SE CONNECTER EN SSH PAR ECHANGE DE CLES SSH

Windows - Linux Server
Utilisation de base

Tutoriel **SSH** - Linux Server

David GOÏTRÉ

Table des matières

Introduction

SSH (Secure Shell) est un protocole de communication sécurisé qui permet de se connecter à un ordinateur distant de façon sécurisée. Par défaut, la connexion SSH s'effectue avec un mot de passe. Cette méthode d'authentification n'est pas la plus sûre car votre mot de passe peut être dérobé ou deviné par des pirates qui pourraient dès lors accéder à votre serveur et voler vos données personnelles.

Pour plus de sécurité, il est conseillé de se connecter à un ordinateur distant en utilisant l'authentification par échange de clés SSH.

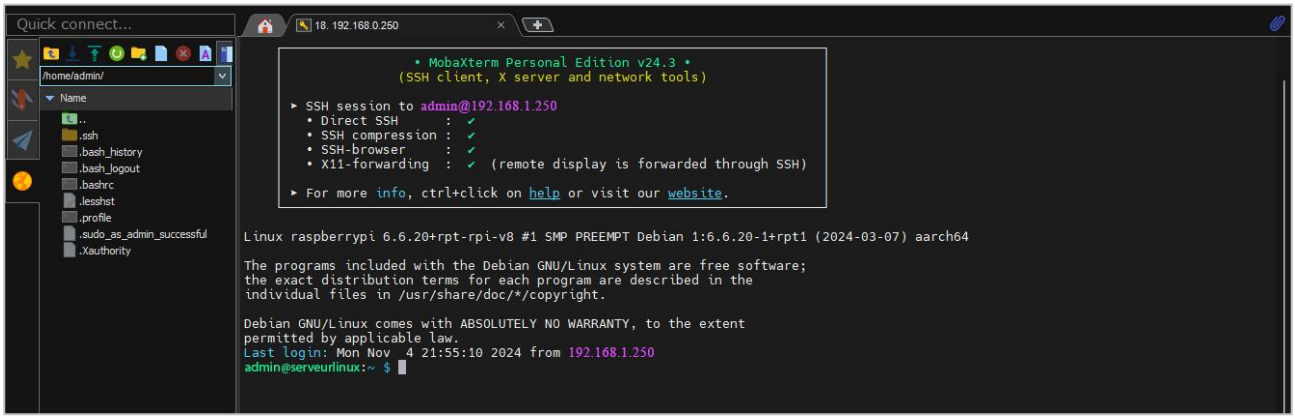
1. Pré requis

On a besoin des différents matériels et logiciels pour la création d'une connexion SSH par certificat.

- Un PC client sous Windows
- Un Serveur Linux pour la prise en main en SSH
- Le logiciel Putty ou MobaXterm pour se connecter en SSH au serveur
- Connaitre le logiciel **PowerShell** de Windows et sa ligne de commandes
- Les identifiants de connexion au serveur (nom d'hôte*, nom utilisateur et mot de passe)

  *Le nom d'hôte (**hostname**) est à remplacer par l'**@IP** ou le **nom** du serveur.

Voici un exemple d'interface que l'on doit obtenir une fois connecter avec MobaXterm



2. Les clés SSH ?

L'authentification par échange de clés SSH fonctionne en plaçant une clé publique sur l'ordinateur distant et en utilisant une clé privée depuis son ordinateur.

Ces deux clés (publique et privée) sont liées l'une à l'autre. C'est seulement en présentant la clé privée à la clé publique qu'il est possible se connecter.

Chaque clé se présente sous la forme d'une longue chaine de caractères enregistrée dans un fichier. Pour plus de sécurité, on peut également protéger la clé privée avec une phrase secrète. Autrement dit, pour pouvoir utiliser la clé privée, il faudra saisir un mot de passe, ce qui renforce encore davantage la sécurité.

