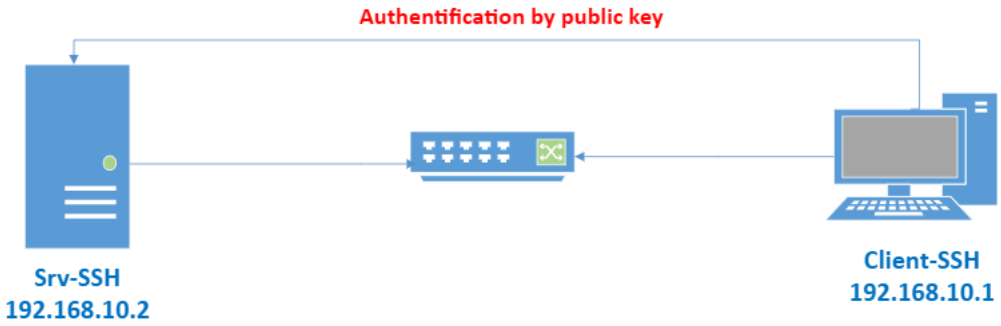
Ameth fall

Administration systéme linux

Dans le cadre de ce mini-projet portant sur SSH, l'objectif est de mettre en place un système d'authentification chiffré basé sur une clé publique.
Le but recherché est de garantir que seule la machine détentrice de la clé privée correspondante puisse se connecter de manière sécurisée à notre serveur 192.168.10.2

Architecture :



Authentification par clé publique :

Vérification du paquet openssh-server :

```
[root@srv-prod ameth]# yum install openssh-server
Dernière vérification de l'expiration des métadonnées effectuée il y a 0:23:23 le mer. 14 août 2
024 23:00:32.
Le paquet openssh-server-8.7p1-43.el9.x86_64 est déjà installé.
Dépendances résolues.
Rien à faire.
Terminé !
[root@srv-prod ameth]#
```

A. Configuration du serveur :

- **Générons une clé RSA privée pour le srv-prod :**

```
[root@srv-prod ~]# ssh-keygen -b 2048 -t rsa -f /etc/ssh/ssh_host_rsa_key
Generating public/private rsa key pair.
/etc/ssh/ssh_host_rsa_key already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /etc/ssh/ssh_host_rsa_key
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub
The key fingerprint is:
SHA256:WGNVIkdZN1LNu+6joiqQZ+fQVettCp/qogBgQjsNuF8 root@srv-prod
The key's randomart image is:
+---[RSA 2048]----+
|.o       ..=++.+o |
|o +       +.o o .o|
|o= .     + . .   .|
|= . E  + o .    . |
| o .. o S . .    .|
|  oo + o . . o . |
|   .+ +    o + .  |
|    .. o    =    o |
|      .o.++o ..o..|
+----[SHA256]-----+
[root@srv-prod ~]#
```

- **Configuration des paramètres du serveur au niveau du fichier /etc/ssh/sshd_config :**

Nous allons définir le port d'écoute du service SSH à 22 et désactiver l'authentification par mot de passe.

```
  GNU nano 5.6.1                  /etc/ssh/sshd_config                  Modifié
#
Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
```

- Activation de l'authentification par clé public **PubKeyAuthentification= YES**

```
#MaxSessions 10

PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_k
# but this is overridden so installations will only check .ssh/authorize
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hos
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
```

- Désactivation de l'authentification par mot de passe :

```
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

- Redémarrage du service démon sshd pour recharger la nouvelle configuration :

