**Architecture mise en place dans le cadre de ces playbooks Ansible servant à redémarrer un groupe de machines.**

Dans le cadre de ce projet, j’ai mis en place trois serveurs dont 2 sous la distribution Linux Debian10 et un sous Windows Serveur2016.

Serveur SQL: *srvsql* ; Serveur node manager: *nodemng* ; Serveur Windows: *winsrv*

***Préparation des serveurs Windows***

* Activation et configuration de WinRM

En jouant les commandes suivantes dans powershell

winrm set winrm/config/client/auth '@{Basic="true"}'

winrm set winrm/config/service/auth '@{Basic="true"}'

* Authorisation de WinRM dans le Pare-feu

netsh advfirewall firewall add rule Profile=Any name="Autoriser WinRM HTTPS" dir=in localport=5986 protocol=TCP action=allow

***Création de l’utilisateur ansible avec mot de passe chiffré sur les nodes***.

Sous ansible on créait d’abord le mot de passe du compte utilisateur à créer.

ansible localhost -i inventaire.ini -m debug -a "msg={{ 'secret' | password\_hash('sha512', 'sceretsalt') }}"

ici avec l’utilisation du module *debug* avec l’argument *msg* pour transformer le mot de passe *secret* en une châine chiffrée avec l’algorithme *sha512.* On ajoute ensuite *« scertsalt »* pour compliquer le mot de passe. L’option *localhost* indique à ansible de lancer cette commande sur le localhost donc le poste node manager.

On peut ensuite créer l’utilisateur « *ansible* » qui lui aura les droits *root* afin de ne pas utiliser le compte root.

Commande : ansible -i inventaire.ini -m user -a 'name=ansible password=passphrase générée au chiffrement du mot de passe' --user root --ask-pass all

Explication : utilisation du module *user* avec l’argument *name* et  *password* en demandant à ansible de lancer la commande sur *all* c’est-à-dire tous les nodes déclarés sur le fichier inventaire.

***Donner les droits sudo à l’utilisateur ansible***

Utilisant la même distribution Linux, la configuration de sudo ici est contenu dans le groupe sudo auquel j’ajoute l’utilisateur *ansible.*

ansible -i inventaire.ini -m user -a 'name=ansible groups=sudo append=yes ' --user root --ask-pass all

Puis on vérifie que le compte a été ajouté au groupe « sudo » sur tous les node de l’inventaire.ini en exécutant la commande avec le compte *ansible*

ansible -i inventaire.ini -m user -a 'name=ansible groups=sudo append=yes ' --user ansible --ask-pass --become --ask-become-pass all

Explication : on utilise le module *user* pour ajouter *ansible* dans le groupe *sudo* et on se sert du module *become* pour devenir un autre utilisateur ‘*ansible*’ en utilisant sudo. Ceci fait, on n’utilisera plus le compte *root* mais le compte *ansible* en mode sudo pour passer les commandes à ansible.

***Ajout des clients dans le fichiers hosts***

Le fichier hosts de définition des utilisateurs se trouve sous */etc/ansible/hosts*

***Création d’un fichier ansible-vault pour les postes Windows***

Ici on créait un fichier yml dans le dossier /etc/ansible/group\_vars/ que je nomme *windows.yml* dans lequel je déclare le compte utilisateur de connexion sur le poste windows.

ansible\_ssh\_user: « compte\_utilisateur\_avec\_privilège »

ansible\_ssh\_pass: « pass\_secret »

ansible\_ssh\_port: 5986

ansible\_connection: winrm

ansible\_winrm\_server\_cert\_validation: ignore

ansible\_winrm\_transport: credssp

***Sur les postes Windows (serveur 2016)***

Configurer le mode d’authentification :

winrm set winrm/config/service/auth @{Basic="true"}

winrm set winrm/config/service @{AllowUnencrypted="true"}

Ouvrir le port 5985 du pare-feu

netsh advfirewall firewall add rule Profile=Any name="Autoriser WinRM HTTPS" dir=in localport=5986 protocol=TCP action=allow

Configurer CredSSP en téléchargeant le script suivant : <https://github.com/ansible/ansible/blob/devel/examples/scripts/ConfigureRemotingForAnsible.ps1>

Lancer powershell as administrator

.\ConfigureRemotingForAnsible.ps1 -CertValidityDays 3650 -EnableCredSSP

***Créer le répertoire “files” qui contiendra les différents scripts***

Sur le *nodemng* sous /etc/ansible/ on créait le dossier files/ dans lequel je mets mes scripts. Ici je mets en premier le script pour rebooter les machines Windows et linux « *reboot.py* » qui est un script écrit en Python, et aussi les script « *mailreboot.py* » qui lui sera utiliser pour renvoyer un mail de confirmation après reboot du serveur.

