



# **Ansible pour professionnels**

## **Linux / Unix**



Le support du cours «Ansible pour professionnel Linux/Unix » est non contractuel ; il ne doit pas être redistribué et/ou reproduit en partie ou en totalité sans permission explicite et écrite de la société Adlere.

Red Hat, le logo Red Hat, OpenShift et Ansible sont des marques déposées ou commerciales de Red Hat, Inc ou ses filiales aux États-Unis et dans d'autre pays. Linux® est une marque déposée de Linus Torvalds aux États-Unis et dans d'autre pays.

UNIX ® est une marque déposée par « The Open Group » aux Etats-Unis et dans d'autres pays.  
Windows® est une marque déposée de Microsoft Corporation aux États-Unis et dans d'autre pays.

Les autres marques citées sont déposées par leurs propriétaires respectifs.

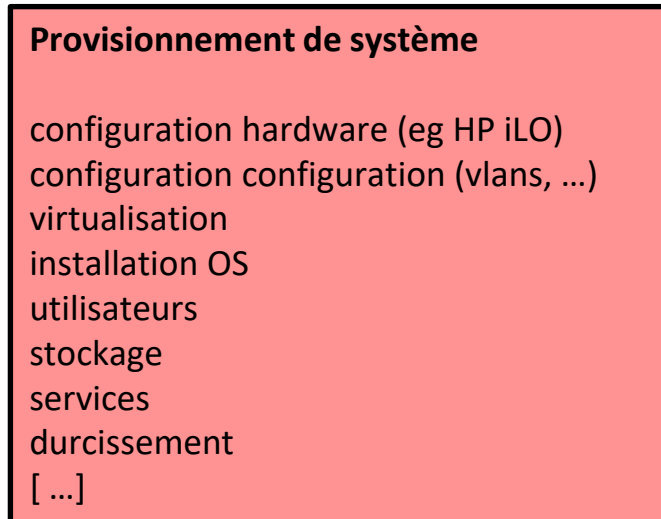


**+adlere**  
DIGITAL EXPERTISE

**Rôles**



# Des rôles, pour quoi faire ?



ROLES



Configuration

Provisionnement

Installation d'applications

Durcissement

Remédiation

- Identifier les unités logiques, les rendre ré-utilisables et agnostiques
- pas de secrets ou de données spécifiques à un sites (partageables)

