

Version : **2023.02**

Dernière mise-à-jour : 2022/09/29 13:15

LCF802 - Les Commandes ansible, ansible-playbook et ansible-galaxy

Contenu du Module

- **LCF802 - Les Commandes ansible, ansible-playbook et ansible-galaxy**
 - Contenu du Module
 - LAB #1 - Débuter avec Ansible
 - 1.1 - La Commande ansible
 - LAB #2 - La Commande ansible-playbook
 - 2.1 - Playbook Files
 - 2.2 - Tasks
 - 2.3 - Handlers
 - 2.4 - Modules
 - 2.4.1 - Modules pour des Paquets
 - 2.4.2 - Modules pour des Fichiers
 - 2.4.3 - Modules pour le Système
 - 2.5 - Inventory Files
 - 2.6 - Privilèges
 - 2.6.1 - Création de Groupes
 - LAB #3 - La Commande ansible-galaxy
 - 3.1 - Utiliser des Rôles avec un Play Book
 - 3.2 - Ansible Galaxy

LAB #1 - Débuter avec Ansible

1.1 - La Commande ansible

Utilisez la commande **ansible** pour vérifier si la machine **TargetA** fonctionne :

```
[trainee@centos8 ~]$ ansible all -i 10.0.2.52, -m ping
10.0.2.52 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```



Important : Cette commande indique à Ansible qu'elle doit exécuter le module indiqué par l'option **-m** dans **tous** (*all*) les serveurs indiqués par l'option **-i**. La virgule placée après l'adresse IP indique à la commande ansible que la valeur est une adresse IP et non le nom d'un fichier.

Les options de la commande **ansible** sont :

```
[trainee@centos8 ~]$ ansible --help
usage: ansible [-h] [--version] [-v] [-b] [--become-method BECOME_METHOD]
              [--become-user BECOME_USER] [-K] [-i INVENTORY] [--list-hosts]
              [-l SUBSET] [-P POLL_INTERVAL] [-B SECONDS] [-o] [-t TREE] [-k]
              [--private-key PRIVATE_KEY_FILE] [-u REMOTE_USER]
              [-c CONNECTION] [-T TIMEOUT]
              [--ssh-common-args SSH_COMMON_ARGS]
              [--sftp-extra-args SFTP_EXTRA_ARGS]
```



```
-B SECONDS, --background SECONDS
    run asynchronously, failing after X seconds
    (default=N/A)
-C, --check
    don't make any changes; instead, try to predict some
    of the changes that may occur
-D, --diff
    when changing (small) files and templates, show the
    differences in those files; works great with --check
-M MODULE_PATH, --module-path MODULE_PATH
    prepend colon-separated path(s) to module library (def
    ault=~/.ansible/plugins/modules:/usr/share/ansible/plu
    gins/modules)
-P POLL_INTERVAL, --poll POLL_INTERVAL
    set the poll interval if using -B (default=15)
-a MODULE_ARGS, --args MODULE_ARGS
    The action's options in space separated k=v format: -a
    'opt1=val1 opt2=val2'
-e EXTRA_VARS, --extra-vars EXTRA_VARS
    set additional variables as key=value or YAML/JSON, if
    filename prepend with @
-f FORKS, --forks FORKS
    specify number of parallel processes to use
    (default=5)
-h, --help
    show this help message and exit
-i INVENTORY, --inventory INVENTORY, --inventory-file INVENTORY
    specify inventory host path or comma separated host
    list. --inventory-file is deprecated
-l SUBSET, --limit SUBSET
    further limit selected hosts to an additional pattern
-m MODULE_NAME, --module-name MODULE_NAME
    Name of the action to execute (default=command)
-o, --one-line
    condense output
-t TREE, --tree TREE
    log output to this directory
-v, --verbose
    verbose mode (-vvv for more, -vvvv to enable
    connection debugging)
```

Privilege Escalation Options:

control how and which user you become as on target hosts

- become-method BECOME_METHOD
privilege escalation method to use (default=sudo), use
`ansible-doc -t become -l` to list valid choices.
- become-user BECOME_USER
run operations as this user (default=root)
- K, --ask-become-pass
ask for privilege escalation password
- b, --become
run operations with become (does not imply password
prompting)

Connection Options:

control as whom and how to connect to hosts

- private-key PRIVATE_KEY_FILE, --key-file PRIVATE_KEY_FILE
use this file to authenticate the connection
- scp-extra-args SCP_EXTRA_ARGS
specify extra arguments to pass to scp only (e.g. -l)
- sftp-extra-args SFTP_EXTRA_ARGS
specify extra arguments to pass to sftp only (e.g. -f,
-l)
- ssh-common-args SSH_COMMON_ARGS
specify common arguments to pass to sftp/scp/ssh (e.g.
ProxyCommand)
- ssh-extra-args SSH_EXTRA_ARGS
specify extra arguments to pass to ssh only (e.g. -R)
- T TIMEOUT, --timeout TIMEOUT
override the connection timeout in seconds
(default=10)
- c CONNECTION, --connection CONNECTION
connection type to use (default=smart)
- k, --ask-pass
ask for connection password

```
-u REMOTE_USER, --user REMOTE_USER
    connect as this user (default=None)
```

Some actions do not make sense in Ad-Hoc (include, meta, etc)

Les quatre machines **WebXX** mises à votre disposition sont dans le même réseau que la machine centos8 et ont la même configuration ssh et sudo que la machine **debian10** :

Machine	Nom d'hôte	Adresse IP
Web01	web01.i2tch.loc	10.0.2.54
Web02	web02.i2tch.loc	10.0.2.55
Web03	web03.i2tch.loc	10.0.2.56
Web04	web04.i2tch.loc	10.0.2.57

Utilisez la commande **scp** pour copier le fichier **authorized_keys** de la machine virtuelle **ansible** vers les machines virtuelles **Web01**, **Web02**, **Web03** et **Web04** :

```
[trainee@centos8 ~]$ scp .ssh/authorized_keys trainee@10.0.2.54:/home/trainee/.ssh/authorized_keys
The authenticity of host '10.0.2.54 (10.0.2.54)' can't be established.
ECDSA key fingerprint is SHA256:sEfHBv9azmK60cjqF/aJgUc9jg56slNaZQdAUcvB0vE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.54' (ECDSA) to the list of known hosts.
Debian GNU/Linux 9
trainee@10.0.2.54's password: trainee
authorized_keys
100% 888    1.8MB/s   00:00
[trainee@centos8 ~]$ scp .ssh/authorized_keys trainee@10.0.2.55:/home/trainee/.ssh/authorized_keys
The authenticity of host '10.0.2.55 (10.0.2.55)' can't be established.
ECDSA key fingerprint is SHA256:sEfHBv9azmK60cjqF/aJgUc9jg56slNaZQdAUcvB0vE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.55' (ECDSA) to the list of known hosts.
Debian GNU/Linux 9
trainee@10.0.2.55's password: trainee
authorized_keys
```

```
100% 888      2.1MB/s   00:00
[trainee@centos8 ~]$ scp .ssh/authorized_keys trainee@10.0.2.56:/home/trainee/.ssh/authorized_keys
The authenticity of host '10.0.2.56 (10.0.2.56)' can't be established.
ECDSA key fingerprint is SHA256:sEfHBv9azmK60cjQF/aJgUc9jg56slNaZQdAUcvB0vE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.56' (ECDSA) to the list of known hosts.
Debian GNU/Linux 9
trainee@10.0.2.56's password: trainee
authorized_keys
100% 888      1.7MB/s   00:00

[trainee@centos8 ~]$ scp .ssh/authorized_keys trainee@10.0.2.57:/home/trainee/.ssh/authorized_keys
The authenticity of host '10.0.2.57 (10.0.2.57)' can't be established.
ECDSA key fingerprint is SHA256:sEfHBv9azmK60cjQF/aJgUc9jg56slNaZQdAUcvB0vE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.57' (ECDSA) to the list of known hosts.
Debian GNU/Linux 9
trainee@10.0.2.57's password: trainee
authorized_keys
100% 888      1.3MB/s   00:00
```

Modifier le fichier **/etc/hosts** dans **centos8** :

```
[trainee@centos8 ~]$ su -
Password: fenestros
[root@centos8 ~]# vi /etc/hosts
[root@centos8 ~]# cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1        localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.2.45 centos8.ittraining.loc centos8
10.0.2.54 web01.i2tch.loc web01
10.0.2.55 web02.i2tch.loc web02
10.0.2.56 web03.i2tch.loc web03
10.0.2.57 web04.i2tch.loc web04
```

```
[root@centos8 ~]# exit
logout
[trainee@centos8 ~]$
```

Vérifiez que vous pouvez vous connecter aux machines virtuelles **Web01**, **Web02**, **Web03** et **Web04** sans entrer de mot de passe et que la configuration de sudo est correcte :

