Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

Hes·so

Téléinformatique – Ch. 22

Automatisation

Vincent Magnin
vincent.magnin@hefr.ch

Objectifs

- Comprendre les enjeux des réseaux actuels et la nécessité d'automatiser.
- Comprendre le principe d'une NFV.
- Comprendre le principe de SDN et du protocole OpenFlow.
- Comprendre le principe d'IaC.
- Connaître quelques outils d'automatisation.

Introduction

Pourquoi automatiser ?

« *L'automatisation permet d'augmenter la productivité des équipes informatiques, de réduire le nombre d'erreurs, de mieux collaborer et de consacrer davantage de temps aux tâches stratégiques.* »

Red Hat, 24 mars 2023

En informatique, on cherche à automatiser des processus pour accélérer la productivité et réduire les erreurs humaines potentielles.

Les besoins d'automatiser

Les besoins des réseaux informatiques, et plus globalement des infrastructures informatiques sont nombreux. Une architecture moderne et complète doit être :

Scalable

La scalabilité fait référence à la capacité d'un système ou d'une application à s'adapter et à évoluer pour répondre à une augmentation ou à une diminution de la demande.

Par exemple, une infrastructure web peut être scalable en dupliquant le nombre de serveurs web prêts à accepter les demandes, et à réduire leur nombre lorsque le nombre de demandes baisse.

Les besoins d'automatiser (2)

Les besoins des réseaux informatiques, et plus globalement des infrastructures informatiques sont nombreux. Une architecture moderne et complète doit être :

Sécurisée

La sécurité peut être automatisée par la détection des menaces en temps réel, la configuration dynamique et automatisée des pare-feu ainsi qu'en utilisant de l'analyse comportementale.

Les systèmes automatisés de sécurité peuvent prendre des décisions sur les accès à certains sites, ressources, dispositifs, en fonction de différentes règles mises en place.

Les besoins d'automatiser (3)

Les besoins des réseaux informatiques, et plus globalement des infrastructures informatiques sont nombreux. Une architecture moderne et complète doit être :

Sans latence

La latence est un phénomène qui doit être le moins présent possible dans les réseaux. L'automatisation permet d'analyser le trafic en temps réel et de définir de nouvelles politiques de routage plus efficientes, grâce notamment aux SDN.

Les réseaux automatisés peuvent également gérer automatiquement la bande passante et l'allouer dynamiquement en fonction des besoins et des quotas des entités les utilisant.

Les besoins d'automatiser (4)

Les besoins des réseaux informatiques, et plus globalement des infrastructures informatiques sont nombreux. Une architecture moderne et complète doit être :

Fiable

La fiabilité d'un réseau réside dans sa capacité à fonctionner de manière continue, sans interruption, même lorsque des pannes ou de la surcharge réseau sont présentes.

L'automatisation permet de répliquer des équipements réseaux si ces derniers sont surchargés ou en panne. Elle est également en capacité de déployer des correctifs et des mises à jour sur les équipements beaucoup plus rapidement que si il était nécessaire de les mettre à jour manuellement.

Il existe de nombreux autres besoins que les réseaux ont, et que l'automatisation permet de résoudre.

Network Function Virtualization

Une NFV (Network Function Virtualization) est une approche qui consiste à virtualiser les fonctions réseau traditionnellement exécutées par des équipements matériels dédiés. Un switch, routeur, pare-feu, proxy... peut être représenté par une machine virtuelle.

Cela permet de dupliquer les équipements réseaux facilement (il suffit de créer une deuxième, ou 3ème, ou Nème machine virtuelle), et d'économiser de l'argent.

Une NFV permet de répondre au besoin de centraliser la gestion des fonctions réseaux, étant donné que le nombre d'équipements et de fonctions ne fait que croître. Les NFV sont également beaucoup plus rapides à déployer que leurs équivalents matériels, et sont plus évolutifs.

