

PROVISIONNING

Dans cette partie, vous allez utiliser des outils de provisionning. Dans chaque cas, le serveur sera implanté sur l'hôte.

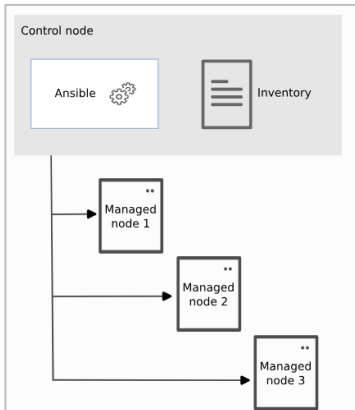
Vous utiliserez en priorité les outils offerts par les solutions de provisionning plutôt que des scripts externes en python ou en bash.

I. Table des matières

II.	Ansible	2
1.	Présentation d'Ansible	2
2.	Installation	2
3.	Créer un inventaire	3
4.	Configurer des connexions SSH afin qu'Ansible puisse se connecter aux nœuds gérés	3
5.	Réaliser un premier playbooks	4
a.	Présentation des Playbooks	4
b.	Playbook pour ping	5
6.	Playbooks (pas réellement testé)	5
a.	LXC (non testé)	5
b.	Synchronisation	6
c.	Installation de paquets	6
d.	Configuration du réseau	6
III.	SaltStack	7

II. Ansible

1. Présentation d'Ansible



Cette section se base de [cette documentation](#) d'Ansible.

Ansible automatise la gestion des systèmes distants et contrôle leur état souhaité.

Il y a trois composants principaux :

- 1- **Nœud de contrôle** : système sur lequel Ansible est installé. C'est ici que les commandes Ansible sont exécutées.
- 2- **Nœud géré** : système distant ou hôte contrôlé par Ansible
- 3- **Inventaire** : liste des nœuds gérés

2. Installation

Les nœuds de contrôles et les nœuds gérés nécessitent Python 3 pour exécuter le code de bibliothèque Ansible. Le nœud géré a également besoin d'un compte utilisateur qui peut SSH.

- 1- Rechercher et mémoriser le chemin d'accès à l'interpréteur Python. Il faut le spécifier à Ansible.
- 2- Vérifier que pip est installé « `python3 -m pip -V` »
- 3- Installer Ansible :

```
sudo python3 -m pip install --user ansible
```

Si l'avertissement suivant apparaît rentrer les commandes qui suivent :

```
WARNING: The scripts ansible, ansible-config, ansible-connection, ansible-console, ansible-doc, ansible-galaxy, ansible-inventory, ansible-playbook, ansible-pull and ansible-vault are installed in '/root/.local/bin' which is not on PATH. Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The script ansible-community is installed in '/root/.local/bin' which is not on PATH. Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed MarkupSafe-2.1.3 ansible-6.7.0 ansible-core-2.13.10 jinja2-3.1.2 packaging-23.1 resolvelib-0.8.1
```

```
nano ~/.bashrc
export PATH="$PATH:/root/.local/bin"
source ~/.bashrc
```

- 4- Vérifier la version du paquet installé :

```
sudo python3 -m pip show ansible
```

```
toto@srv:~$ sudo python3 -m pip show ansible
Name: ansible
Version: 6.7.0
Summary: Radically simple IT automation
Home-page: https://ansible.com/
Author: Ansible, Inc.
Author-email: info@ansible.com
License: GPLv3+
Location: /root/.local/lib/python3.8/site-packages
Requires: ansible-core
Required-by:
```

J'ai tout de même téléchargé ansible en plus avec la commande `sudo apt install ansible` car sinon les commandes n'étaient pas reconnues.

- 5- Optionnel : rajouter la saisie-automatique de l'interface de commande dans le shell avec la commande :

```
python3 -m pip install --user argcomplet
```

Pour plus d'information cf. [la documentation d'Ansible](#).

3. Créer un inventaire

D'après la documentation on rajoute dans le fichier « `/etc/ansible/hosts` » les IP des 3 conteneurs créés dans les parties précédentes.

Cependant ce fichier n'existe pas encore puisque `pip` a installé Ansible dans le dossier `/root/.local/bin`. Ce dossier est utilisé pour stocker les scripts exécutables installés localement pour l'utilisateur 'root'.

Puisque par défaut Ansible recherche le fichier dans `/etc/ansible/hosts` nous allons créer ce fichier. (Il est cependant possible de spécifier le chemin en utilisant l'option `-i`).

```
[myvirtualmachines]
10.0.3.2
10.0.3.111
10.0.3.244
```

Pour vérifier les hôtes de l'inventaire :

```
ansible all --list-hosts
```

```
toto@srv:/etc$ ansible all --list-hosts
hosts (3):
 10.0.3.2
 10.0.3.111
 10.0.3.244
```

4. Configurer des connexions SSH afin qu'Ansible puisse se connecter aux nœuds gérés

- 1- Il faut générer une paire de clefs SSH avec la commande « `ssh-keygen -t rsa` » sur l'hôte qui exécute les commandes Ansible. Cela crée une paire de clefs dans le répertoire `~/ssh/id_rsa`.

```
toto@srv:/etc$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/toto/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/toto/.ssh/id_rsa
Your public key has been saved in /home/toto/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:DufIqvRwnRIYIL50IYcEquHq4qvZssdwaBPnfUHNPI toto@srv
```

- 2- Copier la clef publique générée dans chaque conteneur dans le fichier « `~/ssh/authorized_keys` » .

