WINDOWS :

- Albatros Albatros
- auxilienpereiral admin.AMLES/2022

ARCHLINUX :

- root 2022/root.AMLES
- alice alice
- bob bob

ssh :
bob : bob
alice : alice

I. Installation et configuration du serveur et client Linux.

1. En utilisant l'authentification par mot de passe.

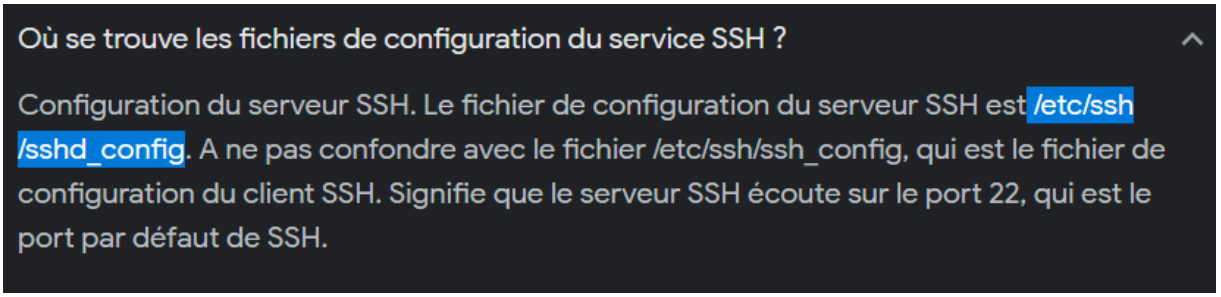
(a) Installer le package openssh. Avant d'installer un nouveau package…

pacman -Syu

pacman -S openssh

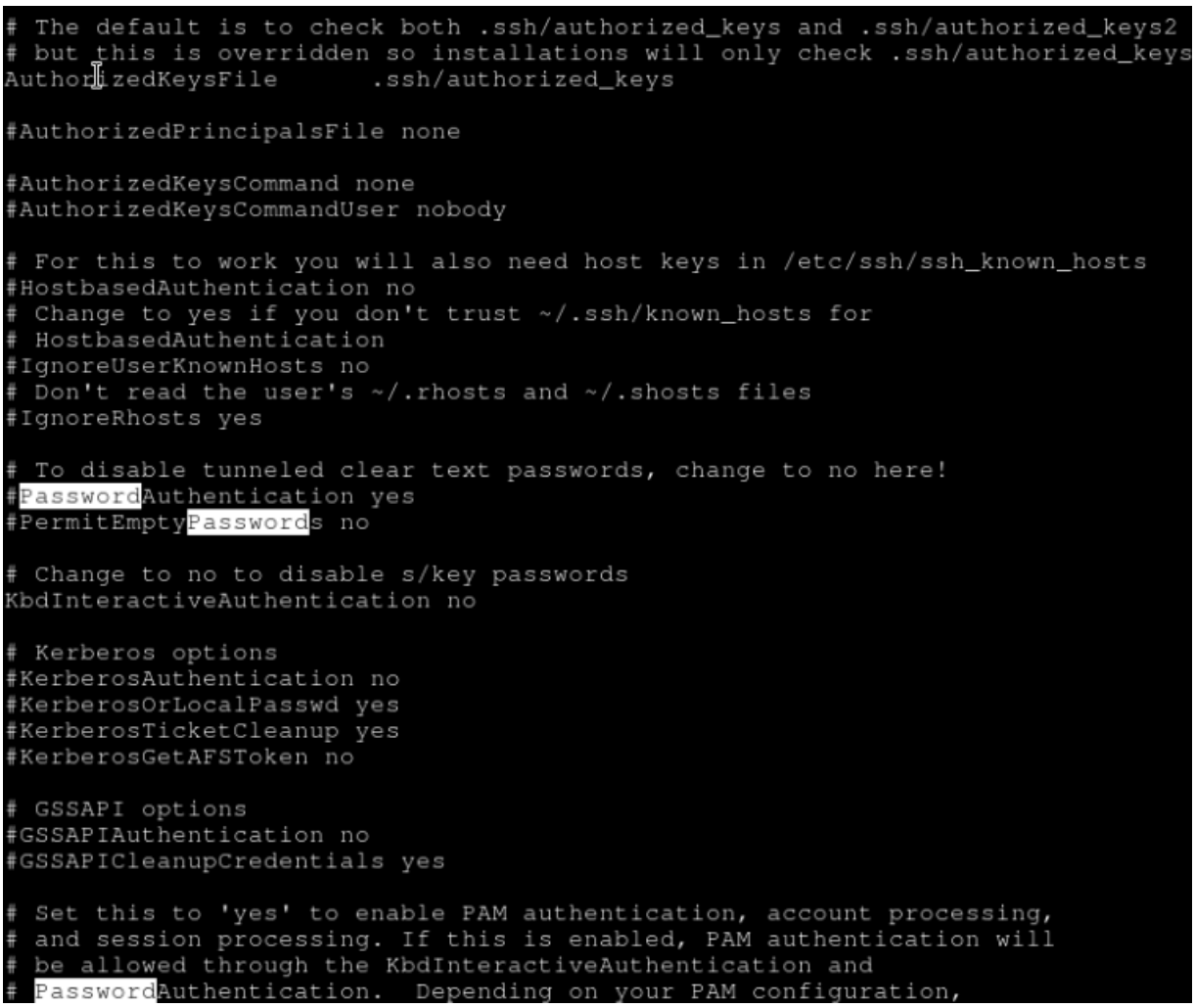
(b) Quel est le fichier de configuration de sshd (le dæmon ssh) ? Dans ce fichier, vérifier que l'authentification du client se fait par mot de passe uniquement. Noter qu'il y a une page man dédiée à ce fichier de configuration.