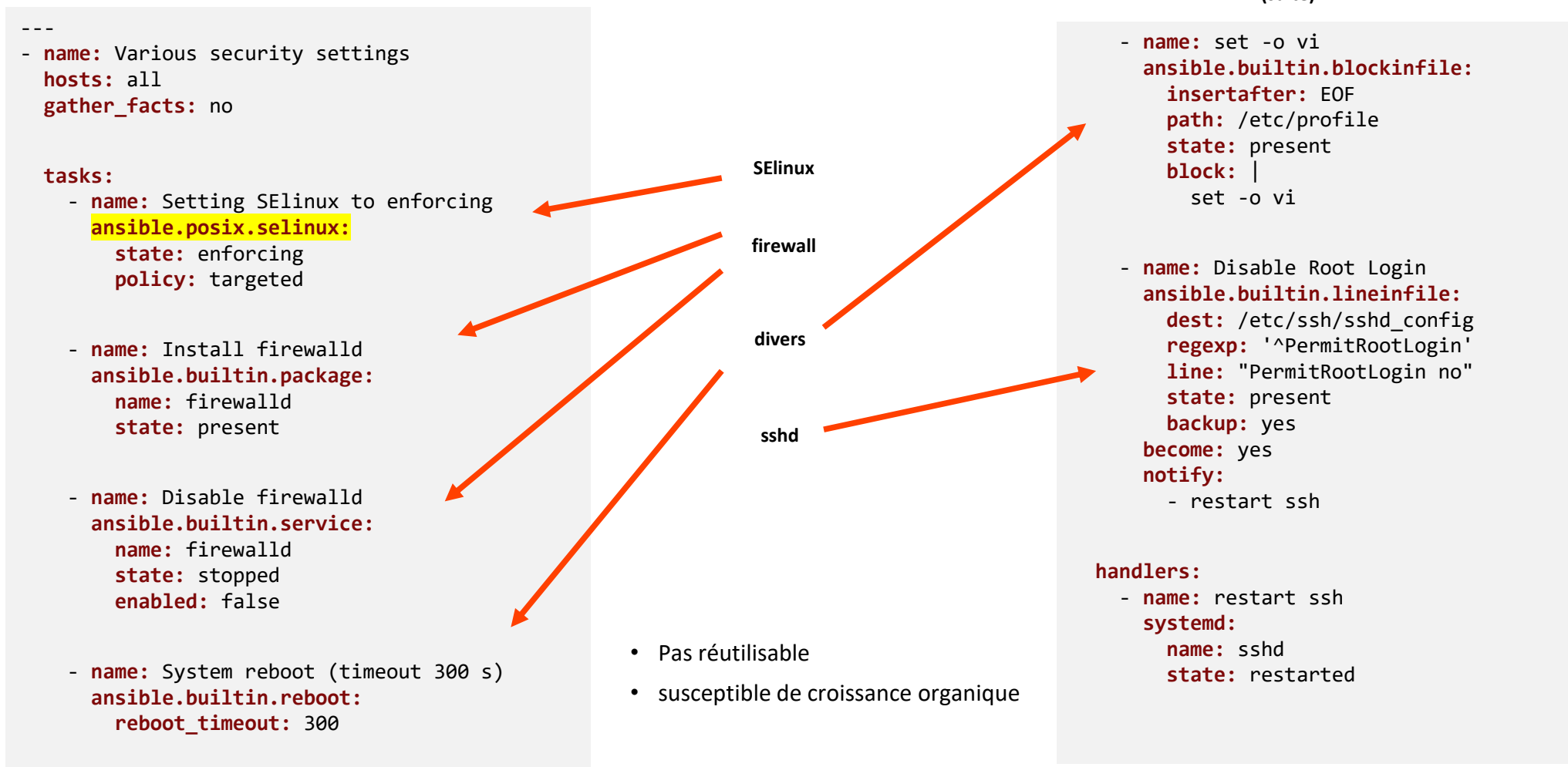




# Playbook d'origine : sec\_settings.yml

5

- Playbook utilisé dans le provisionnement d'une VM; concerne la sécurité en général





# Évolution #1 : rôles embarqués dans le projet

6

- On embarque un répertoire ./roles dans le projet

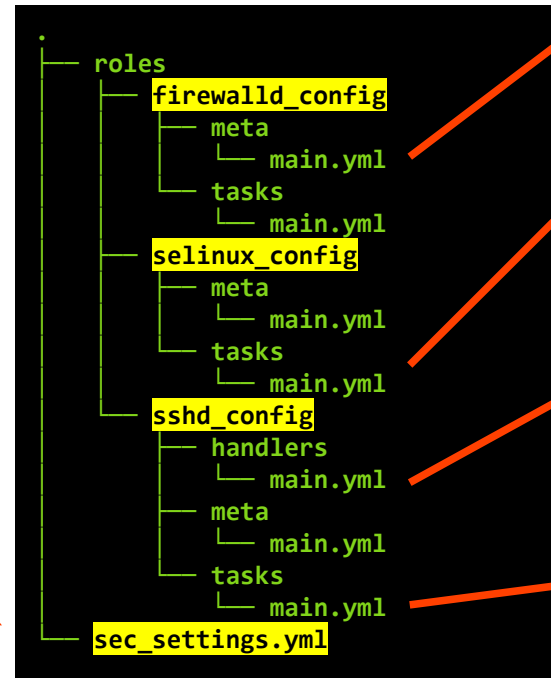
```
---
- name: Various security settings
  hosts: all
  gather_facts: no

  roles:
    - firewallld_config
    - selinux_config
    - sshd_config

  tasks:
    - name: System reboot (timeout 300 s)
      ansible.builtin.reboot:
        reboot_timeout: 300

    - name: set -o vi
      blockinfile:
        insertafter: EOF
        path: /etc/profile
        state: present
        block: |
          set -o vi
```

playbook  
principal



```
- name: Disable firewallld
  ansible.builtin.service:
    name: firewallld
    state: stopped
    enabled: false
```

```
- name: Setting SELinux to enforcing
  ansible.posix.selinux:
    state: enforcing
    policy: targeted
```

```
- name: restart ssh
  ansible.builtin.systemd:
    name: sshd
    state: restarted
```

```
- name: SSHD ROLE | Enable Root Login
  ansible.builtin.lineinfile:
    dest: /etc/ssh/sshd_config
    regexp: '^PermitRootLogin'
    line: "PermitRootLogin no"
    state: present
    backup: yes
  become: yes
  notify:
    - restart ssh
```

