

DOCUMENTATION TECHNIQUE

Installation & Configuration

Ansible

Automatisation réseau Cisco IOS · Ubuntu Server

Date de rédaction : 07 avril 2026

1. Installation d'Ansible

L'installation d'Ansible sur Ubuntu se réalise en quatre étapes successives. Il est recommandé d'exécuter ces commandes en tant que root ou via sudo.

1.1 Mise à jour des paquets

Avant toute installation, il est nécessaire de mettre à jour la liste des paquets disponibles :

```
# Met à jour la liste des paquets
root@ansible:/home/ansible# apt update
```

1.2 Installation des dépendances

Installation du paquet software-properties-common, qui permet la gestion de sources de téléchargement supplémentaires :

```
# Installation de la nouvelle source de téléchargement
root@ansible:/home/ansible# apt install software-properties-common
```

1.3 Ajout du dépôt officiel Ansible

Ajout du PPA (Personal Package Archive) officiel d'Ansible pour obtenir la dernière version stable :

```
# Ajout du dépôt officiel d'Ansible
root@ansible:/home/ansible# add-apt-repository --yes --update ppa:ansible/ansible
```

1.4 Installation d'Ansible

Une fois le dépôt ajouté, procéder à l'installation du paquet ansible :

```
# Installation d'Ansible
root@ansible:/home/ansible# apt install ansible
```



Une connexion Internet active est requise pour télécharger les paquets. L'installation nécessite les droits root.

2. Vérification de l'installation

Après l'installation, vérifiez que Ansible est correctement installé et récupérez les informations de version :

```
root@ansible:/etc/ansible# ansible --version
```

Résultat attendu lors d'une installation réussie :

```
GNU nano 7.2                               ansible.cfg
# Since Ansible 2.12 (core):
# To generate an example config file (a "disabled" one with all default settings, commented out):
#     $ ansible-config init --disabled > ansible.cfg
#
# Also you can now have a more complete file by including existing plugins:
# ansible-config init --disabled -t all > ansible.cfg
#
# For previous versions of Ansible you can check for examples in the 'stable' branches of each version
# Note that this file was always incomplete and lagging changes to configuration settings
# for example, for 2.9: https://github.com/ansible/ansible/blob/stable-2.9/examples/ansible.cfg
[defaults]
collections_path = /usr/lib/python3/dist-packages/ansible_collections:/root/.ansible/collections
inventory = /etc/ansible/hosts
host_key_checking = False
```

Détail des informations retournées par la commande :

Paramètre	Valeur / Description
ansible [core 2.20.4]	Version installée d'Ansible Core
config file	/etc/ansible/ansible.cfg
executable location	/usr/bin/ansible
python version	3.12.3 — /usr/bin/python3
jinj2 version	3.1.2
pyyaml version	6.0.1 (avec libyaml v0.2.5)

3. Configuration du fichier d'inventaire (hosts)

Le fichier /etc/ansible/hosts liste tous les équipements réseau gérés par Ansible, organisés en groupes fonctionnels.

3.1 Déclaration des hôtes

Chaque équipement est déclaré avec son alias et son adresse IP de gestion :

```
sw_net          ansible_host=192.168.3.250
swdmz_interne  ansible_host=10.12.10.253
swdmz_intermediaire  ansible_host=10.11.10.253
rnet1          ansible_host=192.168.3.251
rnet2          ansible_host=192.168.3.252
```

3.2 Groupes d'équipements

Les équipements sont regroupés par type (switchs et routeurs) pour faciliter le ciblage des playbooks :

```
[switchs]
sw_coeur
sw_2e
sw_rc
sw_1e
sw_net
swdmz_interne
swdmz_intermediaire

[routeurs]
rnet1
rnet2
```

3.3 Variables de groupe — Authentification locale (admin)

Variables appliquées aux équipements utilisant l'authentification locale admin :

```
[auth_admin:vars]
ansible_connection=network_cli
ansible_network_os=cisco.ios.ios
ansible_user=admin
ansible_password=admin
ansible_become=yes
ansible_become_method=enable
ansible_become_password=admin
```

3.4 Variables de groupe — Authentification RADIUS

Variables appliquées aux équipements utilisant l'authentification RADIUS :

```
[auth_radius:vars]
```

```

ansible_connection=network_cli
ansible_network_os=cisco.ios.ios
ansible_user=CISCO
ansible_password=Password123.
ansible_become=yes
ansible_become_method=enable
ansible_become_password=admin

```



La variable `ansible_network_os=cisco.ios.ios` indique qu'Ansible utilisera la collection Cisco IOS pour communiquer avec les équipements réseau.

