

Tutorial 2 Configuration d'Ansible.

Table des matières

- [Objectifs](#)
- [Ressources](#)
- [Pré-Requis](#)

Objectifs

Pour notre premier TP, nous allons travailler avec un template de projet. (simple cad sans utiliser les rôles). Cela nous permettra de nous familiariser avec les concepts clés d'Ansible.

Ainsi nous allons :

- Éditer un fichier de configuration.
- Tester notre configuration avec la commande `ansible-config`.
- Écrire un fichier d'inventaire pour cataloguer notre infrastructure.
- Tester notre inventaire avec la commande `ansible-inventory`.
- Utiliser le module Ping en mode commande pour tester notre inventaire.
- Utiliser le module Debug en mode commande pour tester nos variables d'inventaire.

Ressources

- Environnement
- Temps : 60 mn.

Pré-Requis

- avoir mis en place l'environnement dans cloud shell.
[TP01 - Mise en place du Lab](#)

Énoncé

Etape 1 Initier le projet TP01

- dans l'éditeur créer un nouveau dossier dans myWork du control-node ubuntu-c. Ce sera notre repertoire de travail pour ce TP.

Etape 2 Configurer Ansible.

On peut spécifier des preferences de configuration dans le fichier `ansible.cfg`.

Celui ci peut se retrouver à plusieurs endroits,mais il est fortement conseillé de le placer dans le repertoire du projet.

1. tester la configuration actuelle.
 - Nous allons utiliser la commande `ansible-config` pour connaître la configuration.
 - Ouvrir un terminal sur Ubuntu-c avec le profil `ansible/password`.

- Placer vous dans notre repertoire de travail.
`cd ~/ansibleMelodie/myWork/TP02`
 - afficher le fichier de configuration utilisé.
`ansible-config --version`
 - afficher la configuration utilisée.
`ansible-config view`
2. copiez le fichier ansible.cfg du projet template via la commande dans le terminal.
- ```
cp ../../templates/ansible_simple/ansible.cfg ansible.cfg
```
3. visualiser ce fichier dans l'éditeur.
- Nous conservons les valeurs par défaut.
  - Nous ne contrôlons pas l'empreinte de la cible lors d'une connexion SSH.
4. tester la configuration actuelle.
- Nous allons utiliser la commande ansible-config pour connaître la configuration.

- afficher le fichier de configuration utilisé.

`ansible-config --version`

```
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible-config --version
ansible-config [core 2.14.2]
 config file = /home/ansible/ansibleMelodie/myWork/TP01/ansible.cfg
 configured module search path = ['/home/ansible/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
 ansible python module location = /usr/local/lib/python3.10/dist-packages/ansible
```

- afficher la configuration utilisée.

`ansible-config view`

```
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible-config view
[defaults]
host key checking = False
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$
```

## Etape 3 Configurons nos machines hôtes.

Le fichier d'inventaire permet de cataloguer notre infrastructure.

Il permet de lister les différentes machines hôtes que l'on veut administrer.

1. copiez le fichier 00\_inventory.yml du projet template via la commande dans le terminal.  
`cp ../../templates/ansible_simple/00_inventory.yml 00_inventory.yml`
2. Affichez le contenu du fichier 00\_inventory.yml
3. tester la configuration actuelle de nos machines hôtes.
  - Nous allons utiliser la commande ansible-inventory pour connaître la configuration.
  - afficher les hôtes inscrits dans l'inventaire.  
`ansible-inventory -i 00_inventory.yml --list --yaml`

ou ✓ `ansible-inventory -i 00_inventory.yml --graph`

Applications gmail SG ABA English - The b... catilou accord : Piano blues Buildin

```
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible-inventory -i 00_inventory.yml --graph
@all:
|--@ungrouped:
|--@control:
| |--ubuntu-c
|--@managed:
| |--@centos:
| | |--centos1
| | |--@ubuntu:
| | | |--ubuntu1
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$
```

#### 4. tester la connexion vers nos machines hôtes.

- Nous allons le module ping pour se connecter à nos hôtes.
- Consulter la [documentation Ansible](#)
- Ce module envoie un ping à l'hôte et reçoit en retour un pong si tout va bien.
- faire du ping/pong avec les hôtes inscrits.