```
[root@srv-prod ameth]# systemctl reload sshd
[root@srv-prod ameth]#
```

B. Configuration du client :

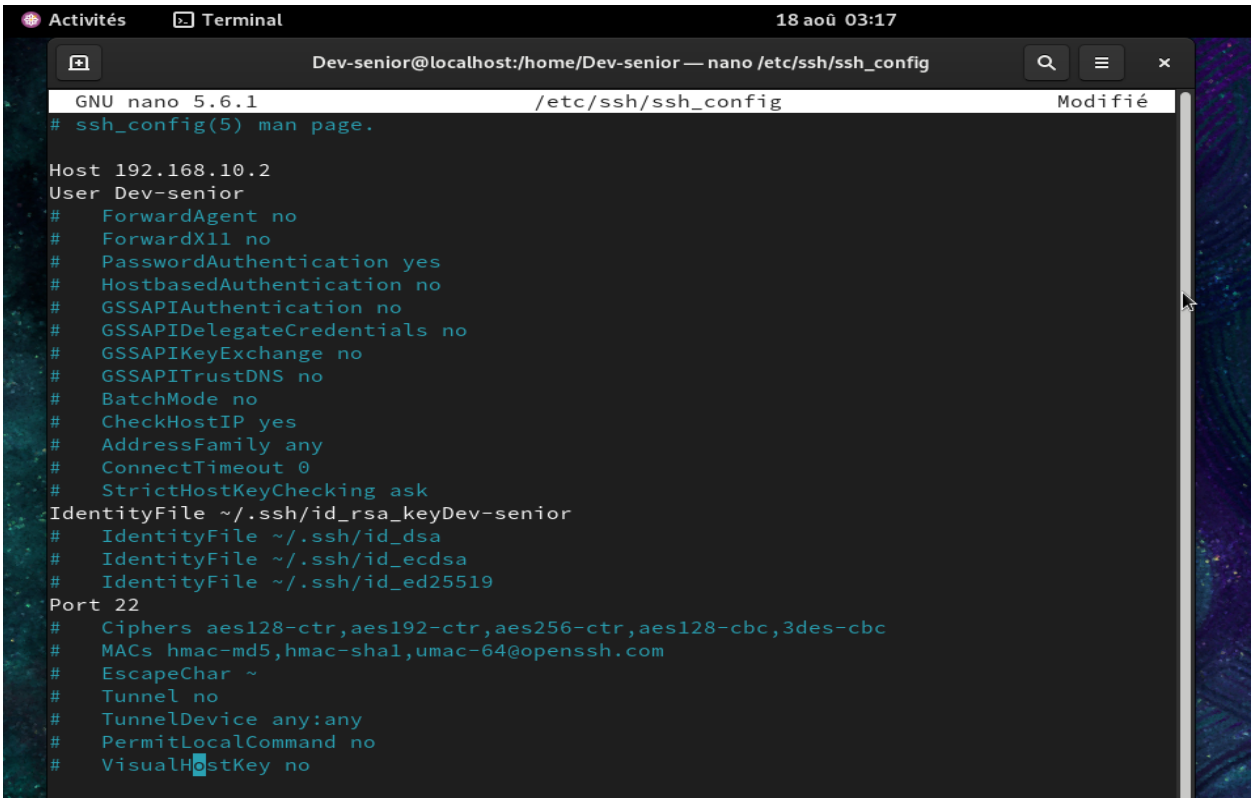
- Ajustement des permissions sur le dossier ssh/ :

```
[Dev-senior@Poste-1 ~]$ mkdir .ssh/
[Dev-senior@Poste-1 ~]$ chmod 700 .ssh/
[Dev-senior@Poste-1 ~]$
```

- Générons une paire de clés pour notre machine cliente dans le fichier **id_rsa_keyrDev-senior :**

```
[Dev-senior@localhost ~]$ ssh-keygen -b 2048 -t rsa -f .ssh/id_rsa_keyDev-senior
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in .ssh/id_rsa_keyDev-senior
Your public key has been saved in .ssh/id_rsa_keyDev-senior.pub
The key fingerprint is:
SHA256:SrrXcAP0s0pysdxWt8eaSFQ7tEpmnEfEZucC5aIFi7A Dev-senior@localhost.localdomain
The key's randomart image is:
+---[RSA 2048]----+
|     .   .  +o   |
|      o.. o..B .  |
|     E.....oO.=   |
|       o ooO.B .  |
|      ..=SO + =   |
|     .o*.* o . o  |
|     .+.* o . +   |
|      .o . . o    |
|     ..           |
+----[SHA256]-----+
[Dev-senior@localhost ~]$
```

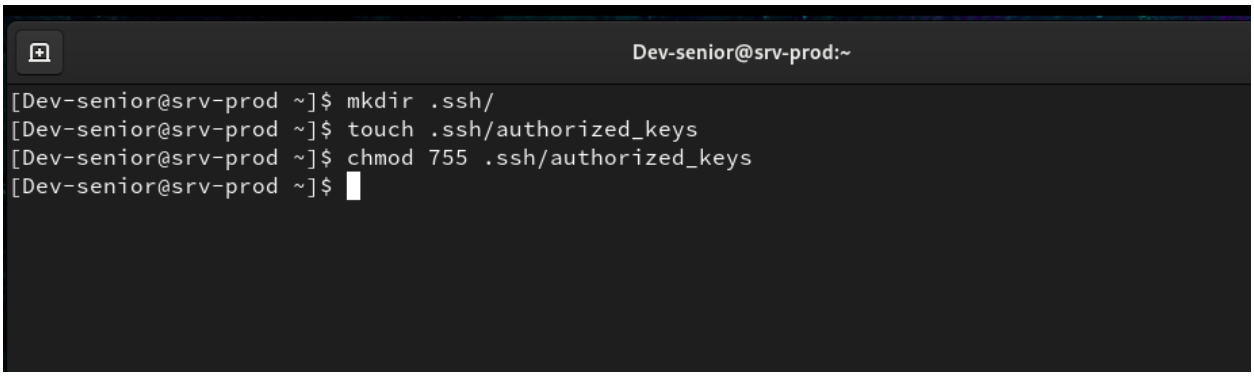
- Création d’un profil de connexion pour l’utilisateur Dev-senior pour le serveur 192.168.10.2 avec sa clé privée qui se trouve dans le fichier id_rsa_keyDev-senior.



- Ajout d’un compte utilisateur **Dev-senior** sur le serveur afin qu’il puisse s’authentifier depuis une autre machine via **SSH** :

