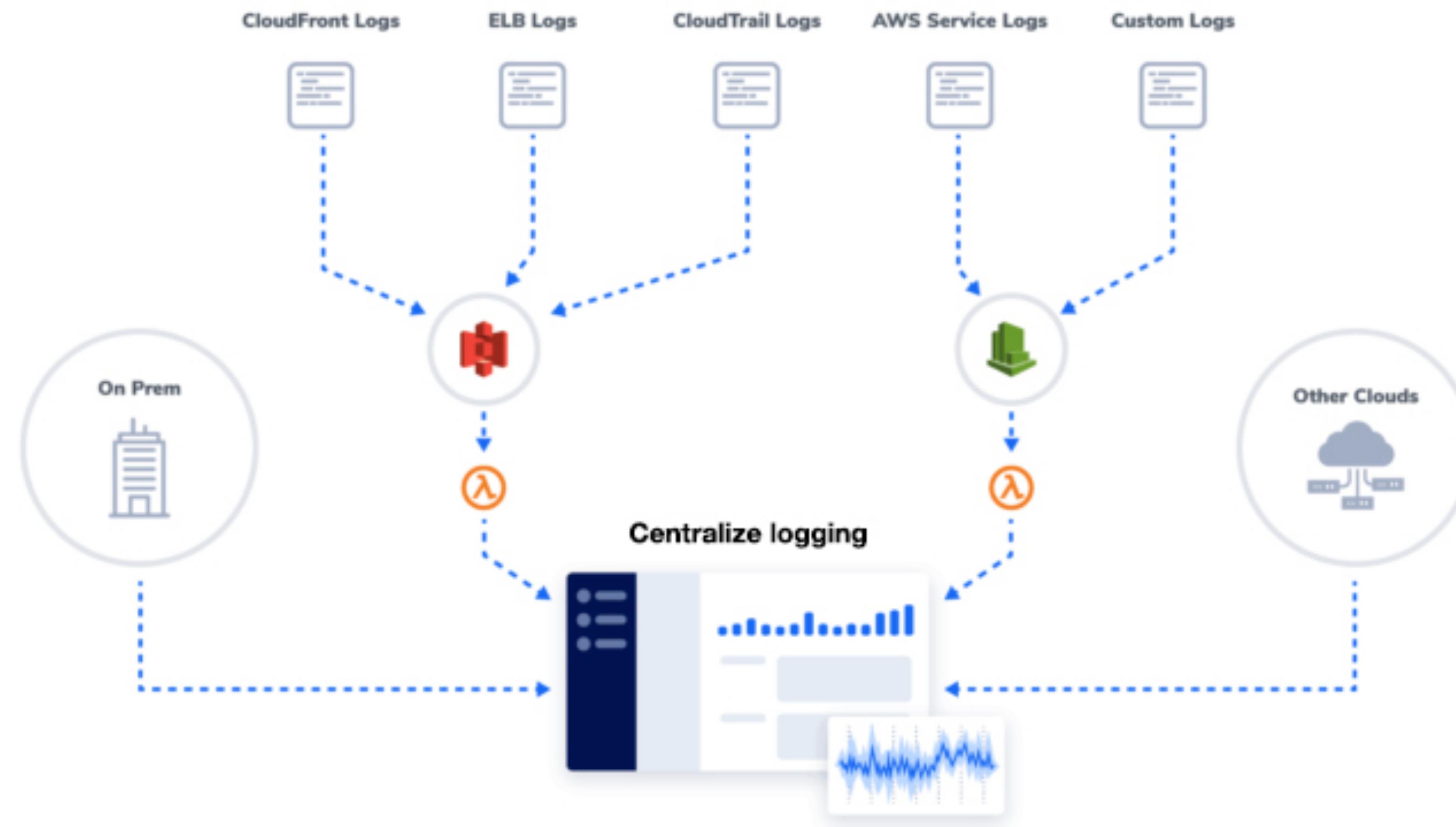
- Đảm bảo hệ thống chạy ổn định, 01 website thường xuyên gặp vấn đề (người dùng không thể sử dụng tính năng mà họ muốn hoặc tốc độ xử lý của trang web cực kì chậm) có thể khiến người dùng mất kiên nhẫn và tìm kiếm các sản phẩm khác thay thế.
- Hệ thống phải luôn ổn định 24/7 để phục vụ khách hàng.

