 Orange Labs recherche & développement	Installation d'Ansible.		1
	AQS/QSI/NIS	B. ROMANOW 28/04/2016	
		Version 1	

Installation d'Ansible.

Ce mémo présente l'installation d'Ansible qui permet d'automatiser certaines tâches via SSH.

Date : 28/04/2016

Version : 1

Status : Draft

Derrière maj. : 28/04/2016- Création

Auteur : Boris ROMANOW



Sommaire

Schéma et explications	3
Ansible	4
Pré requis	4
Installation d'Ansible :	4
Configuration des hosts :	4
Premiers pas avec Ansible :	5
Test de communication :.....	5
Premier playbook	6

Schéma et explications

Ansible est un programme python qui permet la configuration / le déploiement de services sur une machine distante via SSH. Ansible établit une connections SSH avec la liste de ses client et peut déployer l'installation et la configuration de paquets (ou programmes,fichiers...) de manière automatique. Il est aussi possible, par exemple, d'ajouter des utilisateurs sur les machines clientes. Les programmes qui sont écrits pour Ansible sont appelés « playbook » : ils sont généralement écrits en *yaml*.

L'architecture consiste à utiliser une machine dite « serveur » sur laquelle est installer Ansible et d'autres machines ditent « hosts » sur lesquelles le serveur déploie ses scripts *yaml*.

Ansible

Pré requis

Mettre à jour les repository ainsi que les paquets installés sur la machine :

```
sudo apt-get update  
sudo apt-get upgrade
```

Installation d'Ansible :

Installation des paquets nécessaires à Ansible :

```
sudo apt-get install -y ansible
```

C'est tout ! Ansible est maintenant installé sur la machine.

Configuration des hosts :

Pour qu'Ansible puisse se connecté et lancer sans playbooks il faut installer 2 paquets sur les hosts.

```
sudo apt-get install -y python openssh-server
```

Le paquet « python » permet à Ansible de déployer son script et le paquet « openssh-server » permet au serveur Ansible de se connecter sur l'host.

Il suffit maintenant de modifier le fichier de configuration ssh dans `/etc/ssh/sshd_config` puis dans la partie « # Authentication » mettre :

```
PermitRootLogin yes
```

Les machines hosts sont maintenant prêtes à recevoir et exécuter des playbooks.

Premiers pas avec Ansible :

Afin qu'Ansible puisse se connecter sur les machines sans utiliser de mot de passe on va générer un couple clé publique/privée. Il est important de bien rentrer une *passphrase*.

```
ssh-keygen
```

Cette opération est à faire une seule fois (par serveur Ansible).

Ensuite on copie notre clé publique sur les hosts

```
ssh-copy-id root@<ip_host>
```

Cette opération est à répéter pour chaque machine host sur laquelle le serveur Ansible est susceptible de se connecter.

Enfin on se connecte sur l'host : (Opération à réaliser sur les machines hosts de l'étape précédente)

```
ssh root@<ip_host>
```

Le premier mot de passe à rentrer est le mot de passe root du host. Le second est la *passphrase* rentré au moment de la création des clés publique et privée. Ensuite on édite le fichier de configuration de l'host et on change la ligne :

```
- PasswordAuthentication yes  
+ PasswordAuthentication no
```

Ensuite on redémarre le service ssh pour prendre en compte les modifications :


```
service ssh restart
```

Maintenant il n'est plus possible de se connecter avec les mots de passe des comptes utilisateurs, il faut disposer de la clé privée correspondant à la clé publique que l'on a enregistré sur l'host lors de la commande *ssh-copy-id*.

Test de communication :

Maintenant le serveur ansible est capable de communiquer avec ses machines hosts. Vérifions cela à l'aide de la commande suivante :

```
ansible <ip_host> -u root -m ping
```

 Orange Labs recherche & développement	Installation d'Ansible.		6
	AQS/QSI/NIS	B. ROMANOW 28/04/2016	
		Version 1	

Si cela fonctionne un message du type suivant apparaît :

```
10.194.0.8 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

Evidemment il serait fastidieux de répéter cette commande pour chaque host. Pour aller plus vite il faut éditer le fichier `/etc/ansible/hosts` et y ajouter par exemple :

```
[Cluster1]
<ip_host_1>
<ip_host_2>

[Cluster2]
<ip_host>

[web-service]
<ip_host>

Etc...
```

Grâce à ce fichier il est possible de regrouper des machines par catégories et ainsi lancé les playbooks sur les machines concernées.

Pour tester tous les hosts il suffit de lancer la commande suivante :

```
ansible all -u root -m ping
```

Cette commande teste toutes les machines présente dans `/etc/ansible/hosts` mais on peut aussi tester seulement les hosts de « web-service » via la commande :

```
ansible web-service -u root -m ping
```

Premier playbook

Maintenant que les machines communiquent testons notre premier playbook qui installe simplement un paquet. Editer le fichier `first.yml` puis y insérer :


```
---
- hosts: TEST
  remote_user: root
  tasks:
    - name: Installation paquet "sl"
      apt: name=sl update_cache=yes
```

Attention, la syntaxe est très importante !

Ici :

- « --- » signifie le début du script YAML
- « hosts » signifie le groupe de machines sur lesquelles installer le paquet
- « name » : optionnel permet d'être plus verbeux lors de l'exécution du playbook
- « apt » : commande ansible servant à la gestion des paquets



 Orange Labs recherche & développement	Installation d'Ansible.		7
	AQS/QSI/NIS	B. ROMANOW 28/04/2016	
		Version 1	

- « name » : nom du paquet
- « update_cache » : indique s'il faut lancer *apt-get update* avant

Pour exécuter le playbook, se rendre à l'endroit où est situé le *.yaml* et lancer la commande suivante :

```
ansible-playbook first.yaml
```

Si tout se passe bien vous devriez obtenir le résultat suivant :

```
PLAY [SNORT] *****
TASK [setup] *****
ok: [10.194.0.8]

TASK [Installation paquet "sl"] *****
changed: [10.194.0.8]

PLAY RECAP *****
10.194.0.8      : ok=2    changed=1    unreachable=0    failed=0
```

Toutes les commandes ansible nécessaires à la réalisation de playbooks sont disponibles [ici](#).

