

INSTALLATION D'ANSIBLE

PROCÉDURE D'INSTALLATION

INSTALLATION SUR LINUX

UTILISATION DE PACKAGE MANAGER

- **apt-get** (Debian/Ubuntu)
- **yum** (Fedora/CentOS/RHEL)
- **zypper** (OpenSUSE/SLES)

APT-GET (DEBIAN/UBUNTU)

```
sudo apt-get update  
sudo apt-get install ansible
```

YUM (FEDORA/CENTOS/RHEL)

```
sudo yum install ansible
```

Distribution	Commande d'installation
Fedora	<code>sudo dnf install ansible</code>
CentOS/RHEL	<code>sudo yum install ansible</code>

ZYPPER (OPENSUSE/SLES)

```
sudo zypper install ansible
```

INSTALLATION À PARTIR DU SOURCE

- **Cloner** le dépôt git
- Utilisation de `**pip**`

CLONER LE DÉPÔT GIT

```
git clone https://github.com/ansible/ansible.git  
cd ansible
```

UTILISATION DE `pip`

```
sudo pip install ansible
```

INSTALLATION SUR MACOS

UTILISATION DE `brew`

```
brew install ansible
```

INSTALLATION À PARTIR DU SOURCE

Suivez les étapes pour cloner le dépôt git et utiliser `pip` comme décrit dans la section "Installation sur Linux".

- Étape 1 : Cloner le dépôt git

```
git clone https://github.com/ansible/ansible.git
```

- Étape 2 : Se déplacer dans le répertoire

```
cd ansible
```

- Étape 3 : Installer les dépendances

```
sudo apt-get install python-pip python-dev build-essen
```

- Étape 4 : Installer Ansible avec pip

```
sudo pip install .
```

Note : Vous pouvez également utiliser `pip3` et les commandes associées pour une installation avec Python 3.

INSTALLATION SUR WINDOWS

UTILISATION DE WINDOWS SUBSYSTEM FOR LINUX (WSL)

1. Installez WSL à partir du **Microsoft Store**
2. Installez une distribution Linux de votre choix (Ubuntu, Debian, etc.)
3. Suivez les étapes pour installer **Ansible** sur Linux

INSTALLATION VIA `pip`

```
pip install ansible
```

Ansible est souvent installé dans des environnements virtuels Python pour éviter les conflits avec d'autres bibliothèques.

CONFIGURATION DE BASE

PRÉREQUIS

SERVEUR DE CONTRÔLE

EXIGENCES MATÉRIELLES ET LOGICIELLES

Le **serveur de contrôle** doit avoir **Ansible** installé et être capable de se **connecter en SSH** aux **nœuds gérés**.

NŒUDS GÉRÉS

PRÉPARATION DES SYSTÈMES D'EXPLOITATION

Assurez-vous que **Python** soit installé sur les **nœuds gérés**.

SSH ET AUTORISATION SANS MOT DE PASSE

Configurer l'**authentification par clé SSH** entre le **serveur de contrôle** et les **nœuds gérés**.

- Générer une paire de clés SSH sur le serveur de contrôle :

```
ssh-keygen -t rsa -b 4096
```

- Copier la clé publique sur les nœuds gérés :

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user@managed_node_ip
```

- Tester l'authentification sans mot de passe :

```
ssh user@managed_node_ip
```

INVENTAIRE DES HÔTES

FICHIERS D'INVENTAIRE

HOSTS.INI

Créez un fichier `hosts.ini` pour définir les **nœuds gérés**.

Exemple :

```
[nodes]
node1 ansible_host=node1.example.com
node2 ansible_host=node2.example.com
```

GROUPES D'HÔTES

Regroupez les **nœuds gérés** dans des **groupes** pour faciliter la gestion.

Exemple :

Groupe	Hôte	Adresse
web	web1	ansible_host=web1.example.com
web	web2	ansible_host=web2.example.com
db	db1	ansible_host=db1.example.com

VARIABLES D'INVENTAIRE

DÉFINITION DES VARIABLES

Définissez des variables pour personnaliser le comportement d'**Ansible** sur les **nœuds gérés**.

Exemple :

```
[web]
web1 ansible_host=web1.example.com ansible_user=webuser
web2 ansible_host=web2.example.com ansible_user=webuser
```

PORTÉE DES VARIABLES

La **portée** des variables peut être définie au niveau du **nœud**, du **groupe** ou **globale**.

Niveau de portée	Description
Nœud	La variable est spécifique à un nœud
Groupe	La variable est partagée par un groupe
Globale	La variable est accessible à tous les nœuds et groupes

FICHER DE CONFIGURATION ANSIBLE

ANSIBLE.CFG

EMPLACEMENT

Le fichier `ansible.cfg` doit être situé dans le **répertoire courant** ou dans `/etc/ansible/`.

PARAMÈTRES

FORKS

Nombre de **connexions simultanées** :

```
[defaults]  
forks = 5
```

UTILISATEUR DISTANT

Utilisateur par défaut pour se connecter aux **nœuds gérés** :

```
[defaults]  
remote_user = ansible
```

CHEMIN DES RÔLES

Emplacement des rôles **Ansible** :

```
[defaults]  
roles_path = ./roles
```

JOURNALISATION

Activez la **journalisation** des actions d'Ansible :

```
[defaults]  
log_path = ./ansible.log
```

GESTION DES ERREURS DE CONNEXION

TIMEOUT

Configurer le délai d'attente pour les **connexions SSH** :

```
[ssh_connection]  
timeout = 10
```


RETRIES

Configurer le nombre de tentatives de connexion en cas d'échec :

```
[defaults]  
retries = 3
```

Note : Augmenter la valeur de `retries` peut être utile dans les environnements avec des connexions réseau instables. Cependant, une valeur trop élevée peut entraîner des délais importants lors de l'application de la configuration.

UTILISATION DES MODULES AD HOC

EXÉCUTION DE COMMANDES SIMPLES

EXEMPLE : PING

Vérifiez la **connectivité** des nœuds gérés :

```
ansible all -m ping
```

EXEMPLE : COMMAND

Exécutez une **commande shell** sur les **nœuds gérés** :

```
ansible all -m command -a "uptime"
```

MODULES COMPLEXES

EXEMPLE : USER

Créez un utilisateur sur les **nœuds gérés** :

```
ansible all -m user -a "name=jdoe uid=1001 state=present" -b
```

EXEMPLE : COPY

Copiez un fichier sur les nœuds gérés :

```
ansible all -m copy -a "src=/path/to/source dest=/path/to/destination" -b
```


EXEMPLE : APT

Installez un **paquet** sur les nœuds gérés :

```
ansible all -m apt -a "name=nginx state=present" -b
```

