

# **Automatisation des tâches avec Ansible**

# I - Consignes

- Définir **/home/ansible/projet** comme **répertoire de travail ansible**

Après avoir copié le fichier de configuration dans le répertoire de travail **/home/ansible/projet**, on vérifie qu'il est bien pris en compte

```
[ansible@ansible_master projet]$ ansible --version
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller starting with Ansible 2.12. Current
version: 3.6.8 (default, Sep 10 2021, 09:13:53) [GCC 8.5.0 20210514 (Red Hat 8.5.0-3)]. This feature will be removed from
ansible-core in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ansible [core 2.11.7]
  config file = /home/ansible/projet/ansible.cfg
  configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.6/site-packages/ansible
  ansible collection location = /home/ansible/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.6.8 (default, Sep 10 2021, 09:13:53) [GCC 8.5.0 20210514 (Red Hat 8.5.0-3)]
  jinja version = 3.0.3
  libyaml = True
[ansible@ansible_master projet]$
```

## II - Les consignes :

- Le chemin par défaut des rôles doit être **/home/ansible/projet/roles**
- L'inventaire par défaut doit être **/home/ansible/projet/hosts**
- Les paramètres d'escalade de privilège doivent être automatiques

Pour cela, on renseigne le fichier de configuration **/home/ansible/projet/ansible.cfg** de la façon suivante :

```
roles_path    = /home/ansible/projet/roles
```

```
inventory     = /home/ansible/projet/hosts
```

```
[privilege_escalation]  
become=True  
become_method=sudo  
become_user=root  
become_ask_pass=False
```

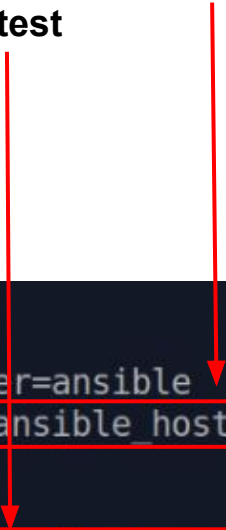
### III - Consignes - inventaire -

- Ajout de la machine **projet\_ansible** dans le groupe **projet**
- Ajout de la machine **formation** dans le groupe **test**

Contenu de l'inventaire

```
[ansible@ansible_master projet]$ cat hosts
[projet]
localhost ansible connection=local ansible user=ansible
ansible_connection=ssh ansible_user=ansible ansible_host=172.19.10.158

[test]
ansible_connection=ssh ansible_user=ansible ansible_host=172.19.10.72
```



## IV - Consignes

- Désactiver l'obligation de la saisie des mots de passe sudo pour la totalité des machines cibles

On renseigne le fichier **/etc/sudoers** de la façon suivante, pour permettre à l'utilisateur **ansible** d'exécuter **toute commande sans mot passe**.

```
ansible ALL=(root) NOPASSWD:ALL
```

Information à spécifier sur toutes les machines cibles

## V - Consignes - gestion des utilisateurs -

- Créer les utilisateurs de type dev dans les machines du groupe projet (ayant comme groupe primaire dev)
- Créer les utilisateurs de type test dans les machines du groupe test (ayant comme groupe secondaire test).

Commençons par créer les groupes dans les machines projet et test

```
tasks:
- name: Création du groupe dev dans les machines du groupe projet
  group:
    name: dev
    state: present
  when: ("projet" in group_names)
  tags:
    - ajout_groupe_projet

- name: Création du groupe testeur dans les machines du groupe test
  group:
    name: test
    state: present
  when: ("test" in group_names)
  tags:
    - ajout_groupe_test
```

## V - Consignes - gestion des utilisateurs - suite -



- Les développeurs ont le mot de passe « ansible »
- et les testeurs ont le mot de passe « testeur ».
- Les mots de passe doivent être dans un fichier à part sécurisé par le mot de passe ansible.

