Automatisation avec Ansible

Préparer la communication avec les nodes

24 octobre 2024

Table des matières

1 Préparer la communication avec les nodes

Dans le module précédent, on a fait connaissance avec Ansible, on a préparé notre architecture pour déployer MediaWiki sur 2 serveurs à l'aide d'un node manager, on a installé les outils Ansible sur le node manager et regardé d'un peu plus près les 3 outils Ansible dont on va avoir besoin par la suite.

Dans ce module, on va préparer la communication avec les nodes. C'est-à-dire qu'on va installer les prérequis pour que le node manager puisse communiquer avec les nodes et leur envoyer les commandes Ansible.

1.1 Préparer la communication

On va **déployer automatiquement** MediaWiki sur les 2 nodes, en utilisant Ansible et les scripts d'automatisation **depuis le node manager**.

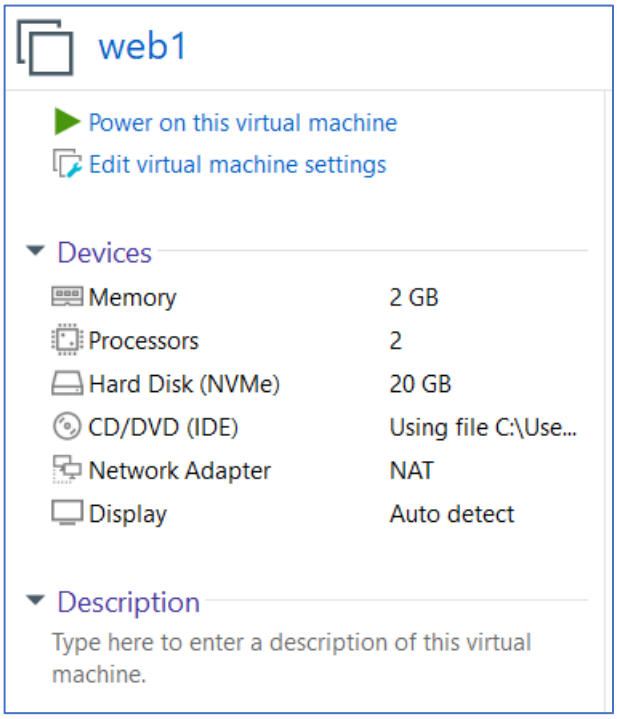
Ansible travaille avec des outils déjà très utilisés sur les systèmes Linux. Il besoin uniquement de **SSH** et de **Python** pour fonctionner. Il faut donc au minimum que ces 2 outils soient installés et fonctionnels sur le node manager et les nodes.

Pour établir la communication et avoir le droit de lancer des commandes à distance, on a besoin de:

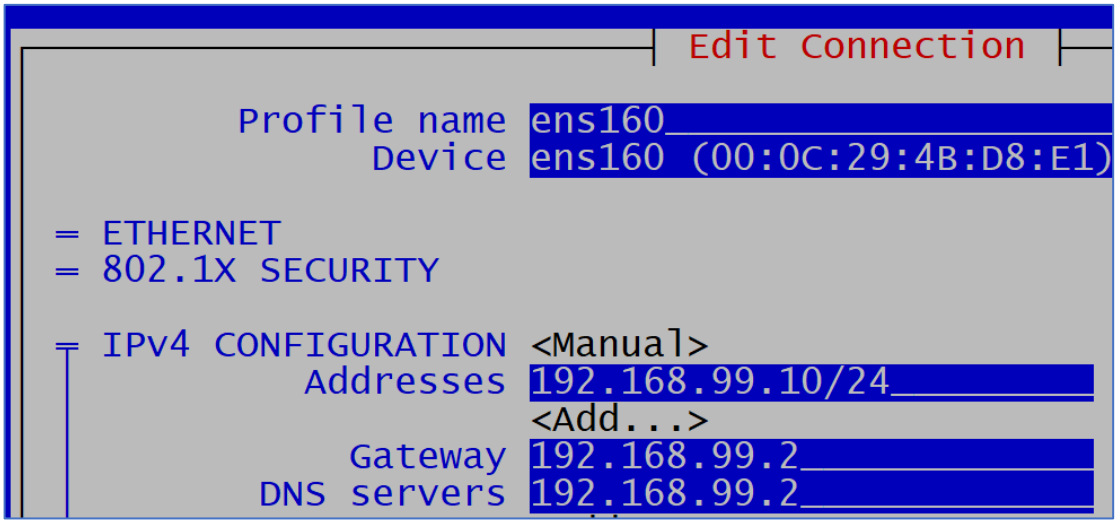
1) Préparer le premier node
2) Préparer le deuxième node.
3) Définir un **inventaire** des nodes.
4) Vérifier la **connexion SSH** entre le node manager et les nodes.
5) Lancer un **ping** avec Ansible sur les nodes.
6) Vérifier que **Python** est installé sur les nodes.
7) Créer un **simple utilisateur** sur les nodes.
8) Attribuer les **droits sudo** à cet utilisateur.
9) Créer une **paire de clés SSH** pour cet utilisateur.
10) Copier la **clé publique SSH** sur les nodes.