- on aurait pu mettre chaque rôle directement dans le répertoire du projet
- les rôles peuvent croître indépendamment l'un de l'autre et s'enrichir
- meilleure organisation, plus lisible



# Évolution #2 : rôles partagés et réutilisables

7

- Chaque rôle est (dé)placé dans un repository git spécifique

```
.
├── collections.yml
├── README.md
├── roles_requirements.yml
└── sec_settings.yml
```

```
$ cat collections.yml
collections:
  - ansible.posix
```

(ansible.posix nécessité par le rôle selinux )

```
---
- name: Various security settings
  hosts: all
  gather_facts: no

  roles:
    - firewallld_config
    - selinux_config
    - sshd_config

  tasks:
    - name: System reboot (timeout 300 s)
      ansible.builtin.reboot:
        reboot_timeout: 300

    - name: set -o vi
      blockinfile:
        insertafter: EOF
        path: /etc/profile
        state: present
        block: |
          set -o vi
```

```
$ cat roles_requirements.yml
---
- src: https://gitlab.com/myrepo/roles/firewalld_config
  scm: git
  name: firewallld_config

- src: https://gitlab.com/myrepo/roles/selinux_config
  scm: git
  name: selinux_config

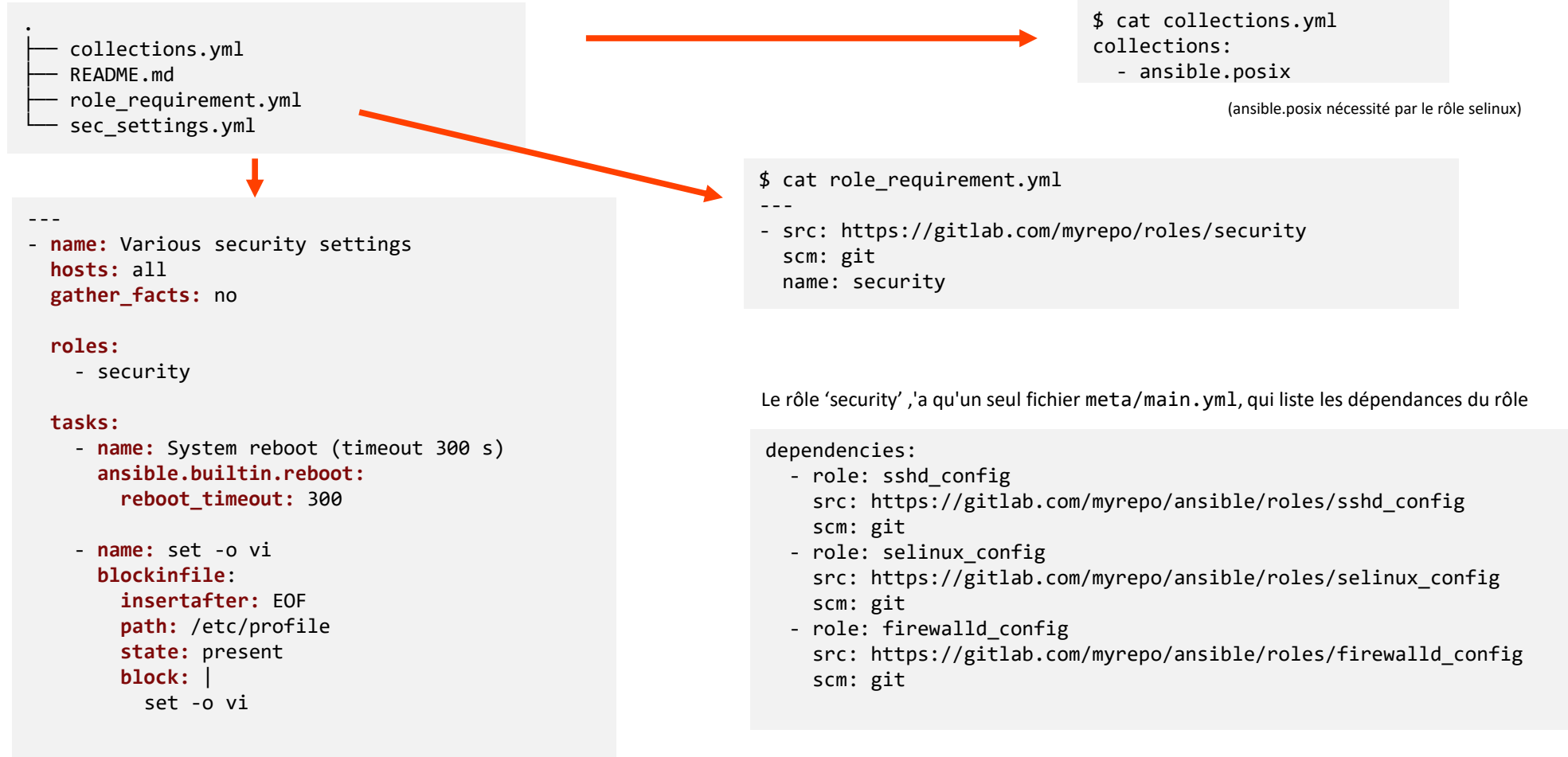
- src: https://gitlab.com/myrepo/roles/sshd_config
  scm: git
  name: sshd_config
```

