Installation & utilisation d'Ansible

Installation de l'environnement virtuel python

Création de l'utilisateur d'ansible :

```
adduser user-ansible
```

Installation de l'environnement virtuel python :

```
apt-get install python-virtualenv sshpass
```

> sshpass sert au bon fonctionnement de la suite !

Changement d'utilisateur :

```
su - user-ansible
```

Lancement de l'environnement virtuel :

```
virtualenv ansible4.6.0
```

Activation de l'environnement virtuel :

```
source ansible4.6.0/bin/activate
```

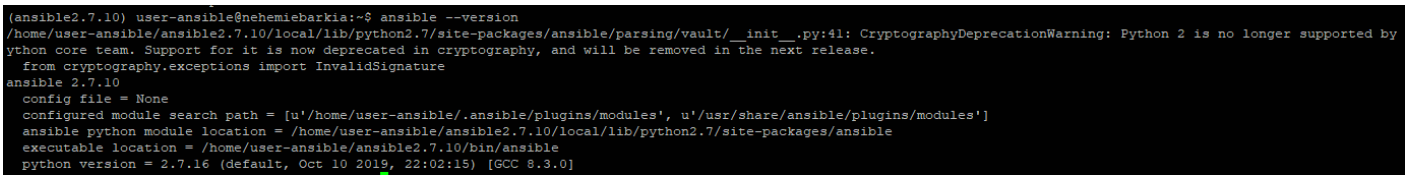
Installation d'ansible

Installation d'ansible :

```
pip install ansible==4.6.0
```

Est-ce que ansible est bien installé ?

```
ansible --version
```



Préparation de la communication avec les nodes

Définition des nodes :

Création du fichier de définition des nodes :

```
nano inventaire.ini
```

```
http1
bdd1
```

Modification de notre fichier host :

```
nano /etc/hosts
```

On ajoute :

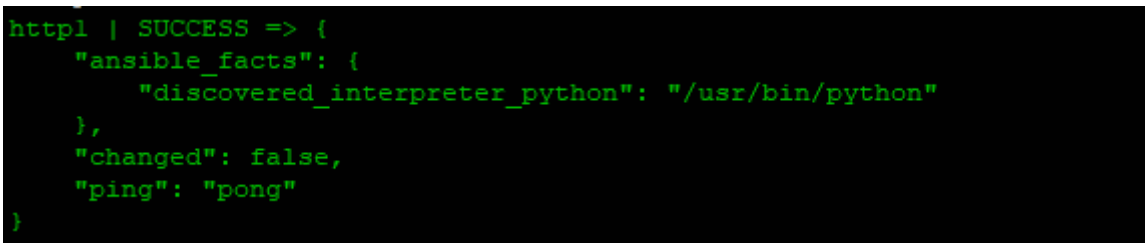
```
# le node http1
34.88.193.243 http1
# le node bdd1
35.228.198.62 bdd1
```

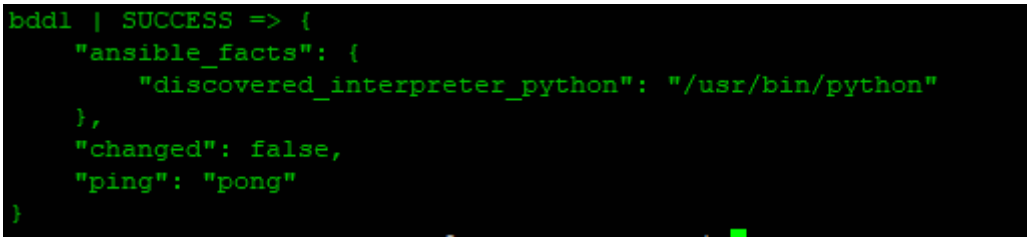
Test du bon fonctionnement SSH des nodes :

```
ssh root@http1
ssh root@bdd1
```

Réalisation de pings sur nos nœuds :

```
ansible -i inventaire.ini -m ping http1 --user root --ask-pass
ansible -i inventaire.ini -m ping bdd1 --user root --ask-pass
```





Installation de python à distance sur nos nœuds :

```
ansible -i inventaire.ini -m raw -a "yum install -y python3" http1 --user root --ask-pass
ansible -i inventaire.ini -m raw -a "yum install -y python3" bdd1 --user root --ask-pass
```