```
[trainee@centos8 ~]$ ssh -l trainee web01
Debian GNU/Linux 9
Linux web01.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar  9 11:47:41 2022 from 10.0.2.45
trainee@web01:~$ sudo apt update
Réception de:1 http://security.debian.org/debian-security stretch/updates InRelease [53,0 kB]
Ign:2 http://ftp.fr.debian.org/debian stretch InRelease
Réception de:3 http://ftp.fr.debian.org/debian stretch-updates InRelease [93,6 kB]
Réception de:4 http://ftp.fr.debian.org/debian stretch Release [118 kB]
Réception de:5 http://ftp.fr.debian.org/debian stretch Release.gpg [3 177 B]
Réception de:6 http://security.debian.org/debian-security stretch/updates/main Sources [396 kB]
Réception de:7 http://security.debian.org/debian-security stretch/updates/main amd64 Packages [759 kB]
Réception de:8 http://security.debian.org/debian-security stretch/updates/main Translation-en [354 kB]
Réception de:9 http://ftp.fr.debian.org/debian stretch/main Sources [6 736 kB]
Réception de:10 http://ftp.fr.debian.org/debian stretch/main amd64 Packages [7 080 kB]
Réception de:11 http://ftp.fr.debian.org/debian stretch/main Translation-en [5 377 kB]
21,0 Mo réceptionnés en 32s (638 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
```

```
341 packages can be upgraded. Run 'apt list --upgradable' to see them.
trainee@web01:~$ exit
déconnexion
Connection to web01 closed.
[trainee@centos8 ~]$ ssh -l trainee web02
Debian GNU/Linux 9
Linux web02.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64
```

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

```
Last login: Wed Mar  9 11:40:49 2022 from 10.0.2.45
```

```
trainee@web02:~$ sudo apt update
```

```
Réception de:1 http://security.debian.org/debian-security stretch/updates InRelease [53,0 kB]
```

```
Ign:2 http://ftp.fr.debian.org/debian stretch InRelease
```

```
Réception de:3 http://ftp.fr.debian.org/debian stretch-updates InRelease [93,6 kB]
```

```
Réception de:4 http://ftp.fr.debian.org/debian stretch Release [118 kB]
```

```
Réception de:5 http://ftp.fr.debian.org/debian stretch Release.gpg [3 177 B]
```

```
Réception de:6 http://security.debian.org/debian-security stretch/updates/main Sources [396 kB]
```

```
Réception de:7 http://security.debian.org/debian-security stretch/updates/main amd64 Packages [759 kB]
```

```
Réception de:8 http://security.debian.org/debian-security stretch/updates/main Translation-en [354 kB]
```

```
Réception de:9 http://ftp.fr.debian.org/debian stretch/main Sources [6 736 kB]
```

```
Réception de:10 http://ftp.fr.debian.org/debian stretch/main amd64 Packages [7 080 kB]
```

```
Réception de:11 http://ftp.fr.debian.org/debian stretch/main Translation-en [5 377 kB]
```

```
21,0 Mo réceptionnés en 32s (639 ko/s)
```

```
Lecture des listes de paquets... Fait
```

```
Construction de l'arbre des dépendances
```

```
Lecture des informations d'état... Fait
```

```
341 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
trainee@web02:~$ exit
```

```
déconnexion
```

Connection to web02 closed.

```
[trainee@centos8 ~]$ ssh -l trainee web03
```

```
Debian GNU/Linux 9
```

```
Linux web03.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64
```

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

```
Last login: Wed Mar  9 11:40:49 2022 from 10.0.2.45
```

```
trainee@web03:~$ sudo apt update
```

```
Réception de:1 http://security.debian.org/debian-security stretch/updates InRelease [53,0 kB]
```

```
Ign:2 http://ftp.fr.debian.org/debian stretch InRelease
```

```
Réception de:3 http://ftp.fr.debian.org/debian stretch-updates InRelease [93,6 kB]
```

```
Réception de:4 http://ftp.fr.debian.org/debian stretch Release [118 kB]
```

```
Réception de:5 http://ftp.fr.debian.org/debian stretch Release.gpg [3 177 B]
```

```
Réception de:6 http://security.debian.org/debian-security stretch/updates/main Sources [396 kB]
```

```
Réception de:7 http://security.debian.org/debian-security stretch/updates/main amd64 Packages [759 kB]
```

```
Réception de:8 http://security.debian.org/debian-security stretch/updates/main Translation-en [354 kB]
```

```
Réception de:9 http://ftp.fr.debian.org/debian stretch/main Sources [6 736 kB]
```

```
Réception de:10 http://ftp.fr.debian.org/debian stretch/main amd64 Packages [7 080 kB]
```

```
Réception de:11 http://ftp.fr.debian.org/debian stretch/main Translation-en [5 377 kB]
```

```
21,0 Mo réceptionnés en 40s (524 ko/s)
```

```
Lecture des listes de paquets... Fait
```

```
Construction de l'arbre des dépendances
```

```
Lecture des informations d'état... Fait
```

```
341 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
trainee@web03:~$ exit
```

```
déconnexion
```

Connection to web03 closed.

```
[trainee@centos8 ~]$ ssh -l trainee web04
```

```
Debian GNU/Linux 9
```

```
Linux web04.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.
```

```
Last login: Wed Mar  9 11:40:49 2022 from 10.0.2.45
```

```
trainee@web04:~$ sudo apt update
```

```
Réception de:1 http://security.debian.org/debian-security stretch/updates InRelease [53,0 kB]
```

```
Ign:2 http://ftp.fr.debian.org/debian stretch InRelease
```

```
Réception de:3 http://ftp.fr.debian.org/debian stretch-updates InRelease [93,6 kB]
```

```
Réception de:4 http://ftp.fr.debian.org/debian stretch Release [118 kB]
```

```
Réception de:5 http://ftp.fr.debian.org/debian stretch Release.gpg [3 177 B]
```

```
Réception de:6 http://security.debian.org/debian-security stretch/updates/main Sources [396 kB]
```

```
Réception de:7 http://ftp.fr.debian.org/debian stretch/main Sources [6 736 kB]
```

```
Réception de:8 http://security.debian.org/debian-security stretch/updates/main amd64 Packages [759 kB]
```

```
Réception de:9 http://ftp.fr.debian.org/debian stretch/main amd64 Packages [7 080 kB]
```

```
Réception de:10 http://security.debian.org/debian-security stretch/updates/main Translation-en [354 kB]
```

```
Réception de:11 http://ftp.fr.debian.org/debian stretch/main Translation-en [5 377 kB]
```

```
21,0 Mo réceptionnés en 39s (535 ko/s)
```

```
Lecture des listes de paquets... Fait
```

```
Construction de l'arbre des dépendances
```

```
Lecture des informations d'état... Fait
```

```
341 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
trainee@web04:~$ exit
```

```
déconnexion
```

```
Connection to web04 closed.
```

```
[trainee@centos8 ~]$
```

Utilisez ensuite la commande **ansible** avec le module **ping** :

```
[trainee@centos8 ~]$ ansible all -i 10.0.2.54, -m ping
```

```
10.0.2.54 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[trainee@centos8 ~]$ ansible all -i 10.0.2.55, -m ping
10.0.2.55 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