https://wiki.archlinux.org/title/OpenSSH



/etc/ssh/sshd_config

L'option PasswordAuthentication est yes



```
# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
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_hosts
#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
#IgnoreRhosts yes

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

# Change to no to disable s/key passwords
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication.  Depending on your PAM configuration,
```

Pour configurer ce fichier attention ! :

sshd -t => test mode de sshd (pour tester la configuration avant la mise en marche. Pas de retour = truc qui marche)

IMPORTANT :

.ssh : ssh du client

ssh : ssh du serveur

key rsa : nos empreintes.

(c) Dans le wiki de Arch Linux, on indique le répertoire qui contient les fichiers correspondant aux paires de clés (privée, publique) du serveur. Noter le contenu de ce répertoire.

https://www.ssh.com/academy/ssh/keygen

Check these simple issues before you look any further.

1. The configuration directory `~/.ssh`, its contents should be accessible only by the user (check this on both the client and the server), and the user's home directory should only be writable by the user:

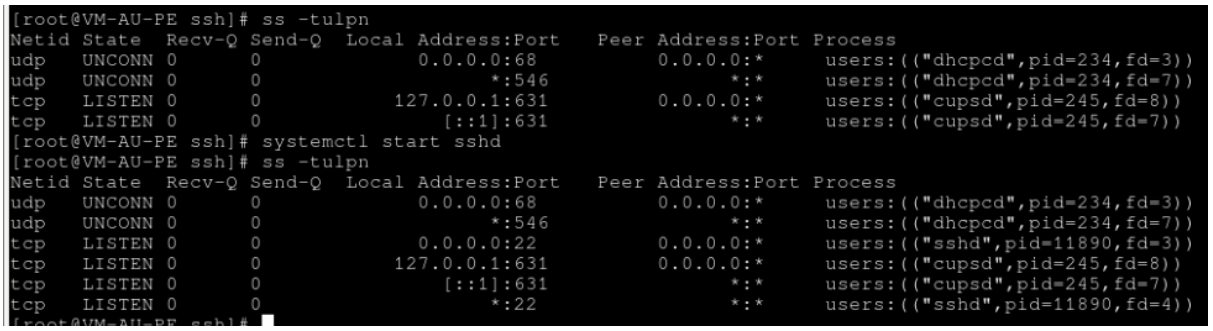
```
$ chmod go-w ~
$ chmod 700 ~/.ssh
$ chmod 600 ~/.ssh/*
$ chown -R $USER ~/.ssh
```

2. Check that the client's public key (e.g. `id_rsa.pub`) is in `~/.ssh/authorized_keys` on the server.
3. Check that you did not limit SSH access with `AllowUsers` or `AllowGroups` in the **server config**.
4. Check if the user has set a password. Sometimes new users who have not yet logged in to the server do not have a password.
5. **Append** `LogLevel DEBUG` to `/etc/ssh/sshd_config`.
6. Run `journalctl -xe` as root for possible (error) messages.
7. **Restart** `sshd` and logout/login on both client and server.

Il n'y a rien dans /etc/ssh

(d) Lancer le service sshd. Vérifier que le port correspondant est bien ouvert. Noter de nouveau le contenu du répertoire précédent.

port par défaut : 22.
systemctl start sshd
ss -tulpn
-> le 22 est bien présent quand on fait ss -tulpn

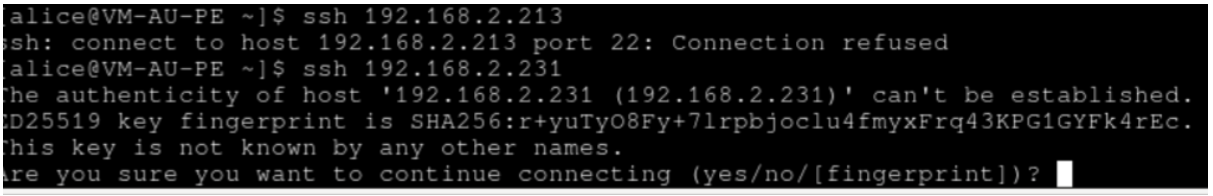


(e) On se met par binôme de vms Linux, et on teste une connexion ssh avec alice, bob. Sur la machine client, lors de la première connexion à un serveur ssh, un message avertit que l'authenticité du serveur ne peut pas être établie. Pourquoi ?

Hint : les serveurs que le client considère comme légitimes sont, par défaut, dans le fichier ~/.ssh/known_hosts. Ce fichier

existe-t-il ? Le serveur auquel on tente de se connecter est-il enregistré dans ce fichier ?

ssh 192.168.2.213



4. Se logguer par SSH

4.1. Authentification par mot de passe

C'est la méthode la plus simple. Depuis la machine cliente, tapez :

```
% ssh login@nom_DNS_du_serveur_SSH
```

- Si c'est la première connexion SSH depuis ce client vers ce serveur, il vous demande si le fingerprint de la clé publique présentée par le serveur est bien le bon. Pour être sûr que vous vous connectez au bon serveur, vous devez connaître de façon certaine le fingerprint de sa clé publique et la comparer à celle qu'il vous affiche. Si les deux fingerprints sont identiques, répondez *yes*, et la clé publique du serveur est alors rajoutée au fichier `~/.ssh/known_hosts`.
- Si vous vous êtes déjà connecté depuis ce client vers le serveur, sa clé publique est déjà dans le fichier `~/.ssh/known_hosts` et il ne vous demande donc rien.

