

## I Présentation

### II Installer Ansible sur votre poste

### III Création d'une recette & du fichier inventaire

# I Présentation

## 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 suffit. Bref, Ansible est très pratique pour pouvoir installer et configurer un serveur voire toute une architecture en un temps-record !!

**i** Info : il existe des outils similaires (Chef, Puppet, SaltStack, Fabric)

## II Installer Ansible sur votre poste

Il faut installer Ansible sur l'ordinateur qui sera le pc orchestrateur, "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 une alliée de qualité pour trouver ce genre d'information.

[Doc officiel](#)

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
```

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

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

Maintenant, que les préparatifs sont finis c'est le moment d'installer Ansible 🥳

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

## Vérifier l'installation

Petite vérification, pour être sûr qu'Ansible est bien installé.

```
ansible --version
```

Incroyable la version 2.12.2 est installé ! Le plus facile est passé c'est l'heure de comprendre ansible et de créer des playbooks 🐙

## III Création d'une recette & du fichier inventaire

Bon après quelques heures de documentation et d'inspection de playbook existant voici un résumé des éléments majeurs à connaître pour utiliser ansible.

- **Playbook** : Fichier YAML qui va contenir les actions à effectuer, par exemple installer le paquet wget, copier un fichier etc...
- **Inventaire** : Fichier qui contient les informations relatives aux machines à administré IP, hostname, shell, username, etc...
- **Clef ssh** : Par défaut l'authentification ssh s'effectuera par clefs ssh, c'est d'ailleurs recommandé. (il est cependant possible de forcer la connexion par mot de passe avec le fichier Inventaire)
- **Templates jinja** : Fichier Jinja2 c'est le modèle d'un fichier de configuration, qui intègre la notion de variable, liste, boucle et condition.

Enfin il est préférable de générer une arborescence par défaut pour chaque playbook avec la commande ansible-galaxy (Playbook, Templates, script, etc..) cette arborescence permet de standardiser, c'est playbook.

```
ansible-galaxy init zabbix
```

Ansible galaxie va alors générer cette arborescence de fichier.

```
├─ 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 extrait de l'un de mes 1ers rôles Ansible qui permet d'installer 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

Voilà la tête de mon fichier inventaire, la connexion ssh ce fait par mot de passe et le binaire python3 est indiqué en chemin absolu.

⚠ En production il faut utiliser des clefs ssh et non des mots 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 plusieurs test en environnement virtuel et quelques cheveux arrachés le rôle fonctionne, Il m'est enfin possible :

- Installer le paquet zabbix\_agent depuis le dépôt officiel
- Configurer zabbix\_agent
- Exécuter quelle script .sh

Le tout en une seule 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'exécution du **Playbook** à certaines machines*


Quelques liens vers les vidéos et site que j'ai consulté 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](#)