Utilisez de nouveau la commande **ansible** afin de vérifier le statut des machines :

```
[trainee@centos8 ~]$ ansible all -i web01,web02,web03,web04 -m ping
web02 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
web01 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
web03 | SUCCESS => {
```

```
"ansible_facts": {
  "discovered_interpreter_python": "/usr/bin/python"
},
"changed": false,
"ping": "pong"
}
web04 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```



Important : Notez que les résultats ne sont pas retournés dans un ordre séquentiel par rapport à la commande exécutée. En effet, ansible a paralléliser l'exécution des commandes.

Exécutez maintenant la commande **uptime** dans les machines web01, web02, web03 et web04 en utilisant le module **command** :

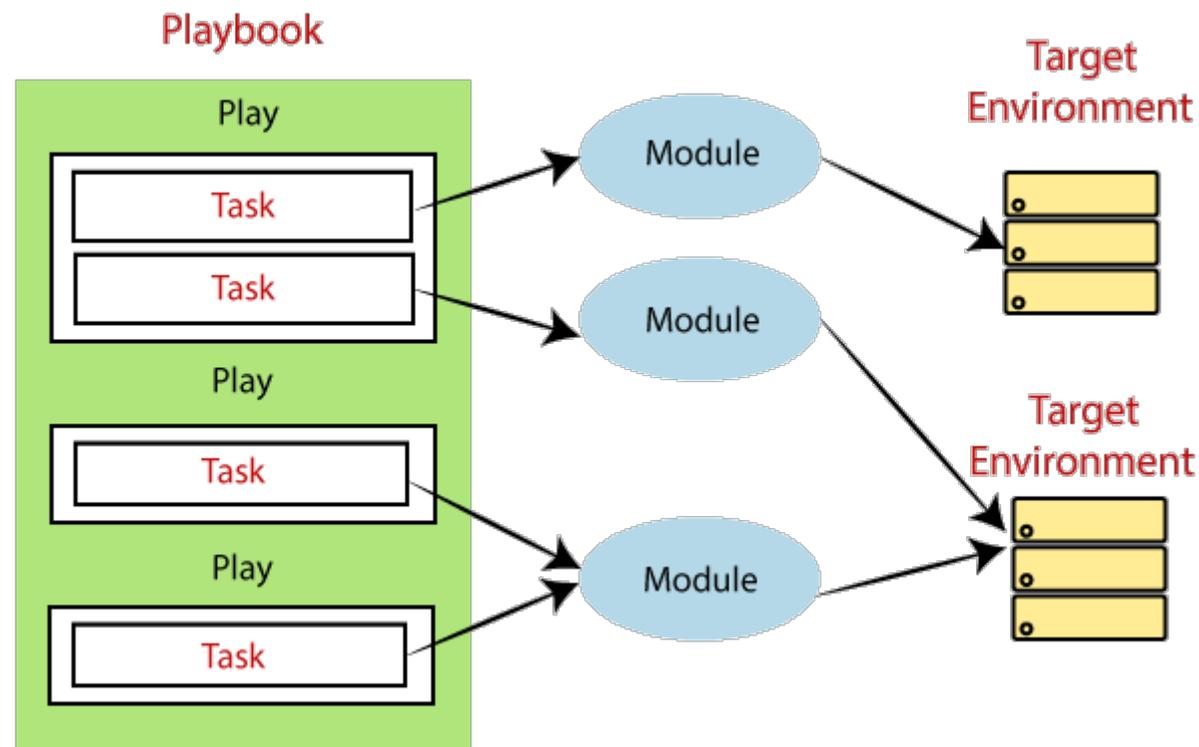
```
[trainee@centos8 ~]$ ansible all -i web01,web02,web03,web04 -m command --args 'uptime'
web02 | CHANGED | rc=0 >>
 11:40:49 up 18:26,  1 user,  load average: 0,24, 0,11, 0,03
web03 | CHANGED | rc=0 >>
 11:40:49 up 18:26,  1 user,  load average: 0,10, 0,03, 0,01
web01 | CHANGED | rc=0 >>
 11:40:49 up 18:26,  1 user,  load average: 0,00, 0,00, 0,00
web04 | CHANGED | rc=0 >>
 11:40:49 up 18:26,  1 user,  load average: 0,00, 0,00, 0,00
```

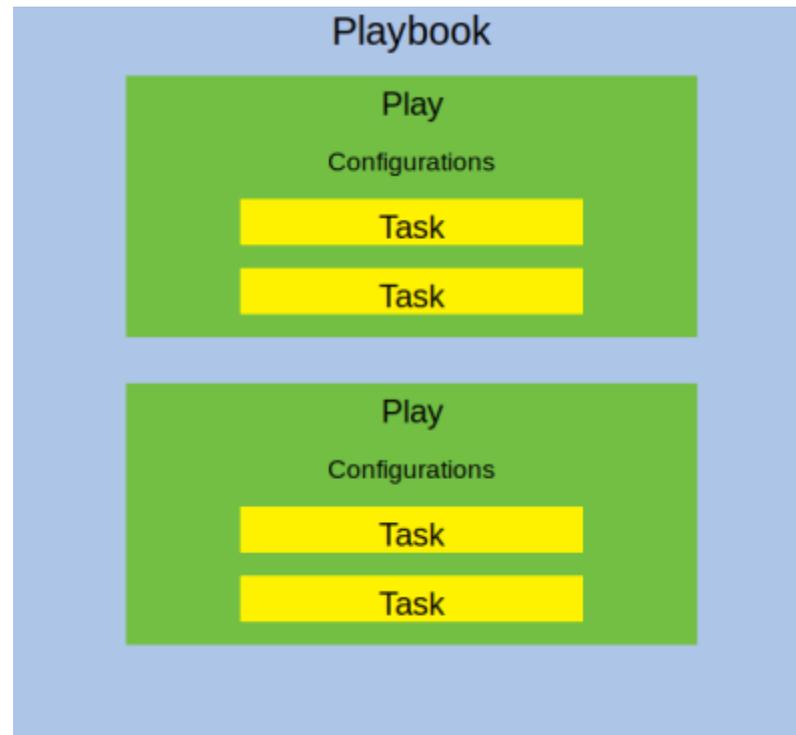
LAB #2 - La Commande ansible-playbook

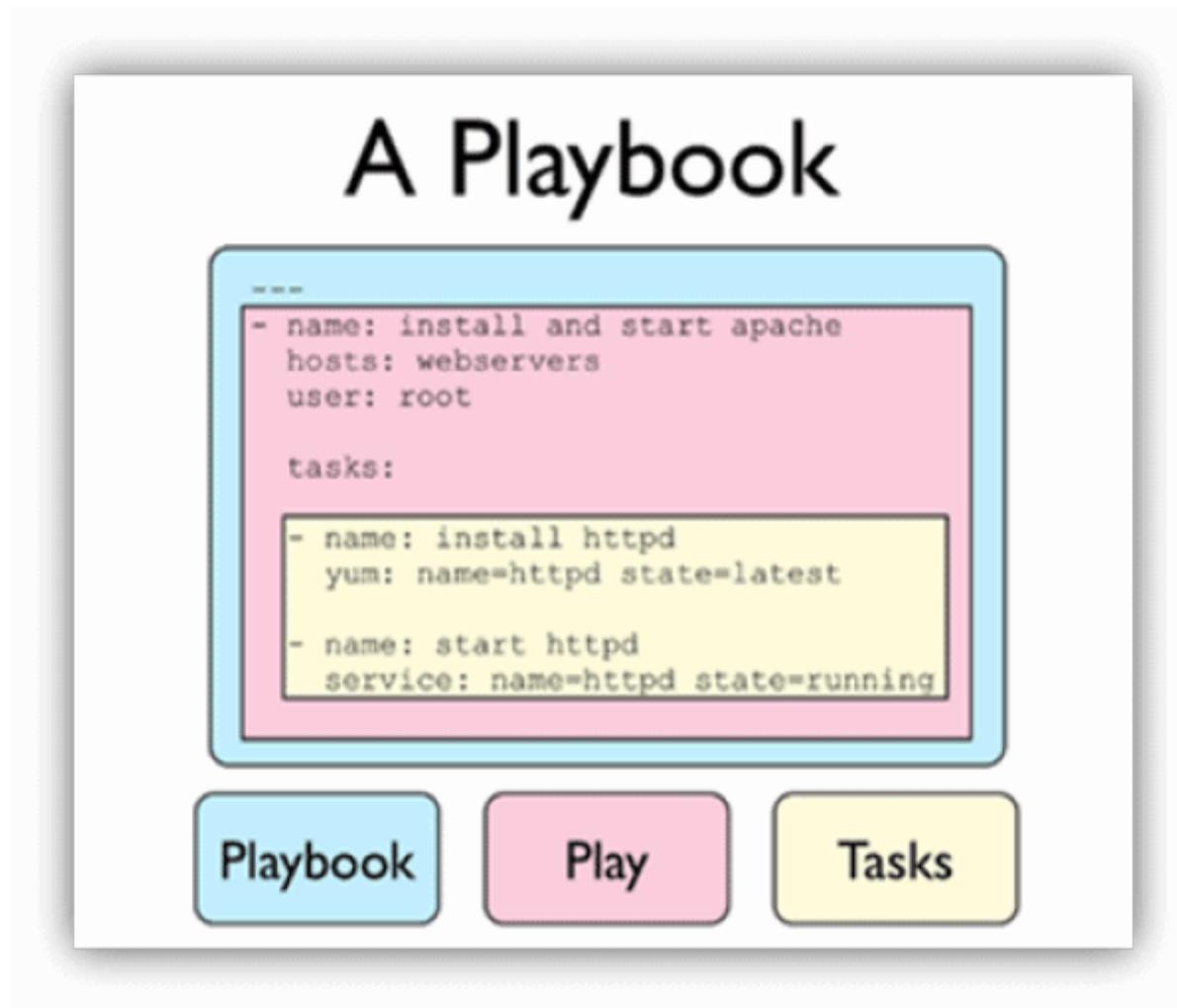
La commande **ansible-playbook** permet d'insérer la configuration d'Ansible dans des fichiers afin de pouvoir ré-exécuter les mêmes jeux de commandes à chaque fois. La commande **ansible-playbook** utilisent des **Playbook Files** ainsi que des **Inventory Files**.

2.1 - Playbook Files

Un Playbook File contient une liste de **Plays** qui regroupent des tâches, appelées des **tasks**. Chaque task est une commande à exécuter sur une machine cible :







Dans l'exemple suivant, le playbook contient deux **Play** commençant par le caractère - :