Résultat :

```
================================================================================
 Package                 Arch         Version               Repository     Size
================================================================================
Installing:
 python3                 x86_64       3.6.8-18.el7          updates        70 k
Installing for dependencies:
 libtirpc                x86_64       0.2.4-0.16.el7        base           89 k
 python3-libs            x86_64       3.6.8-18.el7          updates       6.9 M
 python3-pip             noarch       9.0.3-8.el7           base          1.6 M
 python3-setuptools      noarch       39.2.0-10.el7         base          629 k

Transaction Summary
================================================================================
Install  1 Package (+4 Dependent packages)

Total download size: 9.3 M
Installed size: 48 M
Downloading packages:
(1/5): libtirpc-0.2.4-0.16.el7.x86_64.rpm                   |  89 kB   00:00
(2/5): python3-3.6.8-18.el7.x86_64.rpm                      |  70 kB   00:00
(3/5): python3-setuptools-39.2.0-10.el7.noarch.rpm          | 629 kB   00:00
(4/5): python3-pip-9.0.3-8.el7.noarch.rpm                   | 1.6 MB   00:00
(5/5): python3-libs-3.6.8-18.el7.x86_64.rpm                 | 6.9 MB   00:00
--------------------------------------------------------------------------------
Total                                           25 MB/s | 9.3 MB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : libtirpc-0.2.4-0.16.el7.x86_64                               1/5
  Installing : python3-setuptools-39.2.0-10.el7.noarch                      2/5
  Installing : python3-pip-9.0.3-8.el7.noarch                               3/5
  Installing : python3-3.6.8-18.el7.x86_64                                  4/5
  Installing : python3-libs-3.6.8-18.el7.x86_64                             5/5
  Verifying  : libtirpc-0.2.4-0.16.el7.x86_64                               1/5
  Verifying  : python3-setuptools-39.2.0-10.el7.noarch                      2/5
  Verifying  : python3-libs-3.6.8-18.el7.x86_64                             3/5
  Verifying  : python3-3.6.8-18.el7.x86_64                                  4/5
  Verifying  : python3-pip-9.0.3-8.el7.noarch                               5/5

Installed:
  python3.x86_64 0:3.6.8-18.el7

Dependency Installed:
  libtirpc.x86_64 0:0.2.4-0.16.el7   python3-libs.x86_64 0:3.6.8-18.el7
  python3-pip.noarch 0:9.0.3-8.el7   python3-setuptools.noarch 0:39.2.0-10.el7

Complete!
Shared connection to httpl closed.
```

Génération d'une chaine SHA-512 :

```
ansible localhost -i inventaire.ini -m debug -a "msg={{ 'passforce' | password_hash('sha512', 'sceretsalt') }}"
```

> Le mot de passe pour l'utilisateur user-ansible est donc "passforce", choisissez-en un différent !

Résultat :

```
localhost | SUCCESS => {
    "msg":
"$6$sceretsalt$tBcfGEgifQpQZsg5CIGZ79XC55h5vHy7UWrys7cAF37KNCQQbm7iCvy58MlLQLaS2fLF6ZjqDVHhVr
kMdRi0f0"
}
```

> Notre clef SHA-512 a bien été générée !

Création d'un utilisateur nommé "ansible" sur tous les nodes :

```
ansible -i inventaire.ini -m user -a 'name=user-ansible
password=$6$sceretsalt$tBcfGEgifQpQZsg5CIGZ79XC55h5vHy7UWrys7cAF37KNCQQbm7iCvy58MlLQLaS2fLF6Z
jqDVHhVrkMdRi0f0' --user root --ask-pass http1
ansible -i inventaire.ini -m user -a 'name=user-ansible
password=$6$sceretsalt$tBcfGEgifQpQZsg5CIGZ79XC55h5vHy7UWrys7cAF37KNCQQbm7iCvy58MlLQLaS2fLF6Z
jqDVHhVrkMdRi0f0' --user root --ask-pass bdd1
```

Résultat :

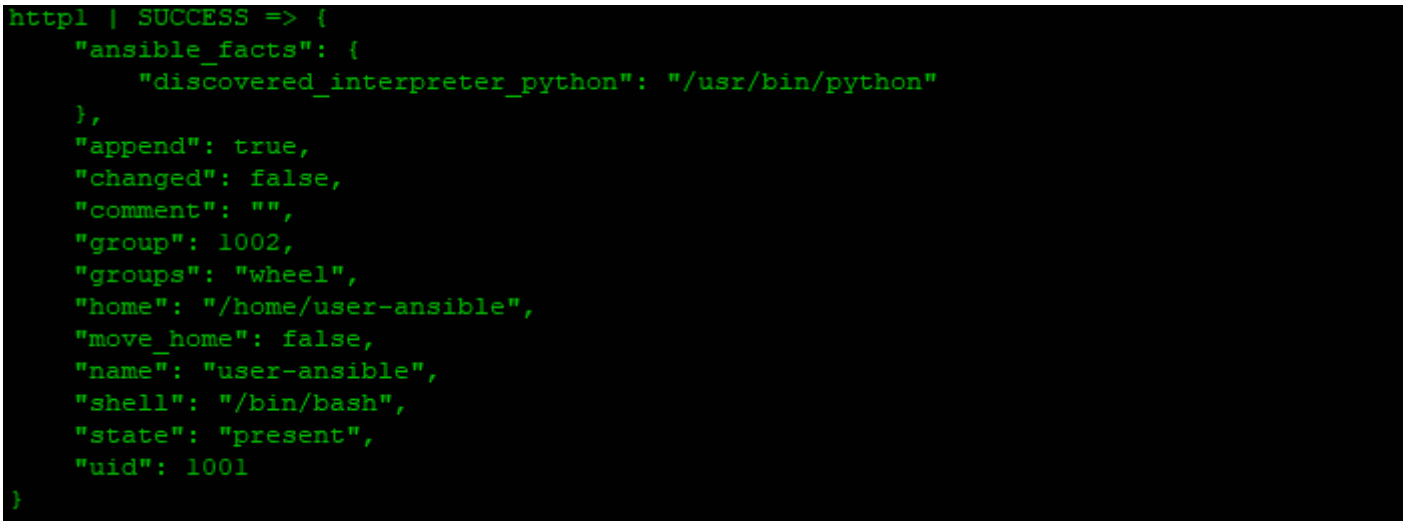
```
http1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1002,
    "home": "/home/user-ansible",
    "name": "user-ansible",
    "password": "NOT_LOGGING_PASSWORD",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 1001
}
```

```
bdd1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1002,
    "home": "/home/user-ansible",
    "name": "user-ansible",
    "password": "NOT_LOGGING_PASSWORD",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 1001
}
```

> Mettez évidement votre chaine à vous !