- rôles et collections sont décrits dans des fichiers spécifiques
- on les installe avec

```
ansible-galaxy collection install -r collections.yml
ansible-galaxy role install -r roles_requirements.yml
```
- Un repository par rôle
- Partageable, réutilisable, bonne lisibilité, peuvent croître indépendamment

**le playbook ne change pas**

# +a Évolution #3 : appel d'un "meta"-rôle et de ses dépendances

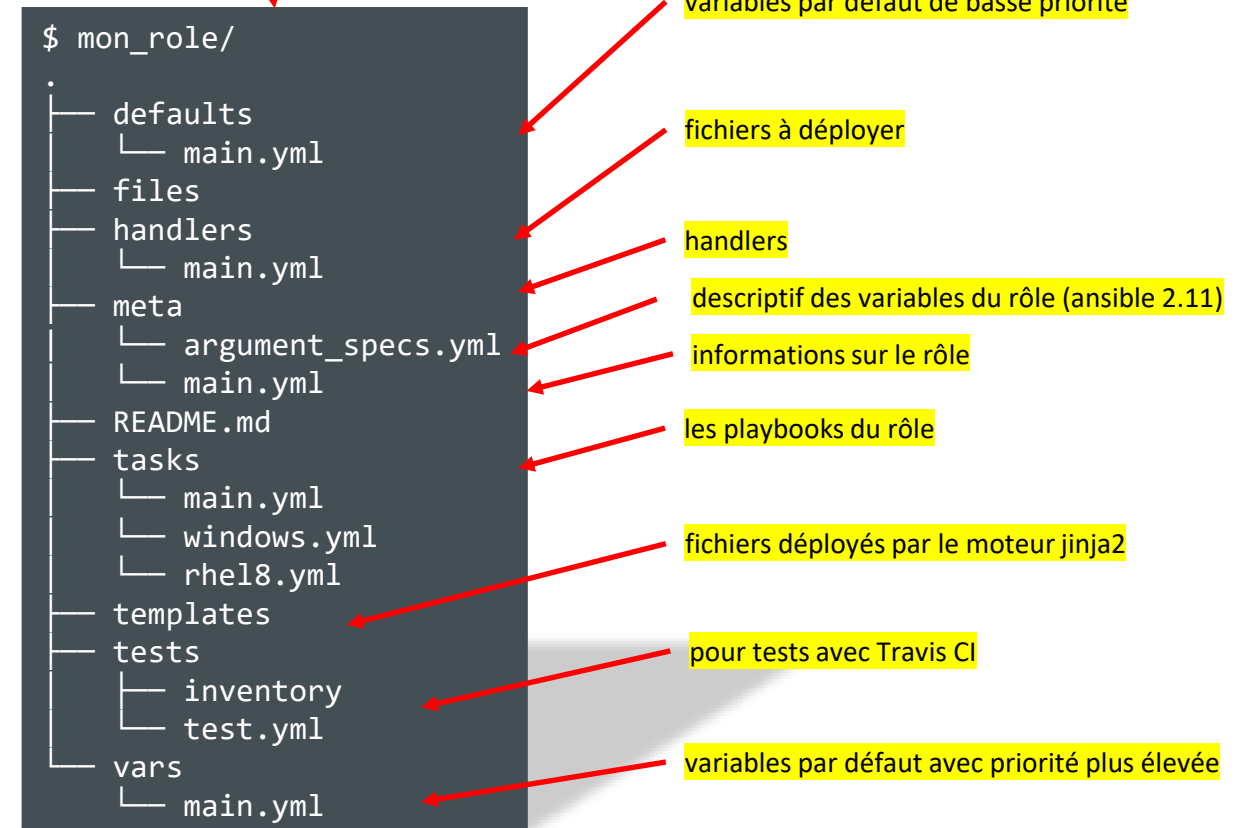


le playbook référence juste un rôle



- Ensemble de playbooks et leurs ressources
- code générique, ré-utilisable, distribuable
- apportent structure et organisation logique aux playbooks
- nécessitent de la réflexion
- appelés à partir d'un playbook
- doivent respecter une structure définie
- **ansible-galaxy [role] init** initialise la structure
  - adapter README.md et meta/main.yml
    - cf. les licences
  - enlever les objets (répertoires) non utilisés