1.2 Préparer le premier node

1) Faire un clone lié du modèle Alma Linux et le nommer web1.



2) Attribuer une configuration statique à l'interface réseau de web1.

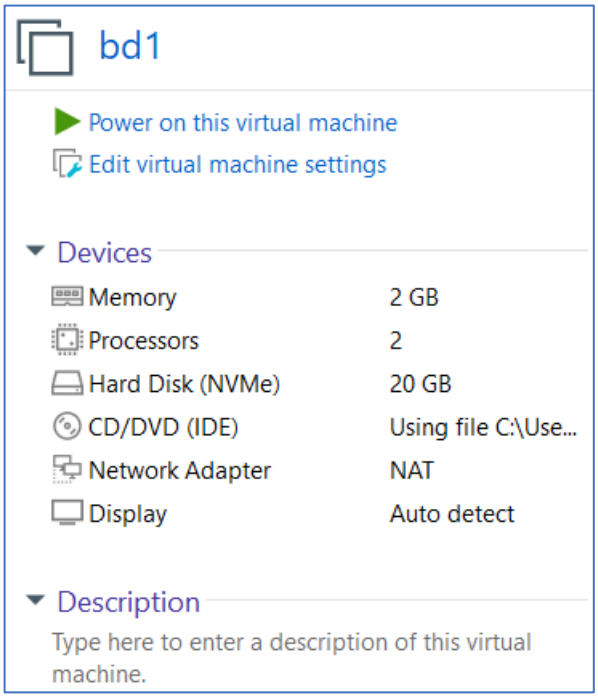


3) Attribuer le nom d'hôte web1.

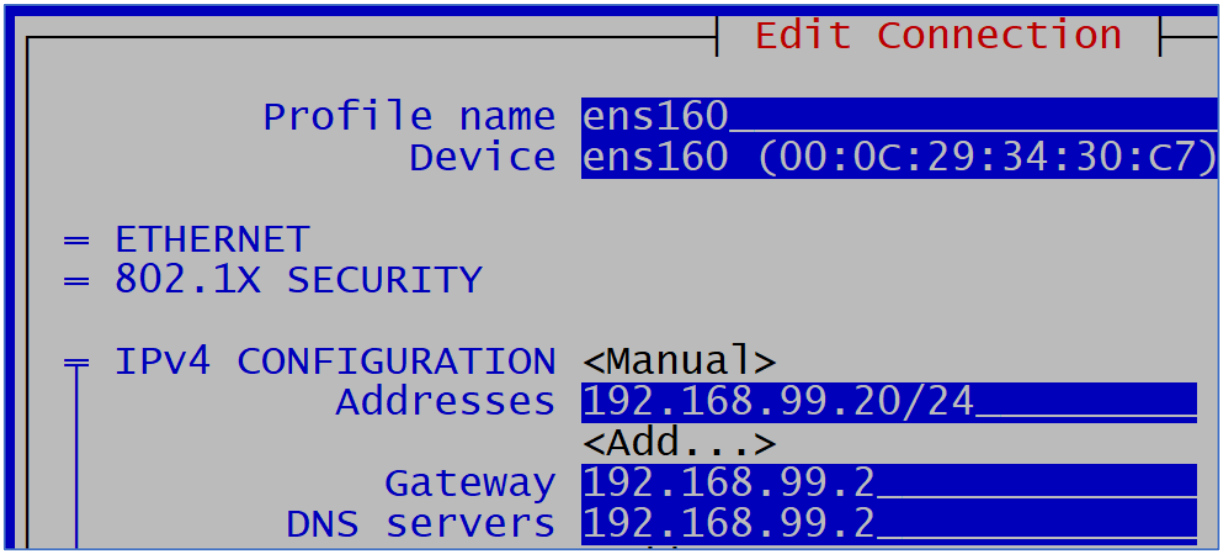
```
[root@localhost ~]# hostnamectl hostname web1
```

1.3 Préparer le deuxième node

4) Faire un clone lié du modèle Alma Linux et le nommer bd1.



5) Attribuer une configuration statique à l'interface réseau de web1.



6) Attribuer le nom d'hôte bd1.

```
[root@localhost ~]# hostnamectl hostname bd1
```

1.4 Définir l'inventaire des nodes

Il est temps de baptiser nos serveurs :

1) Sur le serveur 1 seront installés **Apache, PHP et MediaWiki**; nous l'appellerons donc **web1**.
2) Sur le serveur 2 sera installé **MariaDB** ; nous l'appellerons donc **bd1**.