Ensuite, entrez votre mot de passe... et vous verrez apparaître le prompt, comme si vous vous êtiez loggué en local sur la machine.

L'authenticité du serveur ne peut pas être établie car il s'agit de la première connexion, ainsi le serveur ne peut pas vérifier le "fingerprint" car l'empreinte n'a pas encore été définie.

Le fichier ~/.ssh/known_hosts n'existe pas encore, donc le serveur n'est pas dedans puisque jamais personne ne s'y est connecté. Par conséquent, l'authenticité du serveur n'est pas vérifiée.

(f) Pour permettre au client d'établir l'authenticité du serveur, celui-ci envoie au client l'empreinte (un condensé) du contenu du fichier correspondant à la clé publique du serveur.

Noter, côté client, l'empreinte envoyée par le serveur. **Sur le serveur, on va utiliser la commande ssh-keygen pour générer cette empreinte. Consulter la page manuel de cette commande pour trouver l'option adéquate.** Comparer les deux empreintes. Si elles ne sont pas identiques, il faudra répondre "no" à la finalisation de la connexion.

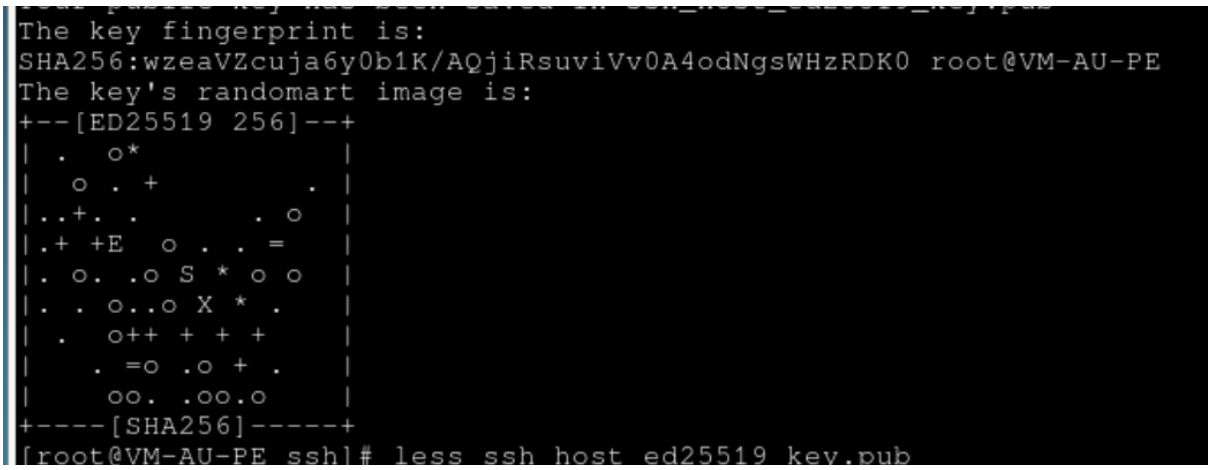
on fait **ssh-keygen -t ed25519**

Il y a maintenant :

- **ssh_host_ed25519_key**
- **ssh_host_ed25519_key.pub**

https://linux.developpez.com/formation_debian/ssh.html

(g) Si les empreintes sont identiques, on répond “yes” `à la finalisation de la connexion du client. Une fois connectée, consulter de nouveau, le fichier ~/.ssh/known_hosts (du client ou du serveur ?)



On regarde le fichier du client, car les informations de connexion du client vers le serveur sont dans ~/.ssh alors que les informations de connexion du serveur vers les clients sont dans /etc/ssh.

On a désormais un fichier known_hosts, qui contient les clés publiques du serveur de l’autre binôme (3 clés pour les 3 types de connexions).

(h) Inverser les rôles dans le binôme, et refaire le travail précédent.

2. Le client s’authentifie par la méthode de paire de clés SSH. Référence : SSH KEYS dans le wiki de Arch Linux.

(a) À l'aide de ssh-keygen, chacun des utilisateurs, alice et bob, génère sa paire de clé en utilisant les paramètres par défaut.