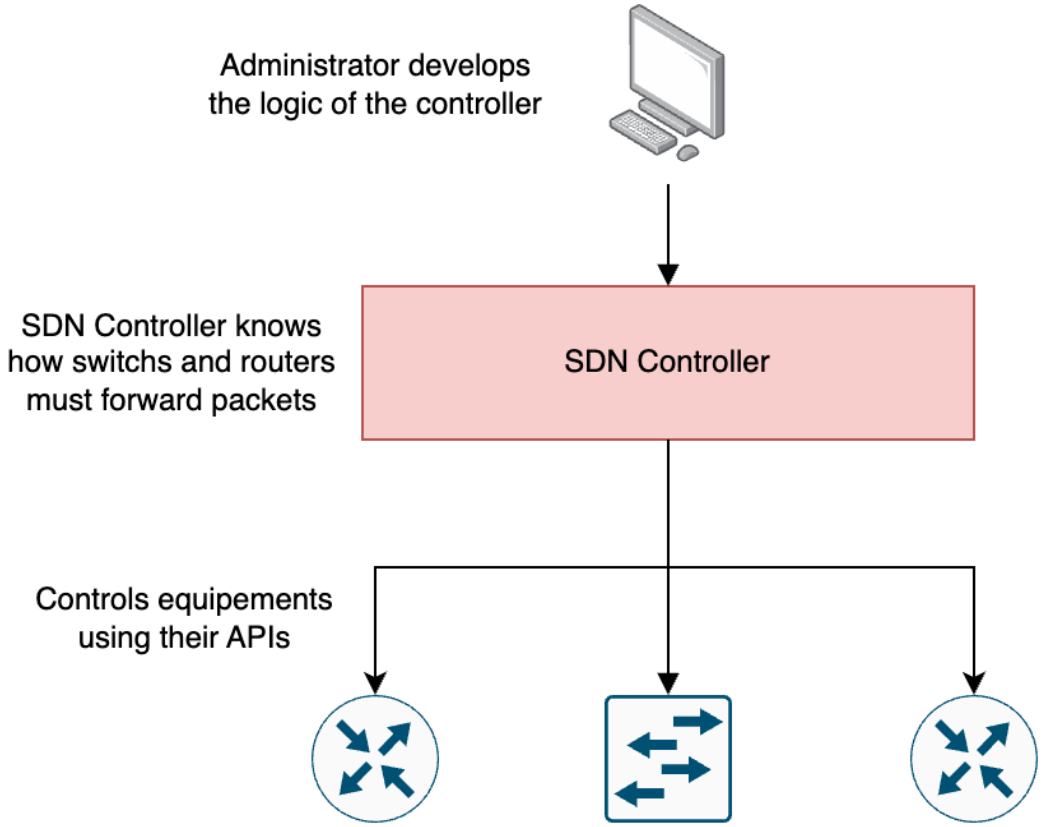
Software Defined Network

Un SDN (Software Defined Network) est un réseau dont le comportement global des équipements est défini par un contrôleur.

Le comportement des équipements est développé puis placé dans un contrôleur. Les switchs, routeurs et autres équipements pourront alors être pilotés par ce contrôleur.

Si un équipement ne sait pas comment router un paquet, il peut demander à son contrôleur.

Un exemple de protocole SDN est OpenFlow.



OpenFlow

OpenFlow est un protocole utilisant le port 6633/TCP qui sert à gérer des équipements réseaux en suivant le principe des Software Defined Networks.

Un équipement OpenFlow analyse les entêtes des paquets pour les associer à des flux : ensembles de paquets qui partagent les mêmes caractéristiques (entêtes identiques).

Le contrôleur SDN OpenFlow configure les équipements en indiquant comment réagir en fonction des flux entrants. La gestion des flux remplace la *forwarding table* traditionnelle des switchs et la *routing table* des routeurs par une nouvelle table qui s'appelle la *flow table* (table de flux).

MAC src	MAC dst	IP src	IP dst	TCP src	TCP dst	Instructions
*	00:11:22:33:44	*	*	*	*	Output port 1
*	*	*	10.2.0.0/24	*	*	Output port 2

OpenFlow (2)

Avec OpenFlow, les équipements réseaux ont la possibilité d'être configurés de 3 manières :

Configuration réactive : Chaque équipement remplit sa *flow table* au fur et à mesure. Dès qu'un nouveau flux arrive et qu'il ne sait pas quoi en faire, il demande à son contrôleur. Ce dernier lui répond, et l'équipement réseau gardera en mémoire la décision pour toute la durée du flux.

