2023-2024

BTS SIO1

TP- ADMIN À DISTANCE SSH

Nicolas Debut

Qu’est ce que le ssh?

Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé permettant entre autres de se connecter à un ordinateur à distance. SSH est indispensable pour l'administration de serveur sous Linux.

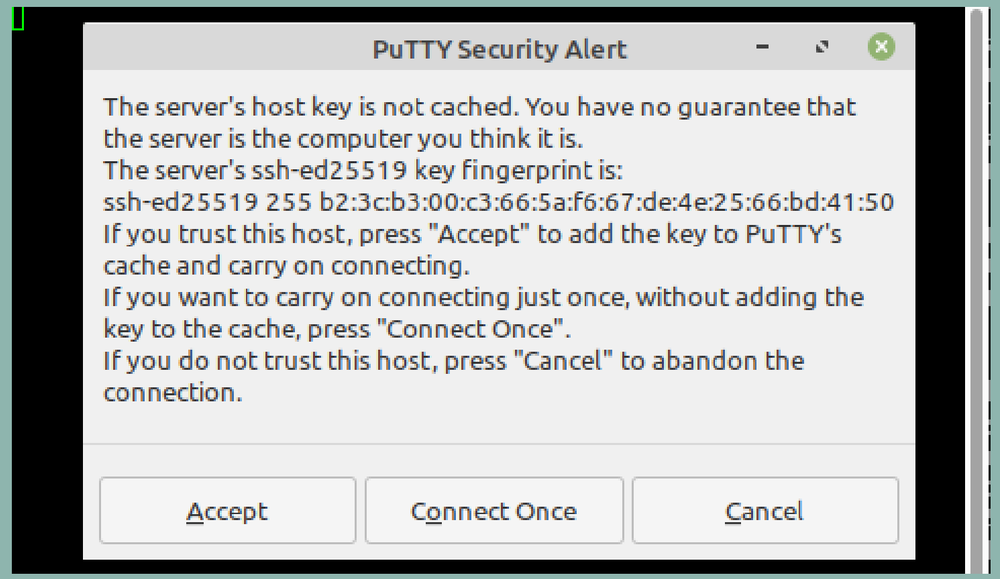
Voyons comment le configurer

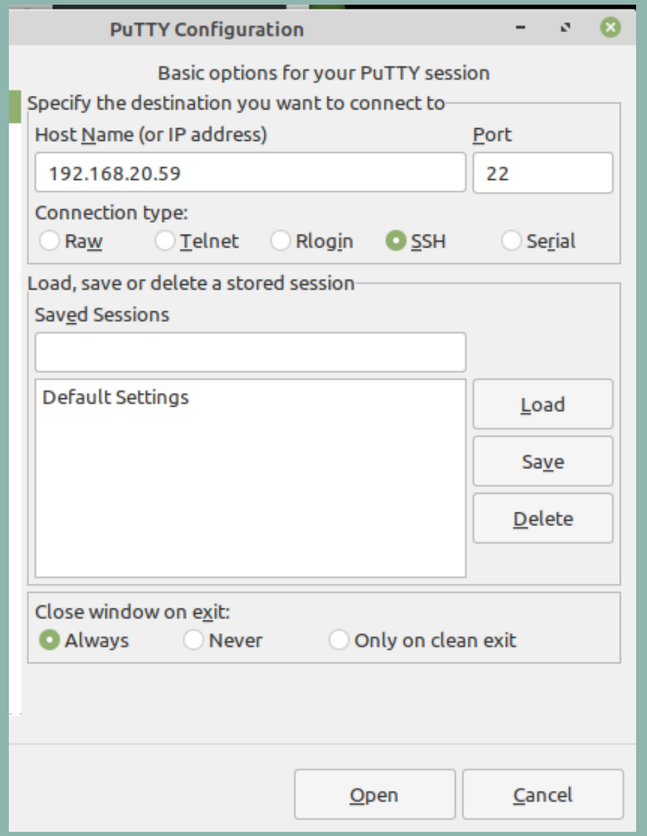
Tout d’abord nous allons effectuer la commande which ssh pour voir si le service n’est pas déjà installé en occurrence il l’est sinon effectuez un apt-get install ssh.

```
root@nicolasserver:~# which ssh
/usr/bin/ssh
```