Les playbooks sont créés directement à la racine du dossier « /etc/ansible ». Ils sont créés au format Yaml et sont nommés : « *reboot-win.yml* » me permettant de lancer le script sur les groupes d’ordinateur Windows et « *reboot-lin.yml* » pour le groupe d’ordinateur Linux.

*Playbook pour le groupe Windows* : “/etc/ansible/reboot-win.yml”

---

- name: "Run reboot Python script" #*nom du PlayBook*

hosts: "windows" #*On indique ici le groupe d’application du Playbook*

gather\_facts: "false" #*mettre cette attribu à false pour ne pas collecter les infos des hosts*

tasks: #*Définition des taches à accomplir*

- name: " Reboot des serveurs" #*nom de la tâche*

script: "files/reboot.py" #*Ici script est le module Ansible qui va lancer le script voulu*

args: #*argument pour dire à Ansible quel programme exécuter le script*

executable: python #*on spécifie le programe d’exécution du script. Ici Python car V2*

- name: "Wait for the reboot to complete" #*Tâche d’attente avant de couper la connexion*

wait\_for\_connection: #*Attendre le rétablissement de la connexion*

connect\_timeout: 20 # *Relancer la connexion de transport après un délai de 20 seconde*

delay: 60 #*Attendre 60 secondes avant de commencer le check*

timeout: 300 #*délais maximum d’attente*

- name: "Sending an e-mail after reboot" #*Tâche d’envoie d’e-mail après reboot*

script: "files/mailreboot.py" #*Script qui sera executer sur l’hote pour confirmer le reboot*

args:

executable: python

*Playbook pour le groupe Linux* : “/etc/ansible/reboot-lin.yml”

---

- name: "Run reboot Python script" #*nom du PlayBook*

hosts: "linux" #*On indique ici le groupe d’application du Playbook*

gather\_facts: "false" #*mettre cette attribu à false pour ne pas collecter les infos des hosts*

tasks: #*Définition des taches à accomplir*

- name: "test" #*nom de la tâche*

script: "files/reboot\_lin.py" #*Ici script est le module Ansible qui va lancer le script voulu*

***Script de redémarrage pour le groupe des machines Linux et Windows***

/etc/ansible/files/reboot.py

#!/usr/bin/python

# Script de reboot de serveurs Linux

import sys, platform

import socket

import smtplib

sock\_name = socket.gethostname()

from email.MIMEMultipart import MIMEMultipart

from email.MIMEText import MIMEText

Fromadd = "yarez.leroy@gmail.com "

Toadd = "yarez@outlook.com"

message = MIMEMultipart()

message['From'] = Fromadd

message['To'] = Toadd

message['Subject'] = sock\_name + " : " socket.gethostbyname(sock\_name) + " , reboot"

msg = "Le serveur " + sock\_name + " ( " + socket.gethostbyname(sock\_name) + " ) " + " va rebooter"

message.attach(MIMEText (msg.encode('utf-8'), 'plain', 'utf-8'))

serveur = smtplib.SMTP('smtp.gmail.com:587')

serveur.starttls()

password = "Nathan0208\*Raphael"

serveur.login(Fromadd, password)

texte = message.as\_string().encode('utf-8')

Toadds = [Toadd]

serveur.sendmail(Fromadd, Toadds, texte)

serveur.quit()

if sys.platform == 'win32':

import ctypes

user32 = ctypes.WinDLL('user32')

user32.ExitWindowsEx(0x00000002, 0x00000001)

else:

import os

os.system("systemctl stop mysqld")

os.system("shutdown -r now")

***Script d’envoi d’e-mail de confirmation du redémarrage pour le groupe des machines Linux et Windows***

/etc/ansible/files/mailreboot.py

import os

import sys

import socket

import smtplib

sock\_name = socket.gethostname()

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

Fromadd = "mail expediteur "

Toadd = "mail destinataire"

message = MIMEMultipart()

message['From'] = Fromadd

message['To'] = Toadd

message['Subject'] = sock\_name + " : " socket.gethostbyname(sock\_name) + " , reboot"

msg = "Le serveur " + sock\_name + " ( " + socket.gethostbyname(sock\_name) + " ) " + " vien de redémarrer avec succès “

message.attach(MIMEText (msg.encode('utf-8'), 'plain', 'utf-8'))

serveur = smtplib.SMTP('smtp.gmail.com', 587)

serveur.starttls()

serveur.login(Fromadd, "Nathan0208\*Raphael")

texte = message.as\_string().encode('utf-8')

Toadds = [Toadd]

serveur.sendmail(Fromadd, Toadds, texte)

serveur.quit()