

---

# Ansible

## Table des matières

<b>I - Ansible (overview).....</b>	<b>3</b>
1. Présentation de Ansible.....	3
1.1. Fonctionnalités d'Ansible.....	3
1.2. Historique et situation d'Ansible.....	4
<b>Ansible (Configuration).....</b>	<b>5</b>
2. Installation et configuration de ansible.....	5
2.1. Différentes installations possibles.....	5
2.2. configurations "ssh".....	6
2.3. Configurations complémentaires.....	7
2.4. Configuration de l' inventory.....	7
<b>II - Ansible (utilisation).....</b>	<b>8</b>
1. Ansible en mode "ad hoc".....	8
1.1. Exemples sans module (ici pour linux debian/ubuntu).....	8
1.2. Exemples avec modules.....	8
2. Ansible en mode "playbook".....	10
2.1. Exemple.....	10
<b>III - Annexe – Bibliographie, Liens WEB + TP.....</b>	<b>12</b>

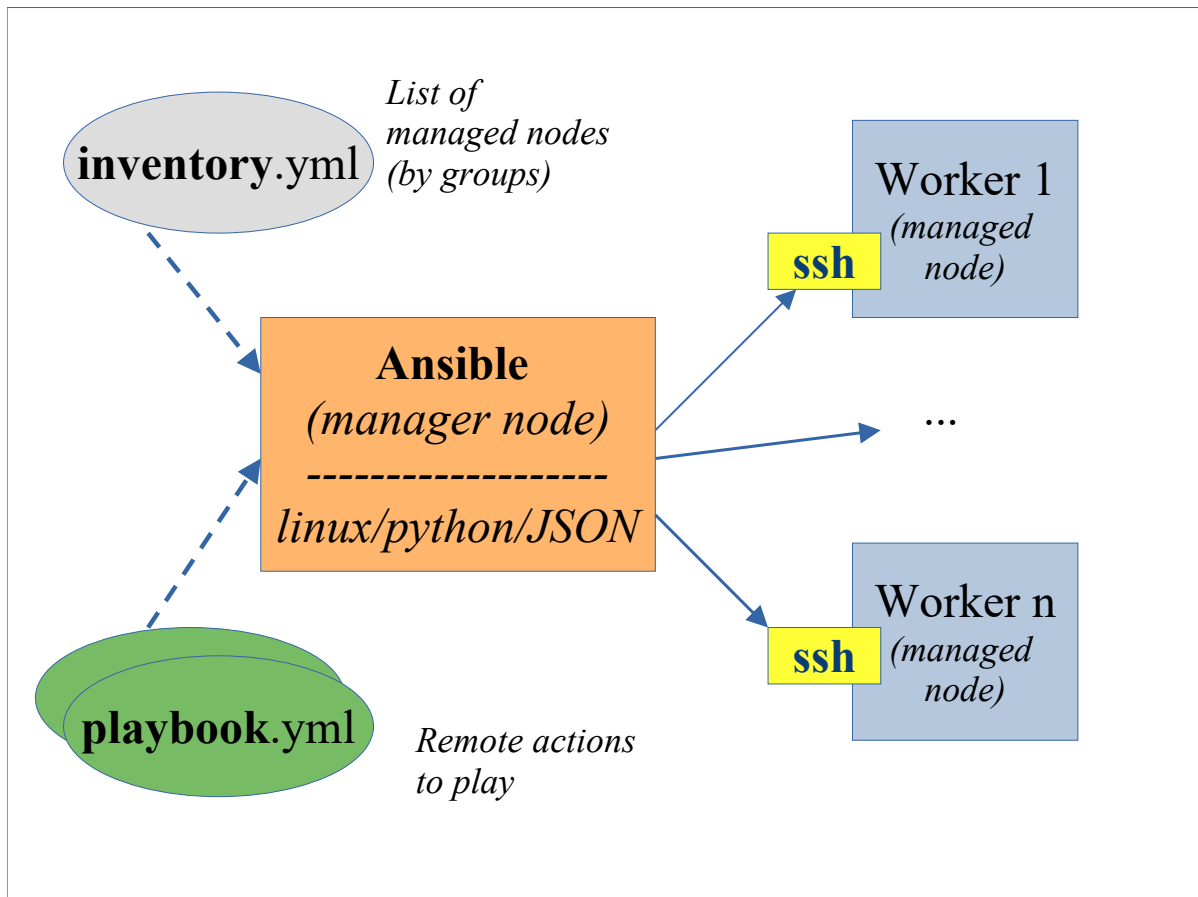
---

1. Bibliographie et liens vers sites "internet" .....	12
2. TP.....	12

# I - Ansible (overview)

## 1. Présentation de Ansible

### 1.1. Fonctionnalités d'Ansible



**Ansible** est un **logiciel open source / libre** (écrit en langage *python*) permettant de **configurer en même temps différentes machines de manière centralisée** .