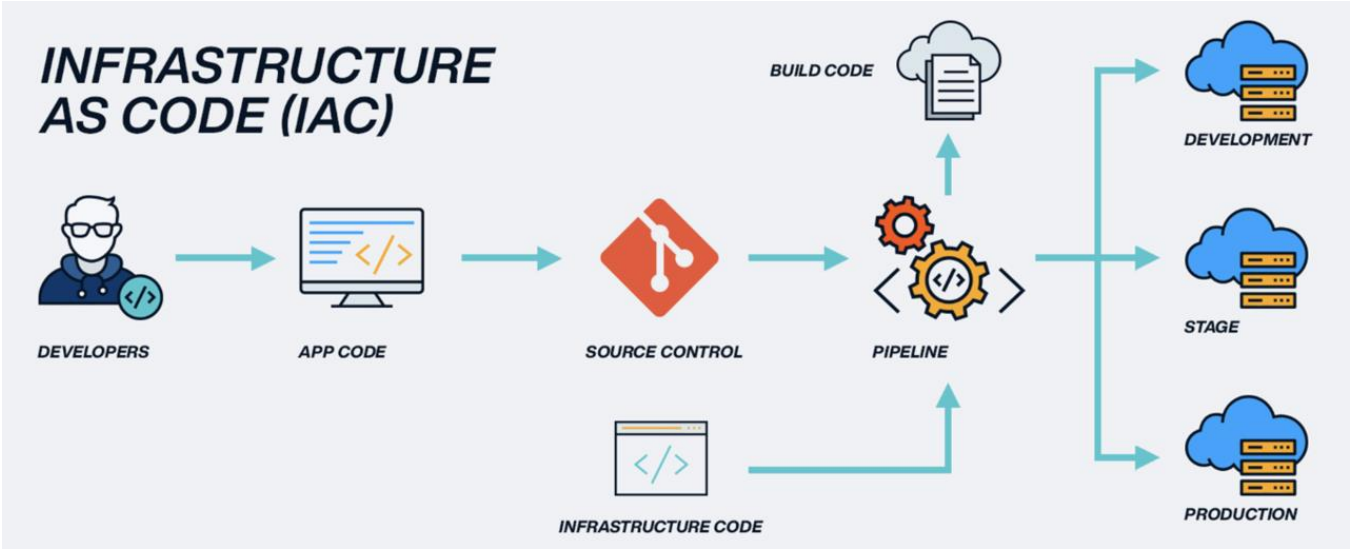
Configuration pro-active : Chaque équipement est pré-configuré par le contrôleur, qui pré-remplit la *flow table* de l'équipement à l'avance. Si un paquet qui arrive ne correspond à aucune entrée dans la *flow table*, alors l'équipement *drop* le paquet.

Configuration hybride : Allie les avantages de la configuration pro-active pour les flux importants et connus d'avance aux avantages de la configuration réactive pour le trafic granulaire.

Infrastructure as Code

L'IaC (Infrastructure as Code) est un principe selon lequel les administrateurs peuvent coder leur infrastructure, puis la déployer avec des outils spécialisés.

Le développeur d'infrastructure peut définir des machines virtuelles, des routeurs, réseaux, firewalls, load-balancers... Tous les éléments nécessaires à une infrastructure informatique. Ces infrastructures deviennent reproductibles facilement, versionnable (par exemple avec Git), documentables, ré-utilisables...

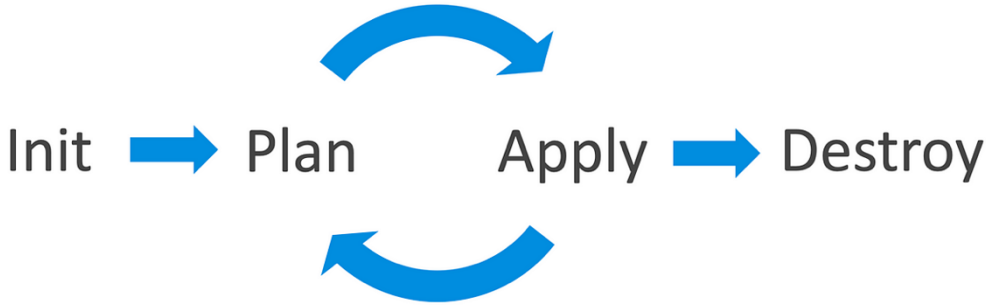


Outils d'automatisation – Terraform

Terraform est un outil d'IaC développé par HashiCorp, et qui se connecte aux APIs de différents *cloud providers* pour déployer une infrastructure.

