

INSTALLATION ET CONFIGURATION PLAYBOOK APACHE2 ANSIBLE SUR UBUNTU 22.04

Prérequis : Serveur Ubuntu 22.04, une infrastructure fonctionnelle

Installation Ansible

```
sudo apt update && sudo apt upgrade -y  
sudo apt install software-properties-common  
sudo add-apt-repository --yes --update ppa:ansible/ansible  
sudo apt install ansible -y
```

Configuration de l'inventaire

On aura une arborescence globale comme suit :

/etc/ansible/

├── inventories/

| └── inventory.ini # Fichier d'inventaire des serveurs à gérer

```
├── roles/
|   ├── apache/ # Rôle pour installer et configurer Apache2 (Intralab)
├── playbooks/
|   ├── deploy_web.yml # Playbook pour déployer le serveur web (Apache)
|   └── deploy_all.yml # Playbook global qui appelle les autres playbooks
```

- Se rendre dans `/etc/ansible/`

```
sudo mkdir inventories
cd inventories/
sudo touch inventory.ini
sudo nano inventory.ini
```

Il faudra ici configurer l'inventaire des serveurs, avec les informations nécessaires pour qu'Ansible puisse se connecter.

```
[web_servers]
"adresse ip de votre serveur web" ansible_user="username" ansible_ssh_private_key_file="chemin"/.ssh/id_rsa
```

- Exemple :

```
[web_servers]
192.168.x.x ansible_user=vegeta ansible_ssh_private_key_file=/home/vegeta/.ssh/id_rsa
```

- username = le nom d'utilisateur que tu utilises pour te connecter via SSH (par exemple, 'admin' ou 'root')
- "chemin" = le chemin vers la clé privée SSH qui doit être utilisée pour la connexion

Configuration playbook pour Apache

Un playbook Ansible est un modèle de tâches d'automatisation, qui sont des opérations informatiques exécutées sans intervention humaine ou presque dans plusieurs solutions informatiques. Les playbooks indiquent à Ansible les actions à réaliser et sur quels dispositifs.

```
sudo mkdir apache
cd apache
sudo mkdir defaults handlers tasks templates
# Ensuite il faut créer un fichier main.yml dans chaque répertoire hormis templates
cd defaults
sudo touch main.yml
# Faire la même manoeuvre pour les autres répertoires sauf templates
# Ensuite on se rend dans tasks
# Les tâches (tasks en anglais) représentent les actions à déployer sur les serveurs cibles.
cd tasks/
sudo nano main.yml
# On doit avoir une configuration de ce style :
```

```
#roles/apache/tasks/main.yml
```

```
---
```

```
- name: Installer Apache2
  apt:
    name: apache2
    state: present
    update_cache: yes

- name: Assurer qu'Apache est démarré et activé au démarrage
  service:
    name: apache2
    state: started
    enabled: yes
```

- Se rendre dans `/etc/ansible/roles/apache/defaults/`

```
cd /etc/ansible/roles/apache/defaults/  
sudo nano main.yml  
# On doit avoir quelque chose de ce type :  
  
#roles/apache/defaults/main.yml  
---  
apache_port: 80
```

- Se rendre dans /etc/ansible/roles/apache/handlers/

Un handler est une tâche qui est appelée uniquement lorsqu'il y a eu un changement et est identifié par son nom. La notification est envoyée dans la section 'notify' de la tâche. Les handlers sont exécutés à la fin d'un play le cas échéant. Ici, notre handler va redémarrer le serveur apache si nécessaire.

```
sudo nano main.yml  
# On doit avoir quelque chose de ce genre :  
  
# roles/apache/handlers/main.yml  
---  
- name: Redémarrer Apache2  
  service:  
    name: apache2  
    state: restarted
```

- Se rendre dans /etc/ansible/playbooks
- Il faut ensuite créer le playbook dont il nous faut, dans notre cas nous nous pencherons sur un playbook qui nous garantira la présence d'un serveur apache2 et son installation sur le serveur que l'on a défini auparavant dans l'inventaire.

```
sudo touch deploy_web.yml
```

```
sudo nano deploy_web.yml
```

Nous devons avoir dans notre configuration quelque chose de ce style :

```
#playbooks/deploy_web.yml
```

```
---
```

- name: Déployer le serveur web Apache (Une description brève du playbook)
 hosts: web_servers (le nom qu'on a défini dans notre inventaire)
 become: yes
 roles:
 - apache

Vérifier la présence du SSH sur les serveurs

Pour se connecter aux serveurs et exécuter des tâches, Ansible utilise le protocole SSH. Par défaut, Ansible utilise des clés SSH avec ssh-agent et se connecte aux machines distantes à l'aide de votre nom d'utilisateur actif.

- Dans le but de faire des tests, il faut avant tout s'assurer que le ssh est fonctionnel sur nos serveurs
- On lance `sudo systemctl status ssh`
- On lance la commande `ls -la ~/.ssh/`
- Si l'on ne trouve rien, alors il faudra installer le package ssh, sinon on aura les fichiers ssh de l'utilisateur ainsi que ses clés
- Pour installer SSH :
 - `sudo apt update`
 - `sudo apt install openssh-server`

```
sudo systemctl enable ssh
```

```
sudo systemctl start ssh
```

- Faire un test manuel : `ssh -i ~/.ssh/id_rsa "username"@adresse ip du serveur`
- S'il n'existe pas de clé, alors on lance : `ssh-keygen -t rsa -b 2048 -f ~/.ssh/id_rsa`

Démarrer le playbook

Pour lancer le playbook, il faut utiliser cette commande :

```
ansible-playbook -i inventories/inventory.ini playbooks/deploy_web.yml --ask-become-pass
```

`--ask-become-pass` fera en sorte qu'un mot de passe sera demandé lors de la tentative de démarrage du playbook, cela rapporte plus de sécurité. Ce mot de passe est celui que vous avez configuré pour votre serveur.

On remarque que en fonction du playbook qu'on veut lancer, on changera le fichier `.yml` et on fera attention à bien spécifier les chemins en fonction de notre arborescence.