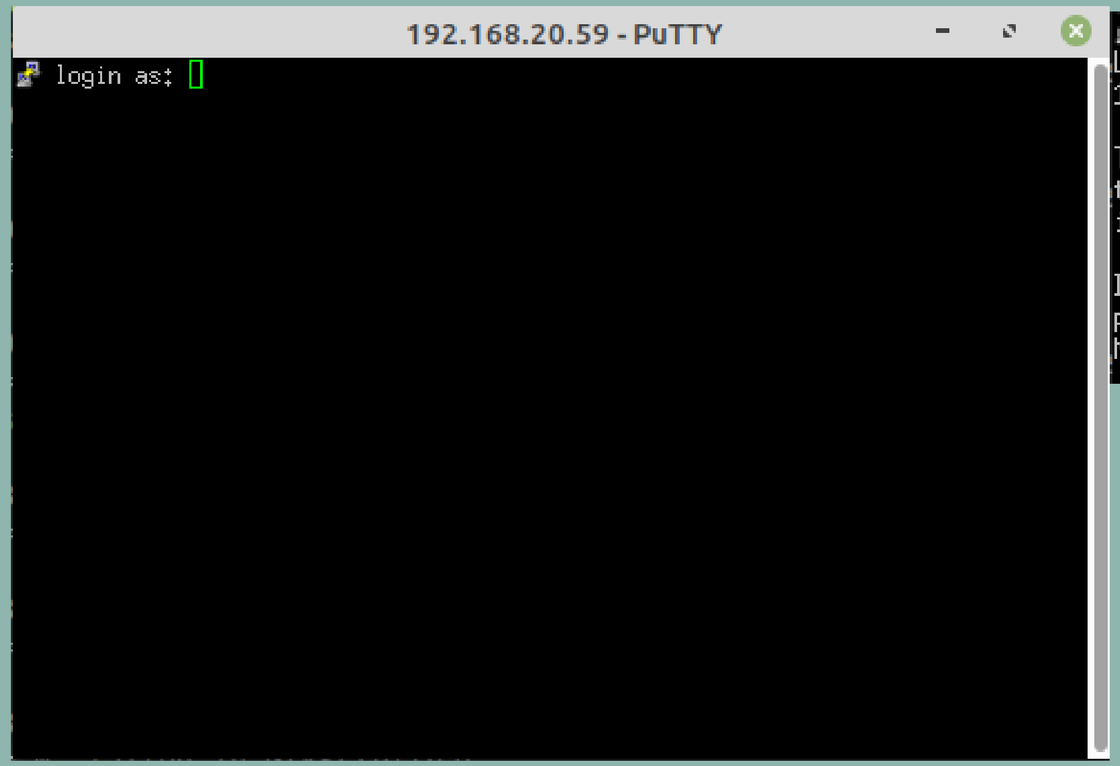
Une fois que vous l'aurez installé repérez votre adresse IP puis allez dans putty afin de tenter la connexion.

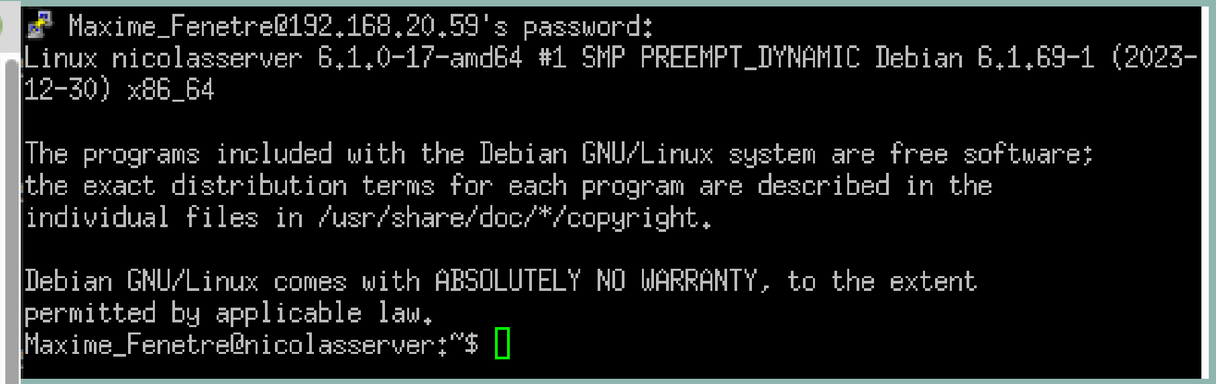
```
root@nicolasserver:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 9e:63:32:e8:f3:97 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.20.59/24 brd 192.168.20.255 scope global dynamic ens18
       valid_lft 6826sec preferred_lft 6826sec
    inet6 fe80::9c63:32ff:fee8:f397/64 scope link
       valid_lft forever preferred_lft forever
root@nicolasserver:~#
```

