Version : **2024.01**.

Dernière mise-à-jour : 2024/11/30 10:45

LDF901 - Installation d'Ansible

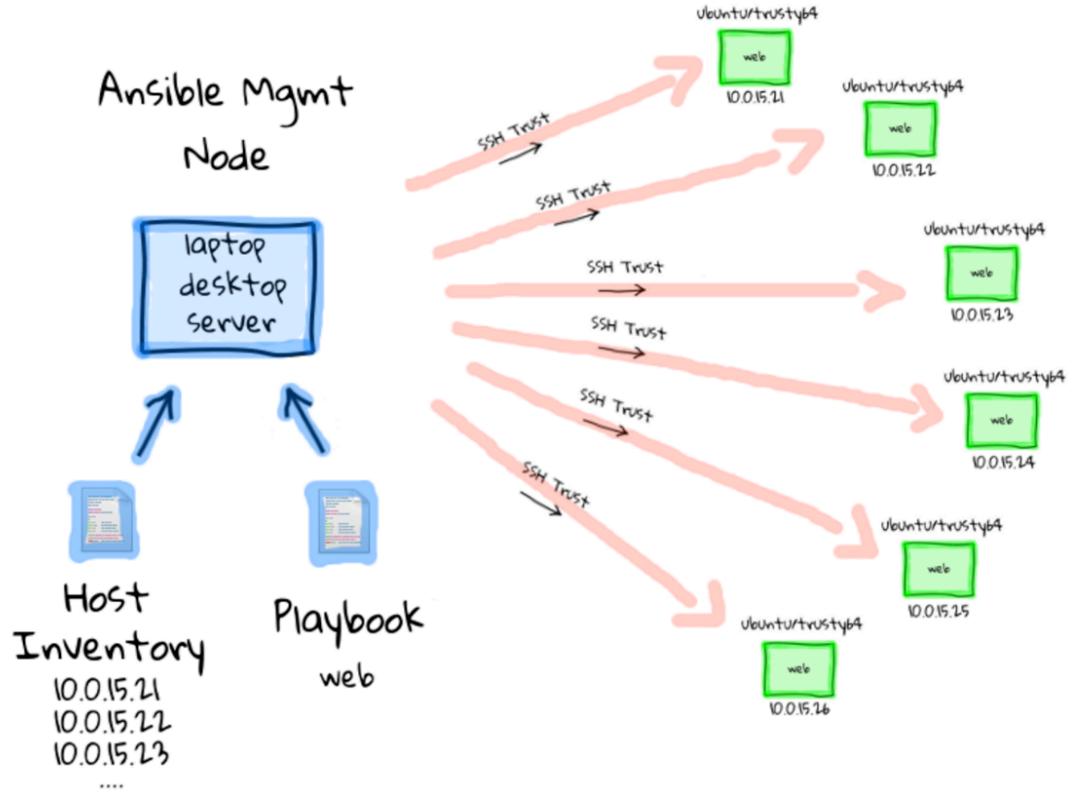
Contenu du Module

Qu'est-ce Ansible ?

Ansible est un outil d'automatisation qui permet d'automatiser les installations et les configurations répétitives de logiciels de manière fiable en utilisant une bibliothèque ré-utilisable d'instructions.

Ansible :

- est installé sur un **contrôleur** qui communique avec les machines cibles en utilisant le protocole **SSH**,
- ne nécessite ni l'utilisation d'un agent ni l'utilisation d'un service sur les cibles,
- utilise le langage **YAML** (**Y**et **A**other **M**arkup **L**anguage), plus simple que JSON, le code Ruby utilisé par **Chef** et le langage propriétaire de **Puppet**.



LAB #1 - Installation d'Ansible

Connectez-vous à la VM **Debian11_10.0.2.46_SSH** à partir de Guacamole.

Pour installer Ansible, il convient d'utiliser **apt** :

```
trainee@debian11:~$ su -
Password: fenestros

root@debian11:~# apt update
```

```
root@debian11:~# apt install software-properties-common

root@debian11:~# apt install ansible
```

Important : Notez que le mot de passe **fenestros** ne sera pas en clair.