Ajout des droits de sudoers à nos nouveaux utilisateurs :

```
ansible -i inventaire.ini -m user -a 'name=user-ansible groups=wheel append=yes ' --user root --ask-pass http1
ansible -i inventaire.ini -m user -a 'name=user-ansible groups=wheel append=yes ' --user root --ask-pass bdd1
```



> Nos deux utilisateurs sont maintenant sudoers !

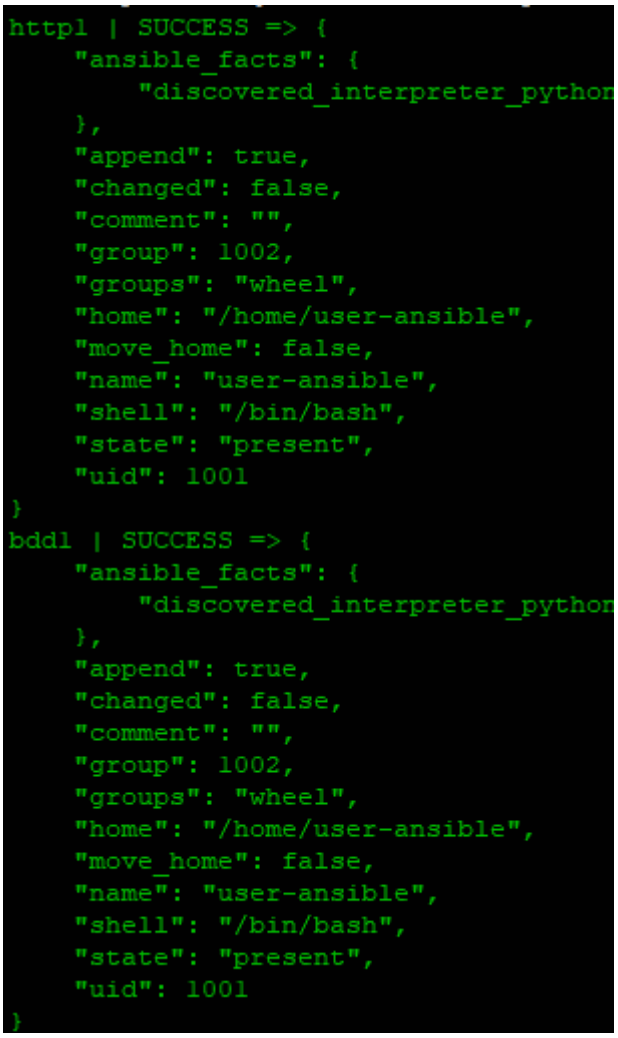
Nous vérifions maintenant que l'utilisateur user-ansible dispose bien des droits sudo :

```
ansible -i inventaire.ini -m user -a 'name=user-ansible groups=wheel append=yes ' --user user-ansible --ask-pass --become --ask-become-pass all
```

Nous répondons :

```
SSH password: passforce
BECOME password[defaults to SSH password]: passforce
```

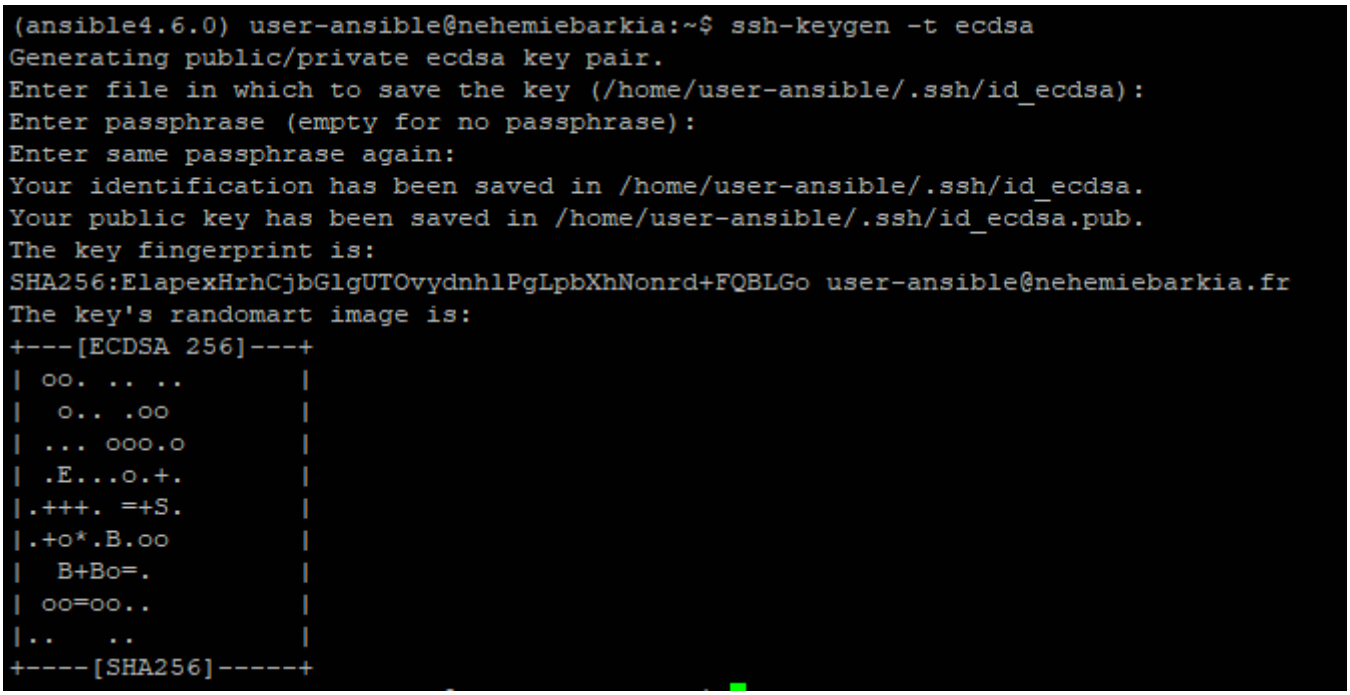
En sortie nous avons :