Le nom du répertoire est important



Documenter le rôle : **meta/main.yml** et **README.md**

[https://docs.ansible.com/ansible/latest/playbook\\_guide/playbooks\\_reuse\\_roles.html](https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_reuse_roles.html)





- les rôles peuvent aller dans un sous-répertoire 'roles' du projet
- ou directement dans le projet
- ce sont des bouts de code réutilisables et partageables
- ils peuvent exister dans d'autres emplacements (chemins spécifiques à définir)

```
mon_projet/
├── ansible.cfg
├── inventory/
├── group_vars/
├── host_vars/
├── roles/
│   ├── common/
│   │   ├── tasks/
│   │   │   └── main.yml
│   │   ├── handlers/
│   │   ├── templates/
│   │   ├── files/
│   │   ├── vars/
│   │   └── defaults/
│   └── webserver/
│       └── [...]
├── playbooks/
├── files/
├── templates/
└── README.md
```

# Fichier de configuration  
# Répertoire de fichiers d'inventaire  
# Répertoires des variables de groupe  
# Répertoire de variables par système  
# Répertoire des rôles  
# Rôle 'common'  
# Playbooks du rôle  
  
# Handlers  
# Fichier template Jinja  
# Fichiers à recopier  
# Variables; internes du rôle  
# Variables par défaut du rôle  
# Rôle 'webserver'  
  
# Répertoire de playbooks  
# Fichiers à recopier  
# Fichiers pour template Jinja2  
# Documentation

- décrit le rôle et ses caractéristiques à travers des mots-clefs
- les tags et autres caractéristiques (platform, documentation, license, ...) catégorisent le rôle sur galaxy.ansible.com ou dans un Private Automation Hub
- définit les dépendances du rôle

Dans meta/main.yml :  Dépendances  Dans meta/requirements.yml :

```
[...]
dependencies:
  - geerlingguy.haproxy
  - name: geerlingguy.ansible
    version: 2.0.2
[...]
```

```
---
- geerlingguy.haproxy
- name: geerlingguy.ansible
  version: 2.0.2
```



# Installation (avec les dépendances)

- Depuis galaxy.ansible.com :

```
ansible-galaxy install nom.role
```

```
$ ansible-galaxy install role1.tar.gz
Starting galaxy role install process
- extracting role1.tar.gz to /home/admijkl/.ansible/roles/role1.tar.gz
- role1.tar.gz was installed successfully
- adding dependency: geerlingguy.haproxy
- adding dependency: geerlingguy.ansible (2.0.2)
- downloading role 'haproxy', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-haproxy/archive/1.3.0.tar.gz
- extracting geerlingguy.haproxy to /home/admijkl/.ansible/roles/geerlingguy.haproxy
- geerlingguy.haproxy (1.3.0) was installed successfully
- downloading role 'ansible', owned by geerlingguy
- downloading role from https://github.com/geerlingguy/ansible-role-ansible/archive/2.0.2.tar.gz
- extracting geerlingguy.ansible to /home/admijkl/.ansible/roles/geerlingguy.ansible
- geerlingguy.ansible (2.0.2) was installed successfully
$
```

- Depuis un repository Git :

```
ansible-galaxy role install git+https://github.com/geerlingguy/ansible-role-certbot
```



# Installation avec un fichier de prérequis

- S'installe avec : `ansible-galaxy role install -r requirements.yml`

```
---
roles:
  # Installe la dernière version disponible depuis Ansible Galaxy
  - name: geerlingguy.firewall

  # Installe une version spécifique, depuis Ansible Galaxy
  - name: geerlingguy.php
    version: 4.3.1

  # Installe une version spécifique depuis GitHub
  - src: https://github.com/geerlingguy/ansible-role-passenger
    name: passenger
    version: 2.0.0

  # Installe une archive depuis un serveur web
  - src: https://www.example.com/ansible/roles/my-role-name.tar.gz
    name: my-role
```



