Tự Động Hóa Cấu Hình Mạng: Khi YAML Gặp Switch và Router!

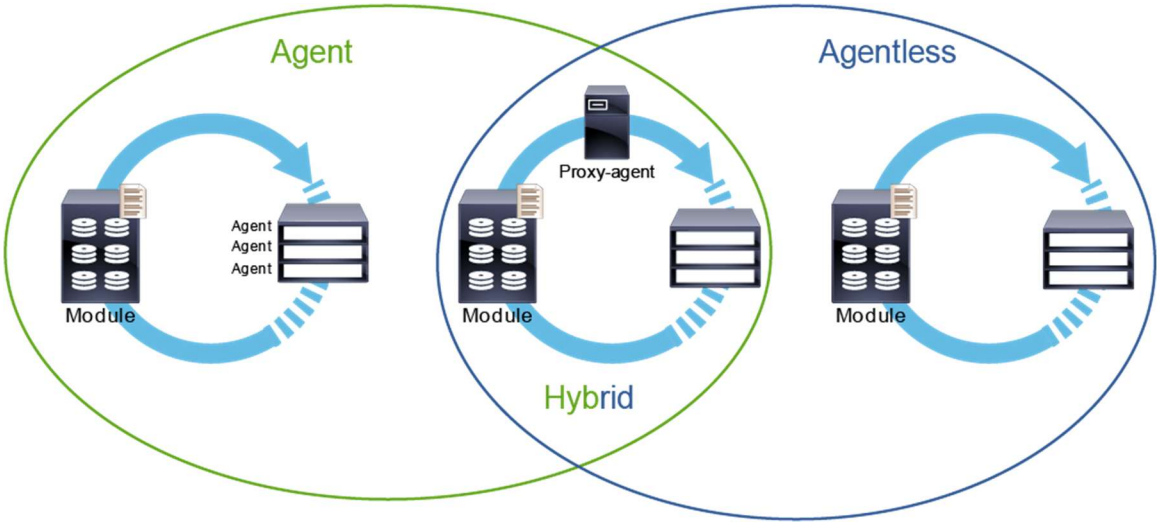
Bài viết dành cho anh em NetDevOps, Automation Engineer và những ai đã từng "mò mẫm CLI giữa đêm khuya chỉ để sửa một cái VLAN. Trong một buổi tối yên tĩnh, bạn nhận được cuộc gọi từ bộ phận vận hành: “Anh ơi, cần thêm VLAN 200 vào 20 thiết bị Core trong vòng 30 phút nữa!” Nếu bạn vẫn đang làm tay, copy-paste CLI từ notepad rồi ssh từng con switch một – thì bài viết này là dành cho bạn.

Vấn đề đặt ra:

Triển khai mạng thủ công = chậm, dễ sai, khó mở rộng.

Giải pháp? Công cụ Quản lý Cấu hình (Configuration Management Tools). Những công cụ này chính là "trợ lý tự động" cho hạ tầng của bạn. Chúng giúp:

- Triển khai cấu hình cho hàng trăm/thậm chí hàng ngàn thiết bị cùng lúc
- Đảm bảo trạng thái cuối mong muốn (Desired State)
- Tự động rollback, kiểm tra lỗi, push config, kiểm tra version
- Tích hợp vào CI/CD workflow như một phần của NetDevOps pipeline



Phân loại công cụ quản lý cấu hình:

- Agent-based: Cần cài "tác nhân" trên thiết bị. → Khó áp dụng cho thiết bị mạng truyền thống (không hỗ trợ OS/Linux).
- Agentless: Giao tiếp qua SSH, API → Phù hợp với switch, router, firewall. Ví dụ: Ansible.
- Proxy-agent: Không cài lên từng thiết bị, nhưng cần một node trung gian (proxy) để điều phối, thường thấy ở SaltStack.

Ví dụ thực tế:

Bạn muốn kiểm tra version IOS của 500 thiết bị Cisco, sau đó update nếu phát hiện phiên bản lỗi thời?

Với Ansible + YAML:

```yaml
- name: Check and upgrade IOS
  hosts: core_switches
  gather_facts: no
  tasks:
    - name: Get IOS version
      ios_facts:
    - name: Upgrade if version is old
      ios_upgrade:
        src: ios_image.bin
        force: yes
      when: ansible_net_version != "17.9.4"
```

Vì sao phải dùng những công cụ này?