> L'utilisateur a bien le droit de sudo !

Génération de clefs SSH :

```
ssh-keygen -t ecdsa
```

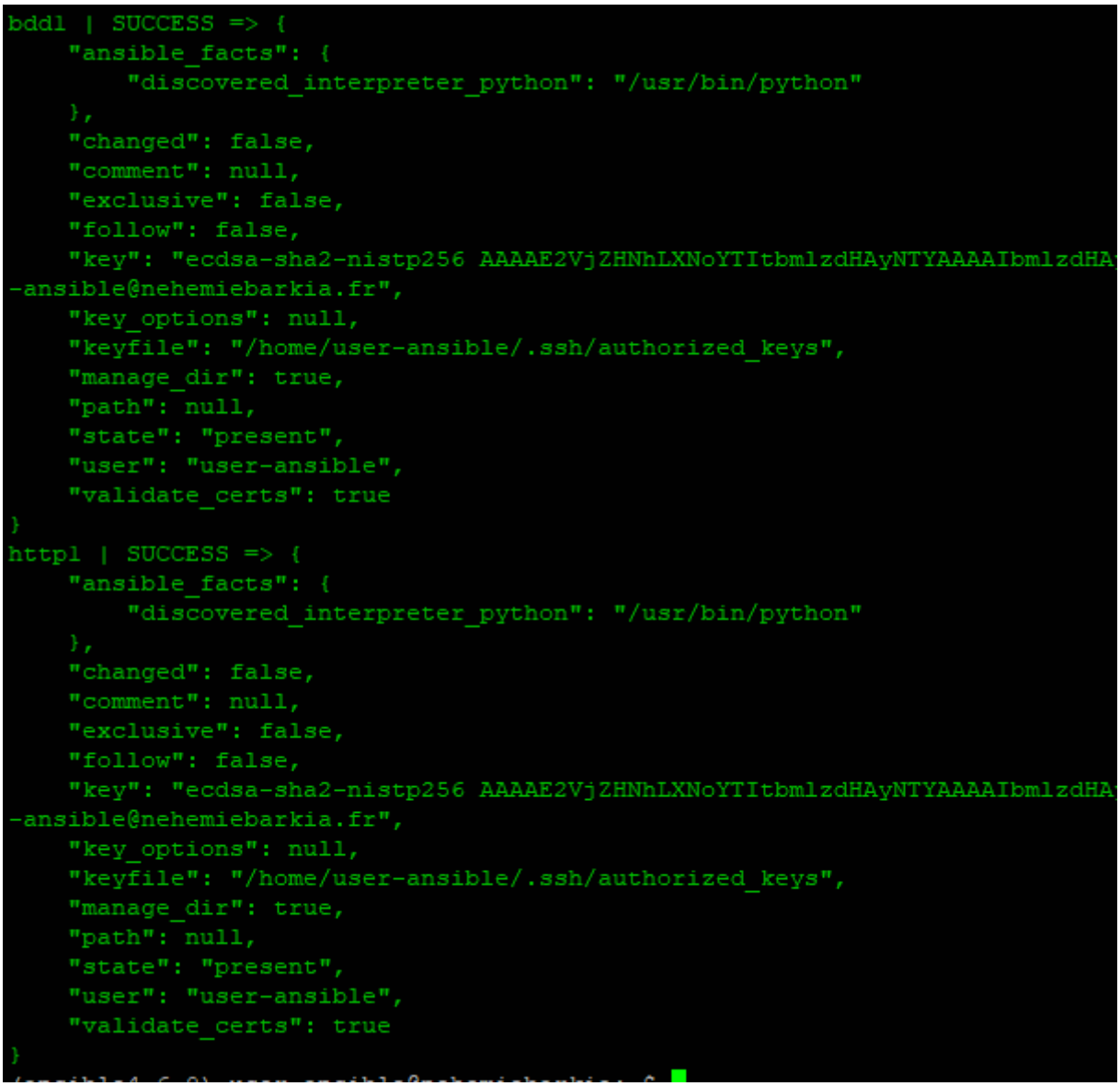


Ajoutez la clé publique de l’utilisateur user-ansible sur les nodes :

```
ansible -i inventaire.ini -m authorized_key -a 'user=user-ansible state=present key="{{ lookup("file", "/home/user-ansible/.ssh/id_ecdsa.pub") }}"' --user user-ansible --ask-pass all
```

> Le mot de passe demandé est "passforce" !