Comme on n'a pas accès au **DNS** de l'entreprise, on va configurer la résolution de nom via le fichier **/etc/hosts** sur le node manager.

Dans la vraie vie, les noms des nodes sont enregistrer dans le DNS. On veut se concentrer sur la pratique de Ansible, on va prendre un raccourci en utilisant le fichier /etc/hosts. Ce fichier permet de mettre en place des correspondances entre les noms et leurs adresses IP.

Se connecter sur le node manager en **root** et le fichier /etc/hosts

```
root@debian:~# nano /etc/hosts
```

Ajouter dans le fichier **/etc/hosts** sur le node manager l'enregistrement des 2 nodes :

```
127.0.0.1 localhost node-manager
# le node web1
192.168.199.10 web1
# le node bd1
192.168.199.20 bd1
```

Cette action permettra au node manager d'utiliser les noms **web1** et **bd1** pour communiquer avec les nodes.

1.5 Créer le fichier inventaire Ansible

Pour fonctionner, Ansible a besoin d'un **fichier inventaire**. Ce fichier contient la liste des nodes. On va donc enregistrer le nom des nodes dans ce fichier.

```
root@debian:~# su - user-ansible
user-ansible@debian:~$
```

Activer l'environnement virtuel :

```
user-ansible@debian:~$ source ansible10.5.0/bin/activate
(ansible10.5.0) user-ansible@debian:~$
```

Créer le fichier **inventaire.ini**:

```
(ansible10.5.0) user-ansible@debian:~$ nano inventaire.ini
```

Mettre dans ce fichier les 2 noms des nodes :

```
web1
bd1
```

Le fichier **inventaire** est au **format INI** par défaut, il suit donc la syntaxe de ce format.

Ansible sait travailler avec d'autres formats de données, il suffit d'activer les bons plugins.

Voir la liste des plugins d'inventaire.

1.6 Vérifier la communication avec les nodes

Avant d'utiliser Ansible pour commencer à automatiser des tâches, on doit lancer une connexion **SSH** sur les nodes pour enregistrer la **fingerprint** (l'empreinte du serveur) sur le node manager.

```
(ansible10.5.0) user-ansible@debian:~$ ssh root@bd1
The authenticity of host 'bd1 (192.168.99.20)' can't be established.
ED25519 key fingerprint is
SHA256:0W46Id5elr1xr+tX3X4k9qKD6BuFuYOrhBCujGTh6iw.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'bd1' (ED25519) to the list of known hosts.
root@bd1's password:
Last login: Thu Oct 24 10:12:00 2024 from 192.168.99.1
[root@bd1 ~]#
```

Fermer la session ssh.

```
[root@bd1 ~]# exit
logout
Connection to bd1 closed.
(ansible10.5.0) user-ansible@debian:~$
```

Répondre **yes** et faire la même chose pour l'autre node:

```
(ansible10.5.0) user-ansible@debian:~$ ssh root@web1
The authenticity of host 'web1 (192.168.99.10)' can't be established.
ED25519 key fingerprint is
SHA256:0W46Id5elr1xr+tX3X4k9qKD6BuFuYOrhBCujGTh6iw.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'web1' (ED25519) to the list of known hosts.
root@web1's password:
Last login: Thu Oct 24 10:10:50 2024 from 192.168.99.1
[root@web1 ~]#
```

Fermer la session ssh.

```
[root@web1 ~]# exit
logout
Connection to web1 closed.
(ansible10.5.0) user-ansible@debian:~$
```

1.7 La commande ansible en mode ad-hoc

On va commencer à utiliser Ansible (en mode ad-hoc, c'est-à-dire avec des commandes manuelles plutôt que des scripts) pour mettre en place les prérequis. Ceci nous permettra de les automatiser et de les appliquer à tous les nodes en même temps.

Les commandes **ad-hoc** sont des actions rapides qui ne nécessitent de les sauvegarder pour plus tard. Se référer à [la documentation d'Ansible sur les commandes ad-hoc] pour approfondir.

Lancer maintenant un **ping** avec **Ansible** dans votre **environnement de travail virtuel** :

```
(ansible10.5.0) user-ansible@debian:~$ ansible -i inventaire.ini -m ping
web1 --user root --ask-pass
SSH password:
[WARNING]: Platform linux on host web1 is using the discovered Python
interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter
could change the
meaning of that path. See https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more
information.
web1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
(ansible10.5.0) user-ansible@debian:~$
```

On vient d'utiliser la commande **ansible** avec plusieurs options :

-i	Indique à Ansible l'emplacement du fichier **i**nventaire
-m	Indique à Ansible d'utiliser le module ping
web1	Indique à Ansible de faire l'action sur le node **tweb1**
--user	Indique à Ansible d'utiliser l'utilisateur root pour se connecter au node (pas le choix pour le moment, car c'est le seul compte dont on dispose)
--ask-pass	Indique à Ansible de demander le **mot de passe SSH**

Le retour de la commande nous indique que l'action est un succès et répond **pong** au **ping** ! Le node web1 est bien joignable.

