Notes perso formation Ansible

## Introduction

### Mon expérience avec Ansible :

Utilisation à Auchan E-Commerce -> Mise en place de l’utilisation d’Ansible de l’administration complète des serveurs BO et de la migration vers l’automatisation des mises en prod via Ansible.

Aujourd’hui, c’est la quasi totalité de mon travail.

A noter que je n’ai jamais utilisé les autres outils de provisionning.

Les opinions sont informées par la lecture d’avis d’autres utilisateurs et mon expérience personnelle.

En aucune manière l’objectif de cette formation est de présenter la totalité des fonctionnalités d’Ansible. Il existe plus de 2800 modules à l’heure de la rédaction.

Il s’agit de montrer la logique de fonctionnement d’Ansible et de concepts de base et essentiels, sur lesquels tous les autres sont bâtis.

La documentation officielle d’Ansible est à mes yeux la meilleure. Explique à la fois bien et souvent avec de nombreux exemples.

### Ansible, c’est quoi ?

#### Ansible: un outil de provisionning

\* Agit sur des machines déjà existantes afin de les \*\*configurer\*\*

\* Similaire à Puppet ou Chef par exemple -> \_Config Management\_

\* Différent de Terraform -> \_Orchestration Management\_

(vont provisionner les serveurs eux-mêmes)

\* !! Ansible peut faire de l'Orchestration, mais ce n'est pas son rôle premier (Notamment pour Docker)

\* Est aujourd’hui sous la responsabilité de RedHat

\* Open Source : les sources sont disponibles sur GitHub.

Ursula K. Le Guin

### Avantages

\* Idempotence de la plupart des modules sur la plupart des plateformes (Linux, Windows, Mac, BSD)

C’est-à-dire que dans 95% des cas, Ansible se comportera de la même manière peut importe la plateforme. Dans les autres cas, il existe des modules spécifiques, par exemple pour la gestion de paquets (apt, yum)

\* Ne nécessite qu'une connection ssh et Python sur la machine distante. Pas d'agent à installer partout.

Par conséquent nécessite quand même un peu de préparation, pour déployer des clefs ssh, mais l’utilisation de mot de passes est possible

\* Pas besoin d'expérience de développement (même si ça aide...)

\* Permet d'avoir très rapidement un aperçu de l'infrastructure via les playbooks

Si les playbooks sont bien écrits, on peut voir tout ce qui est installé et comment en un clin d’œil.

\* Possible d'écrire ses propres modules, ou d’ajouter des modules externes

Si des besoins particuliers existent (par exemple pour la gestion des paquets sous ArchLinux)

### Comment ça marche ?

\* Génère un script python avec l'ensemble des commandes à effectuer

\* Se connecte aux machines via SSH et dépose ce script dans les fichiers temporaires

\* Exécute et monitore le déroulement du script

### Le fichier inventaire

#### Référencer les machines

\* Sert simplement à signaler à Ansible les machines sur lesquelles agir -> hosts.

\* Existe deux formats : ini et yaml

\* permet d'associer des variables à chaque host.

\* permet de regrouper les hosts en groupe, en fonction de leurs affinités.

Faire plusieurs exemples, simples et plus complexes

* Faire monter un serveur web avec les outils et les langages de leur choix.
* Monter une plateforme de monitoring des serveurs utilisés