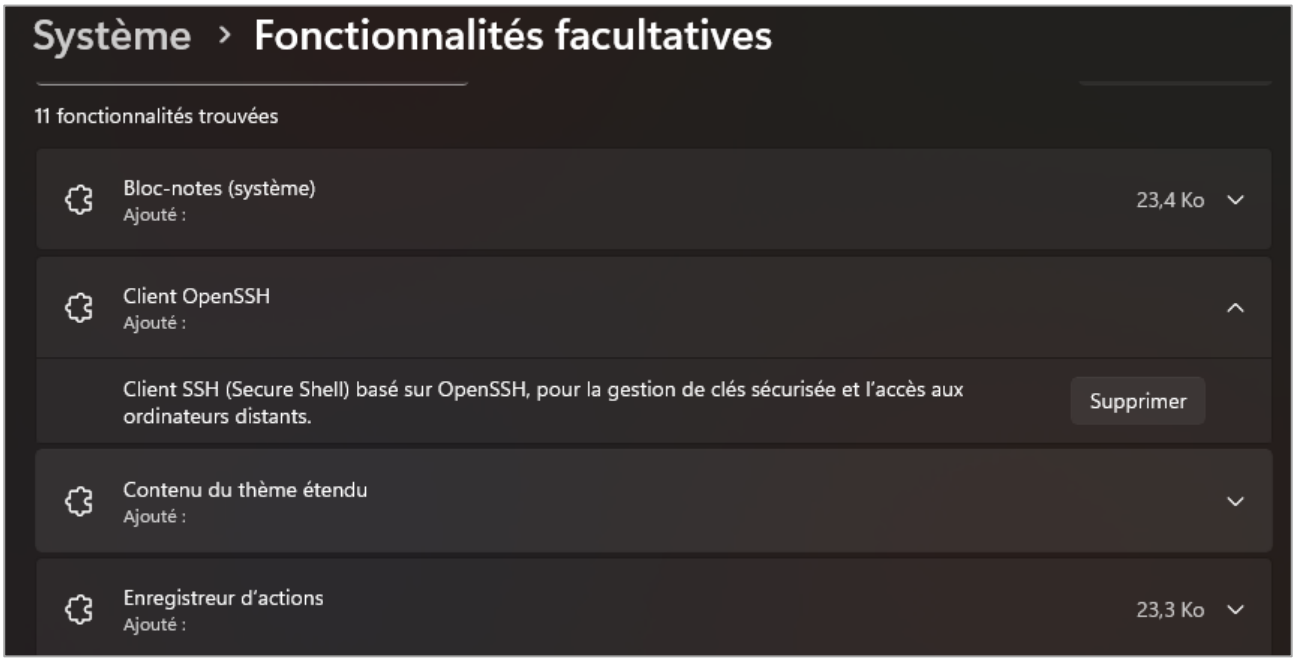
Les clés SSH peuvent être créées avec différents algorithmes de chiffrement :

- **DSA** : dangereux, il n'est plus pris en charge depuis OpenSSH 7.0
- **RSA** : acceptable si la longueur de la clé est de 3072 ou 4096 bits
- **Ed25519** : le plus sûr, c'est l'algorithme à privilégier aujourd'hui

3. Générer une paire de clés SSH avec PowerShell

Pour générer une paire de clés en ligne de commandes sous Windows, on a besoin du Client OpenSSH. Pour vérifier son installation, il faut :

- Ouvrir les **paramètres de Windows**
- Cliquer sur le menu **Système**
- Cliquer sur **Fonctionnalités facultatives**



- Voici une capture représentant le **Client OpenSSH** installé

a) Ouvrir **PowerShell**

- Créer **un dossier .ssh** sur le poste client

```
# c:\> (taper une autre lettre pour sélectionner le disque dur souhaité)
# mkdir .ssh (créé le dossier .ssh)
# cd .ssh (rentre dans le dossier créé)
```

b) Générer la paire de clés

```
# ssh-keygen -t rsa -b 4096 -f C:\.ssh\nomclé (génère une paire de clés rsa)
ou
# ssh-keygen -t ed25519 -f C:\.ssh\nomclé (génère une paire de clés Ed25519)
```

c) Vérifier l'emplacement du dossier (ex : **c:\.ssh**)

```
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\.ssh\id_ed25519):
```

d) Entrer une **phrase secrète** pour **protéger la clé privée**

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

e) Résultat de la paire de clés générées

```
Your identification has been saved in C:\.ssh\id_ed25519
Your public key has been saved in C:\.ssh\id_ed25519.pub
The key fingerprint is:
SHA256:4Yjb63lZzyRw+ADKaZ6nwZDA7jBrtorVR4mkgXRGWN0 nomutilisateur@nomdupc
The key's randomart image is:
+--[ED25519 256]--+
|..+=. .          |
|o+o  ..E         |
|o..o.o ...       |
|o.o+=o ++..      |
|oo.=o.+ S=       |
|.+ .=+.   + .    |
|o o o+o  o =     |
|.o  .. oo   o    |
|+    .+.         |
+-----[SHA256]------+
```

f) La paire de clés se trouve dans le dossier suivant

```
c:\.ssh
id_ed25519 (clé privée)
id_ed25519.pub (clé publique)
```