Résultat :



Vérification du bon fonctionnement des clefs SSH :

```
ansible -i inventaire.ini -m authorized_key -a 'user=user-ansible state=present key="{{ lookup("file",
"/home/user-ansible/.ssh/id_ecdsa.pub") }}"' --user user-ansible --become --ask-become-pass all
BECOME password: passforce
```

> Nous constatons, que on nous as demandé uniquement le mot de passe de sudo (BECOME password). Cela confirme que la liaison fonctionne bel et bien sans mot de passe !

Organisation de déploiement :

> A titre d'exemple, nous allons installer la solution "MediaWiki".

Pour ce faire, nous commençons par définir les différentes étapes - **que l'on nommera des rôle** - d'installation :

- Installation d'apache
- Installation de MariaDB
- Configuration d'apache
- Configuration de MariaDB
- Variables Globales

Structure d'un rôle :

- role/
  - files/ ---> Contient tous les fichier à copier sur le node
  - tasks/ ---> Contient les taches à exécuter
    - main.yml
  - handlers/ ---> Contient les taches de notification
    - main.yml
  - defaults/ ---> Contient des variables par défaut
    - main.yml
  - meta/ ---> Contient des dépendances et des informations
    - main.yml

On applique ça à nos rôles à nous :

- **Apache**
  - tasks/
    - taches à exécuter pour installer apache et php
  - handlers
    - tache pour redémarrer le service apache
- **MariaDB**
  - tasks/
    - taches à exécuter pour installer MariaDB
- **Commun**
  - defaults/
    - Définition des variables globales
- **ConfDB**
  - meta/
    - Dépendance avec le rôle **Commun**
  - tasks/
    - Taches à exécuter pour installer la base de donnée
- **ConfApache**

- meta/
  - Dépendance avec le rôle **Commun**
- tasks/
  - Taches à exécuter pour installer apache

Pour simplifier la suite, nous allons créer un groupe de nodes :

```
nano inventaire.ini
```

```
[apache]
http1

[db]
bdd1
```

Création du répertoire roles :

```
mkdir roles
cd roles
```

Nous allons maintenant utiliser ansible-galaxy qui permet de télécharger, créer et gérer les rôles ansible :

```
ansible-galaxy init apache
```

Regardons ce qu'ansible-galaxy a créer :

```
(ansible4.6.0) user-ansible@nehemiebarkia:~/roles$ tree apache
apache
|-- README.md
|-- defaults
|   `-- main.yml
|-- files
|-- handlers
|   `-- main.yml
|-- meta
|   `-- main.yml
|-- tasks
|   `-- main.yml
|-- templates
|-- tests
|   |-- inventory
|   `-- test.yml
`-- vars
    `-- main.yml
```

Cependant, nous allons garder uniquement les répertoires handlers, meta et tasks. Puis nous allons créer le fichier de configuration de la tache php-7 dans le rôle apache :

```
rm ./apache/{README.md,defaults/,files/,templates/,tests/,vars/} -r
touch apache/tasks/php7-install.yml
```

Création du rôle mariadb :

```
mkdir -p mariadb/tasks/
touch mariadb/tasks/main.yml
```

Création du rôle médiawiki :

```
mkdir mediawiki
```

Création du rôle Commun :

```
mkdir -p mediawiki/commun/defaults/
touch mediawiki/commun/defaults/main.yml
```

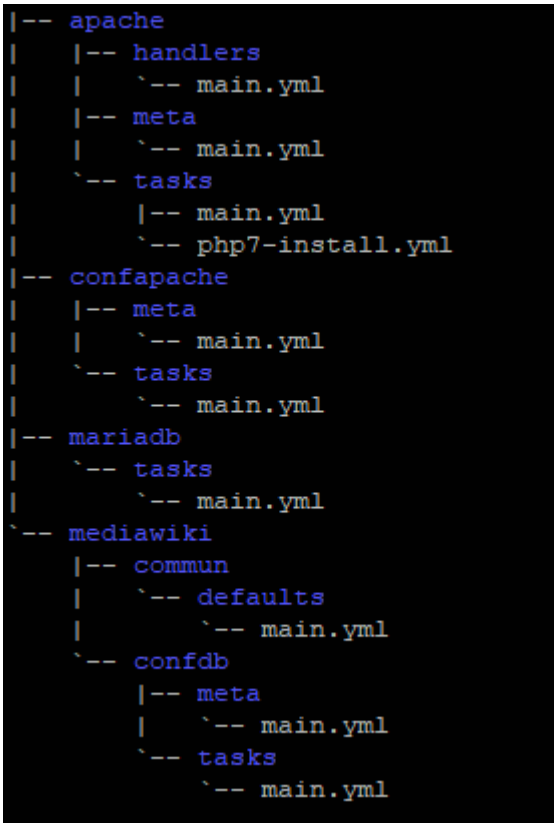
Création du rôle Confdb :

```
mkdir -p mediawiki/confdb/meta mediawiki/confdb/tasks
touch mediawiki/confdb/tasks/main.yml
touch mediawiki/confdb/meta/main.yml
```