Ansible permet d'effectuer du "**provisioning**" (**installation et configuration de logiciels**) ainsi que du **lancement d'exécution de tâches à distance** .

NB: sur les machines pilotées (*workers, managed nodes*) , juste besoin d'un accès "**ssh**" et aucun agent à installer en plus .

Ansible doit être installé sur une machine linux (readHat/centOs ou bien debian/ubuntu ou ...) et les machines pilotées peuvent être très diverses (windows , linux, ...) .

## 1.2. Historique et situation d'Ansible

La première version de Ansible a été créée en **2012** par Michael DeHaan .



En **2015** , **Red Hat** annonce le rachat de Ansible .  
Depuis , la base d'Ansible reste "**open-source / libre**" et l'entreprise **Red Hat propose une extension facultative appelée "Ansible Tower"** permettant de superviser un semble de machine via une belle interface graphique .

Vers 2018/2019 , Red Hat est a son tout rachetée par **IBM** et donc aujourd'hui Ansible appartient donc à la sous marque Red Hat de la grande marque IBM .

# Ansible (Configuration)

## 2. Installation et configuration de ansible

### 2.1. Différentes installations possibles

Installation sous ubuntu 20.04 comportant déjà par défaut python 3.8 :

```
sudo apt update
```

```
sudo apt install ansible
```

```
...répondre y...
```

```
ansible --version
```

```
ansible 2.9.6
```

```
config file = /etc/ansible/ansible.cfg
```

```
configured module search path = ['/home/vagrant/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
```

```
ansible python module location = /usr/lib/python3/dist-packages/ansible
```

```
executable location = /usr/bin/ansible
```

```
python version = 3.8.10 (default, Jun 22 2022, 20:18:18) [GCC 9.4.0]
```

```
ansible-playbook --version
```

```
ansible-playbook 2.9.6
```

Installation d'Ansible via Vagrant :

```
config.vm.provision "ansible_local" do |ansible|
```

```
  ansible.install_mode = "default"
```

```
  ansible.playbook = "playbook.yml"
```

```
    ansible.groups = {
```

```
      "workers" => [ "worker-vm1", "worker-vm2" ],
```

```
      "managers" => [ "va-ansible-manager" ],
```

```
      "all_groups:children" => [ "workers", "managers" ]
```

```
    }
```

```
  end
```

## 2.2. configurations "ssh"

Etant donné que ansible communique avec les différents nœuds en utilisant ssh , il est souvent nécessaire d'effectuer des configurations spécifiques au contexte de manière à pouvoir utiliser ssh avec ou sans mot de passe .

Si les machines à piloter sont configurées "en ssh n'acceptant pas les mots de passe" **on peut éventuellement autoriser (temporairement ou pas) des connexions ssh via mot de passe via des commandes de ce genre :**

```
sudo sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config
sudo systemctl restart sshd.service
```

(à lancer si besoin sur chaque machine à piloter)

De manière à initialiser certains fichiers des mécanismes ssh , il est souvent utile d'établir au moins une fois une connexion ssh vers la machine à piloter en renseignant son mot de passe:

*Exemple:*

**ssh worker-vm2**

*The authenticity of host 'worker-vm2 (192.168.33.72)' can't be established.*

*ECDSA key fingerprint is SHA256:VRtOm0+UeRLfBjwZd4fjbg74DF+wcLlkbx0P0Ncezmo.*

*Are you sure you want to continue connecting (yes/no/[fingerprint])?*

*répondre y*

*saisir le mot de passe*

**exit**

Sur la machine "ansible manager" , on pourra éventuellement se créer des clefs pour ssh via une commande de ce genre :

```
ssh-keygen -t rsa -q -f "$HOME/.ssh/id_rsa" -N ""
```

On pourra ensuite envoyer notre clef publique sur les machines à piloter via une commande de ce genre :

**ssh-copy-id -f username@remoteHost**

(ex : ssh-copy-id -f username@192.168.33.72)

*saisir le mot de passe demandé*

→ Ceci peut quelquefois être utile pour pouvoir ultérieurement se connecter à certains noeuds via ssh sans mot de passe .

*Autres configurations ssh (complémentaires ou alternatives) en fonction du contexte ...*

## 2.3. Configurations complémentaires

Souvent besoin de configurer un nouvel utilisateur (ex : `ansible` ) pour éviter de lancer toutes les commandes via un dangereux compte de type `root` ou autre .

