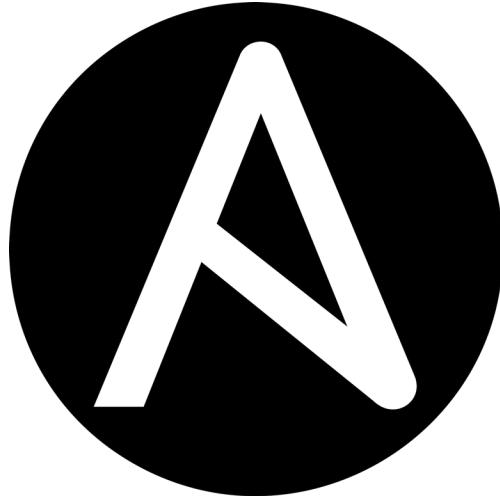


LIGNIERE
Timothée
Classe BTS2 SIO
Date de création
03/04/24
Date de maj
04/04/24

Ansible



Sommaire

1 - Problématique.....	2
2 - Notice d'installation.....	3
2.1 – Installation d'Ansible.....	3
2.2 – Configuration d'Ansible.....	4
3 - Notice d'utilisation.....	5
3.1 – Orthographe d'un playbook.....	5
4 - Annexes.....	6

1 - Problématique

Lors de la gestion d'un grand nombre de machines, il peut devenir compliqué de s'assurer de la configuration de chacune de manière individuelle, car cela peut être la cause de multiples erreurs.

Ansible permet d'éviter cela en centralisant le point de configuration des machines ; En effet, grâce à ce logiciel, vous avez la possibilité de garantir la configuration de vos machines à l'aide de fichiers de consignes qui vont donc ordonner à vos machines des tâches prédéfinies.

Par ce fait, Ansible est une alternative à SaltStack, la différence entre les deux est tout d'abord l'écriture des fichiers de consignes, mais c'est aussi la communication même avec les machines. Ansible repose entièrement sur le ssh, là où SaltStack utilise des agents qui se connectent au master pour récupérer leurs consignes.

(Il est aussi possible de passer par ssh avec Salt-ssh mais ce n'est pas son utilité première)

2 - Notice d'installation

2.1 – Installation d'Ansible

L'installation d'Ansible est très simple sous Debian, si vous utilisez un autre OS, se référer à [la doc officielle](#) :

```
apt update  
apt install ansible
```

Une fois cela fait, vérifier qu'il soit bien installé en vérifiant la version :

```
ansible --version
```

Si tout est en ordre, nous pouvons passer à la configuration.

2.2 – Configuration d'Ansible

Pour commencer, nous allons générer une clé ssh pour permettre à Ansible de se connecter aux machines sans utiliser de mot de passe :

```
ssh-keygen
```

Choisissez le chiffrage que vous préférez, puis envoyer la clé sur chacun des postes :

```
ssh-add <Clé générée>
#A répéter sur chaque machine
ssh-copy-id <Utilisateur>@<IP de la machine>
```

Lorsque toutes les machines auront leurs clés, vous pourrez décrire à Ansible vos hôtes à l'aide d'un fichier que vous allez créer :

```
nano /etc/ansible/hosts
```

Ce fichier doit avoir une construction similaire à la suivante :

```
[servers]
172.16.0.X
172.16.0.Y
```

Ainsi, vous dites à Ansible que le groupe « servers » contient la machine 172.16.0.X et la 172.16.0.Y . Vous pouvez donc tous types de groupes que vous semblez nécessaire pour votre gestion des machines.

La configuration de Ansible suit l'ordre suivant pour déterminer quels paramètres suivre :

```
ANSIBLE_CONFIG
ansible.cfg #Dans le dossier actuel
~/ansible.cfg #Dans le dossier de l'utilisateur
/etc/ansible/ansible.cfg
```

Vous pouvez générer un fichier de configuration dans le dossier actuel de la manière suivante :

```
ansible-config init --disabled > ansible.cfg
```

Chaque paramètre de ce fichier devrait avoir une description afin de comprendre son intérêt.

Une fois que vous aurez terminé de le modifier, vérifier la connexion avec les machines :

```
ansible all -m ping
```

Si les machines vous confirment la communication, tout est en ordre.

3 - Notice d'utilisation

Les fichiers de consignes d'Ansible s'appellent des « playbooks », nous allons voir la construction d'un playbook basique. Il n'est pas possible d'aller dans les détails sachant la liste exhaustive de modules ansible.

3.1 – Orthographe d'un playbook

Tous fichiers playbook doit être en format .yml ou .yaml.

De manière générale, un playbook s'écrit de la manière suivante :

```
- name: <Nom de playbook>
hosts: <Groupe d'hôtes concernés ou « all » pour tous les hôtes >
tasks:
  - name: <Nom de tâche>
    module_name:
      option1: <Valeur X>
      option2: <Valeur Y>
```

« module_name » doit donc varier selon vos besoins, tout comme ses options.

Le fichier playbook suivant permet par exemple de copier des hôtes, de créer un utilisateur ainsi que de mettre à jour les paquets installés :

```
- name: Intro aux playbooks
hosts: all

tasks:
  - name: Copie le fichier hosts avec des permissions
    ansible.builtin.copy:
      src: ./hosts
      dest: /tmp/hosts_backup
      mode: '0644'
  - name: Add the user 'USER'
    ansible.builtin.user:
      name: USER
      become: yes
      become_method: sudo
  - name: Mettre à jour les paquets apt
    apt:
      force_apt_get: yes
      upgrade: dist
      become: yes
```

Puis, pour utiliser le fichier playbook, faites :

```
ansible-playbook <Nom du playbook>.yml
```

Les tâches devraient s'exécuter et vos machines vous renvoyez un retour.

4 - Annexes

Fiche de recette

Vérification de l'opérationnalité de la solution mise en œuvre : Ansible

Description du test :

1. Vérifier ansible --version
2. Ping les machines par Ansible
3. Utilisation d'un playbook basique

Résultats Attendus :

1. Version renvoyée correct
2. Communication avec les machines confirmée
3. Tâches exécutées sans problème

Réception Globale : Ansible

Date: 03/04/24

Auteurs: Lignière Timothée

Reçu :

Reçu avec réserve :

Refusé :

Commentaire :

Recette étape par étape *

* (pour chaque étape, vous devez élaborer dans un fichier distinct un scénario détaillé à faire appliquer au « client » venant valider votre solution)

Réception Etape 1: Vérifier la version d'Ansible pour garantir son installation à l'aide de la commande « ansible --version »

Reçu :

Reçu avec réserve :

Refusé :

Commentaire :

Réception Etape 2 : Vérifier la communication avec les machines à l'aide de la commande « ansible all -m ping »

Reçu :

Reçu avec réserve :

Refusé :

Commentaire :

Réception Etape 3 : Vérifier l'exécution des tâches décrites dans un playbook basique

Reçu :

Reçu avec réserve :

Refusé :

Commentaire :