Attention si la clef est stockée dans le dossier d'un utilisateur (par exemple ici *toto*) il faudra que ce soit cet utilisateur qui exécute la commande ansible. Sinon, il faudra rajouter l'option pour demander le mot de passe.

Si ce fichier n'existe pas créer le sur chaque conteneur avec « touch /var/lib/lxc/C*/rootfs/root/.ssh/authorized_keys » et copie la clef ainsi :

```
cat /home/toto/.ssh/id_rsa.pub >>
/var/lib/lxc/C*/rootfs/root/.ssh/authorized_keys
```

3- Tester la connexion pour savoir si cela fonctionne

```
root@srv:/var/lib/lxc/C1/rootfs/root/.ssh# ssh ubuntu@10.0.3.2
Password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-148-generic x86_64)
```

4- Tester une commande ping sur les nœuds :

```
ansible all -u ubuntu -m ping --ask-pass
```

Si les message d'erreur suivant apparaît voici ce qu'il faut faire :

```
10.0.3.19 | FAILED! => {
  "msg": "to use the 'ssh' connection type with passwords or pkcs11_provider, you must install the sshpass program"
}
root@ubuntu:/etc/ansible# sshpass
Command 'sshpass' not found, but can be installed with:
apt install sshpass
```

Ne pas oublier de rajouter les options pour spécifier le nom d'utilisateur (et de demander le mot de passe si la clef n'est pas chez le bon utilisateur). Voici le résultat :

```
root@srv:/var/lib/lxc/C1/rootfs/root/.ssh# ansible all -u ubuntu -m ping --ask-pass
SSH password:
10.0.3.111 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
10.0.3.244 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
10.0.3.2 | SUCCESS => {
```

5. Réaliser un premier playbooks

a. Présentation des Playbooks

Ansible playbooks

Ansible Playbooks offer a repeatable, re-usable, simple configuration management and multi-machine deployment system, one that is well suited to deploying complex applications. If you need to execute a task with Ansible more than once, write a playbook and put it under source control. Then you can use the playbook to push out new configuration or confirm the configuration of remote systems. The playbooks in the [ansible-examples repository](#) illustrate many useful techniques. You may want to look at these in another tab as you read the documentation.

Cf. [la documentation d'Ansible](#) pour plus d'informations sur ce qu'est un playbook.

A retenir :

- Les playbooks sont exprimés en **YAML**. Voir [ce lien](#) pour comprendre la syntaxe YAML.
- Un playbook s'exécute **dans l'ordre** : du haut vers le bas.

- Un playbook est composé de « **play** » .
- Chaque « play » définit :
 - o Les **nœuds gérés** qui sont ciblés à l'aide d'un modèle
 - o Au **moins une tâche** à exécuter
- Il est possible de rajouter au début des playbooks des « [Keywords](#) » qui permettent d'avoir du contrôle sur le [plug-in de connexion](#), de gérer les erreurs etc.
- Si une tâche **échoue** sur un hôte, Ansible **retire** cet hôte de la **rotation** pour le reste du playbook.

b. Playbook pour ping

On peut créer les playbooks là où l'on souhaite. Voici un exemple de playbook :

```
- name: My first play
  hosts: myvirtualmachines
  remote_user: root
  tasks:
    - name: Ping my hosts
      ansible.builtin.ping:
    - name: Print message
      ansible.builtin.debug:
        msg: Hello world
```

Pour l'exécuter il faut rentrer la commande suivante :

```
ansible-playbook hello_word.yaml
```

Information :

Il existe un outil qui permet d'obtenir des commandes pour obtenir des commentaires détaillés et spécifiques sur les playbooks avant de les exécuter : `ansible-lint`. Voir [la documentation d'Ansible Lint](#) pour plus de détails.

6. Playbooks (pas réellement testé)

a. LXC (non testé)

Pour la gestion de LXC : [community.general.lxc container module – Manage LXC Containers — Ansible Documentation](#)

Ce script n'a pas été testé mais fortement inspiré de la documentation en ligne :

```
---
- name: Download LXC and create Ubuntu containers
  hosts: groupe
  remote_user: root
  tasks:
    - name: Install LXC packages
      ansible.builtin.apt:
        name: lxc
        state: present
        when: ansible_os_family == 'Debian'

    - name: Install LXC packages
      ansible.builtin.yum:
        name: lxc
```

```
    state: present
    when: ansible_os_family == 'RedHat'

- name: Create a started container
  community.general.lxc_container:
    name: test-container-started
    container_log: true
    template: ubuntu
    state: started
    template_options: --release trusty
```

b. Synchronisation

Documentation : [ici](#).

```
- name: Synchronization
  hosts: myvirtualmachines
  remote_user: root
  tasks:
    - name: Synchronization of src on the control machine to dest on the remote hosts
      ansible.posix.synchronize:
        src: /home/adm_jma/ansible/test.txt
        dest: test.txt
```

c. Installation de paquets

Documentation : [ansible.builtin.apt module – Manages apt-packages — Ansible Documentation](#)

d. Configuration du réseau

III. SaltStack

Une fois que vous aurez lancé les scripts de nettoyage des installations, réalisez différents states saltstack réalisant les opérations des scripts développés dans la partie gestion des conteneurs. Vous développerez aussi bien des states pour la gestion de l'environnement que la gestion des conteneurs.

En exploitant les states précédemment écrits, écrivez un ou des states réalisant les mêmes opérations que celle réalisées dans le cadre des scripts de synchronisation de supervision en python

Pas eu le temps