

Molecule – Guide complet (Ansible Testing Framework)

Objectif : Comprendre et utiliser Molecule pour tester des rôles Ansible de manière automatisée, reproductible et fiable.

1. Qu'est-ce que Molecule ?

Molecule est un framework de test pour **rôles Ansible**. Il permet de vérifier automatiquement qu'un rôle fonctionne correctement dans un environnement simulé (Docker, Podman, LXC ou VM).

Il fait partie de l'écosystème Ansible et est utilisé pour :

- tester des rôles Ansible
- valider des infrastructures de manière reproductible
- automatiser les tests CI/CD
- éviter les régressions en production

2. Pourquoi utiliser Molecule ?

Sans Molecule :

- Tests manuels des rôles
- Erreurs découvertes en production
- Aucune reproductibilité

Avec Molecule :

- Tests automatisés
- Environnements jetables
- Validation systématique des rôles

3. Le cycle Molecule (IMPORTANT)

Molecule fonctionne selon un cycle standard :

```
create → converge → verify → destroy
```

Étape	Rôle	Explication
Create	Créer l'environnement	Container ou VM vierge
Converge	Appliquer le rôle	Exécution Ansible
Verify	Tester	Validation du résultat
Destroy	Nettoyer	Suppression de l'environnement

Commande globale : `molecule test` exécute tout automatiquement.

4. Architecture d'un rôle avec Molecule

```
role/  
├─ molecule/  
│   └─ default/  
│       ├── molecule.yml  
│       ├── converge.yml  
│       └── verify.yml
```

- `molecule.yml` → configuration du scénario
- `converge.yml` → exécution du rôle
- `verify.yml` → tests automatisés

5. Installation

```
pip install molecule molecule-docker  
# ou  
pip install molecule molecule-podman
```

6. Création d'un rôle + Molecule

```
ansible-galaxy init my_role
cd my_role
molecule init scenario default
```

7. Exemple molecule.yml

```
driver:
  name: docker

platforms:
  - name: instance
    image: geerlingguy/docker-ubuntu2204-ansible
    privileged: true

provisioner:
  name: ansible

verifier:
  name: ansible
```

8. Exemple converge.yml

```
- name: Converge
  hosts: all
  tasks:
    - name: Include role
      include_role:
        name: my_role
```

9. Exemple verify.yml

```
- name: Verify nginx installation
  hosts: all
  tasks:
    - name: Check nginx version
      command: nginx -v
      register: result
      changed_when: false

    - name: Assert nginx is installed
      assert:
```

```
that:  
  - result.rc == 0
```

10. Commandes essentielles

```
molecule create  
molecule converge  
molecule verify  
molecule destroy  
molecule test
```

11. Debug

```
molecule login  
molecule destroy  
molecule test --debug
```

12. Drivers disponibles

- Docker (standard)
- Podman (rootless)
- LXC (possible mais complexe)
- VM (Vagrant)

Attention : Docker dans LXC nécessite nesting=1 côté Proxmox.

13. Bonnes pratiques

- 1 rôle = 1 scénario Molecule
 - toujours ajouter verify.yml
 - tester plusieurs distributions si possible
 - intégrer CI/CD
-

14. Résumé

Molecule transforme un rôle Ansible en système testable automatiquement.

- ✓ Moins d'erreurs en production
- ✓ Tests reproductibles
- ✓ Infra industrialisée

Revision #3

Created 2026-05-18 20:13:57 UTC by Nico là

Updated 2026-05-26 15:48:02 UTC by Nico là