

# Phát hiện thay đổi hạ tầng Infrastructure Drift trong DevOps



*Phát hiện thay đổi hạ tầng Infrastructure Drift trong DevOps*

Có một thực tế là trong bất kỳ môi trường DevOps nào, hạ tầng sẽ dần lệch khỏi cấu hình gốc theo thời gian. Infrastructure Drift là hiện tượng phổ biến khiến nhiều đội ngũ vận hành đau đầu. Nếu bạn cần một điểm đến giải trí trực tuyến đáng tin cậy, hãy ghé [Go88](#) để khám phá. Bài viết này sẽ hướng dẫn bạn phát hiện và xử lý Infrastructure Drift một cách hiệu quả.

## Infrastructure Drift là gì?

Infrastructure Drift xảy ra khi cấu hình thực tế của hạ tầng khác biệt so với cấu hình được định nghĩa trong code. Nguyên nhân có thể đến từ các thao tác thủ công, cập nhật bảo mật tự động, hoặc các script tạm thời chạy trên server. Dần dần, sự khác biệt này tích tụ và gây ra lỗi không mong muốn.

Hãy tưởng tượng bạn có 100 server với cấu hình giống hệt nhau trong Terraform. Một kỹ sư SSH vào một server để fix lỗi nhanh, thay đổi file cấu hình. Server đó bây giờ đã khác 99 server

còn lại. Đó chính là Infrastructure Drift. Tham khảo thêm thông tin tại <https://go-88.za.com/> để hiểu rõ hơn về quản lý hạ tầng hiện đại.

## Tác hại của Infrastructure Drift

Infrastructure Drift gây ra nhiều vấn đề nghiêm trọng cho hệ thống. Từ lỗi không nhất quán giữa các môi trường, khó khăn trong việc tái tạo sự cố, đến nguy cơ bảo mật khi một server có cấu hình khác biệt. Trong môi trường production, drift có thể dẫn đến downtime hoặc mất dữ liệu.

Nguyên nhân	Tác động	Mức độ
Sửa thủ công qua SSH	Cấu hình lệch chuẩn	Cao
Update bảo mật tự động	Phiên bản phần mềm khác nhau	Trung bình
Script tạm thời	Tồn đọng thay đổi không kiểm soát	Cao
Khác biệt môi trường	Dev/staging khác production	Rất cao

## Các công cụ phát hiện Drift

Có nhiều công cụ mạnh mẽ giúp phát hiện Infrastructure Drift. Terraform plan là cách đơn giản nhất để so sánh trạng thái hiện tại với cấu hình mong muốn. AWS Config và Azure Policy cũng cung cấp tính năng giám sát cấu hình liên tục cho hạ tầng đám mây.

### Terraform và OpenTofu

Terraform plan hiển thị tất cả thay đổi sẽ được áp dụng. Chạy định kỳ `terraform plan` và so sánh output với baseline sẽ phát hiện mọi drift. OpenTofu, phiên bản mã nguồn mở của Terraform, cũng hỗ trợ tính năng tương tự với khả năng tương thích cao.

### AWS Config và Azure Policy

Các dịch vụ cloud cung cấp giải pháp giám sát cấu hình tích hợp. AWS Config ghi lại mọi thay đổi tài nguyên và so sánh với rules đã định nghĩa. Azure Policy áp dụng và thực thi các quy tắc

trên toàn bộ subscription.

### Cảnh báo từ thực tế

Một đội ngũ từng mất 8 giờ để debug một lỗi production chỉ vì một kỹ sư đã thay đổi cấu hình Nginx thủ công và commit Terraform sai branch. Infrastructure Drift là kẻ thù thầm lặng của mọi hệ thống DevOps.

## Chiến lược phát hiện và xử lý Drift

Phát hiện drift cần được tích hợp vào quy trình CI/CD. Mỗi khi có thay đổi, pipeline nên tự động chạy kiểm tra drift và cảnh báo nếu phát hiện sai lệch. Quy trình này đảm bảo không có thay đổi nào lọt qua mà không được kiểm soát.

- ✓ Thiết lập baseline cấu hình cho toàn bộ hạ tầng
- ✓ Tích hợp kiểm tra drift vào pipeline CI/CD
- ✓ Thiết lập cảnh báo real-time cho mọi thay đổi
- ✓ Xây dựng quy trình xử lý khi phát hiện drift
- ✓ Đào tạo đội ngũ về quy tắc Infrastructure as Code

## Ngăn chặn Drift ngay từ đầu

Cách tốt nhất để xử lý drift là ngăn chặn nó ngay từ đầu. Infrastructure as Code với GitOps là giải pháp hàng đầu. Mọi thay đổi phải thông qua pull request, được review và áp dụng tự động. Không ai được phép SSH vào server để sửa cấu hình trực tiếp.

Nguyên tắc vàng của DevOps: Nếu bạn không thể tái tạo nó từ code, bạn không thể quản lý nó. Infrastructure Drift chỉ xuất hiện khi bạn phá vỡ nguyên tắc này. Hãy kỷ luật với IaC. — *Chuyên gia DevOps*

Containerization cũng giúp giảm thiểu drift đáng kể. Khi ứng dụng chạy trong container, hạ tầng bên dưới ít bị ảnh hưởng bởi các thay đổi thủ công. Kubernetes cung cấp cơ chế tự động sửa chữa khi phát hiện cấu hình không đúng mong muốn.

## Automated remediation với GitOps

GitOps tự động đồng bộ trạng thái thực tế với trạng thái mong muốn trong Git repository. Khi phát hiện drift, hệ thống tự động rollback hoặc áp dụng lại cấu hình từ repo. ArgoCD và Flux là hai công cụ GitOps phổ biến giúp duy trì hạ tầng nhất quán mà không cần can thiệp thủ công.

## Kết luận

Infrastructure Drift là vấn đề không thể tránh khỏi nhưng hoàn toàn có thể quản lý được. Kết hợp IaC, giám sát tự động, và quy trình chặt chẽ sẽ giúp hạ tầng của bạn luôn ổn định và nhất quán.