Création du rôle Confapache :

```
mkdir -p roles/mediawiki/confapache/meta roles/mediawiki/confapache/tasks
touch roles/mediawiki/confapache/tasks/main.yml roles/mediawiki/confapache/meta/main.yml
```

Ce qui nous donne cette arborescence là :



Modification du main.yml du rôle apache :

```
nano roles/apache/tasks/main.yml
```

```
---

#1. Cette t  che permet d ^`^yinstaller Apache (httpd)    l ^`^yaide du module yum
- name: "apache installation"
  ansible.builtin.yum:
    name: "httpd"
    state: "present"

#2. Cette t  che active le service Apache
- name: "apache service activation"
  service:
    name: "httpd"
    state: "started"
    enabled: yes

#3. Cette t  che fait appel    un autre fichier de configuration pour installer PHP. Elle est ex  cut  e uniquement si
la variable php_install est    vraie (par d  faut, elle e$
```

```
- name: "install php7 packages"
  include: "php7-install.yml"
  when: php_install|default(False)|bool
```

1. La **première tâche,** "**apache installation**" va installer le serveur Apache avec le module “**yum**”. Le **name: "httpd"** indique le paquet concerné et le **state: "present"** spécifie qu’il faut l’installer.
2. La **deuxième tâche,** “**apache service activation**” va activer le service Apache avec le module “**service**”. Le **name: "httpd"** indique le service concerné, le **state: "started"** indique que le service sera démarré et le **enabled: yes** indique que le service sera activé.
3. La **troisième tâche,** “**install php7 packages**” inclut un fichier de configuration externe pour installer PHP. La tâche fait appel avec l’option “**include**” au fichier **php7-install.yml** qui est placé dans le répertoire **tasks** à coté de main.yml. La condition **when** avec le **filtre** (php_install|default(False)|bool) permettent de conditionner l’installation de PHP.

> Source : https://openclassrooms.com/fr/courses/2035796-utilisez-ansible-pour-automatiser-vos-taches-de-configuration/6373887-controlez-lexecution-des-operations-et-enchainez-plusieurs-actions

Modification du fichier php7-install.yml du rôle apache :

```
---

#1. Cette t  che installe le d  p  t  EPEL (Extra Packages for Enterprise Linux)
- name: "epel activation"
  yum:
    name: "epel-release"
    state: present

#. Cette t  che installe PHP7 et ses extensions
- name: "install php7 packages"
  command: yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
  command: yum -y install https://rpms.remirepo.net/enterprise/remi-release-7.rpm
  command: yum -y install yum-utils
  command: yum-config-manager --enable remi-php74
  command: yum update -y
 # command: yum install php php-cli php-fpm php-mysqlnd php-zip php-devel php-gd php-mcrypt php-mbstring
php-curl php-xml ph$
```

1. La **première tâche,** "**epel activation**", va activer le dépôt **epel** avec le module “**yum**”. Le **name: "epel-release"** indique le dépôt concerné et le **state: present** spécifie qu’il faut l’installer.
   Sur **Centos**, tous les logiciels ne sont pas disponibles par défaut. C’est le cas pour **PHP7** ; il faut donc ajouter le dépôt **EPEL** qui va contenir le paquet **PHP7**.
2. La **deuxième tâche,** “**remi repo activation**”, va installer le paquet remi-release-7.rpm avec le module “**yum**”. Le nom indique le service concerné ; le **state: present** indique que le paquet sera installé.
   Sous **Centos**, les paquets logiciels sont sous la forme **RPM**. Pour installer **PHP7**, vous avez besoin du paquet **remi-release-7.rpm** qui est dans le dépôt **REMI** qui dépend du dépôt **EPEL**.
3. La **troisième tâche,** “**install php7 packages**”, installe les paquets php7 avec le module “**yum**”. Et les options :
   - le **name** indique l’ensemble des paquets à installer ;
   - **state: latest** indique qu’il faut installer les dernières versions disponibles des paquets ;
   - **enablerepo: "remi-php70"** indique d’utiliser le dépôt **remi-php70** pour les installer ;
   - **changed_when** : force le changement d'état, c’est-à-dire qu'avec cette condition à **yes**, l'exécution de la tâche provoquera un changement ;
   - **notify: ["apache restart"]** indique que si la tâche change d'état (et uniquement si la tâche a engendré un changement), **notify** fait appel au **handler** "apache restart" pour relancer le service Apache.

> Source : https://openclassrooms.com/fr/courses/2035796-utilisez-ansible-pour-automatiser-vos-taches-de-configuration/6373887-controlez-lexecution-des-operations-et-enchainez-plusieurs-actions

Création du Handler :

```
nano roles/apache/handlers/main.yml
```

```
---
- name: "apache restart"
  service:
    name: "httpd"
    state: "restarted"
```

Le module **service** est utilisé pour relancer le service **httpd** avec l’option **state: "restarted"**.

Modification du main.yml du rôle mariadb :

```
nano roles/mariadb/tasks/main.yml
```

```
---


# Installation des paquets mariadb serveur et son extension Python
- name: "mariadb-server installation"
  yum:
    name:  "mariadb-server,MySQL-python"
    state: "installed"

# Active le service MariaDB
- name: "start mariadb service"
  service:
    name:  "mariadb"
    state: "started"
    enabled: yes
```

La première tâche "**mariadb-server installation**" va installer **mariadb** serveur et son extension **Python** avec le module “**yum**”. Le **name: "mariadb-server,MySQL-python"** indique les paquets concernés et **state: "installed"** indique que les paquets doivent être installés.