Ansible ne lance pas la commande ping, il lance un module qui fait la même chose que la commande ping.

Module

Un **module** est un programme utilisé pour exécuter une tâche ou une commande Ansible. Chaque tâche utilise **un seul module**, qui peut prendre des arguments pour être exécuté de manière personnalisée. Ansible fournit de nombreux modules, mais on peut créer le vôtre.

Tous les modules sont accessibles sur **la documentation d'Ansible** ou avec la commande :

```
(ansible10.5.0) user-ansible@debian:~$ ansible-doc --list
amazon.aws.autoscaling_group                                                >
amazon.aws.autoscaling_group_info                                           >
amazon.aws.aws_az_info                                                      >
amazon.aws.aws_caller_info                                                  >
amazon.aws.aws_region_info                                                  >
amazon.aws.backup_plan                                                      >
amazon.aws.backup_plan_info                                                 >
amazon.aws.backup_restore_job_info                                          >
amazon.aws.backup_selection                                                 >
amazon.aws.backup_selection_info                                            >
amazon.aws.backup_tag                                                       >
amazon.aws.backup_tag_info                                                  >
amazon.aws.backup_vault                                                     >
amazon.aws.backup_vault_info                                                >
amazon.aws.cloudformation                                                   >
amazon.aws.cloudformation_info                                              >
amazon.aws.cloudtrail                                                       >
amazon.aws.cloudtrail_info                                                  >
amazon.aws.cloudwatch_metric_alarm                                          >
amazon.aws.cloudwatch_metric_alarm_info                                     >
amazon.aws.cloudwatchevent_rule                                             >
amazon.aws.cloudwatchlogs_log_group                                         >
amazon.aws.cloudwatchlogs_log_group_info                                    >
amazon.aws.cloudwatchlogs_log_group_metric_filter                           >
amazon.aws.ec2_ami                                                          >
amazon.aws.ec2_ami_info                                                     >
amazon.aws.ec2_eip                                                          >
amazon.aws.ec2_eip_info                                                     >
amazon.aws.ec2_eni                                                          >
:
```

Tous les modules officiels d'Ansible sont téléchargés sur le node manager lors de l'installation d'Ansible. Lorsqu'on utilise un module, Ansible ira chercher le code à exécuter dans le dossier du module.

Si on a installé Ansible dans un environnement virtuel avec pip, les modules se trouvent dans le dossier:

/home/user-ansible/ansible10.5.0/lib/python3.11/site-packages/ansible/modules

La syntaxe et les options de la commande ansible sont disponibles sur la documentation d'Ansible.