# Emplacements de recherche par défaut

14

1. Dans les collections
2. dans `./roles/`, relativement au fichier du playbook

```
$ tree -L 3
.
├── ansible.cfg
├── inventory
├── main_import.yml
├── main_include.yml
├── main_mix.yml
├── main_traditional.yml
├── roles
│   └── role1
│       ├── defaults
│       ├── files
│       ├── handlers
│       ├── meta
│       ├── README.md
│       ├── tasks
│       ├── templates
│       ├── tests
│       └── vars
```

3. dans la directive `roles_path`, par défaut :  
    `~/.ansible/roles`  
    `/usr/share/ansible/roles`  
    `/etc/ansible/roles`

Sinon personnalisable dans `ansible.cfg` :

```
[defaults]
roles_path = /path/to/roles
```

(ou dans la variable `ANSIBLE_ROLES_PATH`)

4. dans le répertoire du playbook



# +a Appeler des rôles depuis un playbook

```
---
- name: Mon playbook
  hosts: all
  gather_facts: no

  roles:
    - role1
    - role2
    - role: '/chemin/vers/role3'

  tasks:
    - name: Suite des tâches
```

Avec la directive "roles"

```
---
- name: Mon playbook
  hosts: all
  gather_facts: no

  tasks:
    - name: Import du role1
      ansible.builtin.import_role:
        name: role1
      vars:
        myvar: 'valeur'

    - name: Import du role2
      ansible.builtin.import_role:
        name: role2
```

Import (= statique)

```
---
- name: Mon playbook
  hosts: all
  gather_facts: no

  tasks:
    - name: Include du role1
      ansible.builtin.include_role:
        name: role1

    - name: Include du role2
      ansible.builtin.include_role:
        name: role2
```

Include (= dynamique)



- `ansible-playbook -vv` affiche l'emplacement du rôle exécuté

```
PLAYBOOK: main_traditional.yml *****
1 plays in ./main_traditional.yml

PLAY [Main playbook] *****

TASK [role1 : Play 1, task 1] *****
task path: /etc/ansible/roles/role1/tasks/main.yml:2
ok: [localhost] => {
    "msg": "This is play1, task1"
}

TASK [role2 : Play2, task 1] *****
task path: /etc/ansible/roles/role2/tasks/main.yml:2
ok: [localhost] => {
    "msg": "This is play2, task 1"
}
```

- Recommandé de nommer de façon explicite les tâches d'un rôle pour faciliter la compréhension à l'exécution

```
---
- name: APACHE RHEL | Installation Apache
  ansible.builtin.apache:
    name:
      - apache2
```

Objet	Commande
Création de la structure de répertoire	<code>ansible-galaxy [role] init ROLE_NAME</code>
Créera le sous-répertoire <code>ROLE_NAME</code> avec les fichiers nécessaires. Recommandé : enlever le superflu	
Préparer un rôle pour distribution depuis le répertoire parent, le même où on a fait la commande d'init	<code>tar czvf ./ROLE_NAME.tar.gz ./ROLE_NAME</code>
Installer un rôle	<code>ansible-galaxy install ./ROLE_NAME.tar.gz</code>
Par défaut, va dans <code>~/.ansible/roles</code>	<code>ansible-galaxy install ROLE_NAME</code>  <code>[-p destination_path]</code> <code>[--server https://my.server.example.com]</code>

- guide utilisateur ansible-galaxy : <https://docs.ansible.com/ansible/latest/cli/ansible-galaxy.html>
- Documenter les rôles dans **meta/main.yml** et **README.md**



ansible-galaxy role	Description
init	<code>ansible-galaxy role init MON_ROLE</code>
remove	efface un rôle en local
delete	efface un rôle de Galaxy Ansible <code>ansible-galaxy delete github_user github_repo</code>
list	Liste les rôles installés sur le système
search	<code>ansible-galaxy search cyberark</code>
import	Importe un rôle de GitHub vers galaxy.ansible.com ( <code>ansible-galaxy login</code> d'abord)
info	Affiche des informations sur un rôle donné <code>ansible-galaxy role info geerlingguy.haproxy</code>
install	Installe un rôle en local, par défaut dans <code>~/ .ansible/roles</code> Si nom de rôle seul, télécharge depuis Galaxy Ansible ou selon configuration de <code>ansible.cfg</code>

sur le site web Ubuntu : <https://manpages.ubuntu.com/manpages/kinetic/en/man1/ansible-galaxy.1.html>



**Merci**