`ansible -i 00_inventory.yml all -m ping`

```
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible -i 00_inventory.yml all -m ping
ubuntu1 | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python3"
 },
 "changed": false,
 "ping": "pong"
}
ubuntu-c | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/bin/python3"
 },
 "changed": false,
 "ping": "pong"
}
centos1 | SUCCESS => {
 "ansible_facts": {
 "discovered_interpreter_python": "/usr/libexec/platform-python"
 },
 "changed": false,
 "ping": "pong"
}
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$
```

- nous pouvons spécifier un hôte particulier `ansible -i 00_inventory.yml centos1 -m ping`
- un group

`ansible -i 00_inventory.yml ubuntu -m ping` Comme vous pouvez le remarquer nous n'avons pas paramétré toutes les machines hôtes.

Nous allons le faire maintenant.

#### 5. Modifier le fichier d'inventaire pour ajouter les hôtes manquants à savoir

- ubuntu2 et unbuntu3
- centos2 et centos3

#### 6. Tester votre configuration

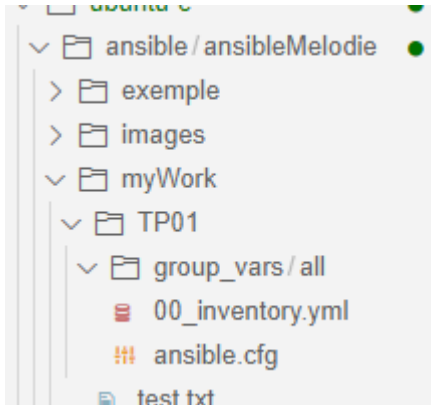
[correction](#)

Etape 3 Configurons les variables de nos machines hôtes.

Le fichier d'inventaire permet de cataloguer notre infrastructure. Il permet de lister les différentes machines hôtes que l'on veut administrer. Il permet aussi de paramétrer nos actions sur les hôtes au moyen de variables. Ces variables peuvent se trouver dans le fichier d'inventaire. Mais en général on préfère les stocker dans les répertoires `group_vars` et `host_vars`.

### 1. Ajout d'une variable pour tous les hôtes.

1. via l'éditeur créer un dossier `<group_vars>` et un sous-dossier `all`



2. dans le dossier `all` créer un fichier `variables.yml`

3. dans ce fichier inscrivez une variable

- nom : `maVariable`
- valeur : `all`

4. sauvegarder

5. tester la configuration actuelle de nos machines hôtes.

- Nous allons utiliser la commande `ansible-inventory` pour connaître la configuration.
- afficher les hôtes inscrits et leurs variables.

```
ansible-inventory -i 00_inventory.yml --graph --vars
```

```
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible-inventory -i 00_inventory.yml --graph --vars
@all:
 |--@ungrouped:
 |--@control:
 | |--ubuntu-c
 | | |--{ansible_connection = local}
 | | |--{maVariable = all}
 |--@managed:
 | |--@centos:
 | | |--centos1
 | | | |--{maVariable = all}
 | | |--centos2
 | | | |--{maVariable = all}
 | | |--centos3
 | | | |--{maVariable = all}
 | |--@ubuntu:
 | | |--ubuntu1
 | | | |--{maVariable = all}
 | | |--ubuntu2
 | | | |--{maVariable = all}
 | | |--ubuntu3
 | | | |--{maVariable = all}
 |--{maVariable = all}
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$
```

6. tester les variables transmises à nos machines hôtes.