4. Copier la clé publique sur le serveur distant

Pour établir la connexion SSH par clés SSH, il faut que la clé publique soit présente dans le fichier **~/.ssh/authorized_keys** d'un utilisateur sur le serveur distant.

a) Créer le **dossier .ssh** sur le serveur

```
# ssh utilisateur@hostname (ex : ssh admin@192.168.1.250)
# cd /home/user
# mkdir .ssh
```

b) Copier la **clé publique** sur le serveur

```
# cat C:\.ssh\id_ed25519.pub | ssh username@hostname "cat >> ~/.ssh/authorized_keys"
Ou
# ssh-copy-id -i C:\.ssh\id_ed25519.pub username@hostname
```

- Saisir le mot de passe utilisateur
- Voilà, la clé publique a bien été copiée dans le **fichier ~/.ssh/authorized_keys** du serveur

5a. Se connecter en SSH par clés SSH avec le Terminal

Il faut maintenant passer à la connexion par clés SSH par défaut.

a) Se connecter au serveur

```
# ssh -i C:\.ssh\id_25519 username@hostname
```

- Entrer la phrase de passe de la clé privée (si elle a été définie)
- Une fois l'authentification par échange de clés SSH configurée, il s'agit de la méthode d'authentification **par défaut** pour se connecter au serveur

b) Créer un fichier de **configuration**

- Pour une connexion automatique, on a besoin de créer un fichier de configuration dans le **dossier .ssh** du poste client nommé **config** (sans l'extension).

```
# cd C:\.ssh
# type nul > config
# notepad config
```

- Copier le contenu suivant dans le fichier **config** en remplaçant les textes en gras

```
Host *
  IgnoreUnknow AddKeysToAgent,UseKeychain
  AddKeysToAgent yes
  UseKeychain yes
Host MonAlias
  HostName 192.168.1.250 (@IP_Serveur)
  User admin
  IdentitiesOnly yes
  IdentityFile ~/.ssh/id_25519
```

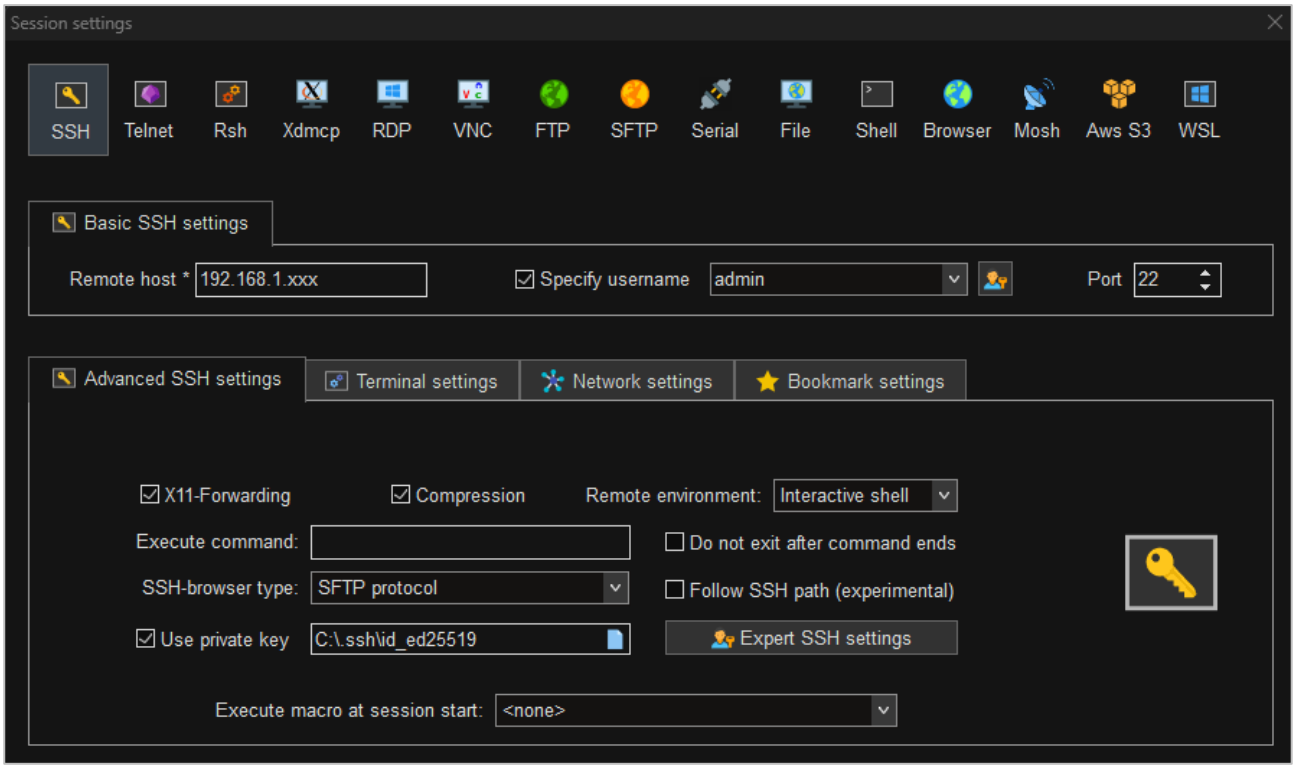
c) Se connecter avec l'alias

```
# ssh MonAlias
```

