

Cours complet sur Ansible : De Zéro à Héros

Installation d'Ansible

Avant de commencer, installons Ansible sur notre machine.

```
sudo apt update  
sudo apt install ansible
```

Vérifions l'installation avec la commande suivante :

```
ansible --version
```

Comprendre le fonctionnement d'Ansible

Pour voir comment Ansible fonctionne, il est nécessaire d'avoir au moins deux serveurs :

1. Un serveur Ansible (serveur de contrôle)
2. Une ou plusieurs machines cibles (serveurs gérés)

Ansible doit pouvoir communiquer avec les machines cibles sans mot de passe. Pour cela, nous allons utiliser **ssh-keygen**.

Générons une clé SSH :

```
ssh-keygen
```

Copions ensuite la clé publique sur la machine cible :

```
ssh-copy-id user@target_server
```

Alternativement, ouvrez le fichier `~/.ssh/authorized_keys` sur la machine cible et collez-y la clé publique.

Qu'est-ce qu'un Ansible Playbook ?

Un **playbook** Ansible est un fichier YAML qui décrit une série de tâches à exécuter sur les machines cibles, similaire à un script shell.

Commandes Ad-Hoc

Les commandes Ad-Hoc permettent d'exécuter des actions simples sans créer un playbook.

Exemple :

```
ansible -i inventory all -m shell -a "touch /tmp/devopsclass"
```

Différence entre les commandes Ad-Hoc et les Playbooks

Ad-Hoc

Exécution unique et rapide

Utilisé pour des tâches ponctuelles

Playbook

Planification d'actions complexes

Meilleure organisation du code

Organisation des serveurs

Dans l'inventaire (`inventory`), les serveurs peuvent être regroupés :

```
[webservers]
192.168.1.10
192.168.1.11
```

```
[dbservers]
192.168.1.20
```

Écriture d'un Playbook Ansible

Créons un playbook pour installer et démarrer Nginx :

```
- name: Installer et démarrer Nginx
  hosts: all
  become: true

  tasks:
    - name: Installer Nginx
      apt:
        name: nginx
        state: present

    - name: Démarrer Nginx
      service:
        name: nginx
        state: started
```

Exécutez ce playbook avec :

```
ansible-playbook -i inventory nginx.yml
```

Ajoutez `-v` pour plus de détails :

```
ansible-playbook -i inventory nginx.yml -v
```

Déploiement Kubernetes avec Ansible

Nous allons déployer un cluster Kubernetes avec Ansible en créant trois instances EC2, une en tant que **Master** et deux en tant que **Workers**.

Créez un fichier `kubernetes.yml` :

```
- name: Configurer Kubernetes
  hosts: all
  become: true
```

```
roles:
  - kubernetes
```

Utilisation des Rôles Ansible

Un **rôle Ansible** est une structure organisée de fichiers facilitant le réemploi du code.

Créons un rôle pour Kubernetes :

```
ansible-galaxy init kubernetes
```

Cela créera un dossier avec plusieurs fichiers :

- `tasks/` : Contient les actions à exécuter
- `handlers/` : Contient les gestionnaires d'événements
- `templates/` : Contient les fichiers Jinja2 pour la configuration
- `vars/` : Contient les variables
- `defaults/` : Contient les valeurs par défaut

Créons un playbook qui appelle ce rôle :

```
- name: Déployer Kubernetes
  hosts: all
  become: true
  roles:
    - kubernetes
```

Exécutez avec :

```
ansible-playbook -i inventory kubernetes.yml
```

Questions d'entretien sur Ansible

1. Qu'est-ce qu'Ansible ?

Réponse : Ansible est un outil open-source d'automatisation IT qui permet de gérer les configurations, déployer des applications et orchestrer des tâches sur plusieurs serveurs sans agent.

2. Quelle est la différence entre une commande Ad-Hoc et un Playbook ?

Réponse : Une commande Ad-Hoc est une commande unique exécutée directement sur les serveurs, tandis qu'un Playbook est un fichier YAML contenant plusieurs tâches organisées.

3. Comment Ansible gère-t-il les connexions SSH sans mot de passe ?

Réponse : Ansible utilise des clés SSH. On génère une clé avec `ssh-keygen` et on copie la clé publique sur la machine cible avec `ssh-copy-id`.

4. Qu'est-ce qu'un inventaire Ansible ?

Réponse : Un fichier contenant la liste des hôtes et groupes de serveurs sur lesquels Ansible exécute ses tâches.

5. Comment grouper des serveurs dans Ansible ?

Réponse : En utilisant des groupes dans le fichier `inventory`, par exemple :

```
[web]
192.168.1.10
192.168.1.11
```

6. Qu'est-ce qu'un rôle Ansible ?

Réponse : Un rôle est une collection organisée de playbooks, de fichiers de variables et de gestionnaires, permettant une meilleure organisation et réutilisation du code.

7. Quelle est la différence entre `tasks/` et `handlers/` dans un rôle ?

Réponse :

- `tasks/` contient les tâches à exécuter.
- `handlers/` contient des actions qui sont appelées uniquement en cas de changement (ex : redémarrer un service après une installation).

8. Comment déboguer un Playbook Ansible ?

Réponse : On utilise l'option `-vvvv` pour obtenir plus de détails sur l'exécution.

Avec ce cours, vous avez une bonne base pour maîtriser Ansible ! 🚀