

Introduction

Ansible c'est quoi ?

Installer Ansible sur votre poste

Vérifier l'installation

Création d'un recette & d'inventaire

Fichier Playbook

Fichier d'inventaire

Introduction

J'ai été amené à créer des rôles Ansible pour automatiser l'installation et la configuration de divers services : openldap, zabbix, telegraf, bacula, rsyslog, squid, isc-dhcp-server etc...

Bref, Ansible est vraiment très pratique pour pouvoir installer et configurer un serveur ou toute une architecture en un temps record !!

Ansible c'est quoi ?

Ansible est un outil qui permet d'installer et de configurer un ou plusieurs serveurs au travers de recettes écrites en YAML. Ces recettes contiennent une série de tâches qui seront lancées séquentiellement. Elles utilisent des modules internes à Ansible qui permettent de décrire les opérations à effectuer et leurs conditions de lancement.

L'une des forces d'Ansible est qu'il n'est pas nécessaire d'installer un agent sur les serveurs à administrer : une connexion SSH et Python3 suffisent.


 *Info : Il existe d'autres outils similaires (Chef, Puppet, SaltStack, Fabric)*

Installer Ansible sur votre poste

C'est bien beau tout ça, mais comment ça fonctionne ?

Déjà, il m'a fallu installer Ansible sur mon ordinateur, qui sera le PC qui va orchestrer tout ça, "node master" en anglais.

Puis, il m'a fallu ajouter le dépôt officiel d'Ansible adapté au système d'exploitation en l'occurrence Debian 11 (bullseye).

 La documentation officielle est toujours un allié de qualité pour trouver le genre d'information ;)

[Doc officielle](#)

Vu que je suis une feignasse, voilà la commande pour ajouter le dépôt ;)

```
echo "deb http://ppa.launchpad.net/ansible/ansible/ubuntu focal main" | sudo tee  
-a /etc/apt/sources.list.d/ansible.list
```

Puis, il ne faut pas oublier le paquet gnupg2, qui est très utile pour ajouter des clés du dépôt. (Oui, pour ajouter un dépôt, il faut une clé, c'est une question de sécurité : histoire de vérifier l'authenticité des paquets)

```
sudo apt-get install gnupg2  
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 93C4A3FD7BB9C367
```

Maintenant que les préparatif sont fini, le moment est enfin venue d'installer Ansible 🐙

```
sudo apt-get update -y && sudo apt-get install ansible -y
```

Vérifier l'installation

Aller une petite vérifie histoire d'être sur à 100% d'Ansible est bien installer.

```
ansible --version
```

Incroyable la version 2.12.2 est installer !

Maintenant que le plus facile est passer c'est l'heure de comprendre ansible et de crée des Playbooks 🐙

Création d'un recette & d'inventaire

Quelque lien vers les vidéos/site que j'ai consulter pour comprendre Ansible.

[📺 Découvrir Ansible \(4min\)](#)

[📺 Mettre en place un serveur web avec ansible \(56min\)](#)

[📺 Fichier YAML \(14min\)](#)

[📺 Fichier Jinja2 \(3min\)](#)

[📄 Repo de playbook ansible galaxy](#)

[📄 Création des clefs ssh](#)

[📄 Docs Ansible](#)

Bon après quelques heures de documentation et d'inspection de playbook existant voici un résumer.

- **Playbook** : Un fichier YAML qui va contenir les action à effectuer, par exemple install le packet wget, copie un fichier etc...
- **Inventaire** : Un fichier qui contient les information relative au machines à administrer ip, hostname, shell, username, etc...
- **Clef ssh** : Par défaut l'authentification ssh s'effectue par clefs, c'est d'ailleurs recommander. (il est cependant possible de forcer la connexion par mot de passe avec le fichier Inventaire)
- **Templates jinja** : Un fichier Jinja2 c'est le modèle d'un fichier de configuration, qui intègre la notion de variable, list, boucle et condition.

Et enfin il est préférable de crée une arborescence par playbook avec la commande ansible-galaxy histoire d'organiser les fichiers qui constitue un Role Ansible (Playbook, Templates, script, etc..)

```
ansible-galaxy init zabbix
```

Ansible galaxy va alors générer cette arborescence.

```
├─ defaults
|   └─ main.yml
├─ files
├─ handlers
|   └─ main.yml
```

```
├─ meta
│   └─ main.yml
├─ README.md
├─ tasks
│   └─ main.yml
├─ templates
├─ tests
│   └─ inventory
│   └─ test.yml
└─ vars
    └─ main.yml
```

Fichier Playbook

Voici un extrais de l'un de mes 1er rôle Ansible qui permet d'installé l'agent zabbix sur toutes les machines qui sont dans le groupe [Zabbix] du fichier **Inventaire**

(Il fait appel à des fichiers pour configurer l'agent que je ne vais pas détailler ici.)

```
---
# Installation de zabbix-agent
- name: Ajout de la clé de dépôt zabbix
  apt_key: url=https://repo.zabbix.com/zabbix-official-repo.key state=present
  tags: [ install ]

- name: Ajout du dépôt zabbix
  template: src=./templates/zabbix.list.j2
  dest=/etc/apt/sources.list.d/zabbix.list owner=root group=root mode='0644'
  tags: [ install ]

- name: Installation PAQUET zabbix_agent
  apt: name=zabbix-agent state=latest update_cache=yes
  tags: [ install ]

# Configuration globale de zabbix_agent
- name: Configuration globale de zabbix_agent
  template: src=./templates/zabbix_agentd.conf.j2
  dest=/etc/zabbix/zabbix_agentd.d/zabbix_agentd.conf owner=zabbix group=zabbix
  mode='0640'
  tags: [ install, config ]
```

Fichier d'inventaire

Voila la tête de mon fichier inventaire, la connexion ssh ce fait par mot de passe et binaire python3 sont indiquer en chemin absolu vu que c'est un environnement de test virtuel.

⚠ En production il faut utiliser des clef ssh et non des mot de passe pour des questions de sécurité.

```
[Zabbix]
server-zabbix ansible_host=192.168.1.30 ansible_user=toor
ansible_ssh_pass=password ansible_sudo_pass=password
ansible_python_interpreter=/usr/bin/python3
client-zabbix ansible_host=192.168.1.10 ansible_user=toor
ansible_ssh_pass=password ansible_sudo_pass=password
ansible_python_interpreter=/usr/bin/python3
server-web ansible_host=192.168.1.32 ansible_user=toor ansible_ssh_pass=password
ansible_sudo_pass=password ansible_python_interpreter=/usr/bin/python3
```

Après moult test en environnement virtuel et quelque cheveux arracher le rôle fonctionne !!

Il m'est enfin possible:

- Installer le paquets zabbix_agent depuis le dépôt officiel
- Configurer zabbix_agent
- Exécuter quelle script .sh

le tout en une seul commande qui peut être scale sur une multitude de machines 😊

```
ansible-playbook main.yml -i inventory --limite server-zabbix:client-zabbix
```

 *--limite permet de limiter l'execution du **Playbook** à certaine machines*