1.8 Vérifier que Python est installé sur les nodes

Il arrive parfois que **Python** ne soit pas installé sur le node; dans ce cas, on peut utiliser un **module spécial : raw**, qui permet de passer des commandes Ansible sans utiliser **Python**:

```
(ansible10.5.0) user-ansible@debian:~$ ansible -i inventaire.ini -m raw -a "
dnf install -y python3.12" all --user root --ask-pass
SSH password:
web1 | CHANGED | rc=0 >>
Last metadata expiration check: 0:03:46 ago on Thu Oct 24 10:33:59 2024.
Dependencies resolved.
================================================================================
 Package                  Arch       Version              Repository     Size
================================================================================
Installing:
 python3.12               x86_64     3.12.1-4.el9_4.3     appstream      25 k
Installing dependencies:
 libnsl2                  x86_64     2.0.0-1.el9          appstream      30 k
 libtirpc                 x86_64     1.3.3-8.el9_4        baseos         93 k
 mpdecimal                x86_64     2.5.1-3.el9          appstream      85 k
 python3.12-libs          x86_64     3.12.1-4.el9_4.3     appstream     9.0 M
 python3.12-pip-wheel     noarch     23.2.1-4.el9         appstream     1.5 M

Transaction Summary
================================================================================
Install  6 Packages

Total download size: 11 M
Installed size: 44 M
Downloading Packages:
(1/6): libnsl2-2.0.0-1.el9.x86_64.rpm           138 kB/s |  30 kB     00:00
(2/6): mpdecimal-2.5.1-3.el9.x86_64.rpm         360 kB/s |  85 kB     00:00
(3/6): python3.12-3.12.1-4.el9_4.3.x86_64.rpm   107 kB/s |  25 kB     00:00
(4/6): libtirpc-1.3.3-8.el9_4.x86_64.rpm        521 kB/s |  93 kB     00:00
(5/6): python3.12-pip-wheel-23.2.1-4.el9.noarch 933 kB/s | 1.5 MB     00:01
(6/6): python3.12-libs-3.12.1-4.el9_4.3.x86_64. 2.0 MB/s | 9.0 MB     00:04
--------------------------------------------------------------------------------
Total                                           2.1 MB/s |  11 MB     00:05
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : libtirpc-1.3.3-8.el9_4.x86_64                          1/6
  Installing       : libnsl2-2.0.0-1.el9.x86_64                             2/6
  Installing       : python3.12-pip-wheel-23.2.1-4.el9.noarch               3/6
  Installing       : mpdecimal-2.5.1-3.el9.x86_64                           4/6
  Installing       : python3.12-3.12.1-4.el9_4.3.x86_64                     5/6
  Installing       : python3.12-libs-3.12.1-4.el9_4.3.x86_64                6/6
  Running scriptlet: python3.12-libs-3.12.1-4.el9_4.3.x86_64                6/6
  Verifying        : libnsl2-2.0.0-1.el9.x86_64                             1/6
  Verifying        : mpdecimal-2.5.1-3.el9.x86_64                           2/6
```

```
  Verifying        : python3.12-3.12.1-4.el9_4.3.x86_64                3/6
  Verifying        : python3.12-libs-3.12.1-4.el9_4.3.x86_64           4/6
  Verifying        : python3.12-pip-wheel-23.2.1-4.el9.noarch          5/6
  Verifying        : libtirpc-1.3.3-8.el9_4.x86_64                     6/6

Installed:
  libnsl2-2.0.0-1.el9.x86_64
  libtirpc-1.3.3-8.el9_4.x86_64
  mpdecimal-2.5.1-3.el9.x86_64
  python3.12-3.12.1-4.el9_4.3.x86_64
  python3.12-libs-3.12.1-4.el9_4.3.x86_64
  python3.12-pip-wheel-23.2.1-4.el9.noarch

Complete!
Shared connection to web1 closed.

bd1 | CHANGED | rc=0 >>
Last metadata expiration check: 0:07:15 ago on Thu Oct 24 10:30:30 2024.
Dependencies resolved.
================================================================================
 Package                  Arch       Version              Repository     Size
================================================================================
Installing:
 python3.12               x86_64     3.12.1-4.el9_4.3     appstream      25 k
Installing dependencies:
 libnsl2                  x86_64     2.0.0-1.el9          appstream      30 k
 libtirpc                 x86_64     1.3.3-8.el9_4        baseos         93 k
 mpdecimal                x86_64     2.5.1-3.el9          appstream      85 k
 python3.12-libs          x86_64     3.12.1-4.el9_4.3     appstream     9.0 M
 python3.12-pip-wheel     noarch     23.2.1-4.el9         appstream     1.5 M

Transaction Summary
================================================================================
Install  6 Packages

Total download size: 11 M
Installed size: 44 M
Downloading Packages:
(1/6): mpdecimal-2.5.1-3.el9.x86_64.rpm          80 kB/s |  85 kB     00:01
(2/6): python3.12-3.12.1-4.el9_4.3.x86_64.rpm    23 kB/s |  25 kB     00:01
(3/6): libnsl2-2.0.0-1.el9.x86_64.rpm            25 kB/s |  30 kB     00:01
(4/6): libtirpc-1.3.3-8.el9_4.x86_64.rpm        123 kB/s |  93 kB     00:00
(5/6): python3.12-pip-wheel-23.2.1-4.el9.noarch 458 kB/s | 1.5 MB     00:03
(6/6): python3.12-libs-3.12.1-4.el9_4.3.x86_64. 1.6 MB/s | 9.0 MB     00:05
--------------------------------------------------------------------------------
Total                                           1.5 MB/s |  11 MB     00:06
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                    1/1
  Installing       : libtirpc-1.3.3-8.el9_4.x86_64                      1/6
  Installing       : libnsl2-2.0.0-1.el9.x86_64                         2/6
```

```
  Installing       : python3.12-pip-wheel-23.2.1-4.el9.noarch           3/6
  Installing       : mpdecimal-2.5.1-3.el9.x86_64                       4/6
  Installing       : python3.12-3.12.1-4.el9_4.3.x86_64                 5/6
  Installing       : python3.12-libs-3.12.1-4.el9_4.3.x86_64            6/6
  Running scriptlet: python3.12-libs-3.12.1-4.el9_4.3.x86_64            6/6
  Verifying        : libnsl2-2.0.0-1.el9.x86_64                         1/6
  Verifying        : mpdecimal-2.5.1-3.el9.x86_64                       2/6
  Verifying        : python3.12-3.12.1-4.el9_4.3.x86_64                 3/6
  Verifying        : python3.12-libs-3.12.1-4.el9_4.3.x86_64            4/6
  Verifying        : python3.12-pip-wheel-23.2.1-4.el9.noarch           5/6
  Verifying        : libtirpc-1.3.3-8.el9_4.x86_64                      6/6

Installed:
  libnsl2-2.0.0-1.el9.x86_64
  libtirpc-1.3.3-8.el9_4.x86_64
  mpdecimal-2.5.1-3.el9.x86_64
  python3.12-3.12.1-4.el9_4.3.x86_64
  python3.12-libs-3.12.1-4.el9_4.3.x86_64
  python3.12-pip-wheel-23.2.1-4.el9.noarch

Complete!
Shared connection to bd1 closed.

(ansible10.5.0) user-ansible@debian:~$
```