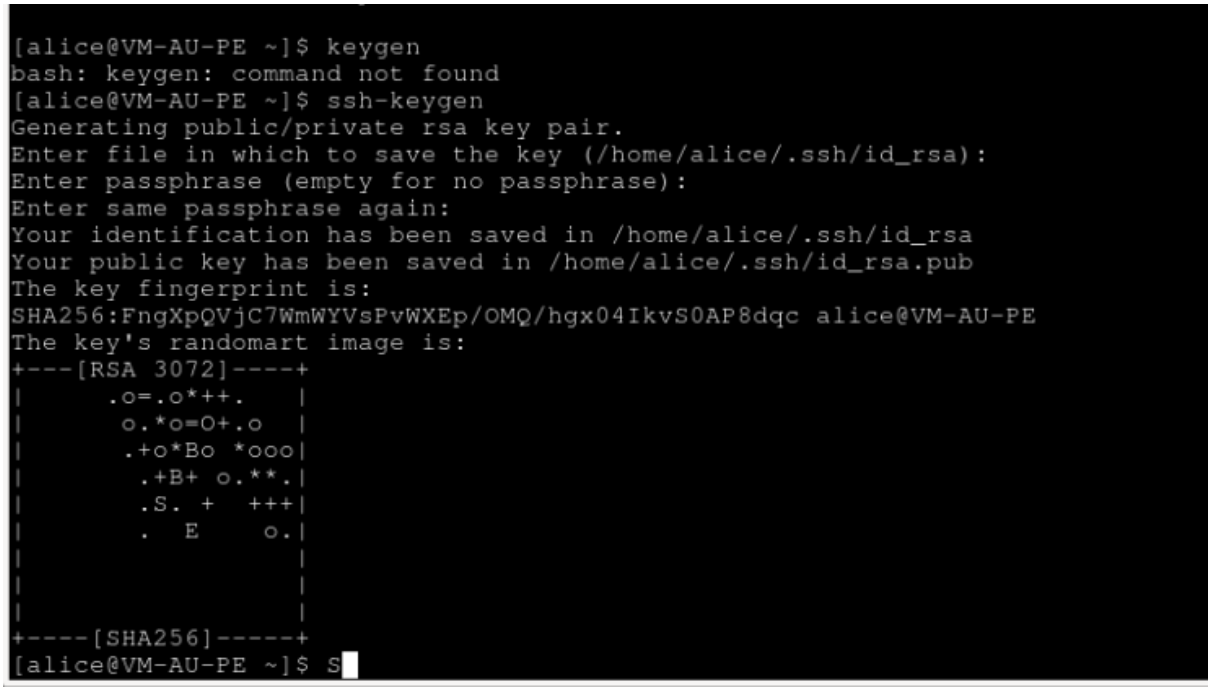
Quel est le type de clés par défaut ?

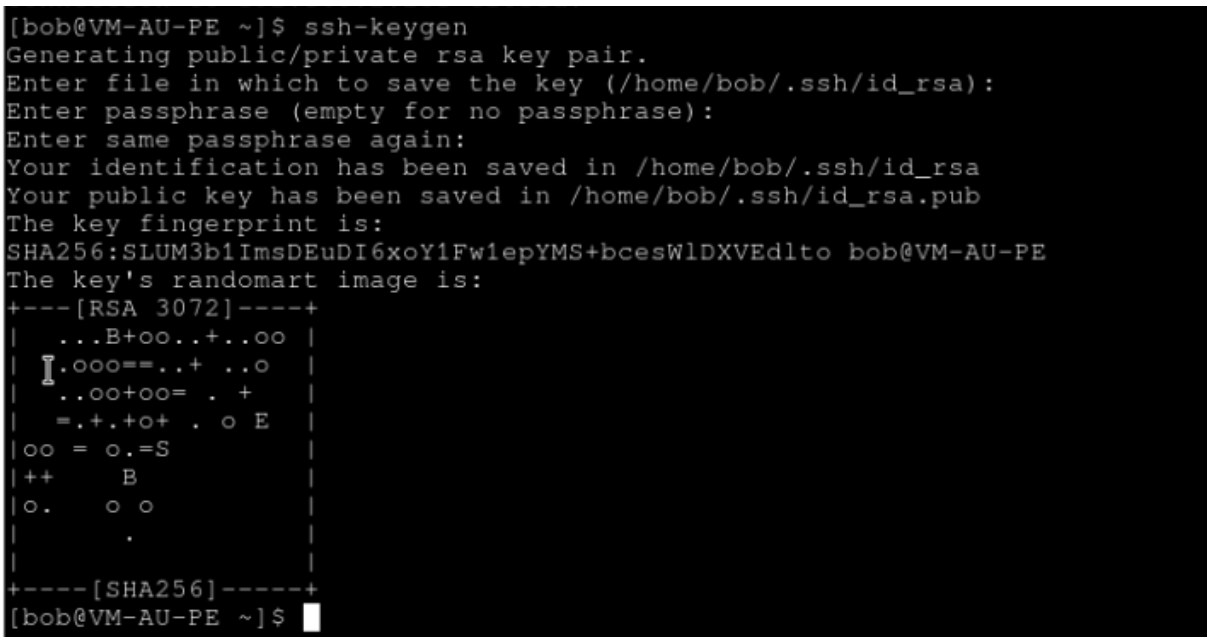
Dans quel répertoire de l’utilisateur la paire de clé est-elle enregistrée par défaut ?

Pour les besoins de cette saé, on ne positionne pas de passphrase.

shh-keygen

*”If invoked without any arguments, **ssh-keygen** will generate an RSA key for use in SSH protocol 2 connections. “*
Le type de clé par défaut est la paire de clé RSA
Les clés générées ont par défaut une longueur de 1024 bits, ce qui est aujourd'hui considéré comme suffisant pour une bonne protection.
Par défaut (il demande confirmation lors du processus de création), la clé privée est stockée dans le fichier **~/.ssh/id_dsa** avec les permissions 600 et la clé publique est stockée dans le fichier **~/.ssh/id_dsa.pub** avec les permissions 644.
Lors de la création, il vous demande une pass phrase qui est un mot de passe pour protéger la clé privée. Cette pass phrase sert à crypter la clé privée. La pass phrase vous sera alors demandée à chaque utilisation de la clé privée, c'est-à-dire à chaque fois que vous vous loguerez en utilisant cette méthode d'authentification. Un mécanisme appelé ssh-agent permet de ne pas rentrer le mot de passe à chaque fois... comme nous le verrons un peu plus loin dans ce chapitre.