```
- hosts: all  
  become : true  
  tasks:  
    - name: installer un paquet  
      package: name=<paquet> state=present
```

```
- name: groupe
  group:
    name: <nom_groupe>
    state: present
- hosts: equilibrage01
  become : true
  tasks:
    - name: installer un paquet
      package: name=<paquet> state=present
```

Dans cet exemple :

- Les deux sections **hosts:** sont dans une liste YAML. Chaque section est un **dictionnaire** YAML, aussi connu en tant que **map** ou **hash**. Le dictionnaire YAML contient trois champs - **hosts:**, **become:** et **tasks:**. Dans le deuxième task, le contenu de hosts est le nom d'un seul hôte. Ansible utilise l'Inventory File pour obtenir des informations concernant cet hôte,
- Le mot clef **hosts:** indique sur quel(s) hôte(s) les tasks doit(vent) être effectué(s). La valeur de hosts: est une valeur scalaire sous la forme d'une chaîne,
- Le mot clef **become:** est une valeur booléenne qui indique que les tâches doivent être accomplies par un utilisateur avec privilèges,
- Le mot clef **tasks:** contient une autre liste YAML qui spécifie les tâches à effectuer.

2.2 - Tasks

Un task est une tâche individuelle qui doit être accomplie :

```
tasks:
  - name: installer un paquet
    package: name=<paquet> state=present
  - name: groupe
    group:
      name: <nom_groupe>
      state: present
```

Dans cet exemple :

- Chaque task est un dictionnaire YAML contenant un champs optionnel - **name** et un **module** Ansible - dans le cas ci-dessus, **package** ou **group**,
- Le **module** Ansible **package** est une valeur scalaire qui contient une chaîne qui stipule le paquet à installer. Le contenu est composé de deux paramètres - **name** et **state**. Dans le cas de ce module la valeur du paramètre **state** peut être **latest**, **present**, **absent** ou le numéro d'une version du paquet à installer,
- Le **module** Ansible **group** est un dictionnaire YAML qui contient deux champs - **name** et **state**.

2.3 - Handlers

Un **Handler** est une tâche **conditionnelle** :

```
tasks:
  - name: configure
    copy:
      src: paquet.conf.local
      dest: /etc/paquet/paquet.conf
      owner: root
      group: root
      mode: 0644
    notify:
      - restart service
```

Le mot clef **notify** appelle un **handler** identifié par **restart service** :

```
tasks:
  - name: configure
    copy:
      src: paquet.conf.local
      dest: /etc/paquet/paquet.conf
      owner: root
      group: root
      mode: 0644
    notify:
```

```
- restart service
handlers:
  - name: restart service
    service: name=<paquet.service> state=restarted
```

Dans cet exemple :

- Ansible compare le fichier local **paquet.conf.local** avec le fichier **/etc/paquet/paquet.conf** dans le système ciblé,
- En cas de différences, Ansible copie le fichier **paquet.conf.local** vers **/etc/paquet/paquet.conf** et appelle le handler **restart service**,
- Dans le cas contraire, le handler n'est pas appelé.



Important : Notez que les handlers sont appelés **après** l'exécution de tous les **tasks**. Pour appeler un handler immédiatement, il convient d'utiliser **meta: flush_handlers** à l'emplacement voulu.

2.4 - Modules

La liste complète des modules se trouve à [cette](#) adresse. Chaque module commence par vérifier si un task doit être effectué avant de l'entreprendre.

2.4.1 - Modules pour des Paquets

- Le **module** Ansible **package** est généralement une valeur scalaire qui contient une chaîne qui stipule le paquet à installer ou bien un dictionnaire. Le contenu est composé de deux paramètres - **name** et **state**. Dans le cas de ce module la valeur du paramètre **state** peut être **latest**, **present**, **absent** ou le numéro d'une version du paquet à installer. Ce module utilise le gestionnaire des paquets de la distribution concernée,
- Le **module** Ansible **pip** est décrit par deux paramètres - **name** et **virtualenv**. Dans le cas de ce module la valeur du paramètre **virtualenv** indique la valeur de l'environnement virtuel,
- Le **module** Ansible **gem** est décrit par deux paramètres - **name** et **state**. Dans le cas de ce module la valeur du paramètre **state** est similaire à celle du module package.

2.4.2 - Modules pour des Fichiers

- Le **module** Ansible **copy** copie un fichier du contrôleur vers un système distant. Ce module peut vérifier les propriétés du fichier telles son contenu, le propriétaire avec **owner:**, le groupe avec **group:** ou les permissions avec **0644**,
- Le **module** Ansible **template** est similaire au module **copy** mais il met à jour le contenu avec des valeurs de variables qui lui sont fournies,
- Le **module** Ansible **files** crée ou supprime des fichiers et/ou des répertoires,
- Le **module** Ansible **lineinfile** trouve une ligne existante dans un fichier selon la valeur d'une expression régulière **regexp:** et compare cette ligne avec la valeur d'une ligne spécifiée par **line:**. Dans le cas où la ligne est différente de la valeur **line:**, Ansible modifie le fichier,
- Le **module** Ansible **unarchive** désarchive une archive sur le système distant. Avant de désarchiver l'archive, Ansible vérifie si le fichier spécifié par la valeur de **creates** existe ou pas. Dans le cas où le fichier n'existe pas, Ansible procède au désarchivage. Dans le cas contraire, Ansible considère que l'archive a déjà été désarchivé.

2.4.3 - Modules pour le Système

Les modules les plus importants dans cette catégorie sont :

- **service**,
- **cron:**,
- **user:**,
- **group**.

2.5 - Inventory Files

Un Ansible Inventory File :

- contrôle quels systèmes sont ciblés par ansible-playbook,
- permet des configurations par système,
- est un fichier au format INI.

L'inventory file par défaut est **/etc/ansible/hosts**.

Dans un tel fichier, on peut trouver des listes simples :

```
web01  
web02  
web03  
web04  
web05
```

des plages :

```
web [01:05]
```

des sections :

```
[basededonnees]  
basededonnees01
```

et des adresses IP :

```
10.0.2.14
```

Par exemple, l'Inventory File pour quatre systèmes différents pourrait ressembler à l'exemple suivant :

```
...  
web01 ansible_hostname=10.0.2.54 ansible_user=trainee  
...  
web02 ansible_hostname=10.0.2.55 ansible_user=trainee  
...  
web03 ansible_hostname=10.0.2.56 ansible_user=trainee  
...  
web04 ansible_hostname=10.0.2.57 ansible_user=trainee  
...
```

2.6 - Privilèges

Ansible a besoin d'obtenir les privilèges d'administration pour la plupart des tâches à accomplir. L'escalade des privilèges peut être configurée dans trois endroits différents grâce à l'utilisation :

- du mot clef **become**: dans un task auquel cas l'escalade des privilèges ne s'applique qu'à la tâche,
- du mot clef **become**: dans un play auquel cas l'escalade des privilèges s'applique à tous les tasks dans le play,
- des directives **ansible_become=yes** et **ansible_become_user=<nom_utilisateur>** dans le fichier Inventory File auquel cas l'escalade des privilèges s'applique à tous les play du serveur spécifié.

Notez qu'Ansible peut également utiliser d'autres mécanismes pour contrôler des systèmes :

- **local** - la connexion n'est pas effectuée par SSH car les tasks sont à effectuer sur le localhost,
- **Winrm** - le module Python Windows Remote Management qui peut être installé à côté d'Ansible et qui permet d'exécuter des commandes PowerShell,
- **Docker** - la commande docker exec. La configuration est effectuée par Ansible au lieu d'un DockerFile traditionnel.

2.6.1 - Création de Groupes

Ansible permet le regroupement d'hôtes dans des groupes. Prenons l'exemple suivant :

```
...
[apache]
web01
web02

[nginx]
web03
web04

[equilibrage]
equilibrage01
```

```
...
```

Le groupe **apache** contient deux machines, le groupe **nginx** contient deux machines tandis que le groupe **equilibrage** n'en contient qu'un. Toute configuration appliquée au groupe apache, par exemple, s'applique aux deux machines dans le groupe.

Dans cet exemple, les quatre machines web0X sont sous Debian et utilisent l'utilisateur trainee pour l'escalade des privilèges tandis que le système equilibrage01 est sous CentOS et utilise l'utilisateur **root**.

Afin qu'Ansible sache quel utilisateur utilisé avec ssh, il convient de créer des groupes supplémentaires :