5b. Se connecter en SSH par clés SSH avec MobaXterm

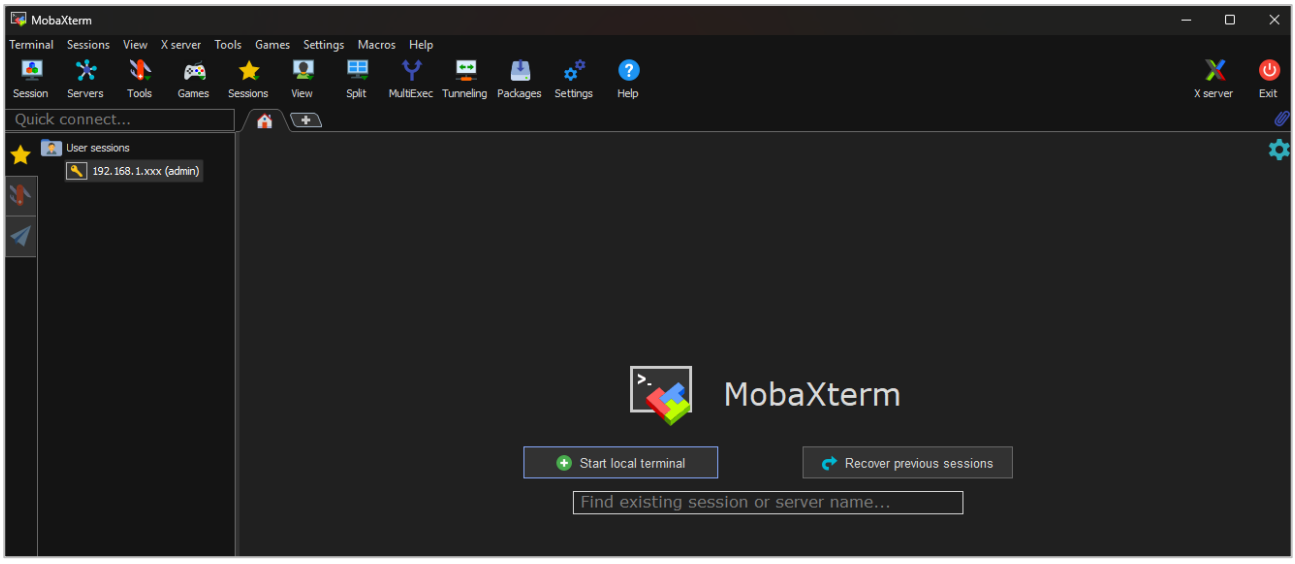
a) Ouvrir MobaXterm

- Cliquer sur le menu **Sessions/New sessions**
- Choisir le type de connexion **SSH**
- Cliquer sur l'onglet **Advanced SSH settings**
- Saisir le **@IP du serveur** et le **nom utilisateur**
- Cocher la case **Use private key**
- Sélectionner le fichier de la **clé privée**

