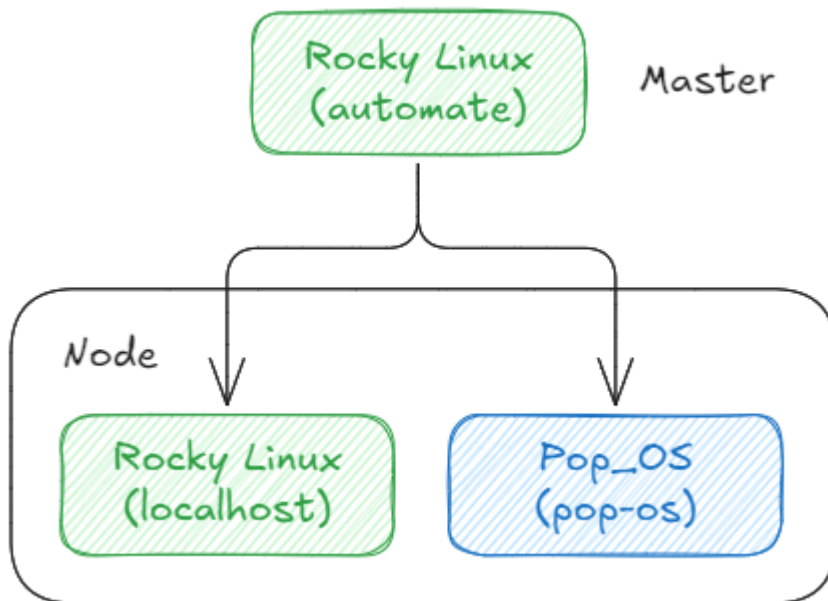


Architecture d'apprentissage

Dans cet projet, l'architecture de notre lab. est le suivant pour bien avoir une courbe d'apprentissage de 0 à la maîtrise du technologie `Ansible`.



Installation de Ansible

`Distribution_RHEL` 📖

```
1 dnf install ansible
```

`Python` 🐍

```
1 # Install dependancies
2 dnf install python3-pip python3-argcomplete rust cargo curl
3
4 # Install via pip Ansible
5 pip3.9 install --user ansible
```

SSH et les clés

Génération de clés public et privée

```
1 ssh-keygen -t ecdsa -b 521
```

Secure SSH

Path : `./ssh/authorized_keys`

```
1 from "192.168.119.163",ecdsa-sha2-nistp521 AAAAE2VjZHNhLXNoYTItbmlzdHA1MjEAAAABbm1zdH...
2 i00EjtgemLfVv2LucA== project@automate
```

Premier test

On va lancer la commande ping un module propre à ansible. Comme illustre sur la figure ci-dessous.

```
[project@automate ~]$ ansible -i "192.168.119.136," -m ping 192.168.119.136
192.168.119.136 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
[project@automate ~]$
```

Avec plusieurs machine, on obtient cet image ci dessous, l'option `-i` spécifie qu'on entre le hôte automatiquement et l'option `-m` représente les modules spécifiques de Ansible.

```
[project@automate ~]$ ansible -i "192.168.119.154,192.168.119.136," -m ping 192.168.119.154,192.168.119.136
Enter passphrase for key '/home/project/.ssh/id_ecdsa':
[WARNING]: Platform linux on host 192.168.119.154 is using the discovered Python interpreter at /usr/bin/python3, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
192.168.119.154 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
test
192.168.119.136 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
t[project@automate ~]$
```

Dans cet section, on va utiliser une autres approche avec un fichier de configuration écrit en `YAML` .

Configuration_YAML (hosts.yml)

```
1  rhel:
2    hosts:
3      192.168.119.136:
4        # Spécifié l'interpreteur Python
5        ansible_python_interpreter: /usr/bin/python3
6
7  debian:
8    hosts:
9      192.168.119.154:
10     ansible_python_interpreter: /usr/bin/python3
```

Pour lancer ansible avec cet configuration écrit en YAML, le commande serait :

Test_connection_hôte

```
1 # Lancer pour toutes les groupes
2 ansible all -i hosts.yml -m ping
3
4 # Lancer pour un groupes seulement
5 ansible rhel -i hosts.yml -m ping
6 ansible debian -i hosts.yml -m ping
```

Résultat :

```
[project@automate ansible_automatisation]$ ansible all -i hosts.yml -m ping
Enter passphrase for key '/home/project/.ssh/id_ecdsa':
192.168.119.154 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
test
192.168.119.136 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
[project@automate ansible_automatisation]$ ansible rhel -i hosts.yml -m ping
192.168.119.136 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
[project@automate ansible_automatisation]$ ansible debian -i hosts.yml -m ping
192.168.119.154 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Commande utilitaire d'ansible

Ici, on test le module `ping` et `shell` avec une résultat : OK

```
1 # Test de connection sur le réseau local
2 ansible localhost -m ping
3
4 # Option
5 # -m : Appel un module
6 # ping : Module propre à ansible
7 # -a : Ajouter le commande à exécuter
8 # shell : Module propre à ansible pour l'exécution de scrip
9
10 # Exemple lancer un script bash
11 ansible [rhel|debian] -m shell -a 'echo $TERM'
```

Résultats :

```
[project@automate ansible_automatisation]$ ansible localhost -m ping
[WARNING]: Unable to parse /home/project/ansible_automatisation/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
localhost | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
[project@automate ansible_automatisation]$ ansible debian -i hosts.yml -m shell -a "echo $SHELL"
Enter passphrase for key '/home/project/.ssh/id_ecdsa':
192.168.119.154 | CHANGED | rc=0 >>
/bin/bash
[project@automate ansible_automatisation]$
```