```
...
[apache]
web01
web02

[nginx]
web03
web04

[debian:children]
apache
nginx

[debian:vars]
ansible_user=trainee

[equilibrage]
equilibrage01

[centos:children]
equilibrage

[centos:vars]
ansible_user=root
```

...

Le groupe **debian** contient les deux groupes **apache** et **nginx** en tant qu'enfants. Toute configuration appliquée au groupe **debian**, s'applique aux quatre machines dans les groupes **apache** et **nginx**. Le groupe **debian** est configuré par une variable **ansible_user=trainee**.

Le groupe **centos** contient un seul groupe **equilibrage** en tant qu'enfant. Le groupe **centos** est configuré par une variable **ansible_user=root**.

Cette configuration permet de modifier le Playbook ainsi :

```
- hosts: all
  become : true
  tasks:
    - name: groupe
      group:
        name: <nom_groupe>
        state: present
- hosts: apache
  become : true
  tasks:
    - name: installer Apache
      package: name=apache state=present
- hosts: nginx
  become : true
  tasks:
    - name: installer Nginx
      package: name=nginx state=present
- hosts: equilibrage
  become : true
  tasks:
    - name: installer HAProxy
      package: name=haproxy state=present
```

Dans cet exemple, la valeur du mot clef **hosts**: est le nom d'un groupe.



Important : Notez qu'il existe un groupe spécial appelé **all**. Ce groupe n'a pas besoin d'être défini et contient l'ensemble des hôtes référencés dans l'Inventory File

Dans la machine virtuelle **centos8**, créez le fichier Inventory File suivant :

```
[trainee@centos8 ~]$ vi inventory
[trainee@centos8 ~]$ cat inventory
[basedonnees]
web01

[dotcms]
web02
web03

[equilibrage]
web04

[debian:children]
basedonnees
dotcms
equilibrage

[debian:vars]
ansible_user=trainee
```

Créez ensuite le Play Book suivant :

```
[trainee@centos8 ~]$ vi playbook.yaml
[trainee@centos8 ~]$ cat playbook.yaml
---
- hosts: basedonnees
```

```
become: true
tasks:
  - name: install
    package: name=postgresql state=present
  - name: service
    service: name=postgresql state=started enabled=yes
    tags: service
- hosts: dotcms
  become: true
  tasks:
    - name: install jre
      package: name=openjdk-8-jre state=present
- hosts: equilibrage
  become: true
  tasks:
    - name: install
      package: name=haproxy state=present
    - name: service
      service: name=haproxy state=started enabled=yes
      tags: service
```

Exécutez la commande **ansible-playbook** :

```
[trainee@centos8 ~]$ ansible-playbook -i inventory playbook.yaml
```

```
PLAY [basededonnees]
```

```
*****
*****
```

```
TASK [Gathering Facts]
```

```
*****
*****
```

```
ok: [web01]
```

TASK [install]

changed: [web01]

TASK [service]

ok: [web01]

PLAY [dotcms]

TASK [Gathering Facts]

ok: [web02]

ok: [web03]

TASK [install jre]

ok: [web03]

ok: [web02]

PLAY [equilibrage]

TASK [Gathering Facts]

ok: [web04]

```

TASK [install]
*****
*****
changed: [web04]

TASK [service]
*****
*****
ok: [web04]

PLAY RECAP
*****
*****
web01           : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02           : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03           : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04           : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

```

Il est possible de limiter l'action de la commande **ansible-playbook** avec l'option **-l**. Exécutez la commande **ansible-playbook** en limitant l'action d'ansible-playbook au groupe **equilibrage** :

```

[trainee@centos8 ~]$ ansible-playbook -i inventory playbook.yaml -l equilibrage

PLAY [basedonnees]
*****
*****
skipping: no hosts matched

PLAY [dotcms]
*****

```

```

*****
skipping: no hosts matched

PLAY [equilibrage]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [web04]

TASK [install]
*****
*****
ok: [web04]

TASK [service]
*****
*****
ok: [web04]

PLAY RECAP
*****
*****
web04          : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

```

Bien évidemment il est aussi possible de limiter à une seule machine :

```

[trainee@centos8 ~]$ ansible-playbook -i inventory playbook.yaml -l web04

PLAY [basedonnees]
*****

```

```
*****
skipping: no hosts matched

PLAY [dotcms]
*****
skipping: no hosts matched

PLAY [equilibrage]
*****

TASK [Gathering Facts]
*****
ok: [web04]

TASK [install]
*****
ok: [web04]

TASK [service]
*****
ok: [web04]

PLAY RECAP
*****
web04          : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Notez que dans le fichier **playbook.yaml** se trouvent deux lignes dont le mot clef est **tags:**. Cette étiquette permet de limiter l'action de la commande

ansible-playbook aux seuls services :

```
...
- name: service
  service: name=postgresql state=started enabled=yes
  tags: service
...
- name: service
  service: name=haproxy state=started enabled=yes
  tags: service
```

Exécutez de nouveau la commande **ansible-playbook** en limitant l'action aux seuls services :

```
[trainee@centos8 ~]$ ansible-playbook -i inventory playbook.yaml --tags service
```

```
PLAY [basededonnees]
```

```
*****
*****
```

```
TASK [Gathering Facts]
```

```
*****
*****
```

```
ok: [web01]
```

```
TASK [service]
```

```
*****
*****
```

```
ok: [web01]
```

```
PLAY [dotcms]
```

```
*****
*****
```

```
TASK [Gathering Facts]
```

```

*****
*****
ok: [web02]
ok: [web03]

PLAY [equilibrage]
*****
*****

TASK [Gathering Facts]
*****
*****

ok: [web04]

TASK [service]
*****
*****

ok: [web04]

PLAY RECAP
*****
*****
web01          : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web02          : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web03          : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
web04          : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

```

Il est aussi possible de combiner les options **-tags** et **-l** :

```
[trainee@centos8 ~]$ ansible-playbook -i inventory playbook.yaml --tags service -l equilibrage
```

```
PLAY [basededonnees]
*****
*****
skipping: no hosts matched

PLAY [dotcms]
*****
*****
skipping: no hosts matched

PLAY [equilibrage]
*****
*****

TASK [Gathering Facts]
*****
*****
ok: [web04]

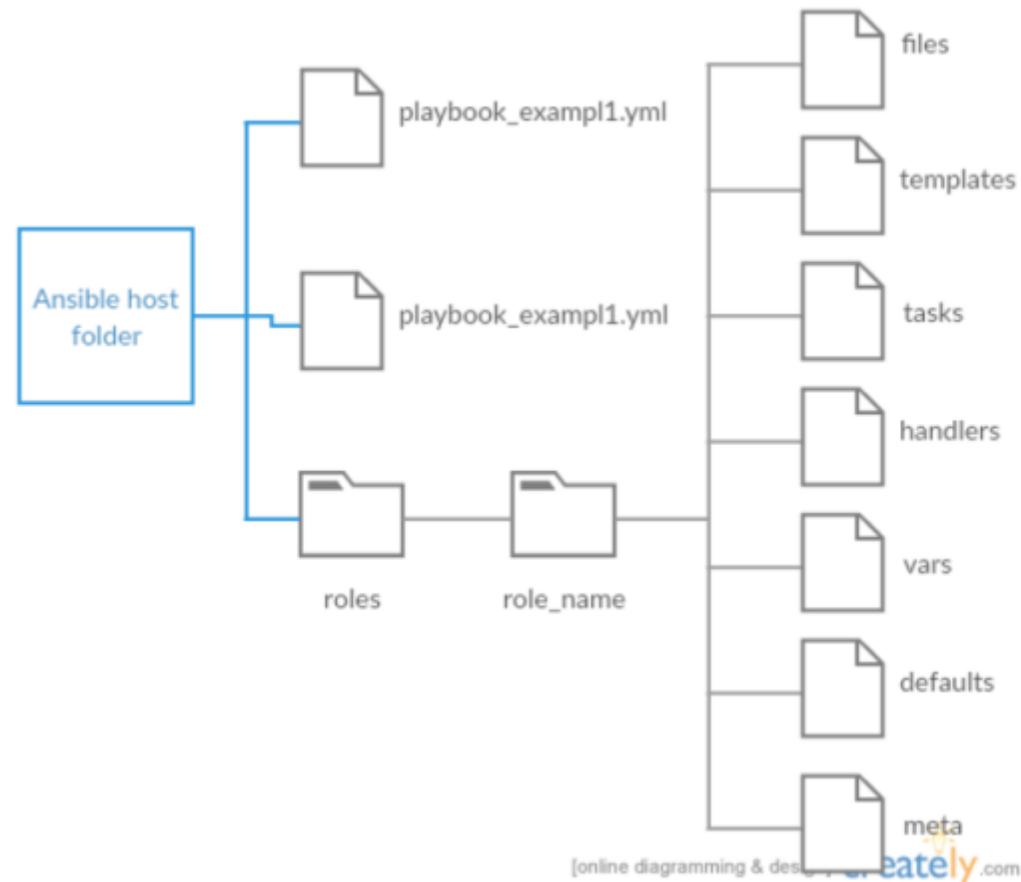
TASK [service]
*****
*****
ok: [web04]

PLAY RECAP
*****
*****
web04          : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

LAB #3 - La Commande ansible-galaxy

3.1 - Utiliser des Rôles avec un Play Book

Les **Rôles** permettent la création d'une bibliothèque réutilisable.



Un **Rôle** est un répertoire contenant les sous-répertoires suivants :

- **defaults**
 - contient des variables par défaut de basses priorités qui peuvent être sur-chargées par des variables ailleurs,

- **files**,
 - contient des fichiers à déployer aux systèmes distants,
- **handlers**,
 - contient les handlers,
- **meta**,
 - contient entre autre des informations sur des dépendances entre les **Rôles**,
- **tasks**,
 - contient les **actions** à effectuer,
- **templates**,
 - contient des fichiers qui peuvent être customisé avec des variables avant d'être déployés,
- **var**,
 - contient des variables de configuration de haute priorité, difficilement sur-chargées par des variables ailleurs.



Important : Notez que seul le répertoire **tasks** est obligatoire.

Notez que le module **copy**: ne référence un fichier que par son nom et son éventuelle extension :