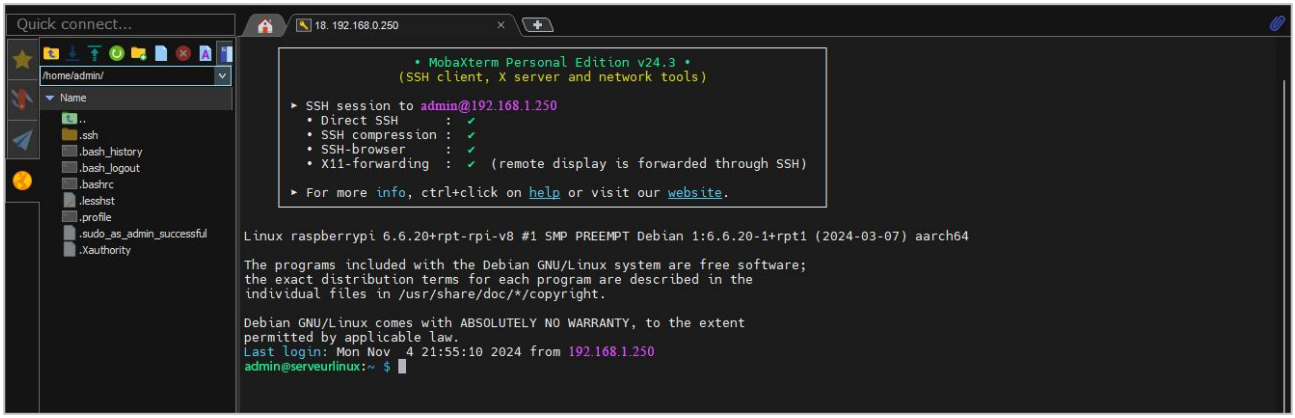


- Ouvrir l'onglet **Bookmark settings** pour modifier le nom du serveur

b) Double-cliquer sur le **nom de la session** dans la colonne de gauche



c) A la première connexion, une question s'affiche nous demandant d'accepter si on fait confiance à l'identité de la session et de continuer la connexion. Cliquer sur le bouton **Accepter**.
d) Saisir l'identifiant et la phrase de passe de la clé
e) Résultat de la connexion



Enregistrement des mots de passe
On a la possibilité d'ouvrir une session automatiquement sans saisir de mot de passe. Pour cela, il faut suivre les étapes suivantes à la première connexion de la session

a) Une fois le mot de passe saisie, **MobaXterm** demande comment on veut gérer les sessions.
Saisir un mot de passe principal
b) Cocher une des **trois lignes**, selon son besoin
- uniquement sur un nouveau compte Windows ou un nouvel ordinateur
- à chaque nouvelle configuration de session
- à chaque démarrage de MobaXterm et après la sortie du mode veille

Tous les paramètres de MobaXterm sont enregistrés dans le fichier **mobaxterm.ini**. Si l'on doit modifier la configuration, il faut supprimer le fichier et recommencer.

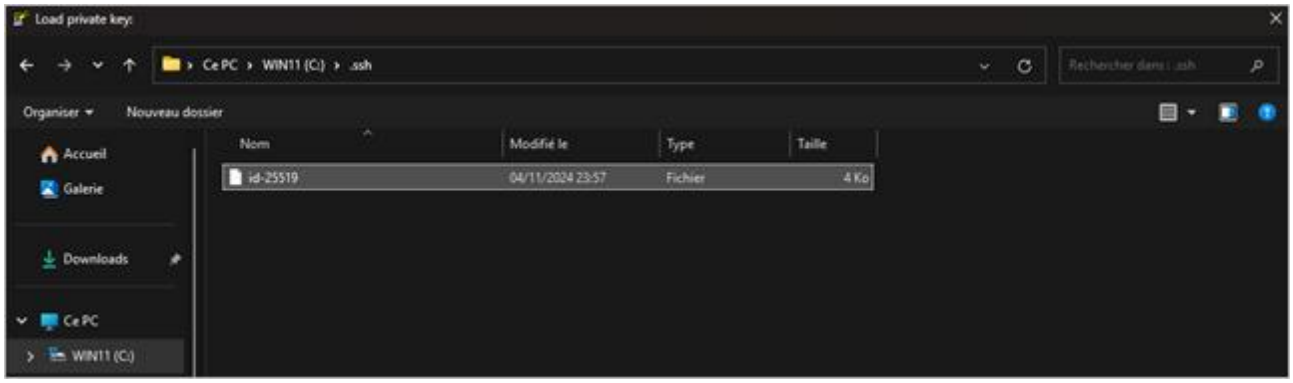
5c. Se connecter en SSH par clés SSH avec Putty