(b) On se remet par binôme de vms Linux : client et serveur. L'utilisateur disons Alice, comme client du serveur, doit avoir, au préalable, le contenu sa cle publique concaténée au contenu de ~alice/.ssh/authorized_keys cote serveur.

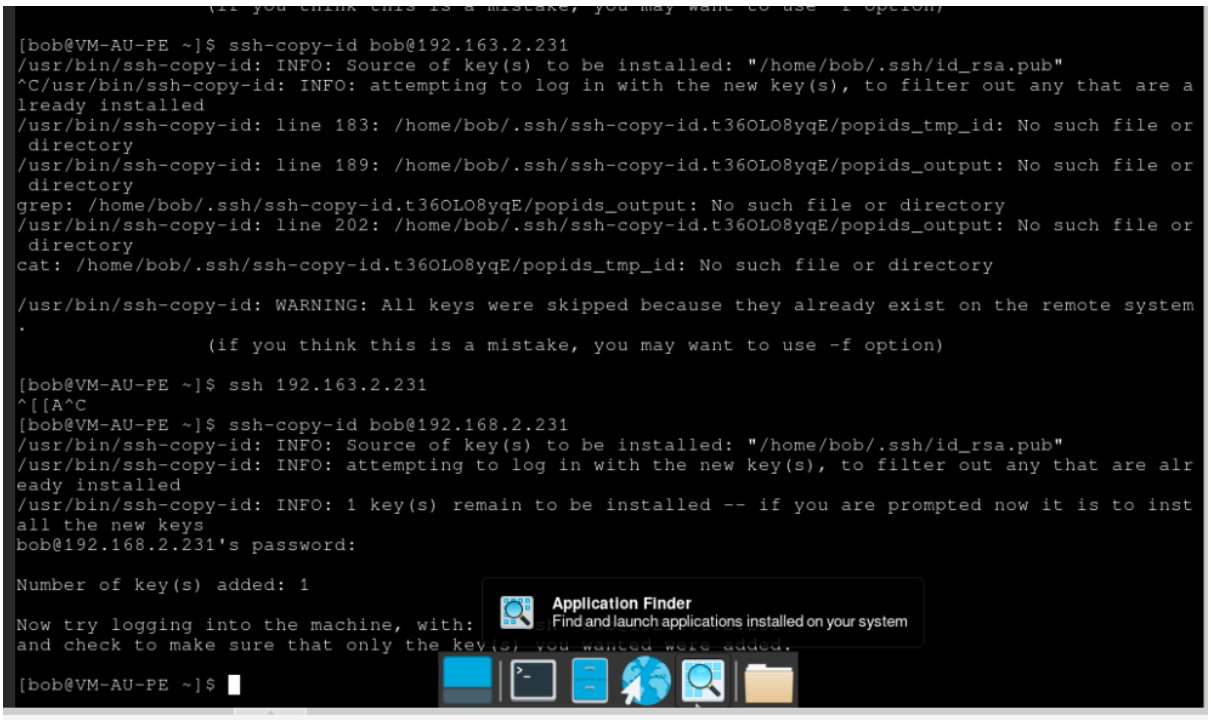
Vérifier que, dans ~alice/.ssh/authorized_keys, chaque clé commence sur une nouvelle ligne.

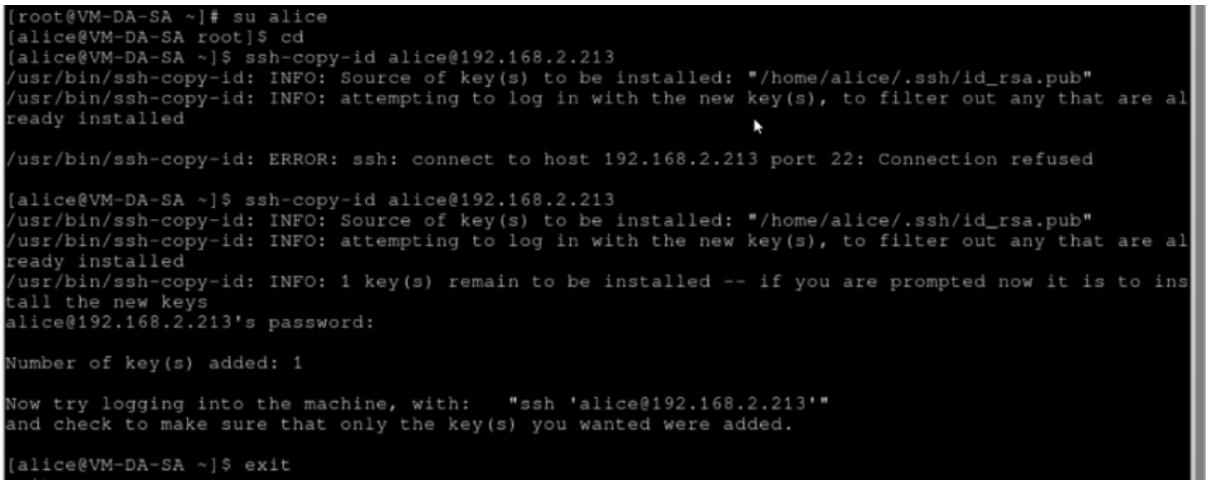
Faire le nécessaire pour que alice et bob aient chacun sa clé publique ajoutée au contenu de son fichier authorized_keys. Vérifier que les permissions d'accès aux fichiers et répertoires sont comme il faut.

https://www.digitalocean.com/community/tutorials/how-to-configure-ssh-key-based-authentication-on-a-linux-server#step-2-copying-an-ssh-public-key-to-your-server

ssh-copy-id alice@192.168.2.231

ssh-copy-id bob@192.168.2.231





C’est tout à fait correct, bob ne peut pas regarder le .ssh de alice et inversement, de même pour le root.

