Projet Ansible v1

2017/2018

1. **Installation et configuration sous centOS 7**
   1. **Configuration réseau liée au réseau de l’école**

Modifier « /etc/profile » et ajouter :

|  |
| --- |
| MY\_PROXY="http://129.194.185.57:3128/"  HTTP\_PROXY=$MY\_PROXY  HTTPS\_PROXY=$MY\_PROXY  FTP\_PROXY=$MY\_PROXY  http\_proxy=$MY\_PROXY  https\_proxy=$MY\_PROXY  ftp\_proxy=$MY\_PROXY  export HTTP\_PROXY HTTPS\_PROXY FTP\_PROXY http\_proxy https\_proxy ftp\_proxy |

Exécuter la commande source /etc/profile

Modifier « /etc/yum.conf » et ajouter :

|  |
| --- |
| proxy = http://129.194.185.57:3128/ |

* 1. **Update et installation**

Update :

|  |
| --- |
| Sudo yum update |

Installation :

|  |
| --- |
| Sudo yum install ansible |

Si problème de dépôt :

|  |
| --- |
| sudo yum install epel-release |

* 1. **Configuration**

Ouvrir le fichier hosts dans *vi /etc/ansible/hosts* et ajouter :

|  |
| --- |
| [group\_name]  alias ansible\_ssh\_host=your\_server\_ip  [servers]  host1 ansible\_ssh\_host= ip address  host2 ansible\_ssh\_host= ip address  host3 ansible\_ssh\_host= ip address |

Exemple:

Server\_web ansible\_ssh\_host = 10.0.0.2

Les hôtes peuvent être dans des groupes multiples et les groupes peuvent configurer des paramètres pour tous leurs membres.

Ensuite faut crée un dossier groupe qui vas contenir nos configuration des groupes crée dans Hosts :

|  |
| --- |
| sudo mkdir /etc/ansible/group\_vars |

Puis crée un fichier au nom du groupe (ici ces « servers ») et l’ouvrir :

|  |
| --- |
| sudo nano /etc/ansible/group\_vars/servers |

Et intégré les informations pour la connexion :

|  |
| --- |
| ---  ansible\_ssh\_user: « nom d’user pour le SSH » |

Ensuite il faut générer des clefs pour le SSH entre le(s) client(s) et le serveur :

|  |
| --- |
| ssh-add //add the ssh key to the agent  ssh-keygen -t rsa -C "user@ip\_ansible"  ssh-copy id user@ipduclient |

Dans notre cas :

|  |
| --- |
| ssh-keygen -t rsa -C "ansible@ 10.194.184.190"  ssh-copy id root@ 10.194.184.191 |

Puis lancer le test :

|  |
| --- |
| ansible -m ping « nom du groupe a ping » |

Si sa a marche :

|  |
| --- |
| host1 | SUCCESS => {  "changed": false,  "ping": "pong"  } |

Ou si sa marche pas :

|  |
| --- |
| host1 | UNREACHABLE! => {  "changed": false,  "msg": "Failed to connect to the host via ssh.",  "unreachable": true  } |

1. **Création et exécution de playbooks** 
   1. **Commande pour lancer un playbooks :**

|  |
| --- |
| ansible-playbook chemindufichier/name.yml |

Avec variable en parameter:

|  |
| --- |
| ansible-playbool chemindufichier/name.yml –e “nomvariable=valueofvariable” |

-e = --extra-vars

Exemple :

|  |
| --- |
| ansible-playbook /playbooks/ping.yml |

* 1. **Création et configuration d’un fichier YML**

Création et ouverture (exemple avec lancement d’un ping) :

|  |
| --- |
| nano /playbooks/ping.yml |

Et le contenu :

|  |
| --- |
| --- Doit toujours commencer par ceci /!\  - hosts: Clients  remote\_user: root  tasks:  - ping:  … |

hosts : on doit indiquer le groupe ou la machine à qui on veut faire exécuter notre futur taches/services

remote\_user : indiqué sous qu’elle utilisateur nous voulons lancer la prochaine taches/services

tasks : pour indiquer que la prochaine ligne sera une taches a faire

-ping : indique le nom de la commande à lancer

On peut aussi démarrer, arrêter, relancer un service (par exemple apache)

|  |
| --- |
| - service:  name: httpd  state: started |

-service : pour indiquer qu’on veut travailler sur un service

name : le nom du service qu’on veut toucher

state : indiquer l’état futur (started, stopped, restarted, reloaded)

Bibliographie :

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-centos-7>

<https://github.com/ansible/ansible/issues/19584>

<https://docs.ansible.com/ansible/latest/yum_module.html>

<https://docs.ansible.com/ansible/latest/service_module.html>

<https://docs.ansible.com/ansible/latest/apt_module.html>

<https://docs.ansible.com/ansible/latest/apt_repository_module.html>