La clé privée générée par la commande ssh-keygen **n'est pas compatible** avec **Putty**. Pour pouvoir se connecter, il faut utiliser **PuttyGen** et **Pageant**. Ces deux outils sont installés avec **Putty**.

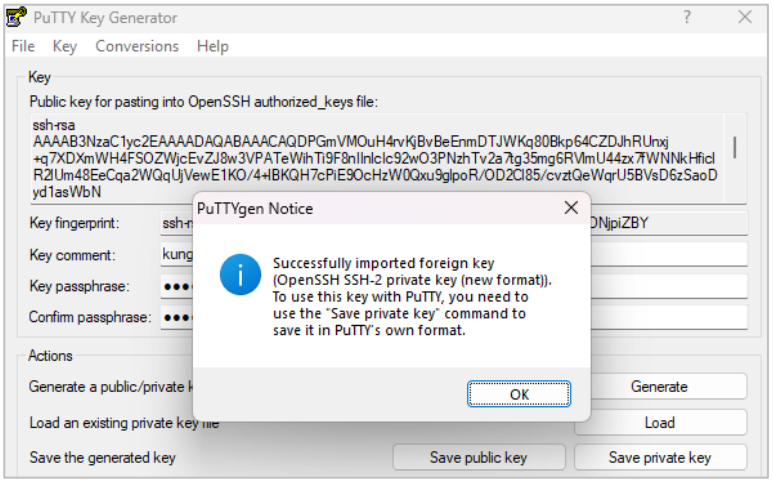
- **PuttyGen** permet de générer ou convertir une clé privée pour putty
- **Pageant** permet de rendre la connexion au serveur automatique

a) Ouvrir **PuttyGen**

- Cliquer sur le menu **File/Load private key**
- Sélectionner la clé privée dans le dossier **c:\.ssh**