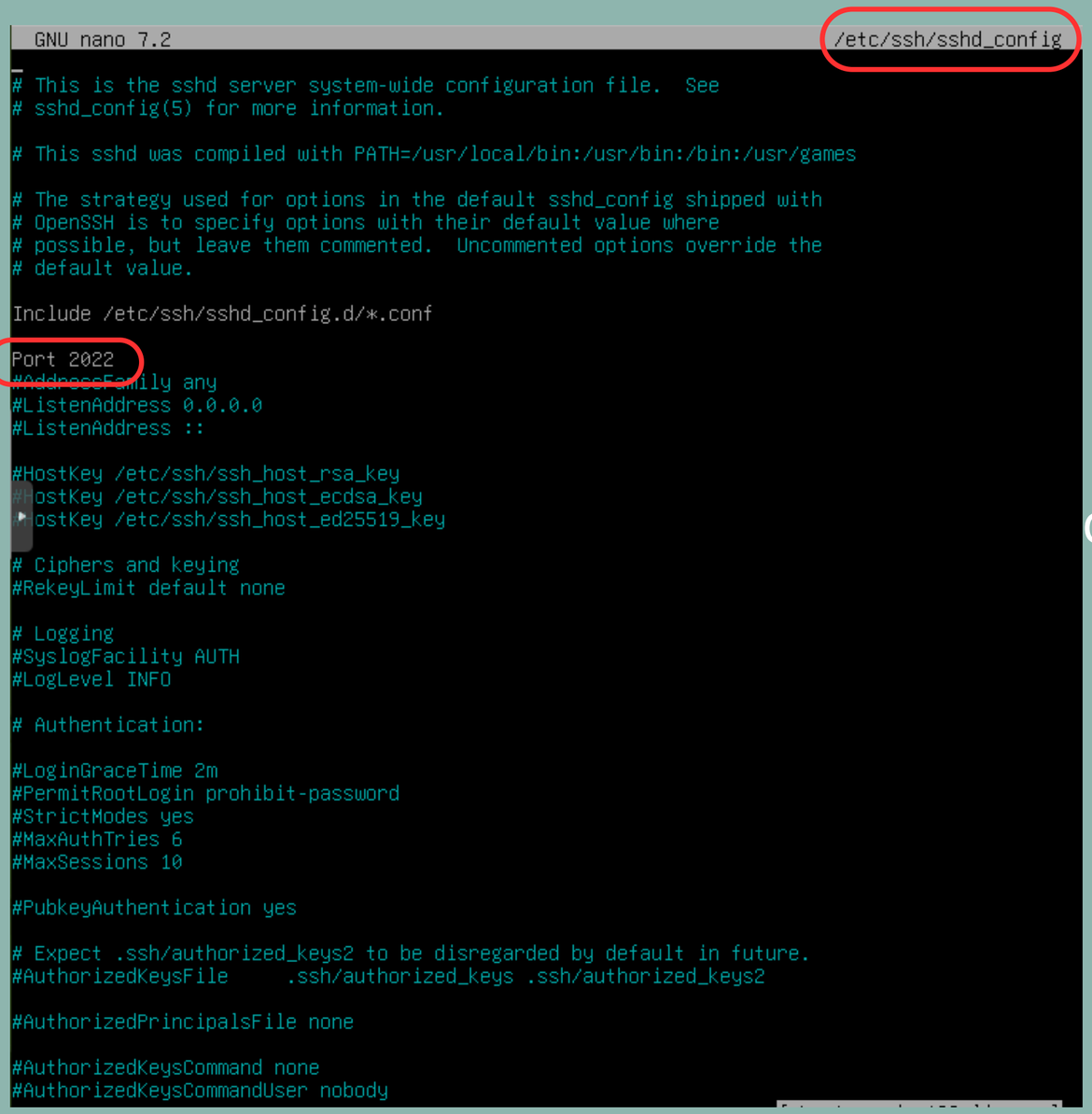




Une fois connectez vous arriverez sur cette page.
Vous devrez entrer des logins et mots de passe.
Vous remarquerez que vous ne pouvez pas vous connecter en root mais nous y reviendrons plus tard.