- Nous allons le module `debug` pour se vérifier sur nos hôtes.
- Consulter la [documentation Ansible](#)
- Ce module permet d'afficher le contenu d'une variable sur une machine hôtes ou un message utilisant un langage de templates `jinja2`.
- afficher le contenu de `maVariable` sur les hôtes inscrits.

```
ansible -i 00_inventory.yml all -m debug -a 'var=maVariable'
```

```

ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible -i 00_inventory.yml all -m debug -a 'var=maVariable'
ubuntu-c | SUCCESS => {
 "maVariable": "all"
}
centos2 | SUCCESS => {
 "maVariable": "all"
}
centos1 | SUCCESS => {
 "maVariable": "all"
}
centos3 | SUCCESS => {
 "maVariable": "all"
}
ubuntu2 | SUCCESS => {
 "maVariable": "all"
}
ubuntu1 | SUCCESS => {
 "maVariable": "all"
}
ubuntu3 | SUCCESS => {
 "maVariable": "all"
}
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$

```

- afficher un message avec le contenu de maVariable sur les hôtes inscrits.

```

ansible -i 00_inventory.yml all -m debug -a "msg='le contenu de ma
variable est {{ maVariable }}.'"

```

```

ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible -i 00_inventory.yml all -m debug -a "msg='le contenu de ma variable est {{ maVariable }}.'"
ubuntu-c | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
centos1 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
centos2 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
centos3 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
ubuntu2 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
ubuntu1 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
ubuntu3 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$

```

## 2. Ajouter de ma même façon une variable pour tous les hôtes du group centos.

- nom : maVariable
- valeur : centos

1. créer un dossier centos dans group\_vars et un fichier maVariable.yml dans centos.  
noter que le nom du fichier n'a pas d'importance..
2. tester la configuration actuelle de nos machines hôtes.
3. tester les variables transmises à nos machines hôtes.

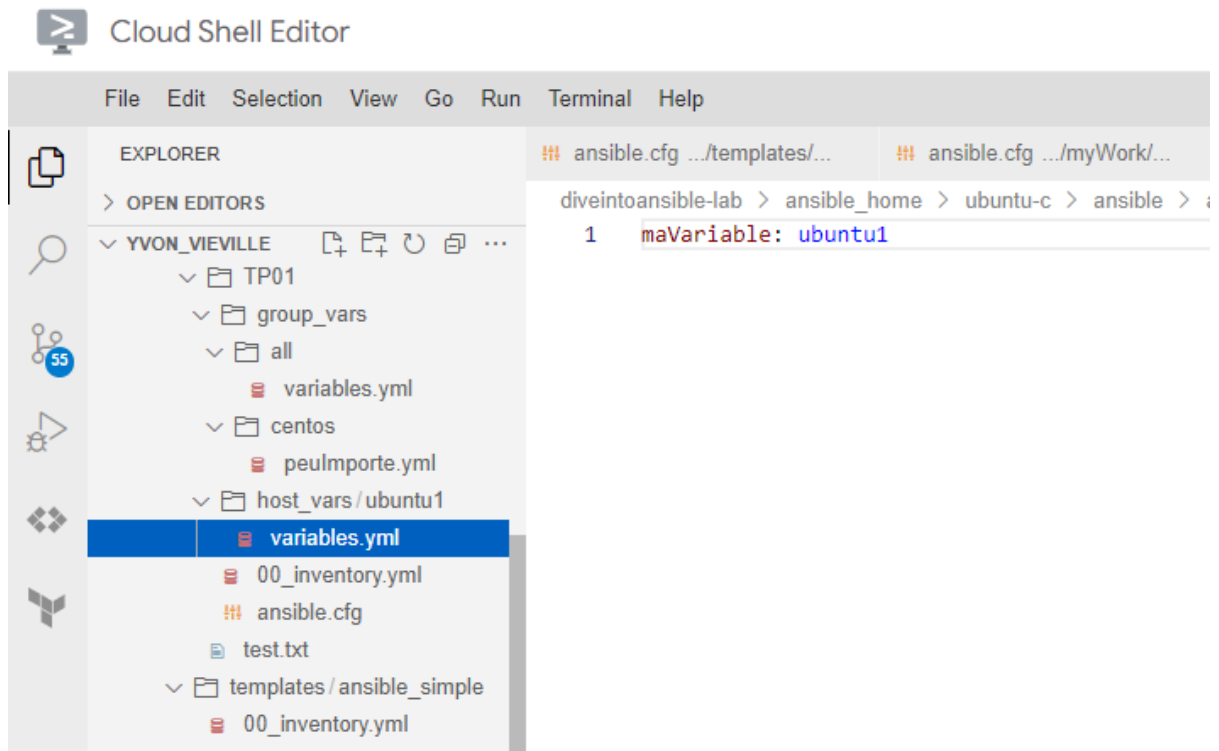
[correction](#)