(c) Une fois le point précédent réalisé, on modifiera le fichier de configuration du serveur ssh afin que l’authentification ne soit plus par mot de passe, mais par la méthode de paire de clés SSH. Tester.

PasswordAuthentication no

```
#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      %h/.ssh/authorized_keys

#AuthorizedPrincipalsFile none
```

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

systemctl restart sshd

(d) alice ajoute une paire de clés de type ecdsa. Réaliser le travail. Tester une session ssh en présentant à la ligne de commande la clé à utiliser lors de l'authentification.

ssh-keygen -t ecdsa

```
[alice@VM-AU-PE ~]$ ls
Desktop  KAMM
[alice@VM-AU-PE ~]$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/alice/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alice/.ssh/id_ecdsa
Your public key has been saved in /home/alice/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:Un2ftJyXnWEYPyQ1P3Rj7pmcMWcfBwNBZgLmHqzmwn8 alice@VM-AU-PE
The key's randomart image is:
+---[ECDSA 256]---+
|        o...*=+B.|
|       + . +  Xo=|
|        = . ...@=|
|       + . . ++=^|
|      + S     *O+|
|   . o .       . |
|    o .          |
|     o  E        |
|      ..         |
+----[SHA256]-----+
[alice@VM-AU-PE ~]$ ssh-copy-id -i ~/.ssh/id_ecdsa.pub alice@192.168.2.231
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/alice/.ssh/id_ecdsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'alice@192.168.2.231'"
and check to make sure that only the key(s) you wanted were added.

[alice@VM-AU-PE ~]$
```

```
[alice@VM-AU-PE .ssh]$ ssh 192.168.2.231
Last login: Wed May  3 08:02:26 2023 from 192.168.2.213
[alice@VM-DA-SA ~]$ ls .ssh/authorized_keys
.ssh/authorized_keys
[alice@VM-DA-SA ~]$ less .ssh/authorized_keys
[alice@VM-DA-SA ~]$
```

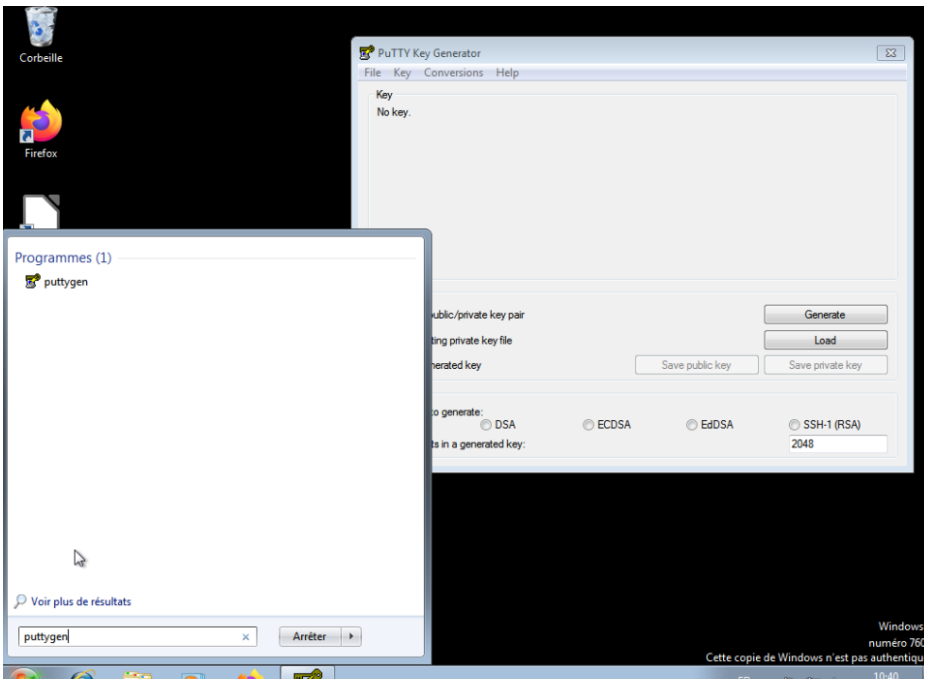
(e) Inverser les rôles dans le binôme, et refaire le travail précédent.