```
...  
copy:  
  src: myfile.conf  
...
```

Ansible recherche ce fichier dans plusieurs répertoires dans l'ordre qui suit :

- le répertoire **files** du Rôle,
- le répertoire **tasks** du Rôle,
- le répertoire du Rôle lui-même,
- le répertoire **files** du répertoire dans lequel se trouve le fichier `playbook.yaml`,
- le répertoire dans lequel se trouve le fichier `playbook.yaml`.

Par exemple :

```
.
├── 5-myfile.conf
├── files
│   └── 4-myfile.conf
├── playbook.yaml
├── roles
│   └── dotcms
│       ├── 3-myfile.conf
│       ├── defaults
│       │   └── main.yaml
│       ├── files
│       │   └── 1-myfile.conf
│       ├── handlers
│       │   └── main.yaml
│       ├── tasks
│       │   ├── 2-myfile.conf
│       │   └── main.yaml
│       └── templates
│           ├── context.xml
│           └── dotcms.service
```

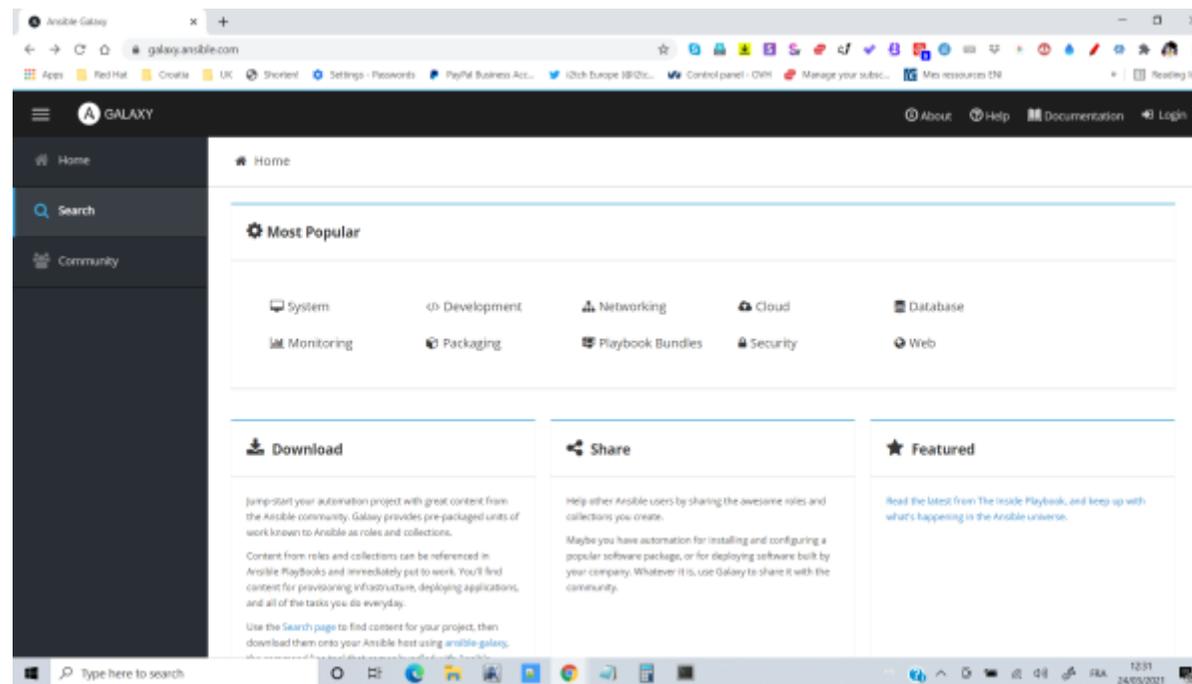
Dans le cas de l'utilisation de **Rôles**, le fichier `playbook.yaml` devient plus simplifié. Par exemple :