On a utilisé l'option **-m** pour appeler le module **raw** avec l'argument **"dnf install -y python39"**, qui a pour effet d'installer Python en version 3.9 sur les deux nodes **web1** et **bd1**.

En réalité, les modules Ansible peuvent être écrits avec un autre langage que Python. Le module **raw**, écrit en Shell, en est la preuve ! Un atout supplémentaire en faveur d'Ansible.

1.9 Créer l'utilisateur user-ansible sur les nodes

On va suivre les bonnes pratiques en créant un **simple utilisateur** pour ne pas travailler directement avec le compte **root**.

Il est déconseillé d'utiliser le compte **root** directement pour faire l'ensemble des opérations. Il faut plutôt privilégier de travailler avec un simple utilisateur et lui donner les droits **sudo**.

Les commandes lancées avec sudo se feront avec un mot de passe préalable et peuvent être limitées à certaines actions. De plus, le compte utilisateur étant nominatif, il sera beaucoup plus facile de tracer les actions dans les logs pour debugger.

Installer le module passlib pour pouvoir générer un mot de passe chiffré.

```
(ansible10.5.0) user-ansible@debian:~$ pip install --break-system-packages
passlib
Collecting passlib
  Downloading passlib-1.7.4-py2.py3-none-any.whl (525 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
525.6/525.6 kB 3.4 MB/s eta 0:00:00
Installing collected packages: passlib
Successfully installed passlib-1.7.4
(ansible10.5.0) user-ansible@debian:~$
```

Mais avant de créer un utilisateur, on va générer un **mot de passe chiffré** (au format reconnu par Linux) avec Ansible :

```
(ansible10.5.0) user-ansible@debian:~$ ansible localhost -i inventaire.ini -
m debug -a "msg={{ 'secret' | password_hash('sha512', 'sel') }}"
localhost | SUCCESS => {
    "msg":
"$6$rounds=656000$sel$u8PTCwwR06J5c2/64S93ncA1qnt.MbTBy2bs7lCNU.nYzdBXFBVgjZ
HNsFYvCe4r0gXbYn4IKRxlci/svmI5Z."
```

Dans cette commande, on a utilisé le module **debug** avec l'argument **msg** pour transformer le **mot de passe** "secret" en une **chaîne chiffrée** avec l'algorithme **sha512**. **"**sel**"** est ce qu'on appelle le ***sel***. C'est un mot qui permet de renforcer la sécurité de l'algorithme en apportant une inconnue en plus dans le processus de cryptage du mot de passe.

L'option **"localhost"** a permis d'indiquer à Ansible de lancer la commande sur **localhost** (en local sur le node manager).

On va maintenant créer l'utilisateur **user-ansible** avec un mot de passe chiffré, grâce à la commande suivante :

```
(ansible10.5.0) user-ansible@debian:~$ ansible -i inventaire.ini -m user -a 'name=user-ansible password=$6$rounds=656000$sel$u8PTCwwR06J5c2/64S93ncA1qnt.MbTBy2bs7lCNU.nYzdBXFBVgjZHNsFYvCe4r0gXbYn4IKRxlci/svmI5Z.' --user root --ask-pass all
SSH password:
[WARNING]: Platform linux on host web1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
web1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1000,
    "home": "/home/user-ansible",
    "name": "user-ansible",
    "password": "NOT_LOGGING_PASSWORD",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 1000
}
[WARNING]: Platform linux on host bd1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
bd1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1000,
    "home": "/home/user-ansible",
    "name": "user-ansible",
    "password": "NOT_LOGGING_PASSWORD",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 1000
}
(ansible10.5.0) user-ansible@debian:~$
```

On a utilisé le module **user** avec les arguments **name** et **password**, et demandé à Ansible de lancer la commande sur **all** (tous les nodes présents dans le fichier inventaire).