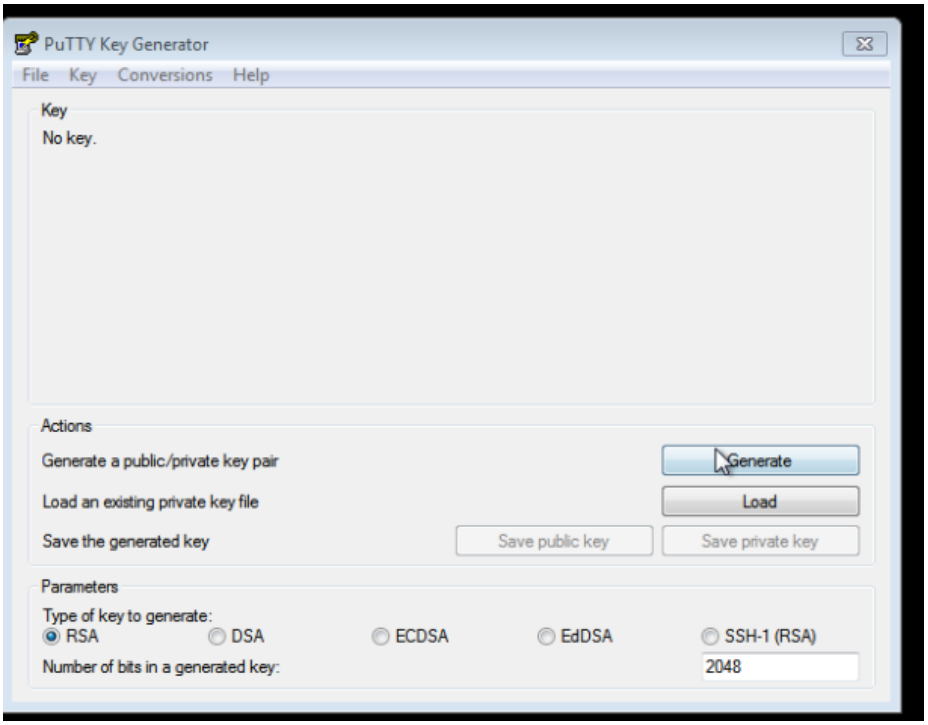
II. Installation et configuration du client Windows

On se remet par binome de vms : client sur Windows et serveur sur Linux. L'utilisateur Albatros va se connecter à son compte alice sur la machine Linux via SSH.

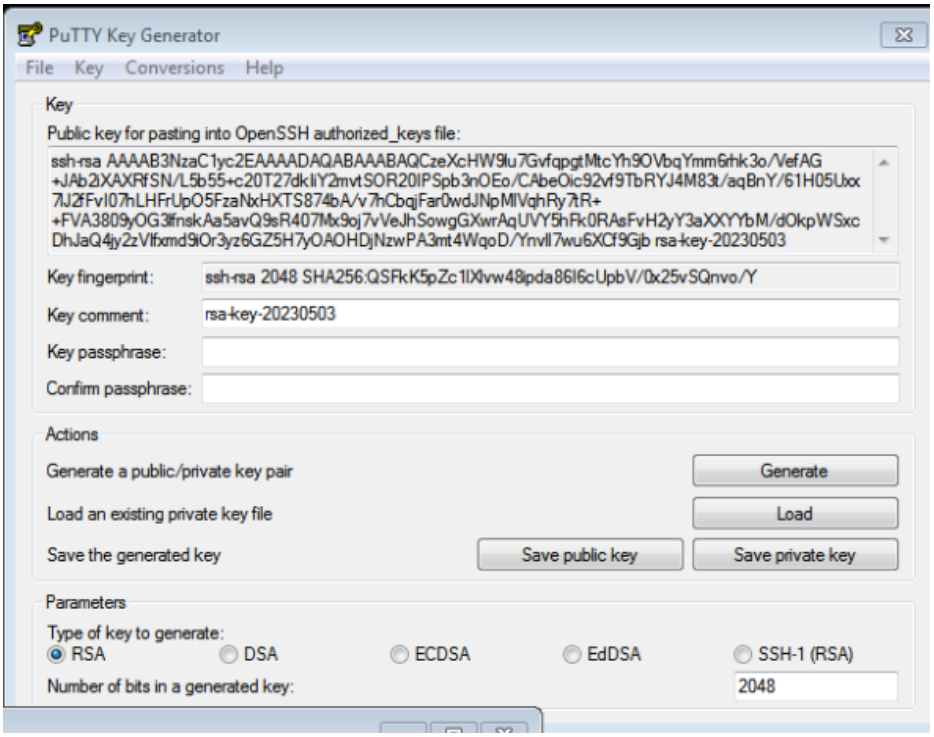
(a) On boote sur Windows et on installe puTTY : https: //www. putty. org/. Dans tout ce qui suit, on ne doit pas utiliser l'agent Pageant de puTTY.

(b) Albatros lance le programme PuTTYgen pour générer sa paire de clés de type rsa et la sauvegarder dans C:\Users\Albatros\ . Ne pas fermer encore la boîte de dialogue PuTTYgen.