Cryptons le fichier de mots de passe: **ansible-vault encrypt mdp.yml**

### Liste des utilisateurs: **users.yml**

- Utilisateurs :
  - Nom : jean  
Type : dev
  - Nom : Philippe  
Type : dev
  - Nom : Christian  
Type : test
  - Nom : Nicolas  
Type : test

### Contenu du fichier (**mdp.yml**) des mots de passe

```
[ansible@ansible_master projet]$ cat mdp.yml
$ANSIBLE_VAULT;1.1;AES256
61633464656538633632363432623933386637633062646332343635386535643733323131363034
3838346665633562653766343430613730303134333664350a303730316565653765623863376164
33336232366164613235666236313534336533396536396233326134326639376266323038613230
6630643537363763300a303830366531316536393966336165353637373063656264333164303439
64373733616363333764633961393864643034373833623730343631326266613437323030356565
3166663234656636636637353531383763666165633766626637
[ansible@ansible_master projet]$
[ansible@ansible_master projet]$ ansible-vault view mdp.yml
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller
2.12. Current version: 3.6.8 (default, Sep 10 2021, 09:13:53) [GCC 8.5.0 20210514]
This feature will be removed from ansible-core in version 2.12. Deprecation warni
setting deprecation_warnings=False in ansible.cfg.
Vault password:
mdp_dev: ansible
mdp_test: testeur
```

crypté

en clair

## V - Consignes - gestion des utilisateurs - suite -

### Création des utilisateurs dev et test

vars\_files:

- users.yml
- mdp.yml

```
- name: Ajout des utilisateurs dev
  user:
    name: "{{ item.nom }}"
    password: "{{ 'mdp_dev' | password_hash('sha512') }}"
    group: dev
    state: present
  loop: "{{ utilisateurs }}"
  when: (item.type == "dev") and ("projet" in group_names)
  tags:
    - ajout_utilisateurs_dev

- name: Ajout des utilisateurs testeurs
  user:
    name: "{{ item.nom }}"
    password: "{{ 'mdp_test' | password_hash('sha512') }}"
    groups: test
    state: present
  loop: "{{ utilisateurs }}"
  when: (item.type == "test") and ("test" in group_names)
  tags:
    - ajout_utilisateurs_testeurs
```



## V - Consignes - gestion des utilisateurs - suite -



**vault\_mdp** étant le fichier contenant le mot de secret ansible, la commande de lancement du playbook est la suivante: **ansible-playbook --vault-password-file=vault\_mdp play\_users.yml**

```
[ansible@ansible_master projet]$ ansible-playbook --vault-password-file=vault_mdp play_users.yml
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller starting with Ansible
2.12. Current version: 3.6.8 (default, Sep 10 2021, 09:13:53) [GCC 8.5.0 20210514 (Red Hat 8.5.0-3)].
This feature will be removed from ansible-core in version 2.12. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.

PLAY [GESTION DES UTILISATEURS] *****

TASK [Gathering Facts] *****
ok: [ansible_connection=ssh]
ok: [localhost]
ok: [ansible_projet]

TASK [Création du groupe dev dans les machines du groupe projet] *****
skipping: [ansible_connection=ssh]
ok: [localhost]
changed: [ansible_projet]

TASK [Création du groupe testeur dans les machines du groupe test] *****
skipping: [localhost]
skipping: [ansible_projet]
ok: [ansible_connection=ssh]

TASK [Ajout des utilisateurs dev] *****
skipping: [ansible_connection=ssh] => (item={'nom': 'jean', 'type': 'dev'})
skipping: [ansible_connection=ssh] => (item={'nom': 'Philippe', 'type': 'dev'})
skipping: [ansible_connection=ssh] => (item={'nom': 'Christian', 'type': 'test'})
skipping: [ansible_connection=ssh] => (item={'nom': 'Nicolas', 'type': 'test'})
changed: [localhost] => (item={'nom': 'jean', 'type': 'dev'})
changed: [localhost] => (item={'nom': 'Philippe', 'type': 'dev'})
skipping: [localhost] => (item={'nom': 'Christian', 'type': 'test'})
skipping: [localhost] => (item={'nom': 'Nicolas', 'type': 'test'})
changed: [ansible_projet] => (item={'nom': 'jean', 'type': 'dev'})
changed: [ansible_projet] => (item={'nom': 'Philippe', 'type': 'dev'})
```