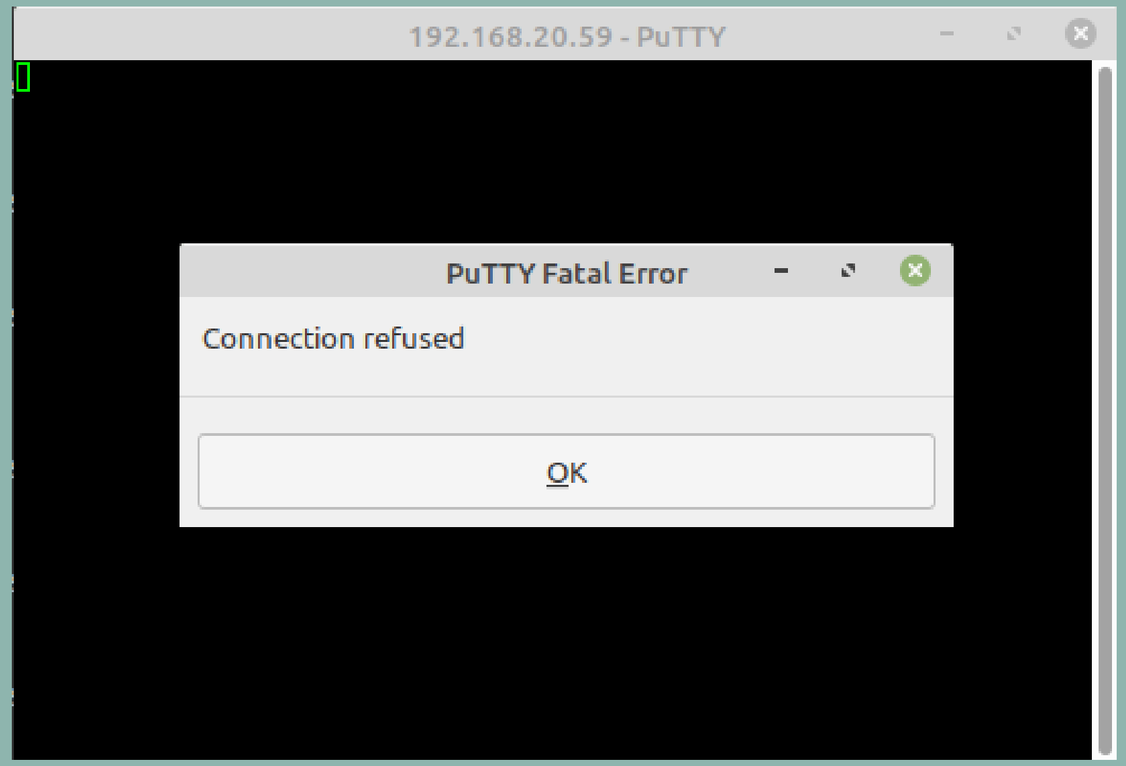


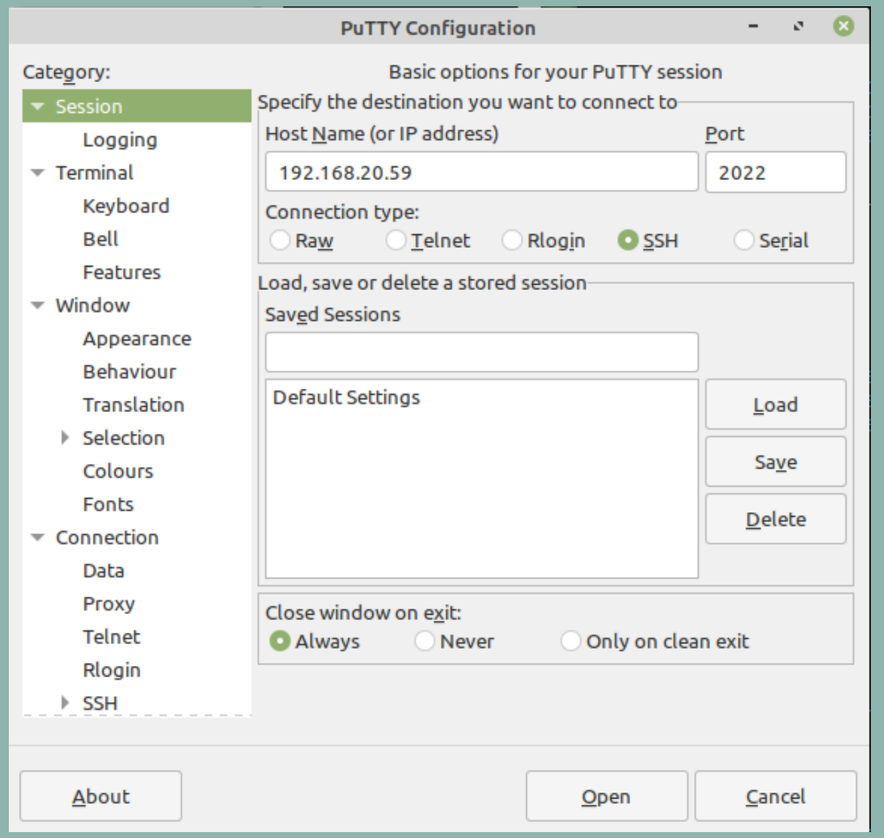




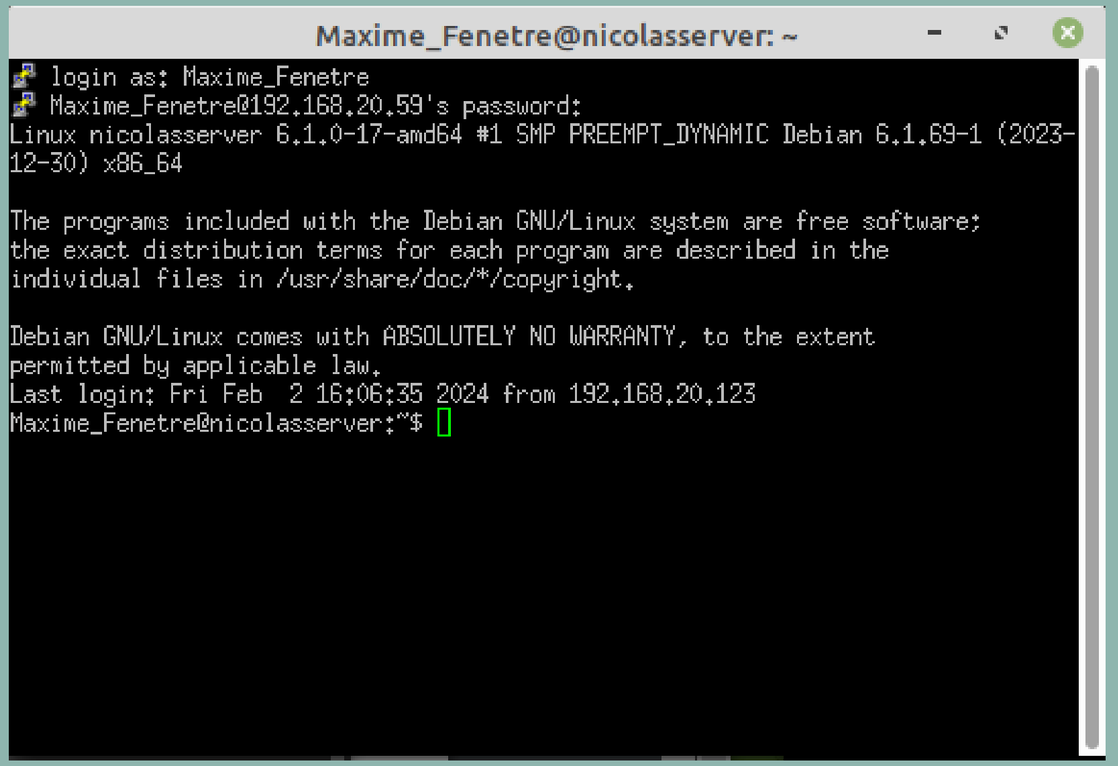
Vous aurez remarqué que le port par défaut du ssh est le port 22 ainsi pour sécuriser un minimum votre connexion ici par exemple nous avons mis le port 2022.
Pour le changer vous devrez modifié le fichier /etc/ssh/sshd_config.