```
- hosts: basededonnees
  become: true
  roles:
    - basededonnees
- hosts: dotcms
  become: true
  roles:
    - java
    - dotcms
- hosts: equilibrage
```

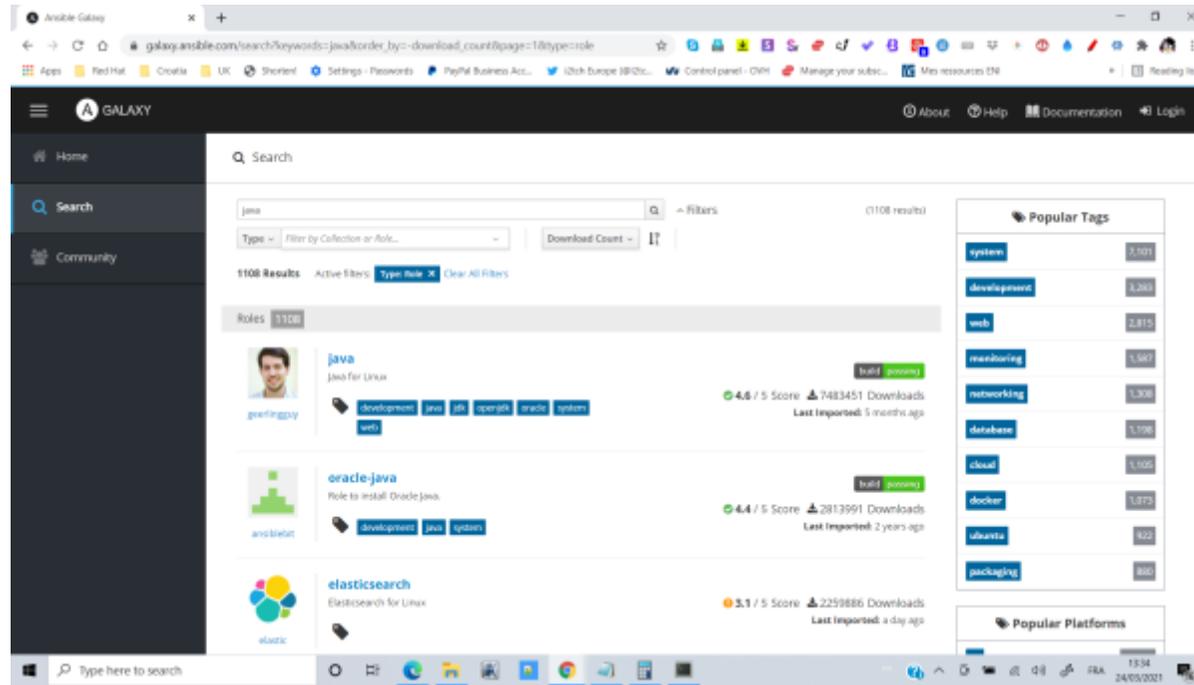
```
become: true
roles:
  - equilibrage
```

3.2 - Ansible Galaxy

Ansible Galaxy est un dépôt de Rôles créés par des tierces personnes. Ce dépôt peut être consulté à l'adresse <https://galaxy.ansible.com/> :



Recherchez le mot clef **java** avec un premier filtre de type **Rôle** et un deuxième filtre de type **Download Count** :



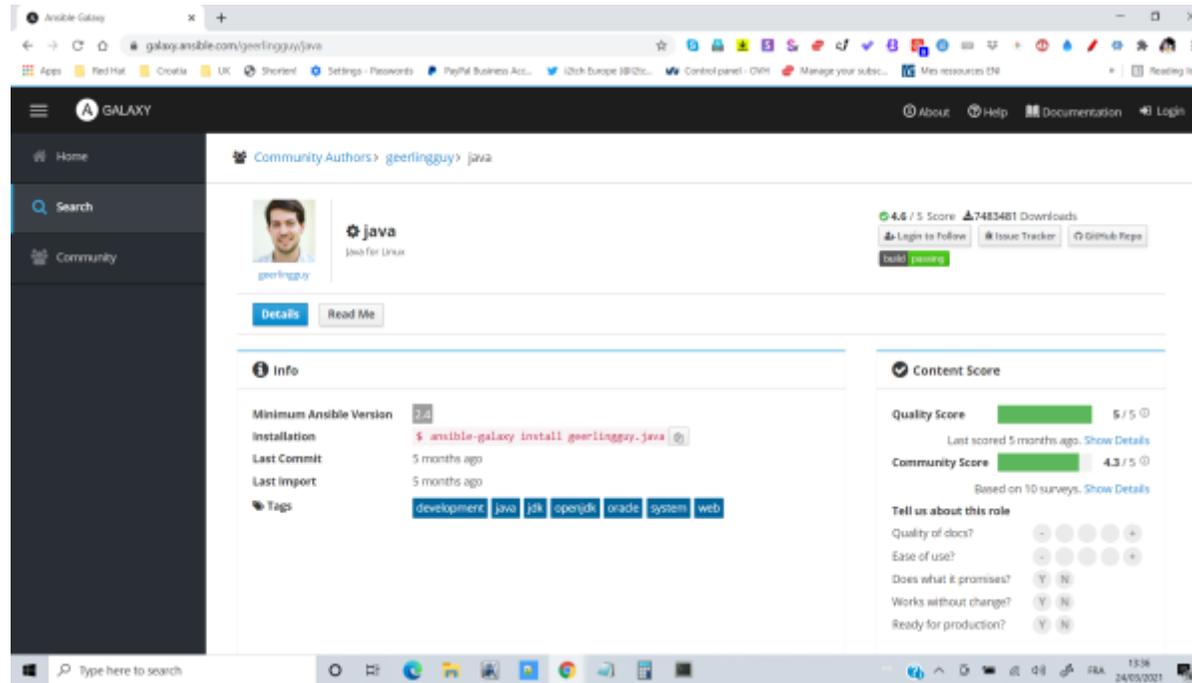
The screenshot shows the Ansible Galaxy search interface. The search bar contains the keyword 'java', resulting in 1108 results. The top three results are:

- java** by [peeringgguy](#): A role for Linux with a 4.6 / 5 score, 7483451 downloads, and last imported 5 months ago. It is tagged with 'development', 'java', 'jdk', 'openjdk', 'oracle', 'system', and 'web'.
- oracle-java** by [ansiblebot](#): A role to install Oracle Java with a 4.4 / 5 score, 2813991 downloads, and last imported 2 years ago. It is tagged with 'development', 'java', and 'system'.
- elasticsearch** by [elastic](#): A role for Elasticsearch for Linux with a 3.1 / 5 score, 2259886 downloads, and last imported a day ago.

On the right side, there is a 'Popular Tags' section with the following data:

Tag	Count
system	3,301
development	3,263
web	2,815
monitoring	1,907
networking	1,308
database	1,196
cloud	1,105
docker	1,073
elasticsearch	622
packaging	880

Consultez la première dans la liste :



The screenshot shows the Ansible Galaxy interface for the 'java' role. The page includes a navigation sidebar with 'Home', 'Search', and 'Community'. The main content area displays the role's details, including the author's profile (geerlingguy), a 4.6/5 score, 7483481 downloads, and a 'build passing' status. The 'Info' section lists the minimum Ansible version (2.4), installation instructions (ansible-galaxy install geerlingguy.java), and last commit/import dates (5 months ago). The 'Content Score' section shows a Quality Score of 5/5 and a Community Score of 4.3/5, based on 10 surveys. A feedback form is also visible at the bottom right.

Community Authors > geerlingguy > java

java
java for Linux

4.6 / 5 Score 7483481 Downloads

Login to Follow Issue Tracker GitHub Page

build passing

Details Read Me

Info

Minimum Ansible Version 2.4

Installation `ansible-galaxy install geerlingguy.java`

Last Commit 5 months ago

Last Import 5 months ago

Tags `development` `java` `jdk` `openjdk` `oracle` `system` `web`

Content Score

Quality Score 5 / 5
Last scored 5 months ago. [Show Details](#)

Community Score 4.3 / 5
Based on 10 surveys. [Show Details](#)

Tell us about this role

Quality of docs?

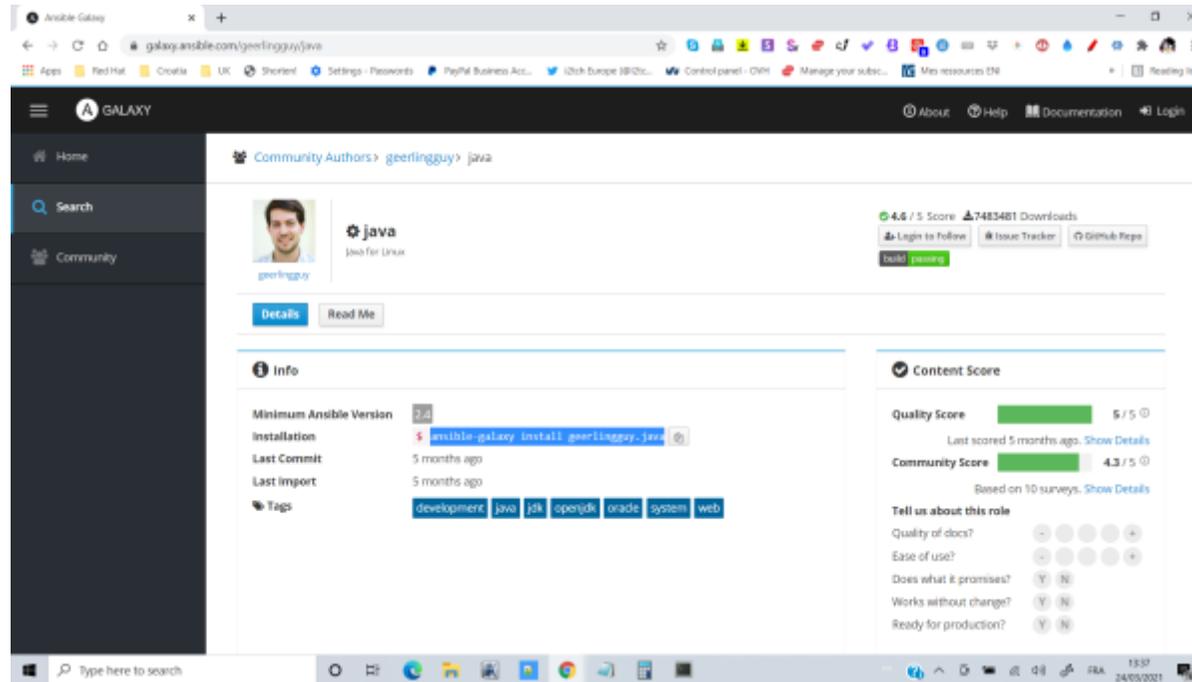
Ease of use?

Does what it promises? Y N

Works without change? Y N

Ready for production? Y N

Copiez la commande d'installation :



Important : Notez que la convention utilisée pour les noms est *nom_utilisateur.rôle*.

Dans la machine **centos8**, collez la commande copiée et exécutez-la :

```
[trainee@centos8 ~]$ ansible-galaxy install geerlingguy.java
Starting galaxy role install process
- downloading role 'java', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-java/archive/2.0.1.tar.gz
- extracting geerlingguy.java to /home/trainee/.ansible/roles/geerlingguy.java
- geerlingguy.java (2.0.1) was installed successfully
```

Installez le paquet **tree** et consultez le répertoire **/home/trainee/.ansible/roles/geerlingguy.java** :

```
[trainee@centos8 ~]$ su -
```

```
Password:
[root@centos8 ~]# which tree
/usr/bin/tree
[root@centos8 ~]# exit
logout
[trainee@centos8 ~]$ tree /home/trainee/.ansible/roles/geerlingguy.java
/home/trainee/.ansible/roles/geerlingguy.java
├── defaults
│   └── main.yml
├── LICENSE
├── meta
│   └── main.yml
├── molecule
│   └── default
│       ├── converge.yml
│       └── molecule.yml
├── README.md
├── tasks
│   ├── main.yml
│   ├── setup-Debian.yml
│   ├── setup-FreeBSD.yml
│   └── setup-RedHat.yml
├── templates
│   └── java_home.sh.j2
└── vars
    ├── Debian-10.yml
    ├── Debian-11.yml
    ├── Debian-8.yml
    ├── Debian-9.yml
    ├── Fedora.yml
    ├── FreeBSD.yml
    ├── RedHat-7.yml
    ├── RedHat-8.yml
    └── Ubuntu-12.yml
```