màn hình theo dõi hệ thống với Grafana

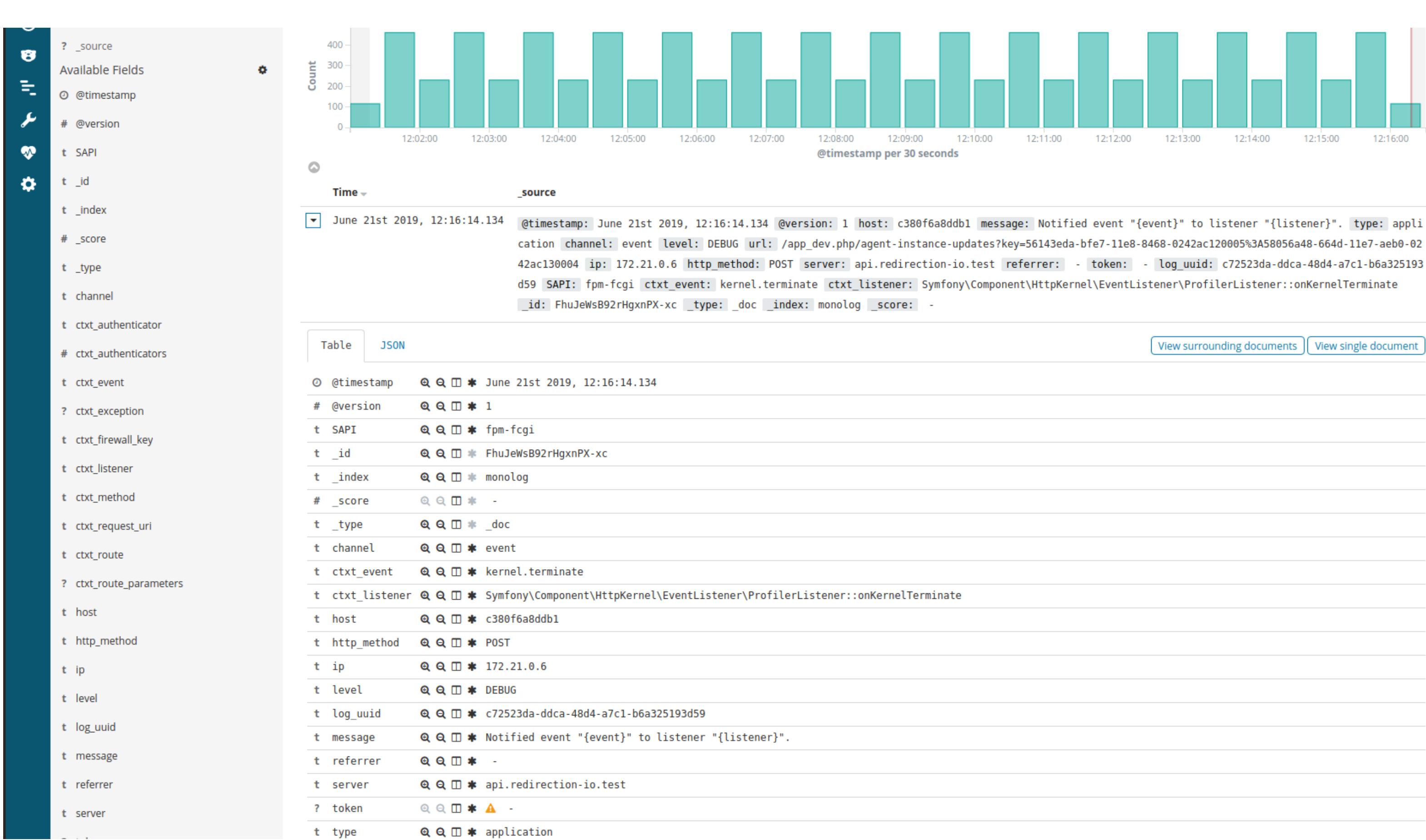


Ví dụ theo dõi các thông số RAM, CPU, trạng thái ứng dụng của máy chủ và hệ thống ứng dụng.

Hệ thống thu thập logs



tìm kiếm và phân tích log ứng dụng



Ví dụ màn hình tìm
kiếm log ứng dụng với
Kibana + Elasticsearch

Thanks for watching

Email: thuongnn666@gmail.com