Après avoir modifié cette ligne vous pourrez voir que vous ne pouvez plus vous connectez via le port 22.
Essayons maintenant de nous connecter avec le port 2022.





Et vous voilà de nouveau connectés via Putty



Comme nous l’avons vu précédemment il n’est pas possible de se connecter en root pour remédier à cela nous allons retourner dans le fichier configuration du ssh et modifier les lignes suivantes, qui, comme vous le verrez sont assez explicites.

Il est important de réfléchir avant d’activer cette option surtout réfléchir à quoi va servir votre ssh veillez à ne pas activer cette option s’il n’y en a pas l’utilité.

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes_
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```

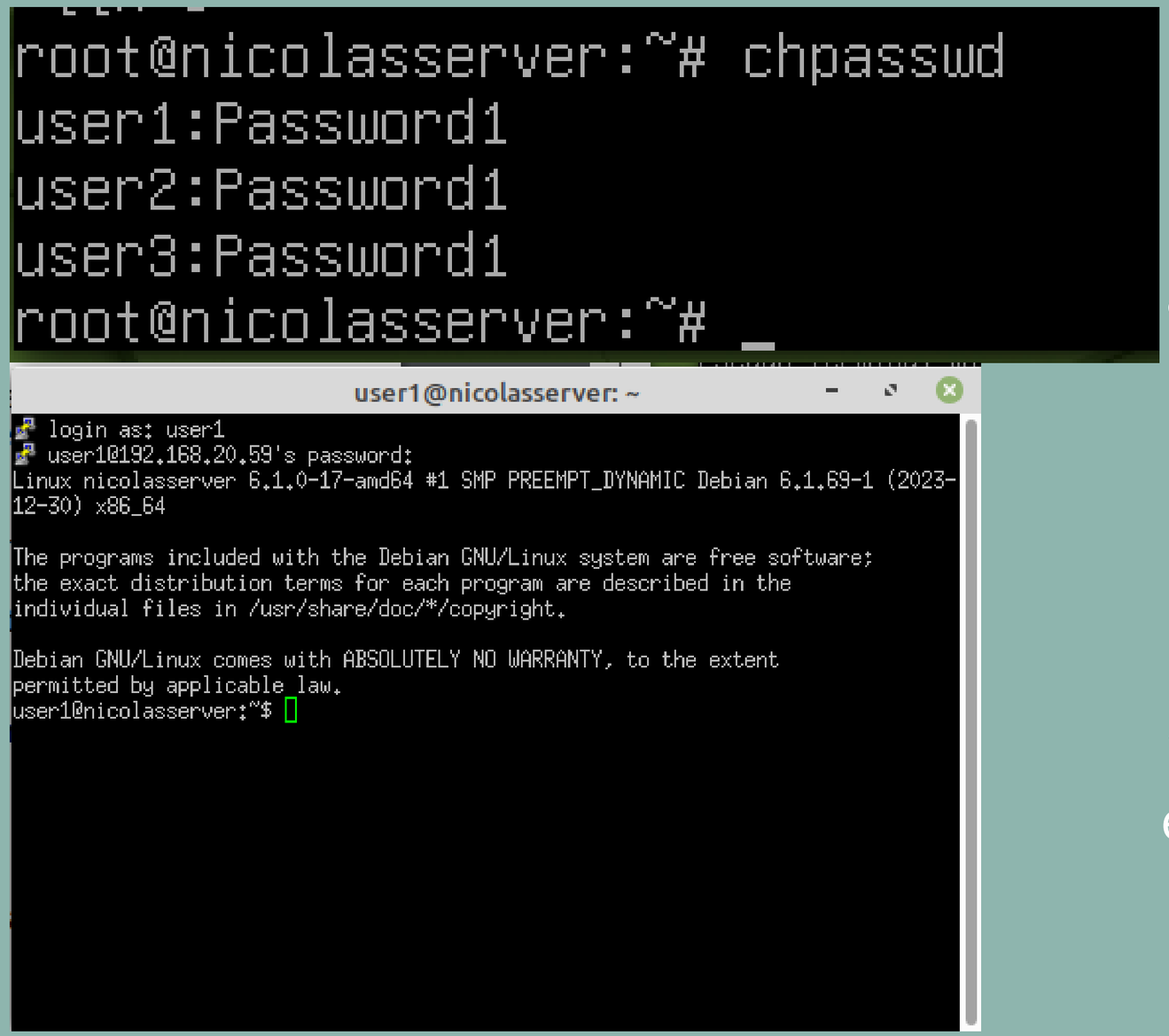
```
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

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

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no
```