Terraform a la capacité de créer des routeurs, des machines virtuelles, de les connecter ensemble, de créer des réseaux, des switchs virtuels...

Le développeur doit coder son infrastructure, puis la déployer avec Terraform avec une commande. De la même manière, il peut la mettre à jour et la supprimer entièrement avec une seule commande.



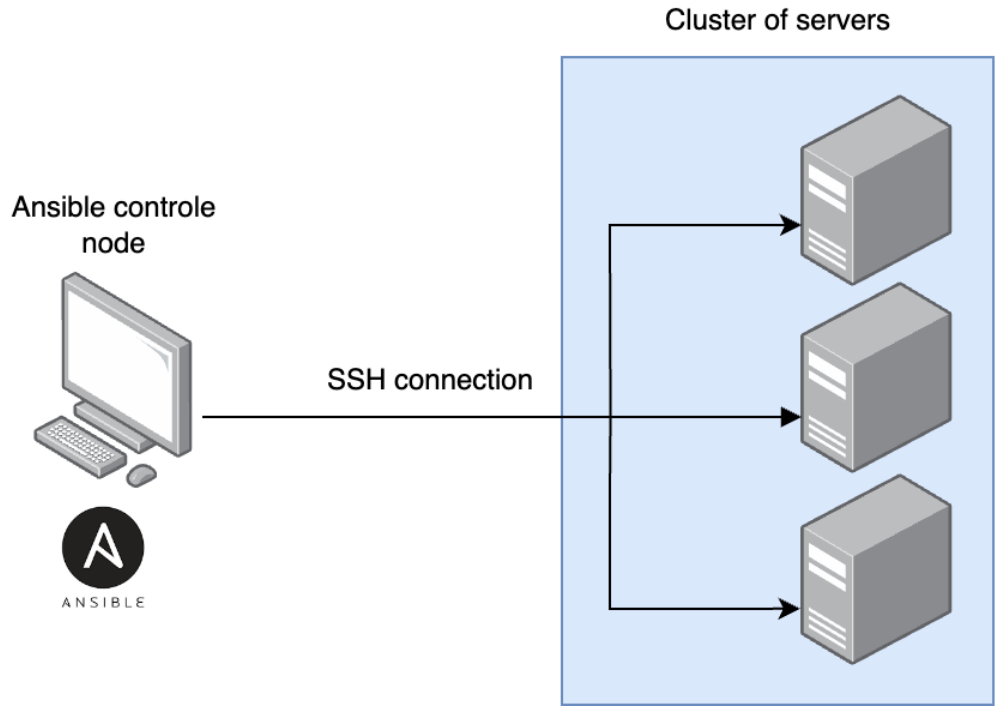
Outils d'automatisation – Ansible

Ansible est un outil qui permet de configurer à distance des machines. Il utilise le protocole SSH pour se connecter aux machines et est basé sur le langage de programmation Python.

Ansible utilise une architecture simple, avec 2 entités :

1. Le noeud Ansible Controller.
2. Les noeuds à configurer.

Ansible doit être installé uniquement sur la machine de contrôle, et doit avoir accès aux machines par SSH pour les configurer.



Outils d'automatisation – Ryu

Ryu est un framework de contrôleur SDN OpenFlow qui permet de configurer des switchs et équipements réseaux à distance, de manière automatisée.

Le développeur code en Python la manière dont les switchs et routeurs doivent se comporter, en fonction des flux qui les traversent.

```
if packet.ip_dst == "10.2.0.0/24" # If IP destination is on 10.2.0.0/24 network
    action = outputPacket(routerPort.2) # Tell router to output packet on port 2
    sendResponseToRouter(action)
```

Une fois que le code est développé, les équipements réseaux peuvent être configurés de 3 manières (réactive, pro-active ou hybride).

Références

- Ancien cours « Téléinformatique » (G. Waeber, S. Paccard, Q. Vaucher, N. Wirth)
- Cours « Architecture des réseaux » (François Buntschu)
- Cours « Administration et supervision de réseaux » (François Buntschu)
- Ancien cours « Téléinformatique » (M. Roch-Neirey)