--> **state** peut prendre indifféremment la valeur **present** ou **installed**. Ce sont deux alias.

La deuxième tâche “**start mariadb service**” va activer et démarrer le service MariaDB avec le module “**service**”. Le **name: "mariadb"** indique le service concerné, **state: "started"** indique de démarrer le service et **enabled:yes**, de l'activer.

> Source : https://openclassrooms.com/fr/courses/2035796-utilisez-ansible-pour-automatiser-vos-taches-de-configuration/6373887-controlez-lexecution-des-operations-et-enchainez-plusieurs-actions

Modification du main.yml du rôle commun (dans mediawiki) :

```
nano roles/mediawiki/commun/defaults/main.yml
```

```
---

# nom de la base de données
mediawiki_db_name: "mediawiki"

# nom de l'utilisateur de la base de données et son mot de passe
mediawiki_db_user: "mediawiki"
mediawiki_db_password: !vault |
      $ANSIBLE_VAULT;1.1;AES256
366639346333039613137616236363366663537663331343061616430366263323330323235383939
393266623063363262643436373462353762346336363132 0a6366623630323765633031636433663
762623738353465303566333562323364636432636532656536636134383566303031653337666 4
663337653966356331 0a386439653931623333316532326262343562326638313266343236346536
      3032

# nom et mot de passe de l'administrateur Mediawiki
mediawiki_admin_user: "admin"
mediawiki_admin_password: !vault |
      $ANSIBLE_VAULT;1.1;AES256
366639346333039613137616236363366663537663331343061616430366263323330323235383939
393266623063363262643436373462353762346336363132 0a6366623630323765633031636433663
762623738353465303566333562323364636432636532656536636134383566303031653337666 4
663337653966356331 0a386439653931623333316532326262343562326638313266343236346536
      3032

# nom du Mediawiki et son titre
mediawiki_name: "mediawiki"
mediawiki_title: "ELS"

# l'emplacement du répertoire d'installation de Mediawiki
mediawiki_directory: "/var/www/html/{{mediawiki_name}}"

# répertoire de maintenance de Mediawiki
mediawiki_maintenance_directory: "{{mediawiki_directory}}/maintenance"

# Definie le premier node du groupe mariadb
mediawiki_db_host: "{{groups.db.0}}"

# l'url des sources Mediawiki
mediawiki_archive_url: "https://releases.wikimedia.org/mediawiki/1.31/mediawiki-1.31.1.tar.gz"
```

Modification du main.yml du rôle confdb (dans mediawiki) :

```
nano roles/mediawiki/confdb/meta/main.yml
```

```
dependencies:
  - role: "mediawiki/commun"
```

Modification du main.yml du rôle confdb (dans mediawiki) :

```
nano roles/mediawiki/confdb/tasks/main.yml
```

```
---

#1. Installation de la base de donnée Mediawiki
- name: "mediawiki database"
  mysql_db:
name: "{{mediawiki_db_name}}"
state: present

#2. Création d’un accès utilisateur et attribution des privilèges sur la base Mediawiki
- name: "mediawiki user+privileges"
  mysql_user:
name: "{{mediawiki_db_user}}"
password: "{{mediawiki_db_password}}"
priv: "{{mediawiki_db_name}}.*:ALL"
host: "{{item}}"
state: present
  with_items: "{{groups.apache}}"
```

```
nano roles/mediawiki/confapache/meta/main.yml
```

```
dependencies:
  - role: "mediawiki/commun"
```

```
nano roles/mediawiki/confapache/tasks/main.yml
```

```
---

#1. Création du repertoire pour l’installation des fichiers Mediawiki
- name: "mediawiki directory"
  file:
path: "{{mediawiki_directory}}"
owner: "apache"
group: "apache"
state: directory

#2. Décompresse le fichier source archive Mediawiki et le formate sans extension
- name: "uncompress mediawiki archive"
  unarchive:
src: "{{mediawiki_archive_url}}"
dest: "{{mediawiki_directory}}"
owner: "apache"
group: "apache"
remote_src: yes
# supprime mediawiki-1.xx.x/ du chemin
extra_opts: --transform=s/mediawiki-[0-9\.]*\///

#3. Exécute la tâche avec l'utilisateur apache, se place dans le répertoire de maintenance et exécute la
commande de configuration si le fichier localsetting.php n’existe pas
- name: "mediawiki configuration"
  become: yes
  become_user: "apache"
  args:
   creates: "{{mediawiki_directory}}/LocalSettings.php"
   chdir: "{{mediawiki_maintenance_directory}}"
  command:
php install.php --scriptpath /{{mediawiki_name}}
    --dbname mediawiki --lang fr
    --dbuser {{mediawiki_db_user}}
    --dbpass {{mediawiki_db_password}}
    --pass {{mediawiki_admin_password}}
    --dbserver {{mediawiki_db_host}}
    {{mediawiki_title}} {{mediawiki_admin_user}}
  run_once: yes
```

```
  delegate_to: "{{item}}"
  with_items: "{{groups.apache}}"

#4. Exécute la tâche avec l'utilisateur apache, se place dans le répertoire de maintenance et exécute la commande de mise à jour de la baseune seule fois
- name: "mediawiki db update"
  become: yes
  become_user: "apache"
  command:
php update.php --quick
  args:
chdir: "{{mediawiki_maintenance_directory}}"
  # La mise à jour à besoin d'être lancée une seule fois
  run_once: yes
  register: resultat
  changed_when: "' ...done.' in resultat.stdout"
```