Exemple de sites à consulter pour cela :

[https://linuxhint.com/install\\_ansible\\_ubuntu/](https://linuxhint.com/install_ansible_ubuntu/)

<https://openclassrooms.com/fr/courses/2035796-utilisez-ansible-pour-automatiser-vos-taches-de-configuration>

## 2.4. Configuration de l' inventory

**Ansible** appelle "*inventory*" la liste des nœuds à piloter (organisés en groupes).

Cet "inventory" peut être paramétré au format INI ou bien au format YAML .

Exemple de fichier `/etc/ansible/hosts` (au format INI) :

```
mail.example.com

[workers]
worker-vm1
worker-vm2

[dbservers]
one.example.com
two.example.com
three.example.com
```

Au sein de l'exemple précédent le groupe **workers** est constitué des nœuds `worker-vm1` et `worker-vm2` et d'autre part , le nœud `mail.example.com` n'est placé dans aucun groupe ; il appartiendra tout de même au groupe prédéfini **all** .

Exemple équivalent au format YAML :

```
all:
  hosts:
    mail.example.com:
  children:
    workers:
      hosts:
        worker-vm1:
        worker-vm2:
    dbservers:
      hosts:
        one.example.com:
        two.example.com:
        three.example.com:
```

**NB:** un fichier `my-inventory1.yml` peut être placé en argument des commandes `ansible` ou bien `ansible-playbook` via l'option `-i` .

Exemple: **ansible** workers -a "hostname" **-i my-inventory1.yml**

Sans cette option, le "inventory" par défaut correspond au fichier `/etc/ansible/hosts` à préparer avec `vi` ou `nano` par exemple .

## II - Ansible (utilisation)

### 1. Ansible en mode "ad hoc"

Ansible peut éventuellement être directement utilisé via des lignes de commandes simples (sans script de type playbook), on parle alors en terme de commande "**ad hoc**" (spécifiquement adaptée à un cas bien précis).

```
$ ansible [pattern] -m [module] -a "[module options]"
```

Principaux patterns (sélection de noeuds/machines):

<b>all</b> (or <b>*</b> )	Tous les nœuds configurés dans l'inventaire
<i>host1</i>	Le nœud <i>host1</i>
<i>host1:host2</i> (or <i>host1,host2</i> )	Plusieurs noeuds
<i>workers</i>	Le groupe <i>workers</i> configuré dans l'inventaire
...	Autres possibilités : plusieurs groupes , intersection de groupes, ...

#### 1.1. Exemples sans module (ici pour linux debian/ubuntu)

Sans précision d'un module, c'est le module *shell* qui est utilisé par défaut .

**ansible workers -a "hostname"** (ou bien **-a "/usr/bin/hostname"**)

```
worker-vm2 | CHANGED | rc=0 >>
```

```
worker-vm2
```

```
worker-vm1 | CHANGED | rc=0 >>
```

```
worker-vm1
```

**ansible workers -a "hostname -i"**

```
worker-vm2 | CHANGED | rc=0 >>
```

```
127.0.2.1 192.168.33.72
```

```
worker-vm1 | CHANGED | rc=0 >>
```

```
127.0.2.1 192.168.33.71
```

#### 1.2. Exemples avec modules

Copie de fichier:

```
ansible workers -m ansible.builtin.copy -a "src=/etc/hosts dest=/tmp/hosts"
```

Changer les attributs d'un fichier:

```
ansible workers -m ansible.builtin.file -a "dest=/tmp/hosts mode=600"
```



ansible workers -m ansible.builtin.file -a "dest=/tmp/hosts mode=600 owner=xx group=yy"

Le module **ansible.builtin.apt** permet d'effectuer des installations sur linux debian et ubuntu

Le module **ansible.builtin.yum** permet d'effectuer des installations sur linux Red Hat et centOs

## 2. Ansible en mode "playbook"

### 2.1. Exemple

```
ansible-playbook -i inventory.yml hostnames_pb.yml
```

avec *inventory.yml*

```
all:
  children:
    workers:
      hosts:
        worker-vm1:
        worker-vm2:
    managers:
      hosts:
        va-ansible-manager:
```

et *hostnames\_pb.yml*

```
---
- name: Playbook for retrieve all hostnames
  hosts: workers
  tasks:
    - name: get hostname
      shell:
        "hostname"
      register: hostname_output
      tags: hostname
    - name: display
      debug:
        msg: "hostname is {{hostname_output.stdout}}"
```

NB : **shell:** et **debug:** font référence aux [modules](#) *ansible.builtin.shell* et ...*debug*

NB:

- un fichier *main.yml* peut inclure des sous fichiers *.yml* via l'instruction ***include*** :  
"xyz.yml"
- Le module "***gathering facts***" permet de récupérer des informations sur les nœuds pilotés sous forme de variables initialisées automatiquement

# ANNEXES

## III - Annexe – Bibliographie, Liens WEB + TP

### 1. Bibliographie et liens vers sites "internet"


### 2. TP