Nous allons maintenant voir comment gérer les utilisateurs sur le serveur et sur le client.

```
root@nicolasserver:~# addgroup etudiant
Ajout du groupe « etudiant » (GID 1015)...
Fait.
root@nicolasserver:~# addgroup ssh
Ajout du groupe « ssh » (GID 1016)...
Fait.
root@nicolasserver:~# _
```

```
root@nicolasserver:~# usermod -a -G etudiant user1
root@nicolasserver:~# usermod -a -G ssh user1
root@nicolasserver:~# usermod -a -G ssh user2
root@nicolasserver:~# usermod -a -G etudiant user3
```

Nous allons commencer par creer nos groupes et nos utilisateurs puis nous allons mettre les utilisateurs dans leurs groupes avec la commande usermod -a -G groupe utilisateur.



Vous devrez ensuite modifier les mots de passes de vos utilisateurs avec la commande chpasswd, puis mettez votre_utilisateur:nouveau_mdp autant de fois que d'utilisateur à modifier.

Vous pourrez ensuite voir que votre commande à fonctionner en essayant de vous reconnecter avec vos nouveau mots de passe.

Nous allons maintenant voir comment gérer l’échange de clés publiques.

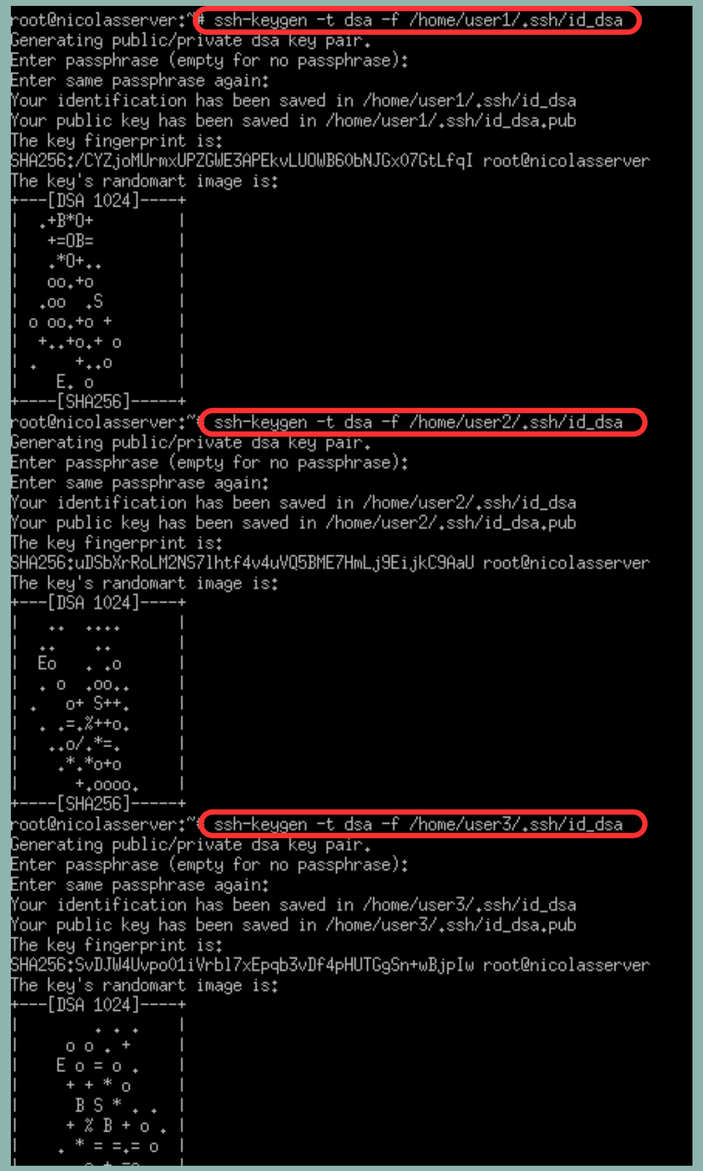
```
root@nicolasserver:/home# cd user1
root@nicolasserver:/home/user1# ls
root@nicolasserver:/home/user1# mkdir .ssh
mkdir: impossible de créer le répertoire « .ssh »: Le fichier existe
root@nicolasserver:/home/user1# mkdir /home/user2/.ssh
root@nicolasserver:/home/user1# mkdir /home/user3/.ssh
root@nicolasserver:/home/user1#