## VI - Consignes - écrire développement ou test dans /etc/type

- Créer un fichier /etc/type contenant :
  - **development** pour les machines du groupe projet
  - **test** pour les machines du groupe test.

```
- name: "Ecriture dans les machines projet et test "  
  hosts:  
    - projet  
    - test  
  tasks:  
    - name: "Ecrire developpement dans les machines projet"  
      copy:  
        content: "development"  
        dest: /etc/type  
        when: ('projet' in group_names)  
    - name: "Ecrire test dans les machines test"  
      copy:  
        content: "test"  
        dest: /etc/type  
        when: ('test' in group_names)
```

## VII - Consignes - gestion des paquets ... -

### Pour les machines du groupe projet :

1. Mettre à jour tous les paquets.
2. Installer httpd pour les machines Centos et apache2 pour les machines Ubuntu
3. Personnaliser la page d'accueil en créant un fichier contenant la chaîne « la machine **nom\_machine** de la famille **OS\_distribution** vous souhaite la bienvenue » où :  
    **nom\_machine** est le nom de la machine distante  
    **OS\_distribution** est la distribution de la machine (redhat,centos,ubuntu,...)
4. Redémarrer les services httpd et apache2 en cas de changement

Mise à jour de tous les paquets et installation de httpd

Mise à jour des paquets et installation de apache2

```
tasks:
- name: Mise à jour de tous les paquets
  yum:
    name: '*'
    state: latest

- name: Installation de httpd
  yum:
    name: httpd
    state: latest
```

```
- name: Mise à jour de tous les paquets
  apt:
    name: '*'
    state: latest

- name: Installation de apache2
  apt:
    name: apache2
    state: latest
```

## VII - Consignes - gestion des paquets ... suite -

```
- name: Personnalisation de la bannière
  template:
    src: motd.j2
    dest: /etc/motd
    mode: 0644
  when: ('projet' in group_names)
```

Contenu dynamique du fichier motd.j2

```
La machine {{ ansible_hostname }} de la distribution {{ ansible_distribution }}
vous souhaite la bienvenue
```

## VII - Consignes - gestion des paquets ... suite -

- Redémarrer les services httpd et apache2 en cas de changement

A spécifier juste après le bloc d'installation de httpd

```
notify:  
- Restart httpd
```

A spécifier juste après le bloc d'installation de apache2

```
notify:  
- Restart apache2
```

Redémarrage des services suite à des changements notifiés par les blocs notify

```
handlers:  
- name: Restart httpd  
  service:  
    name: httpd  
    state: restarted  
    enabled: true  
  
- name: Restart apache2  
  service:  
    name: apache2  
    state: restarted  
    enabled: true
```

## VIII - Consignes - 2 partitions de 100 Mo chacune -

- a - Créer 2 partitions de 100 Mo chacune.
- b - Créer un volume logique de 120Mo en utilisant les 2 partitions précédentes.
- c - Configurer leur montage automatique en faisant en sorte que le playbook monte automatiquement le volume logique.

Création d'un disque de partition de 2Go à partir duquel on crée les 2 partitions de 100 Mo chacune

**Disque /dev/sdd : 2 GiB, 2147483648 octets, 4194304 secteurs**

```
- name: Création de la partition part1 sur sdd
  parted:
    device: /dev/sdd
    number: 1
    state: present
    part_end: 100MB
```

Création de la première  
partition de 100Mo

```
- name: Création de la partition part2 sur sdd
  parted:
    device: /dev/sdd
    number: 2
    state: present
    part_start: 100MB
    part_end: 200MB
```

Création de la deuxième  
partition de 100Mo

## VIII - Consignes - 2 partitions de 100 Mo chacune - suite -

### Création du volume de groupe

```
- name: Création du volume de groupe
  lvg:
    vg: my_vg1
    pvs: /dev/sdd1,/dev/sdd2
    state: present
```

### Création du volume logique de 120 Mo

```
- name: Création du volume logique de 120Mo
  lvol:
    vg: my_vg1
    lv: my_lv1
    size: 120M
```