Consultez la version d'Ansible que vous avez installé :

```
root@debian11:~# ansible --version
ansible 2.10.8
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.9.2 (default, Feb 28 2021, 17:03:44) [GCC 10.2.1 20210110]
```

LAB #2 - Configuration de ssh et de sudo

2.1 - ssh

Ansible a besoin d'une configuration par clef asymétrique avec la machine **targeta** afin de fonctionner correctement. Dans cette configuration :

- La machine **debian11** envoie à la machine **targeta** une requête d'authentification par clé asymétrique qui contient le module de la clé à utiliser,
- La machine **targeta** recherche une correspondance pour ce module dans le fichier des clés autorisés **~/.ssh/authorized_keys**,
  - Dans le cas où une correspondance n'est pas trouvée, la machine **targeta** met fin à la communication,
  - Dans le cas contraire la machine **targeta** génère une chaîne aléatoire de 256 bits appelée un **challenge** et la chiffre avec la **clé publique de la machine debian11**,
- La machine **debian11** reçoit le challenge et le décrypte avec la partie privée de sa clé. Il combine le challenge avec l'identifiant de session et chiffre le résultat. Ensuite il envoie le résultat chiffré à la machine **targeta**.

- La machine **targeta** génère le même haché et le compare avec celui reçu de la machine **debian11**. Si les deux hachés sont identiques, l'authentification est réussie.

Redevenez l'utilisateur **trainee** :

```
root@targeta:~# exit
déconnexion
```

Saisissez maintenant les commandes suivantes en tant que **trainee** dans la machine **debian11** :

Important - Lors de la génération des clefs, la passphrase doit être **vide**.

```
trainee@debian11:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_rsa):
Created directory '/home/trainee/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_rsa
Your public key has been saved in /home/trainee/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:RlCq7kYilWPKZ+NvAD9ZAhmBPAntKxPiZ7WX17WzDrM trainee@debian11
The key's randomart image is:
+---[RSA 3072]----+
|=+=   ...        |
|.*.    o         |
| .o.  . .        |
|o.*. + .     .   |
|+=oo* . S . . .  |
|=oo%.. + . . o   |
| +*o= . .  o  o  |
```

```
|   o..      +.   |
|   .+.     E..   |
+----[SHA256]-----+
trainee@debian11:~$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_ecdsa
Your public key has been saved in /home/trainee/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:MbPPmbvPZSn/dCbeqqFSTcOD4Nz0J42a191wNmMDUEU trainee@debian11
The key's randomart image is:
+---[ECDSA 256]---+
|           .o.oE |
|        . .  .   |
|       o++ + o.  |
|        o=o O +=o|
|        S  = *.==|
|         o+oo ..o|
|         .=.o = +|
|        .  + B =.|
|         .+o+.+oo|
+----[SHA256]-----+
trainee@debian11:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/trainee/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/trainee/.ssh/id_ed25519
Your public key has been saved in /home/trainee/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:sH8JHS05b+AhPR44YKTnjD2ho4GGz1Bz+VIOxrPZCeo trainee@debian11
The key's randomart image is:
```

```
+--[ED25519 256]--+
|     .+          |
|   . + . o o     |
|  o X = + @ .    |
|.o = ^ = * X     |
|+.o B @ S + o    |
|.= o o o . o     |
|  E     . o      |
|         .       |
|                 |
+----[SHA256]-----+
```

Important - Les clés générées seront placées dans le répertoire **~/.ssh/**.

Créez ensuite le fichier **~/.ssh/authorized_keys** :

```
trainee@debian11:~$ cat .ssh/id_rsa.pub >> .ssh/authorized_keys
trainee@debian11:~$ cat .ssh/id_ecdsa.pub >> .ssh/authorized_keys
trainee@debian11:~$ cat .ssh/id_ed25519.pub >> .ssh/authorized_keys
trainee@debian11:~$ cat .ssh/authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDGZkzJU19orBzrwmLpEzgNPsVnvZ/xYSDo4veMbSgTUs5EEtAXtF1rL+VvN7xBESf1y0b/JuXg/y0dF1MsJ
S3Tb+BuzJ65BkaZmf9PqkcDrq96YBmeagxbH3+oxopInL1WXu3EzEDBJUFj/Xn/PybPoM3odDBp6rCSnHylzo1C5MeoN1wB6V5/N2xTR0TLpG60Sm
huDsyvlmRltRrP3LXqORnuS93qOzRTzMuduZJ/Sjowpf4PS/HyXSIFnGV6LLyL9ubyVzUcwgrcQlbrhPDC9BHO8hYxu+Y6fZUL1kOx5Yds1S/I6r+
GUw8DUYUDmMWaggbigdofhMIYfRqmM4ERs/i3jftMOVIv0rO9a42TzP0g4XXwFJQsVcwOXi3D1oSQBFXqUmbz06i2V3e5kvmN/oW4RPADL70v7zU1
IRldo4nqHB8dLWinpYAgCTxzSXBpRKxNLRUPzWXta4F9zZZ0Zo0Ndav5agLV6ZlwY5sXXmFGUAvVko32NQiL3eo0mZ0= trainee@debian11
ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBBWSAQl6zAEmi/Nq9/0CZzb5pXeJqXKnqDPE6p0Vi6R+BP3GzLxXI1h6digVG
7siQ1ArNzrQqxkqvHd6QsdhdZQ= trainee@debian11
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBRchnTXPnQHlHVIUS2RPKCBn2ZKnyw3NLWE0CYsMcc6 trainee@debian11
```