Chiffrer vos mots de passes :

Pour chiffrer le mot de passe qui sera utilisé dans le fichier de variables, vous pouvez utiliser la commande suivante :

```
ansible-vault encrypt_string 'foobar' --name 'mediawiki_db_password'
New Vault password: Ansible2019
Confirm New Vault password: Ansible2019
```

Résultat :

```
mediawiki_db_password: !vault |
        $ANSIBLE_VAULT;1.1;AES256
        34366266383764373965333237376137666134373733313832636334363236633965323938653762
        33613439613831636331613561366631356433346630326 30a366334643939653831626162323331
        63313361623732313436653634646438366161393464623937326161386562363362616665386231
        37356461353665656 10a6365316232626635636535626536373666366165373233383739313264633
        3830
Encryption successful
```

> On va reporter cette chaine dans notre fichier variables.

Création des playbooks :

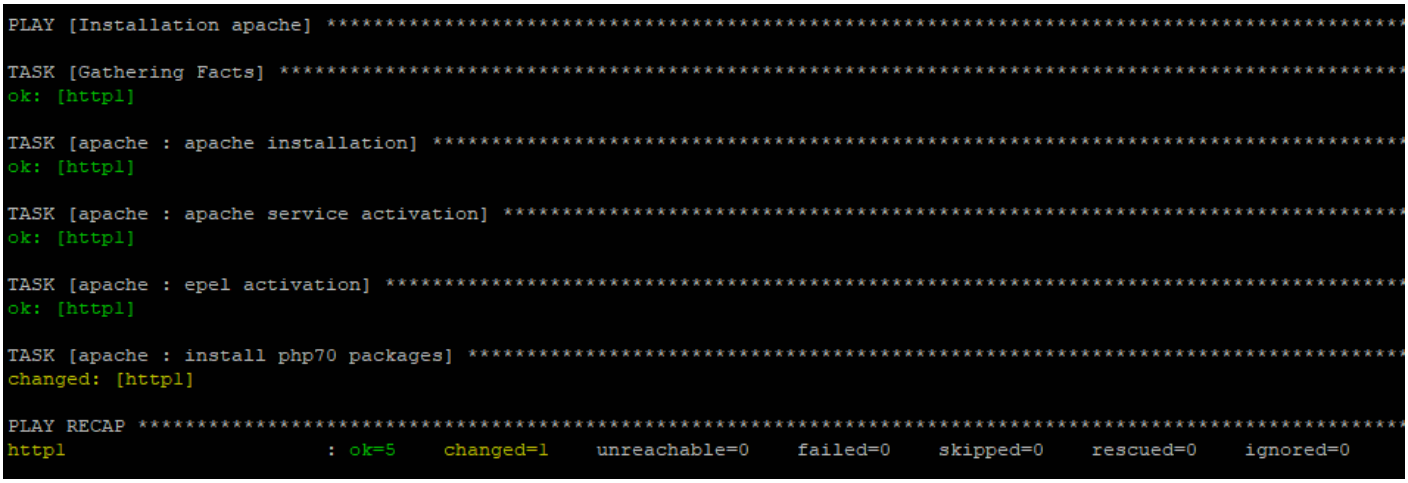
Playbook d'installation d'apache

```
nano install-apache.yml
```

```
---
- name: "Installation apache"
  hosts: http1
  roles:
    - role: "apache"
      php_install: yes
```

```
ansible-playbook -i inventaire.ini --user user-ansible --become --ask-pass --ask-become-pass install-apache.yml
```

Résultat :



On peut maintenant accéder à l'IP d'HTTP1 :

> Apache a bien été installé !

Playbook d'installation de mariadb :

```
nano install-mariadb.yml
```

```
---
- name: "Installation MariaDB"
  hosts: bdd1
  gather_facts: no
  roles:
    - role: mariadb
```

On lance :

```
ansible-playbook -i inventaire.ini --user user-ansible --become --ask-pass --ask-become-pass install-mariadb.yml
```

Ce qui nous donne :

```
PLAY [Installation MariaDB] ****************************************************************************************************************************

TASK [mariadb : mariadb-server installation] ***********************************************************************************************************
changed: [bdd1]

TASK [mariadb : start mariadb service] *****************************************************************************************************************
changed: [bdd1]

PLAY RECAP *********************************************************************************************************************************************
bdd1                       : ok=2    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Pour vérifier, on peut se connecter sur le node bdd1 et faire :

```
systemctl status mariadb
```

> MariaDB est bien fonctionnel !

Playbook d'installation de mediawiki :

```
nano install-mediawiki.yml
```

```
- name: "MediaWiki db configuration"
  hosts: db
  gather_facts: no
  tags: [ "mariadb", "mysql" ]
  roles:
    - role: "mediawiki/confdb"

- name: "MediaWiki apache configuration"
  hosts: apache
  gather_facts: no
  tags: "apache"
  roles:
    - role: "mediawiki/confapache"
```

Revision #78
Created 4 October 2021 21:19:56 by Nehemie
Updated 26 December 2022 13:44:05 by Nehemie