1.10 Donner les droits sudo à user-ansible

Se Connecter au node **web1**, et consulter le fichier **/etc/sudoers**; il contient les configurations de **sudo**. On peut voir la ligne suivante :

```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL
```

Le groupe **wheel** a les droits **sudo**. On va donc ajouter l'utilisateur **user-ansible** dans le groupe **wheel** sur tous les nodes :

```
(ansible10.5.0) user-ansible@debian:~$ ansible -i inventaire.ini -m user -a 'name=user-ansible groups=wheel append=yes' --user root --ask-pass all
SSH password:
[WARNING]: Platform linux on host bd1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
bd1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "append": true,
    "changed": true,
    "comment": "",
    "group": 1000,
    "groups": "wheel",
    "home": "/home/user-ansible",
    "move_home": false,
    "name": "user-ansible",
    "shell": "/bin/bash",
    "state": "present",
    "uid": 1000
}
[WARNING]: Platform linux on host web1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
web1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "append": true,
    "changed": true,
    "comment": "",
    "group": 1000,
    "groups": "wheel",
    "home": "/home/user-ansible",
    "move_home": false,
    "name": "user-ansible",
    "shell": "/bin/bash",
    "state": "present",
    "uid": 1000
}
```

```
(ansible10.5.0) user-ansible@debian:~$
```

Pour valider :

```
(ansible10.5.0) user-ansible@debian:~$ ansible -i inventaire.ini -m raw -a "id user-ansible" all --user root --ask-pass
SSH password:
bd1 | CHANGED | rc=0 >>
uid=1000(user-ansible) gid=1000(user-ansible) groups=1000(user-ansible),10(wheel)
Shared connection to bd1 closed.

web1 | CHANGED | rc=0 >>
uid=1000(user-ansible) gid=1000(user-ansible) groups=1000(user-ansible),10(wheel)
Shared connection to web1 closed.
```

Un moyen pour vérifier que **user-ansible** a bien les droits **sudo** est de relancer la commande précédente, mais en ajoutant de nouvelles options :

```
(ansible10.5.0) user-ansible@debian:~$ ansible -i inventaire.ini -m user -a 'name=user-ansible groups=wheel append=yes' --user user-ansible --ask-pass --become --ask-become-pass all
SSH password:
BECOME password[defaults to SSH password]:
[WARNING]: Platform linux on host web1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
web1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "append": true,
    "changed": false,
    "comment": "",
    "group": 1000,
    "groups": "wheel",
    "home": "/home/user-ansible",
    "move_home": false,
    "name": "user-ansible",
    "shell": "/bin/bash",
    "state": "present",
    "uid": 1000
}
[WARNING]: Platform linux on host bd1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
bd1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "append": true,
    "changed": false,
```

```
    "comment": "",
    "group": 1000,
    "groups": "wheel",
    "home": "/home/user-ansible",
    "move_home": false,
    "name": "user-ansible",
    "shell": "/bin/bash",
    "state": "present",
    "uid": 1000
}
(ansible10.5.0) user-ansible@debian:~$
```

Le retour est un **SUCCESS**

On a utilisé le module **user** pour ajouter **user-ansible** dans le groupe **wheel** et utilisé deux nouvelles options :

--become	Ansible nous permet de "devenir" un autre utilisateur en utilisant sudo
--ask-become-pass	Ansible demande le mot de passe sudo (qui est le même que le mot de passe de user-ansible)

Dorénavant, on utilisera **user-ansible** en mode **sudo** pour passer les commandes **Ansible**.

1.11 Créer les clés SSH

Pour se connecter en **SSH**, il est recommandé d'utiliser une **paire de clés** plutôt que d'utiliser un **mot de passe**.

La communication **SSH** est établie sur la base de **clés SSH**; cette pratique est conseillée, car elle permet un niveau d'authentification beaucoup plus sûr que l'authentification par mot de passe.

On va maintenant créer une paire de **clés SSH** de type **ecdsa** pour l'utilisateur **user-ansible**.

Elliptic Curve Digital Signature Algorithm (**ECDSA**) est un algorithme de signature numérique qu'il est conseillé d'utiliser aujourd'hui pour avoir un niveau de sécurité satisfaisant.