- Đồng nhất cấu hình toàn hệ thống
- Tăng tốc độ triển khai ứng dụng mới
- Giảm thiểu lỗi do con người
- Chuẩn hóa các quy trình NetOps
- Kết nối với CI/CD để auto-deploy mạng theo từng lần commit (NetDevOps thật sự)

Những công cụ phổ biến bạn nên biết:

- Ansible – Agentless, dễ học, rất mạnh cho mạng.
- Terraform – Khai báo hạ tầng cloud, nhưng cũng hỗ trợ networking thông qua provider.
- Puppet & Chef – Mạnh ở server/app nhưng áp dụng cho mạng hơi hạn chế.
- SaltStack – Hybrid giữa push/pull, có thể dùng proxy-agent.

Gợi ý học tập & lab:

Bắt đầu với Ansible để automate cấu hình switch/router. Kết hợp thêm Git + GitLab CI để xây dựng pipeline NetDevOps. Hãy thử build playbook để:

- Backup config định kỳ
- Deploy VLANs theo file CSV
- Tự động kiểm tra lỗi security trên ACL

Tóm tắt bài 1

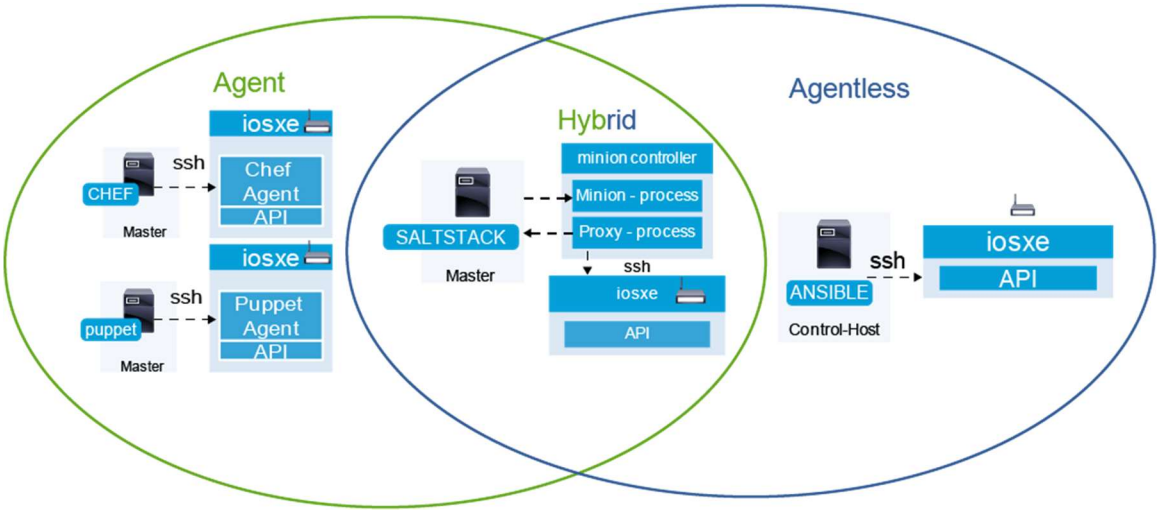
Nếu bạn từng tốn 1 giờ để add một VLAN, thì chỉ cần vài dòng YAML, bạn có thể làm việc đó

trong vài giây, trên toàn mạng. Thế giới đang chuyển mình với Infrastructure as Code – và mạng lưới cũng không nằm ngoài cuộc chơi này. Bạn không cần phải là developer để làm DevOps – bạn chỉ cần công cụ phù hợp và mindset tự động hóa.

So sánh các công cụ quản lý cấu hình cho hạ tầng mạng: Puppet, Chef, Ansible, Salt

Bạn chọn Ansible, Puppet, hay Chef để tự động hóa mạng?

Trong thế giới đang chuyển mình mạnh mẽ sang tự động hóa hạ tầng, đặc biệt là hạ tầng mạng (Network Infrastructure), việc lựa chọn công cụ quản lý cấu hình (Configuration Management) phù hợp là một quyết định chiến lược. Nhưng bạn đã bao giờ tự hỏi: Tại sao Ansible lại được ưa chuộng hơn Puppet, Chef hay Salt trong mảng mạng (networking)?



