



Support pédagogique Ansible - Collections





Les collections Ansible

Les collections - Introduction

Depuis la version 2.10 d'Ansible, l'architecture du projet a été revue pour séparer le cœur du moteur (ansible-core) des contenus supplémentaires (modules, plugins, rôles) distribués via Ansible Galaxy sous forme de collections.

- Apporte de la **modularité** pour intégrer des **changements** plus rapidement sans attendre la prochaine **release**
- Sans les **collection**, chaque fois qu'un nouveau **module** ou une **nouvelle version** d'un module est disponible auprès de la **communauté** ou des **fournisseurs**, les **utilisateurs** doivent **attendre** la prochaine version d'**Ansible** pour obtenir le module mis à jour
- Permet de faire du **packaging** Ansible (partage et réutilisation)
 - Inclus les **modules**, les **roles**, les **plugins**,
- Ce format permet de regrouper et partager facilement du code réutilisable via **Ansible Galaxy** ou d'autres hubs.

Les collections - Introduction

 Fonctionnalité	 Version Ansible	 Date approximative
Introduction de Galaxy (rôles)	1.4	Novembre 2013
Lancement du site Galaxy Web	—	2014
Introduction des collections	2.10	Août 2020
Séparation <code>ansible-core</code> / contenu	2.10+ / 2.11	2020–2021

Les collections

Structure d'une collection Ansible

```
my_namespace/  
└─ my_collection/  
    └─ plugins/  
        └─ modules/  
        └─ filters/  
        └─ actions/  
    └─ roles/  
    └─ playbooks/  
    └─ README.md  
    └─ tests/  
        └─ integration/  
        └─ unit/
```

Nom de l'espace de noms, un identifiant unique pour les collections.
Nom de la collection Ansible
plugins personnalisés d'Ansible.
modules Ansible, scripts personnalisés pour des tâches spécifiques.
filtres Jinja2, utilisés pour formater ou manipuler les données.
plugins d'actions, pour exécuter des tâches avancées
rôles : tâches, handlers et fichiers de configuration.
playbooks Ansible
Fichier README qui décrit la collection, son contenu et son utilisation.
tests, contenant les tests unitaires et d'intégration de la collection.
vérifie que les modules/rôles fonctionnent correctement ensemble.
vérifie que chaque module ou rôle fonctionne comme prévu en isolation.

galaxy.yml – détails du fichier

Fichier obligatoire contenant les métadonnées :

```
namespace: demo
name: moncollection
version: 1.0.0
readme: README.md
authors:
  - "Toi :)"
```

Autres champs : `description` , `tags` , `dependencies` , `repository` , etc.

Notion de Namespace


Un **namespace** identifie l'auteur ou l'organisation d'une collection.

Exemples : `community`, `ansible`, `mycompany`



Permet d'éviter les conflits entre noms de rôles/modules.

Exemple concret de collection publique

 `community.general`

Contient des centaines de modules utiles :

- `community.general.yum_versionlock`
- `community.general.htpasswd`
- `community.general.shutdown`

 Utilisable via :

```
- name: Verrouillage de version
  community.general.yum_versionlock:
    name: nginx
    state: present
```


Où sont stockées les collections installées ?

Par défaut :

```
~/.ansible/collections/ansible_collections/
```

Elles peuvent aussi être installées dans un répertoire de projet via `ANSIBLE_COLLECTIONS_PATHS` .

Dans notre contexte de formation :

```
/root/.local/lib/python3.9/site-packages/ansible_collections/
```

Les collections

- Un exemple d'utilisation d'une collection dans un `playbook` :

```
- name: set SELinux to enforcing
  hosts: localhost
  gather_facts: no
  become: yes
  collections:
    - ansible.posix
  tasks:
    - name: set SELinux to enforcing
      selinux:
        policy: targeted
        state: enforcing
```

- Pour avoir de l'aide sur la `collection`

```
# ansible-doc ansible.posix.selinux
```

Les collections

- Interroger les collections installées en local

```
# ansible-galaxy collection list
```

- Installation d'une collection pour utiliser le module `community.general.dig lookup` (on en aura besoin plustard)

```
# ansible-galaxy collection install community.general
```

Les collections

- On crée une collection `test` avec la commande `ansible-galaxy`

```
$ mkdir /root/ansible-practice/ansible_collections && cd /root/ansible-practice/ansible_collections  
$ ansible-galaxy collection init coll.test  
- Collection coll.test was created successfully
```

- L'arborescence de la collection `test`

```
└─ coll  
  └─ test  
      ├── docs  
      ├── galaxy.yml  
      ├── plugins  
      │   └─ README.md  
      ├── README.md  
      └─ roles
```

Les collections

- On crée un rôle `rtest` dans la collection `test`

```
# ansible-galaxy role init rtest --init-path=coll/test/roles/  
- Role rtest was created successfully
```

- On ajoute une tache qui affiche un débogue

```
# vi coll/test/roles/rtest/tasks/main.yml
```

```
---  
# tasks file for rtest  
  
- name: Debug test.  
  debug: msg="Un role test !!!"
```

Les collections

- On build la collection `test` depuis le répertoire de la collection avec la commande `ansible-galaxy`

```
# cd coll/test/ && ansible-galaxy collection build  
Created collection for coll.test at /root/coll/test/coll-test-1.0.0.tar.gz
```

On vient de créer un artefact (fichier archive compressé)

Les collections

L'intérêt des collections est de pouvoir les réutiliser

- Nous allons donc créer un autre projet dans lequel nous allons faire appel à notre collection `test`

```
# mkdir ~/ansible-practice/projet00 && cd ~/ansible-practice/projet00
```

```
# ansible-galaxy collection install \
/root/ansible-practice/ansible_collections/coll/test/coll-test-1.0.0.tar.gz -p ./collections
```

```
Starting galaxy collection install process
[WARNING]: The specified collections path '/root/ansible-practice/projet00/collections'
is not part of the configured Ansible collections paths
'/root/.ansible/collections:/usr/share/ansible/collections'. The installed collection
will not be picked up in an Ansible run, unless within a playbook-adjacent collections
directory.
Process install dependency map
Starting collection install process
Installing 'coll.test:1.0.0' to '/root/ansible-practice/projet00/collections/ansible_collections/coll/test'
coll.test:1.0.0 was installed successfully
```

Les collections

- Un warning nous signale que le chemin de la collection doit être indiqué, on choisit de le faire dans `ansible.cfg`

```
$ cat ansible.cfg

[defaults]
collections_paths=/root/ansible-practice/projet00/collections
```

- On crée le `site.yml` pour jouer la collection

```
---
- name: test collection import
  hosts: localhost
  gather_facts: false
  tasks:
    - ansible.builtin.import_role:
        name: coll.test.rtest
```


Les collections

- On lance le playbook en question

```
# ansible-playbook site.yml
```

```
PLAY [localhost] *****

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

TASK [coll.test.rtest : Debug test.] *****
ok: [localhost] => {
    "msg": "Un role test !!!"
}

PLAY RECAP *****
localhost                : ok=2
```