4. Configuration du fichier `ansible.cfg`

Le fichier `ansible.cfg` (situé dans `/etc/ansible/`) définit les paramètres globaux de fonctionnement d'Ansible.

```

[defaults]
collections_path = /usr/lib/python3/dist-
packages/ansible_collections:/root/.ansible/collections
inventory = /etc/ansible/hosts
host_key_checking = False

```

Directive	Description
<code>collections_path</code>	Chemins de recherche des collections Ansible installées
<code>inventory</code>	Chemin vers le fichier d'inventaire par défaut
<code>host_key_checking</code>	Désactivation de la vérification des clés SSH (False = désactivé)



`host_key_checking = False` est recommandé en environnement de lab pour éviter les erreurs de vérification SSH. En production, il est conseillé de laisser cette vérification active.

5. Exemple de Playbook

Un playbook Ansible est un fichier YAML qui décrit les tâches à exécuter sur les équipements cibles. Ci-dessous un exemple de playbook adapté à l'infrastructure Cisco IOS configurée :

```

---
- name: Configuration du parc mixte Cisco (Separation Switches/Routeurs)
  hosts: all
  gather_facts: no

  tasks:
    # =====
    # 1. VLANs (UNIQUEMENT POUR LES SWITCHS)
    # =====
    # Pour Les modèles récents (2960-X, 3650)
    - name: VLAN - Methode Moderne (2960-X / 3650)
      cisco.ios.ios_vlans:
        config:
          - name: V_TEST
            vlan_id: 3
            state: merged
            when: "inventory_hostname in ['sw_net', 'sw2serv', 'sw_coeur']"
            ignore_errors: yes

    # Pour Les modèles Legacy (2960 Standard : sw_1e, sw_rc, interne, intermediaire)
    - name: VLAN - Methode Robuste (2960 Standard)
      cisco.ios.ios_config:
        lines:
          - name V_TEST
            parents: vlan 3
            when: "inventory_hostname in ['sw_1e', 'sw_rc', 'swdmz_interne', 'swdmz_intermediaire']"

    # =====
    # 2. BANNIÈRE (POUR TOUT LE MONDE)
    # =====
    - name: Configuration du message d'accueil (MOTD)
      cisco.ios.ios_banner:
        banner: motd
        text: "ACCES SECURISE - EQUIPEMENT {{ inventory_hostname }}"
        state: present

    # =====
    # 3. NTP (POUR TOUT LE MONDE)
    # =====
    # On utilise ios_config simple pour garantir la compatibilité avec Les 2911 et Les vieux 2960
    - name: Configuration du serveur NTP
      cisco.ios.ios_config:
        lines:
          - ntp server 192.168.3.254

    # =====
    # 4. SAUVEGARDE (POUR TOUT LE MONDE)
    # =====
    - name: Ecriture en memoire (copy run start)
      cisco.ios.ios_config:
        save_when: modified

```



Les playbooks doivent être lancés depuis le serveur Ansible avec la commande : `ansible-playbook nom_du_playbook.yml`

6. Commandes utiles

Commande	Description
<code>ansible --version</code>	Afficher la version installée
<code>ansible all -m ping</code>	Tester la connectivité à tous les hôtes
<code>ansible switchs -m ping</code>	Tester la connectivité au groupe switchs
<code>ansible-playbook playbook.yml</code>	Exécuter un playbook
<code>ansible-playbook playbook.yml --check</code>	Simulation sans modification (dry-run)
<code>ansible-inventory --list</code>	Afficher l'inventaire au format JSON
<code>ansible all -m gather_facts</code>	Récupérer les infos des équipements