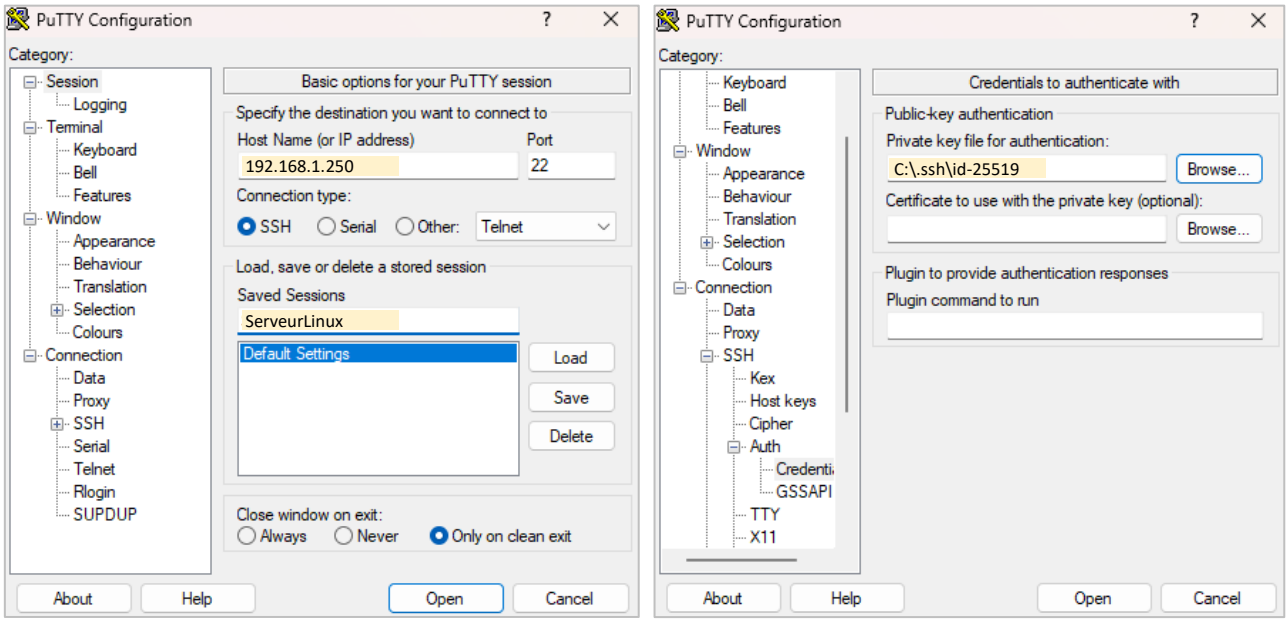
- Résultat du chargement de la clé



- Valider et cliquer sur le bouton **Save private key** pour générer une nouvelle clé privée, **compatible** avec Putty.

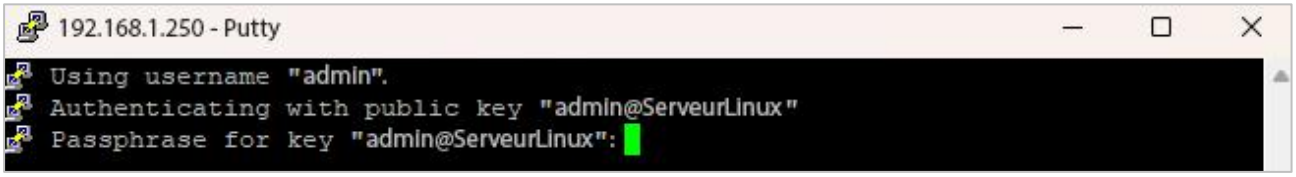
b) Ouvrir **Putty**

- Saisir l'**@IP du serveur** et un **nom de session**
- Cliquer sur le menu **Connection/SSH/Auth/Credentials**
- Dans le champ **Private Key file…**, sélectionner la nouvelle clé privée
- Revenir sur l'écran d'accueil et Cliquer sur le bouton **Save**



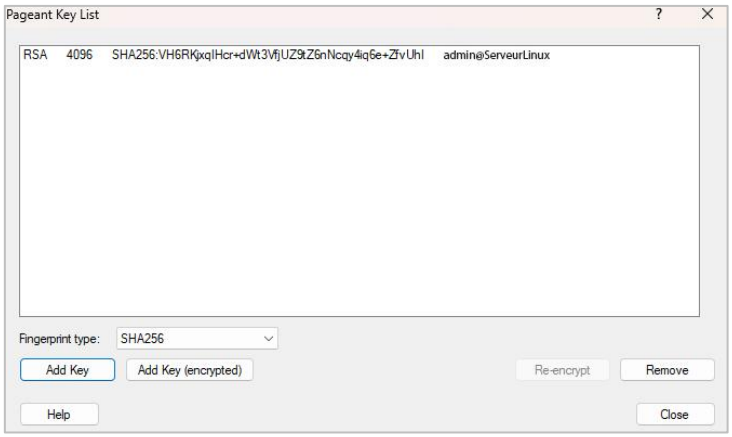
c) Sélectionner la **session** et cliquer sur le bouton **Open**

- Saisir la passphrase
- Résultat de la connexion



d) Automatiser la connexion

- Ouvrir **Pageant**
- Double-cliquer sur **l'icône** Pageant dans la **zone de notification**
- Cliquer sur le bouton **Add Key**
- Résultat du chargement



- Pour se connecter automatiquement à chaque session avec Putty, **Pageant** devra être exécuter en amont.

6. Autres méthodes de connexion

Voici une liste de sites web sur le même sujet

- Connexion SSH avec putty : https://www.it-connect.fr
- Connexion SSH et génération de clés : https://www.malekal.com
- Connexion SSH et generation de clés : https://pixiscreen.fr

7. Conclusion

Les clés SSH privées et publiques sont bien installées et configurées avec succès sur le **serveur** et le **client**. On peut désormais se connecter en SSH par échange de clés SSH.