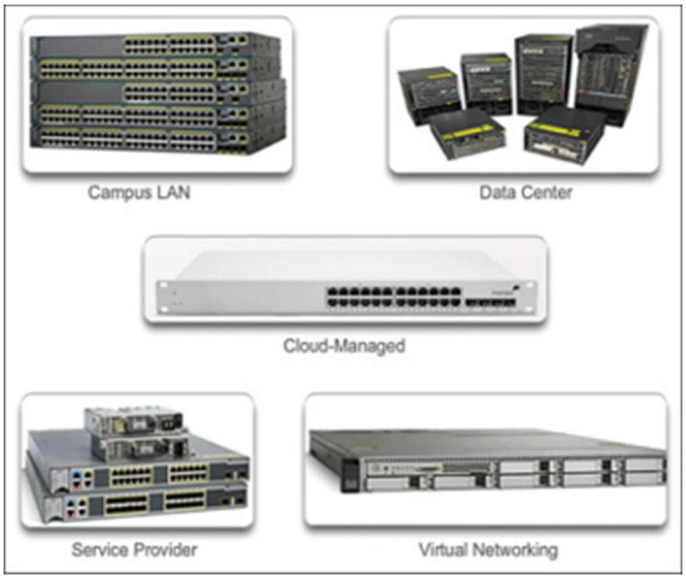
Hãy cùng ôn lại và khám phá từng công cụ nhé.

Puppet – Lão đại kỳ cựu (từ 2005)

- Viết bằng Ruby, dùng manifests để định nghĩa cấu hình.
- Theo mô hình declarative: bạn chỉ cần mô tả trạng thái mong muốn, Puppet sẽ tự xử lý cách đạt được.
- Agent-based: cần cài phần mềm "tác nhân" lên thiết bị được quản lý ⇒ Không phù hợp cho router/switch.
- Có hỗ trợ proxy-agent, nhưng vẫn là rào cản lớn khi triển khai cho thiết bị mạng.

Chef – Đầu bếp của hạ tầng

- Cũng viết bằng Ruby, dùng recipes và cookbooks.
- Cấu trúc giống Puppet, cũng agent-based, declarative.
- Yếu điểm lớn với giới NetDevOps là cần agent trên thiết bị ⇒ khó áp dụng trong mạng truyền thống.



Ansible – Chân ái của anh em NetDevOps

- Viết bằng Python, dùng playbooks (tập hợp các task theo kịch bản).
- Agentless: không cần cài gì lên router/switch ⇒ chỉ cần thiết bị hỗ trợ SSH, REST API, NETCONF, SNMP.
- Học dễ, mở rộng nhanh, tích hợp được với hàng loạt sản phẩm Cisco.

Các nền tảng Cisco hỗ trợ module Ansible:
- ACI, NX-OS, IOS XE, IOS XR, ASA, Meraki, UCS, NSO, AireOS,…
Bạn có thể dùng Ansible để tự động hóa từ campus LAN, datacenter, cloud cho đến nhà cung cấp dịch vụ (SP).
Tài liệu đầy đủ tại: https://docs.ansible.com

SaltStack – Vị mặn đặc biệt

- Viết bằng Python, dùng cơ chế ZeroMQ để truyền thông điệp.
- Mặc định vẫn cần agent (minion) trên thiết bị.
- Có hỗ trợ proxy minion để điều khiển thiết bị mạng không cài được agent, tuy nhiên triển khai phức tạp hơn Ansible.

📌 Vậy công cụ nào cần tiến trình (process) chạy giữa controller và thiết bị?

👉 Đáp án đúng: agent-based configuration management tools

Tóm tắt bài 2

- Nếu bạn làm network automation, chọn Ansible!
- Puppet/Chef/Salt mạnh trong mảng server/cloud, nhưng thiếu lợi thế trong môi trường mạng vật lý.
- Đừng quên: bạn còn có thể tự viết module Ansible cho thiết bị cũ chỉ hỗ trợ Telnet hay SNMP