```
├── Ubuntu-14.yml
├── Ubuntu-16.yml
├── Ubuntu-18.yml
└── Ubuntu-20.yml
```

7 directories, 24 files



Important : Notez que dans ce cas il n'y a pas eu de dépendances. Par contre la commande **ansible-galaxy** sait résoudre et gérer les dépendances.

Créez maintenant un fichier Play Book pour utiliser ce Rôle :

```
[trainee@centos8 ~]$ vi playbook_java.yaml
[trainee@centos8 ~]$ cat playbook_java.yaml
---
- hosts: all
  become: true
  roles:
    - geerlingguy.java
```

Avant d'exécuter ce playbook, créez le fichier **~/.ansible.cfg** :

```
[trainee@centos8 ~]$ vi .ansible.cfg
[trainee@centos8 ~]$ cat .ansible.cfg
[defaults]
deprecation_warnings=False
log_path = ./ansible.log
```



Important : Le comportement d'Ansible peut être configuré par des directives dans la variable d'environnement **ANSIBLE_CONFIG** ou dans le fichier **ansible.cfg** dans le répertoire courant de travail ou dans le fichier **.ansible.cfg** dans le répertoire personnel de l'utilisateur



ou dans le fichier **/etc/ansible/ansible.cfg**, selon le premier qui est trouvé. Pour une liste complète des directives, consultez la page https://docs.ansible.com/ansible/latest/reference_appendices/config.html.

Pour consultez l'ensemble des variables, exécutez la commande suivante :

```
[trainee@centos8 ~]$ ansible-config list | more
ACTION_WARNINGS:
  default: true
  description: [By default Ansible will issue a warning when received from a task
    action (module or action plugin), These warnings can be silenced by adjusting
    this setting to False.]
  env:
  - {name: ANSIBLE_ACTION_WARNINGS}
  ini:
  - {key: action_warnings, section: defaults}
  name: Toggle action warnings
  type: boolean
  version_added: '2.5'
AGNOSTIC_BECOME_PROMPT:
  default: true
  description: Display an agnostic become prompt instead of displaying a prompt containing
    the command line supplied become method
  env:
  - {name: ANSIBLE_AGNOSTIC_BECOME_PROMPT}
  ini:
  - {key: agnostic_become_prompt, section: privilege_escalation}
  name: Display an agnostic become prompt
  type: boolean
  version_added: '2.5'
  yaml: {key: privilege_escalation.agnostic_become_prompt}
ALLOW_WORLD_READABLE_TMPFILES:
  default: false
  deprecated: {alternatives: world_readable_tmp, collection_name: ansible.builtin,
```

```
version: '2.14', why: moved to shell plugins}
description: ['This setting has been moved to the individual shell plugins as a
  plugin option :ref:`shell_plugins`.', 'The existing configuration settings are
  still accepted with the shell plugin adding additional options, like variables.',
  This message will be removed in 2.14.]
name: Allow world-readable temporary files
type: boolean
ANSIBLE_CONNECTION_PATH:
  default: null
  description: [Specify where to look for the ansible-connection script. This location
    will be checked before searching $PATH., 'If null, ansible will start with the
    same directory as the ansible script.']
  env:
  - {name: ANSIBLE_CONNECTION_PATH}
  ini:
  - {key: ansible_connection_path, section: persistent_connection}
  name: Path of ansible-connection script
  type: path
  version_added: '2.8'
  yaml: {key: persistent_connection.ansible_connection_path}
ANSIBLE_COW_ACCEPTLIST:
  default: [bud-frogs, bunny, cheese, daemon, default, dragon, elephant-in-snake,
    elephant, eyes, hellokitty, kitty, luke-koala, meow, milk, moofasa, moose, ren,
    sheep, small, stegosaurus, stimp, supermilker, three-eyes, turkey, turtle, tux,
    udder, vader-koala, vader, www]
  description: White list of cowsay templates that are 'safe' to use, set to empty
    list if you want to enable all installed templates.
  env:
  - deprecated: {alternatives: ANSIBLE_COW_ACCEPTLIST, collection_name: ansible.builtin,
  --More--
```

Exécutez maintenant la commande **ansible-playbook** :

```
[trainee@centos8 ~]$ ansible-playbook -i inventory playbook_java.yaml
```

PLAY [all]

TASK [Gathering Facts]

ok: [web04]
ok: [web01]
ok: [web03]
ok: [web02]

TASK [geerlingguy.java : Include OS-specific variables for Fedora or FreeBSD.]

skipping: [web01]
skipping: [web02]
skipping: [web03]
skipping: [web04]

TASK [geerlingguy.java : Include version-specific variables for CentOS/RHEL.]

skipping: [web01]
skipping: [web02]
skipping: [web03]
skipping: [web04]

TASK [geerlingguy.java : Include version-specific variables for Ubuntu.]

skipping: [web01]
skipping: [web02]
skipping: [web03]

skipping: [web04]

TASK [geerlingguy.java : Include version-specific variables for Debian.]

ok: [web02]
ok: [web03]
ok: [web01]
ok: [web04]

TASK [geerlingguy.java : Define java_packages.]

ok: [web01]
ok: [web02]
ok: [web03]
ok: [web04]

TASK [geerlingguy.java : include_tasks]

skipping: [web01]
skipping: [web02]
skipping: [web03]
skipping: [web04]

TASK [geerlingguy.java : include_tasks]

included: /home/trainee/.ansible/roles/geerlingguy.java/tasks/setup-Debian.yml for web01, web02, web03, web04

TASK [geerlingguy.java : Ensure 'man' directory exists.]


```
skipping: [web01]
skipping: [web02]
skipping: [web03]
skipping: [web04]
```

TASK [geerlingguy.java : Ensure Java is installed.]

```
*****
*****
```

```
changed: [web04]
changed: [web03]
changed: [web02]
changed: [web01]
```

TASK [geerlingguy.java : include_tasks]

```
*****
*****
```

```
skipping: [web01]
skipping: [web02]
skipping: [web03]
skipping: [web04]
```

TASK [geerlingguy.java : Set JAVA_HOME if configured.]

```
*****
*****
```

```
skipping: [web01]
skipping: [web02]
skipping: [web03]
skipping: [web04]
```

PLAY RECAP

```
*****
*****
```

```
web01          : ok=5    changed=1    unreachable=0    failed=0    skipped=7    rescued=0
ignored=0
```

```
web02          : ok=5    changed=1    unreachable=0    failed=0    skipped=7    rescued=0
ignored=0
web03          : ok=5    changed=1    unreachable=0    failed=0    skipped=7    rescued=0
ignored=0
web04          : ok=5    changed=1    unreachable=0    failed=0    skipped=7    rescued=0
ignored=0
```

Notez qu'Ansible a pu localiser le Rôle grâce à la valeur de la variable Ansible **ANSIBLE_ROLES_PATH**. La valeur par défaut de cette variable est **~/.ansible/roles:/usr/share/ansible/roles:/etc/ansible/roles** :

```
[trainee@centos8 ~]$ ansible-config dump |grep ROLES_PATH
DEFAULT_ROLES_PATH(default) = ['/home/trainee/.ansible/roles', '/usr/share/ansible/roles', '/etc/ansible/roles']
```

Le PLAY RECAP contient 7 catégories :

- **ok**
 - le nombre de tâches qui ont été traitées sans erreur
- **changed**
 - le nombre de tâches qui ont effectué des modifications sur un des hôtes concernés
- **unreachable**
 - le nombre de tâches qui n'ont pas été exécutées parce que l'hôte ne pouvait pas être contacté
- **failed**
 - le nombre de tâches en erreur pour lesquelles la valeur de **ignore_errors** était fixée à **false**
- **skipped**
 - le nombre de tâches qui n'ont pas été exécutées parce que la valeur de la clause **when** était **fausse**
- **rescued**
 - le nombre de tâches de type **rescue** qui ont été exécutées suite à une erreur ailleurs dans le playbook
- **ignored**
 - le nombre de tâches en erreur pour lesquelles la valeur de **ignore_errors** était fixée à **true**

Pour mieux expliquer la catégorie **rescued**, considérez le code suivant :

```
---
- name: Ansible Blocks
```

```
hosts: server1
gather_facts: false

tasks:
  - block:
    - name: List home directory content
      command: "ls -l ~/"

    - name: Failing intentionally
      command: "ls -l /tmp/does-not-exist"

  rescue:
    - name: Rescue block (perform recovery)
      debug:
        msg: "Something went wrong, cleaning up.."

  always:
    - name: This will execute always
      debug:
        msg: "I will execute even in failure scenario"
```

Dans le cas ci-dessus, la deuxième tâche dans la section **block** va générer une erreur. Dans ce cas la tâche dans la section **rescue** sera exécutée. Dans **tous** les cas la tâche dans la section **always** sera exécutée.