### Ensuite on procède à la création du système de fichiers xfs

```
- name: Création du système fichiers XFS
  filesystem:
    fstype: xfs
    dev: /dev/my_vg1/my_lv1
    state: present
```



## VIII - 2 partitions de 100 Mo chacune - suite -

Montage automatique du volume logique

```
- name: Montage automatique du volume logique
  mount:
    path: /my_mnt/
    src: /dev/my_vg1/my_lv1
    fstype: xfs
    state: present
```

Montage automatique du volume logique à partir de son **UUID (autre méthode)**

```
- name: Récupération du UUID du volume logique /dev/my_vg1/my_lv1
  command: blkid -s UUID -o value /dev/my_vg1/my_lv1
  register: get_UUID
  tags:
    - get_UUID

- name: Deuxième méthode - Montage automatique du volume logique avec UUID
  mount:
    path: /my_mnt/
    src: UUID="{{ get_UUID.stdout }}"
    fstype: xfs
    state: present
```



## VIII - 2 partitions de 100 Mo chacune - suite -

Résultat dans un fichier /etc/fstab:

```
/dev/mapper/cl-root    /                xfs      defaults    0 0
UUID=f01da4fe-76d3-434b-8b13-0ec150729ba7 /boot            xfs      defaults    0 0
/dev/mapper/cl-swap    none             swap     defaults    0 0

/dev/sdb2              /data            ext4     defaults    0 0
/dev/sdc2              none             swap     defaults    0 0
UUID="1b956f17-a453-4372-bb2c-d9ba6bf10a28" /my mnt/ xfs defaults 0 0
```

### Remarques :

Installation du **module parted**:

ansible-galaxy collection install community.general

Installation du module **mount** inclus dans la collection posix:

ansible-galaxy collection install ansible.posix

## IX - Consignes - **graphical.target** -

- Changer le target par défaut des machines du **groupe projet** en **graphical.target**.

→ Créer un lien symbolique depuis **graphical.target** vers **default.target**

```
- name: Mettre graphical-target par défaut => création de lien symbolique
hosts:
  - projet
  - test
tasks:
  - name: Création du lien symbolique
    file:
      src: /usr/lib/systemd/system/graphical.target
      dest: /etc/systemd/system/default.target
      state: link
    when: ("projet" in group_names)
```

## X - Consignes - **crontab** -



- Mettre à jour les paquets sur les machines du groupe test chaque jour à 1h du matin

```
- name: MISE A JOUR DES PAQUETS SUR LES MACHINES DU GROUPE TEST
hosts:
  - projet
  - test

tasks:
  - name: "Mise à jour des paquets"
    cron:
      user: "ansible"
      minute: "0"
      hour: "1"
      day: "*"
      job: 'apt upgrade "'
      state: present
    when: ("test" in group_names)
```

### Résultat :

```
ansible@ubuntu:~$ crontab -l
#Ansible: None
0 1 * * * apt upgrade "
```

## XI - Consignes - haproxy installation -



- Installer le rôle **geerlingguy.haproxy** sous le nom **haproxy**.

Inscription du rôle dans le fichier **roles.yml** sous le nom **haproxy**:

```
- src: geerlingguy.haproxy
  name: haproxy
```

Pour installer le rôle : **ansible-galaxy install -r roles.yml -p roles**

répertoire d'installation des rôles

## XII - Consignes - haproxy déploiement -



- Déployer le rôle **haproxy** sur la machine locale
- Définir les machines formation et projet\_ansible comme serveur de backend
- Éviter les conflits avec httpd

```
- name: HAPROXY - EQUILIBRAGE DE CHARGE
```

```
vars:
```

```
  haproxy_backend_servers:
```

```
    - name: machine_projet_ansible  
      address: 172.19.10.158:8080
```

```
    - name: machine_formation  
      address: 172.19.10.72:8080
```

```
hosts: localhost
```

```
roles:
```

```
  - haproxy
```

Variable d'entrée

Les serveurs de backend

Déploiement sur la machine locale