## 3. Ajout d'une variable l'hôte ubuntu1.

1. via l'éditeur créer un dossier <host\_vars> et un sous-dossier ubuntu1
2. dans le dossier ubuntu1 créer un fichier variables.yml
3. dans ce fichier inscrivez une variable

- nom : maVariable
- valeur : ubuntu1

## 4. sauvegarder



## 5. tester la configuration actuelle de nos machines hôtes.

```
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible-inventory -i 00_inventory.yml --graph --vars
@all:
|--@ungrouped:
|--@control:
| |--ubuntu-c
| | |--{ansible_connection = local}
| | |--{maVariable = all}
|--@managed:
| |--@centos:
| | |--centos1
| | | |--{maVariable = centos}
| | |--centos2
| | | |--{maVariable = centos}
| | |--centos3
| | | |--{maVariable = centos}
| | |--{maVariable = centos}
| |--@ubuntu:
| | |--ubuntu1
| | | |--{maVariable = ubuntu1}
| | |--ubuntu2
| | | |--{maVariable = all}
| | |--ubuntu3
| | | |--{maVariable = all}
| |--{maVariable = all}
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$
```

## 6. tester les variables transmises à nos machines hôtes.

```
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible -i 00_inventory.yml all -m debug -a "msg='le contenu de ma variable est {{ maVariable }}.'"
ubuntu-c | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
centos1 | SUCCESS => {
 "msg": "le contenu de ma variable est centos."
}
centos2 | SUCCESS => {
 "msg": "le contenu de ma variable est centos."
}
centos3 | SUCCESS => {
 "msg": "le contenu de ma variable est centos."
}
ubuntu2 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
ubuntu1 | SUCCESS => {
 "msg": "le contenu de ma variable est ubuntu1."
}
ubuntu3 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$
```

#### 4. Ajouter de la même façon une variable pour l'hôte centos2.

- nom : maVariable
- valeur : centos2

1. créer un dossier centos2 dans host\_vars et un fichier maVariable.yml dans centos2.  
noter que le nom du fichier n'a pas d'importance..
2. tester la configuration actuelle de nos machines hôtes.
3. tester les variables transmises à nos machines hôtes.

```
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible -i 00_inventory.yml all -m debug -a "msg='le contenu de ma variable est {{ maVariable }}.'"
ubuntu-c | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
centos2 | SUCCESS => {
 "msg": "le contenu de ma variable est centos2."
}
centos1 | SUCCESS => {
 "msg": "le contenu de ma variable est centos."
}
ubuntu1 | SUCCESS => {
 "msg": "le contenu de ma variable est ubuntu1."
}
centos3 | SUCCESS => {
 "msg": "le contenu de ma variable est centos."
}
ubuntu2 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
ubuntu3 | SUCCESS => {
 "msg": "le contenu de ma variable est all."
}
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$
```

#### correction

4. ajouter une variable mavariable2 avec le contenu que vous voulez dans le fichier maVariable de centos2 et retester.

```
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$ ansible-inventory -i 00_inventory.yml --graph --vars
@all:
|--@ungrouped:
|--@control:
| |--ubuntu-c
| | |--{ansible_connection = local}
| | |--{maVariable = all}
|--@managed:
| |--@centos:
| | |--centos1
| | | |--{maVariable = centos}
| | |--centos2
| | | |--{maVariable = centos2}
| | | |--{maVariable2 = toto}
| | |--centos3
| | | |--{maVariable = centos}
| | |--{maVariable = centos}
| |--@ubuntu:
| | |--ubuntu1
| | | |--{maVariable = ubuntu1}
| | |--ubuntu2
| | | |--{maVariable = all}
| | |--ubuntu3
| | | |--{maVariable = all}
| |--{maVariable = all}
ansible@ubuntu-c:~/ansibleMelodie/myWork/TP01$
```

## Conclusion et feed-back

## Correction

## 💡💡💡💡 Idées

•