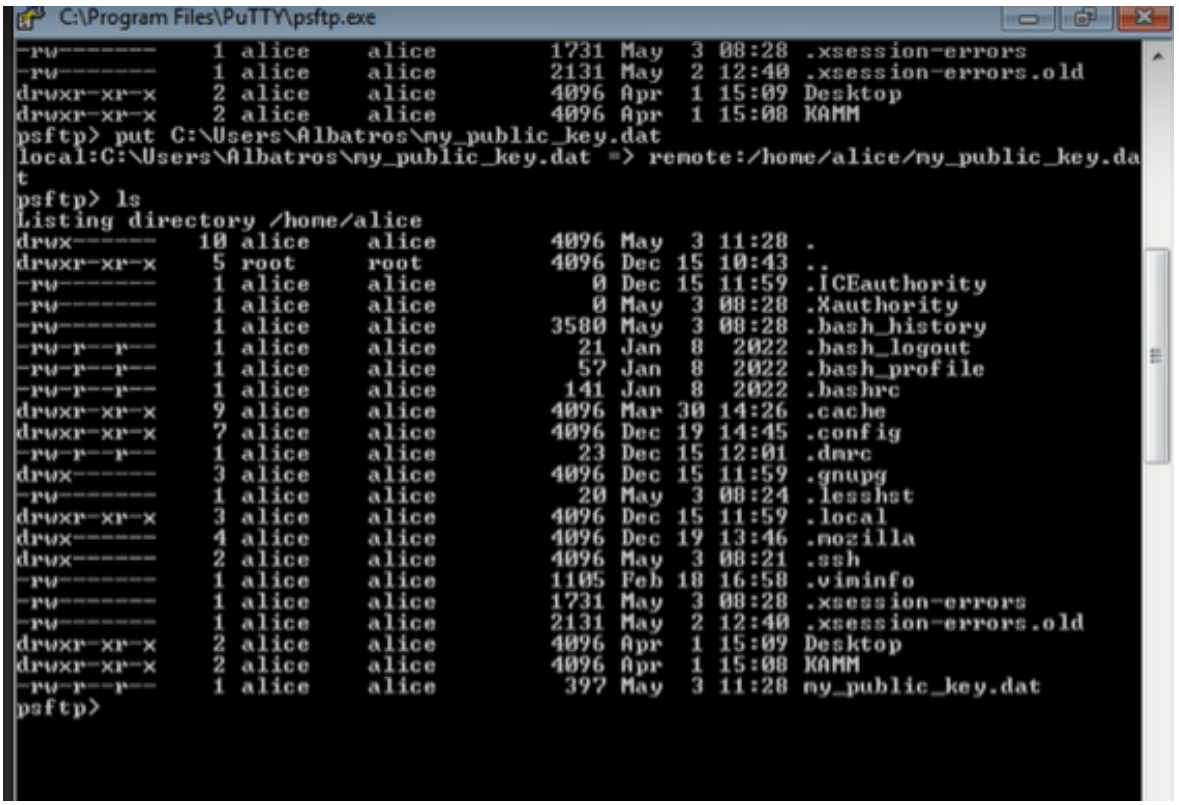
(c) Comme dans la partie I. 2, il faudra ajouter la cl´e publique `a la suite du contenu du fichier ~alice/.ssh/authorized_keys. Pour ce faire, on copie la zone indiqu´ee par PuTTYgen, on la place dans un fichier texte brut d'extension .dat qu'on envoie dans ~alice par le programme PSFTP de PuTTY.



Sur C:\Users\Albatros\ : Pour créer un .dat, on créer un doc text, on l'ouvre avec le bloc note classique, on copie colle la zone prévue par puTTY. On fait enregistrer sous en sélectionnant comme type de fichier tous les fichiers, ainsi, il prendra l'extension de fichier indiqué dans son nom de fichier ici "public_key_rsa.dat"

On fait save private key sur C:\Users\Albatros\ nommé "private_key_rsa.ppk" (laisser le type de fichier par défaut)

Une fois que le binôme a lancé le serveur ssh, pour se connecter :
Barre de recherche windows : PSFTP
On lance la commande : open 192.168.2.231
On saisi "alice" puis son mdp.
Pour le transfert : put C:\Users\Albatros\public_key_rsa.dat
On tape : bye



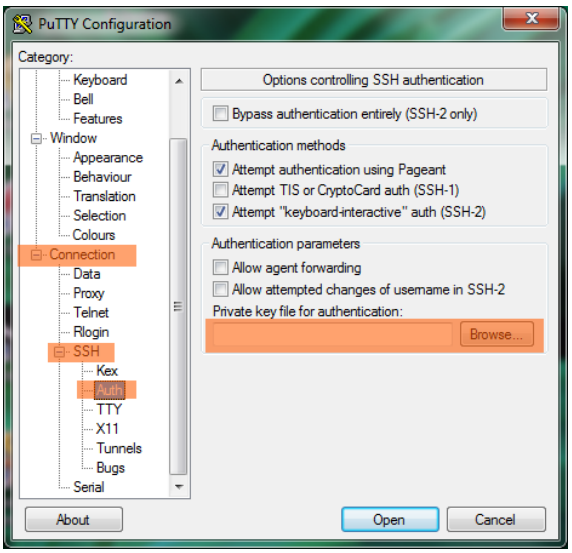
Lancer PuTTY et se connecter. Ensuite, on va ajouter le contenu de public_key_rsa.dat dans authorizedkeys de alice avec : cat public_key_rsa.dat >> .ssh/authorized_keys
Vérifier que tout est correct, on fait “exit”.

(d) Dans PuTTY, d´efinir une session SSH associ´ee `a la cl´e priv´ee sauvegard´ee. Tester la connexion SSH.

Lancer le programme PuTTY.

Aller dans Connection, saisir l’adresse ip sur serv ssh dans host name. Connection type : ssh. Nommé la saved session : VM_DA_SA_RSA (ne pas save encore)

Ensuite, aller dans la sous section de “Connection” “Data” dans « **Auto-login username** » saisir “alice”. Dans la section de “Connection > SSH > Auth>Credential” ensuite indiquer le chemin vers le fichier “private_key_RSA.ppk”.



Retourner dans la section Connection, puis “save” le profil VM_DA_SA_RSA

On se connecte en faisant “open”, si tout fonctionne, aucun mdp n’est nécessaire.

https://akril.net/cle-ssh-sous-windows-avec-putty/

(e) Albatros ajoute une paire de cl´es de type ecdsa, et d´efinit, dans PuTTY, une session SSH associ´ee `a cette cl´e. Tester la connexion SSH.

On fait les questions b) c) d) e) mais cette fois-ci avec _ECDSA.

(f) Inverser les rˆ oles dans le binˆome, et refaire le travail pr´ec´edent.

THE END