```
[root@srv-prod ~]# adduser -m Dev-senior
[root@srv-prod ~]# passwd Dev-senior
Changement de mot de passe pour l'utilisateur Dev-senior.
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mise à jour réussie de tous les jetons d'authentification.
[root@srv-prod ~]#
```

- Création du répertoire **authorized.keys** pour y mettre les clés publiques autorisé par le serveur :

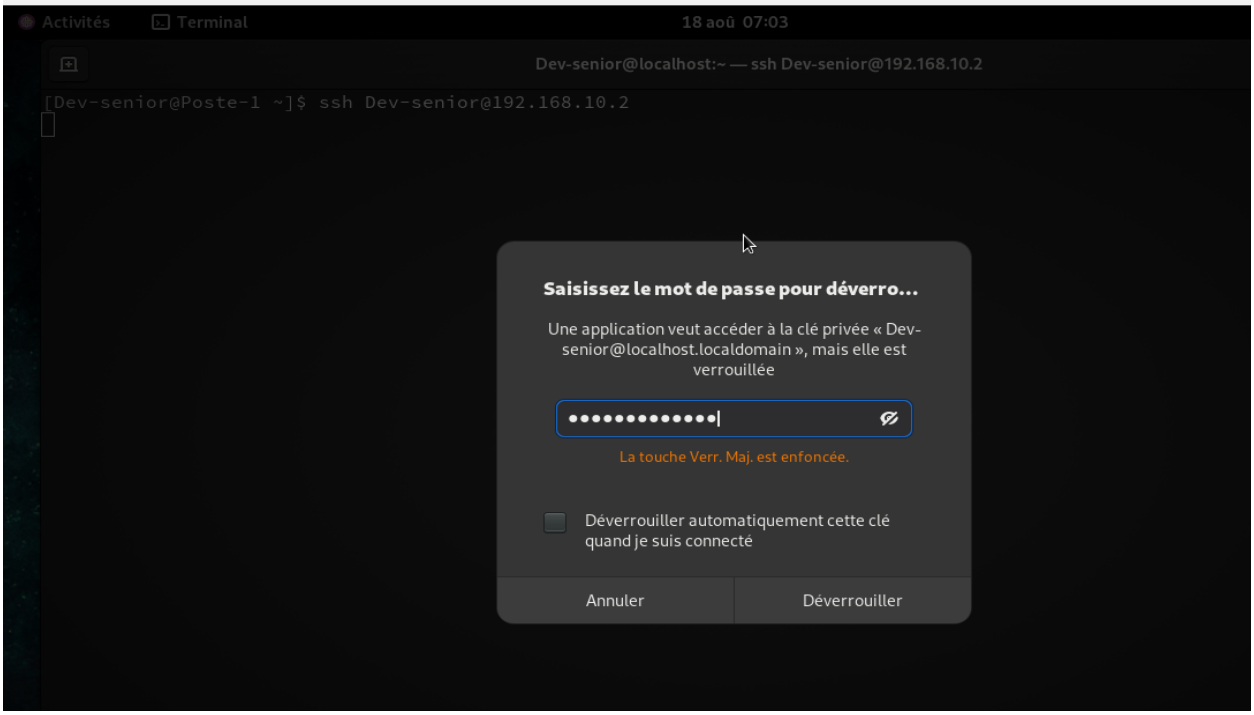


- Démarrons un serveur http pour transférer des fichiers depuis notre client ver le serveur :

- Envoi de la clé public KeyDev-senior pub sur le serveur via Wget :

- Ajout de la clé **publique** de notre machine cliente dans le répertoire des clés **autorisées** par le serveur :

```
[Dev-senior@srv-prod ~]$ cp keyDev-senior.pub .ssh/authorized_keys
[Dev-senior@srv-prod ~]$
```



C. Test de l'authentification sans login sur le serveur :

Essayons de nous connecter depuis la **machine à** notre serveur 192.168.10.2 sans mot de passe.

```
[Dev-senior@Poste-1 ~]$ ssh Dev-senior@192.168.10.2
Last login: Sat Aug 17 19:37:18 2024 from 192.168.10.3
[Dev-senior@srv-prod ~]$
```

Authentification réussie.

Nous sommes à présent **connectés** à notre serveur depuis la machine cliente via SSH, sans identifiant ni mot de passe.