

UA Master 2 : EC914 : Génie Logiciel

Installation de Ansible et écriture de son 1er rôle

Avant de parler de ce qu'est un « rôle » et qu'un « playbook », commençons par définir les pré-requis de ce tp afin de pouvoir utiliser ansible.

Pour utiliser ansible, commençons par créer une clé ssh sur la machine que nous voulons que ansible communique, de ce fait il faut lancer la commande suivante :

- `$ ssh-keygen`

*Cette commande, va nous permettre de créer une paire de clé publique et privée que nous devrions ensuite copier sur le serveur afin qu'ansible puisse se déployer sur la machine. Par ailleurs, nous pouvons donner un nom à notre clé afin qu'il soit identifiable sinon son nom par défaut sera « **id_rsa.pub** ».*

Pour copier la clé ssh :

- `$ ssh-copy-id -i ~/.ssh/cle nom_machine_cible@ip_machine_cible`

Maintenant, que la clé ait été copiée. Nous pouvons passer à la partie de Ansible.

- *Qu'est-ce que Ansible ?*

Ansible, est un logiciel de déploiement et de configuration à distance écrit en python et utilisant seulement comme communication le ssh. Ce logiciel utilise des recettes ou Playbooks écrit en YAML, qui contiennent une série de tâches/rôles qui seront lancés de manière ordonnées

- **Qu'est-ce qu'un rôle ?**

Un rôle, est une directive écrite de manière atomique afin de simplifier les tâches complexes dans notre serveur ansible. Par ailleurs, ils seront listés à travers une liste d'inclure ou bien par le biais d'un référencement du dossier roles/name dans le playbook général (celui de la racine). Le playbook et le rôle sont quasi similaires sauf que chacun a une utilisation particulière. Nous pouvons comparer les rôles à des fonctions en C.

- **Qu'est-ce qu'un playbook ?**

Un playbook est un grand script qui permet de lister les différents rôles à exécutés. De plus, le playbook aura un rôle de gestionnaire de tâches et de « main » intrinsèque à notre serveur ansible lui disant qu'elle tâche que ce dernier doit exécuter.

1^{er} rôle

Écrire un rôle permettant d'installer les dépôts dotdeb et backport sur une GNU/Linux Debian stable fraîchement installée.

Nous allons devoir définir dans un premier temps, dans le fichier **hosts**, les adresses ip sur lequel notre ansible doit se déployer.

Maintenant, que cette modification a été faite, nous allons rédiger notre premier playbook.

- **name:** Installation de dotdeb & backport

hosts: all

become: yes

become_method: su

roles:

- role: repository

...

- **Name :** Nom de la tâche effectuée par le playbook
- **Hosts (all) :** ansible va se déployer sur toutes les machines référencées dans notre hosts. Il est possible de définir des clusters/groupe pour mieux gérer les actions.
- **Become(yes) :** élévation des privilèges sur les machines cibles pour l'exécution des futurs rôles.
- **Become_method(su) :** Toutes les commandes seront lancées en tant qu'administrateur sur les machines Debian.

Maintenant que le playbook de base a été défini, nous allons définir notre rôle permettant d'installer les dépôts dotdeb et backport sur nos machines cibles fraîchement installées.

- name: Ajout de DotDeb

apt_repository: repo= 'deb http://packages.dotdeb.org {{ ansible_lsb['codename'] }} all'

```
#'{{ item }}'
```

```
#with_items:
```

```
# - deb http://packages.dotdeb.org {{ ansible_lsb['codename'] }} all
```

```
# - deb-src http://packages.dotdeb.org {{ ansible_lsb['codename'] }} Installation
```

```
- name: Ajout clef DotDeb
```

```
apt_key: url=https://www.dotdeb.org/dotdeb.gpg
```

```
state: present
```

```
- name: Ajout de backport
```

```
apt-repository: repo= 'deb http://deb.debian.org/debian {{ ansible_lsb['codename'] }}-  
backports main contrib non free'
```

```
- name: APT update
```

```
apt: update_cache=yes
```

```
...
```