On commence par passer à l'utilisateur **user-ansible** avec la commande suivante :

```
root@debian:~# su - user-ansible
user-ansible@debian:~$
```

Lancer la création des clés avec la commande suivante :

```
user-ansible@debian:~$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/user-ansible/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user-ansible/.ssh/id_ecdsa
Your public key has been saved in /home/user-ansible/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:RyjnocHG/dnZcPdJ9eZPOO7g6cOLL9Obv/zafiJmNjI user-ansible@debian
The key's randomart image is:
+---[ECDSA 256]---+
|                 .|
|     o . .      o|
|      * = . . ..+|
|     . * + o =.=o|
|      . S + o +.+|
|         .   . o.|
|           o. . .|
|          EoX*...|
|          .@OB**=|
+----[SHA256]-----+
user-ansible@debian:~$
```

On laisse tout par défaut, sans rien changer. Il n'est pas nécessaire de mettre de **passphrase**. La **passphrase** est utilisée pour renforcer la connexion SSH par un mot de passe.

1.12 Ajouter la clé publique de l'utilisateur user-ansible sur les nodes

On va utiliser le module **authorized_key** pour enregistrer la clé publique de l'utilisateur **user-ansible** sur tous les nodes. Il sera possible par la suite de se connecter aux nodes sans saisir de mot de passe **SSH**.

```
(ansible10.5.0) user-ansible@debian:~$ ansible -i inventaire.ini -m
authorized_key -a 'user=user-ansible state=present key="{{ lookup("file",
"/home/user-ansible/.ssh/id_ecdsa.pub") }}"' --user user-ansible --ask-pass
--become --ask-become-pass all
SSH password:
BECOME password[defaults to SSH password]:
[WARNING]: Platform linux on host bd1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
bd1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": true,
    "comment": null,
    "exclusive": false,
    "follow": false,
    "key":                                                    "ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPLpEf87Tpe9QBQOaXrB+ouBtAMlEmH1
91H/jmVSaE2gw+HOCykJDwK6LoOp7slRsBXjFbqNP5HYHX7p0wNU7p4= user-ansible@debian",
    "key_options": null,
    "keyfile": "/home/user-ansible/.ssh/authorized_keys",
    "manage_dir": true,
    "path": null,
    "state": "present",
    "user": "user-ansible",
    "validate_certs": true
}
[WARNING]: Platform linux on host web1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
web1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": true,
    "comment": null,
    "exclusive": false,
    "follow": false,
    "key":                                                    "ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPLpEf87Tpe9QBQOaXrB+ouBtAMlEmH1
91H/jmVSaE2gw+HOCykJDwK6LoOp7slRsBXjFbqNP5HYHX7p0wNU7p4= user-ansible@debian",
    "key_options": null,
    "keyfile": "/home/user-ansible/.ssh/authorized_keys",
    "manage_dir": true,
    "path": null,
    "state": "present",
```

```
    "user": "user-ansible",
    "validate_certs": true
}
(ansible10.5.0) user-ansible@debian:~$
```

On a utilisé de nouvelles options :

-m authorized_key	module authorized_key(ajoute ou supprime les clés SSH pour des utilisateurs)
user=user-ansible	l'utilisateur concerné est user-ansible
state=present	indique d'ajouter le fichier
key="{{ lookup("file", "/home/user-ansible/.ssh/id_ecdsa.pub") }}"	utilise la commande lookup pour rechercher le fichier concerné.

Relancer la commande mais cette fois sans **--ask-pass** (demande le mot de pass SSH) :

```
(ansible10.5.0) user-ansible@debian:~$ ansible -i inventaire.ini -m authorized_key -a 'user=user-ansible state=present key="{{ lookup("file", "/home/user-ansible/.ssh/id_ecdsa.pub") }}"' --user user-ansible --become --ask-become-pass all
BECOME password:
[WARNING]: Platform linux on host bd1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
bd1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": false,
    "comment": null,
    "exclusive": false,
    "follow": false,
    "key": "ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPLpEf87Tpe9QBQOaXrB+ouBtAMlEmH1
91H/jmVSaE2gw+HOCykJDwK6LoOp7slRsBXjFbqNP5HYHX7p0wNU7p4= user-ansible@debian",
    "key_options": null,
    "keyfile": "/home/user-ansible/.ssh/authorized_keys",
    "manage_dir": true,
    "path": null,
    "state": "present",
    "user": "user-ansible",
    "validate_certs": true
}
[WARNING]: Platform linux on host web1 is using the discovered Python
interpreter at /usr/bin/python3.12, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more information.
web1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": false,
    "comment": null,
    "exclusive": false,
    "follow": false,
    "key": "ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPLpEf87Tpe9QBQOaXrB+ouBtAMlEmH1
91H/jmVSaE2gw+HOCykJDwK6LoOp7slRsBXjFbqNP5HYHX7p0wNU7p4= user-ansible@debian",
    "key_options": null,
    "keyfile": "/home/user-ansible/.ssh/authorized_keys",
    "manage_dir": true,
    "path": null,
    "state": "present",
    "user": "user-ansible",
    "validate_certs": true
}
(ansible10.5.0) user-ansible@debian:~$
```

Un seul mot de passe a été demandé, celui de sudo uniquement.