Il convient maintenant de se connecter sur la machine **targeta** en utilisant ssh et de vérifier que le répertoire ~/.ssh existe :

```
trainee@debian11:~$ ssh -l trainee 10.0.2.52
The authenticity of host '10.0.2.52 (10.0.2.52)' can't be established.
ECDSA key fingerprint is SHA256:sEfHBv9azmK60cjqF/aJgUc9jg56slNaZQdAUcvBOvE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.52' (ECDSA) to the list of known hosts.
Debian GNU/Linux 9
trainee@10.0.2.52's password: trainee
Linux targeta.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Mar 21 08:47:45 2021 from 10.0.2.10

trainee@targeta:~$ ls -la | grep .ssh
drwx------  2 trainee trainee  4096 janv. 28  2019 .ssh
trainee@targeta:~$ exit
déconnexion
Connection to 10.0.2.52 closed.
```

Important : Notez que le mot de passe **trainee** ne sera pas en clair.

Important - Si le dossier distant .ssh n'existe pas dans le répertoire personnel de l'utilisateur connecté, il faut le créer avec des permissions de 700.

Ensuite, il convient de transférer le fichier **.ssh/authorized_keys** de la machine **debian11** vers la machine **targeta** :

```
trainee@debian11:~$ scp .ssh/authorized_keys trainee@10.0.2.52:/home/trainee/.ssh/authorized_keys
Debian GNU/Linux 9
trainee@10.0.2.52's password: trainee
authorized_keys                                   100%  846     1.8MB/s   00:00
```

Important : Notez que le mot de passe **trainee** ne sera pas en clair.

Connectez-vous via ssh de la machine **debian11** à la machine **targeta** :

```
trainee@debian11:~$ ssh -l trainee 10.0.2.52
Debian GNU/Linux 9
Linux targeta.i2tch.loc 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Sep 20 14:19:26 2023 from 10.0.2.46
trainee@targeta:~$
```

Important : Notez que l'authentification a utilisé le couple de clefs asymétrique et aucun mot de passe n'a été requis.

2.2 - sudo

La commande **sudo** permet à un utilisateur autorisé d'exécuter une commande en tant que **root** ou en tant qu'un autre utilisateur. Lors de l'exécution de la commande, l'UID et le GID effectifs et réels sont ceux de l'identité de l'utilisateur cible. L'utilisation de la commande **sudo** est une façon simple de déléguer des tâches administratives à d'autres utilisateurs sans communiquer le mot de passe de **root** et sans placer un SUID bit sur l'exécutable. La commande **sudo** est configurée grâce au fichier **/etc/sudoers**.

Consultez ensuite le fichier **/etc/sudoers** :

```
trainee@targeta:~$ su -
Mot de passe : fenestros

root@targeta:~# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
```

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

> **Important** : Notez la présence de la ligne **%sudo ALL=(ALL) ALL**. Cette ligne possède le format **Qui Où = (En tant que qui) Quoi**. La ligne implique donc que les membres du groupe **sudo** peuvent exécuter à partir de n'importe quel hôte et en tant que n'importe quel utilisateur, toutes les commandes du système. Dans ce fichier donc, un groupe est référencé par un **%**. Un nom sans ce caractère est forcément un utilisateur. Pour éditer le fichier **/etc/sudoers**, il est **nécessaire** d'utiliser la commande **visudo**.

Vérifez le contenu du fichier **/etc/sudoers.d/ansible_users** dans la VM **targeta** qui permettra à **trainee** d'utiliser la commande sudo **sans** entrer son mot de passe :

```
root@targeta:~# cat /etc/sudoers.d/ansible_users
trainee ALL=(ALL)       NOPASSWD:ALL

root@targeta:~# ls -l /etc/sudoers.d/ansible_users
-r--r----- 1 root root 31 janv. 29  2019 /etc/sudoers.d/ansible_users
```

Testez la prise en compte de votre configuration :

```
root@targeta:~# exit
déconnexion
trainee@targeta:~$ sudo su -
root@targeta:~# exit
déconnexion
```

Important : Notez que trainee a pu exécuter la commande **su -** via sudo sans saisir son mot de passe.