```

Nous allons tout d’abord créer un dossier .ssh dans le dossier de chaque utilisateur.

```
root@nicolasserver:~# mkdir /home/.ssh
```

```
root@nicolasserver:/home# chmod 0770 /home/user1/.ssh
root@nicolasserver:/home# chmod 0770 /home/user2/.ssh
root@nicolasserver:/home# chmod 0770 /home/user3/.ssh
root@nicolasserver:/home# chmod 0770 /home/.ssh
```

Nous allons ensuite faire en sorte que les propriétaires et les membres du groupe aient tous les droits sur leurs dossiers.



Nous allons ensuite générer une clé publique à destination du serveur pour chaque utillisateur avec la commande ssh-keygen -t dsa -f /home/utilisateur/.ssh/id_dsa

```
user10@nicolasserver:~$ cd /home/user1/.ssh
user10@nicolasserver:~/.ssh$ ls
id_dsa  id_dsa.pub
```

Vous verrez que dans votre dossier .ssh se situent 2 fichiers et lorsque nous regardons à quoi ressemblent ces fichiers:

```
root@nicolasserver:~# cat /home/user1/.ssh/id_dsa.pub
ssh-dss AAAAB3NzaC1kc3MAAACBANC3ntagwUtzmLfnoO5pSkaBVDzDfAcaLVdcJLKxnsc67PaiJiem
3p5HGzFQ6lHPQpOmGqZtLOpUnVYen9q90A64MrjK4MOzQsqGY1w1DIsz2GbMlyXTcIecvf2JxAbkvkOe
sCSixJ8Yl/XEP46311RqUWAnhLUP8IK7fvbYeLRHAAAAFQDfypej5ops8Mhu5s1tyCAJYUtECwAAAIAp
Cw2+I1e5ySvZvL/Z3NMpcBE1ZhJFfXbF3Oy7FoweEKBvggjKHpVkkTJjcGP9e+u5XONaDsCX99GkMpbw
MB6o3eyalv8WB+CEGmzy/FD/D+R2KukkfP8QPD1PHGffzfWkKXGoFzgrDal54V6O/3SNPx7/DGXPdW2M
1gD9tEx2ZQAAAIApRbaBN+S8t4puDtNEJB5y4FxIiv5aDvKncmJco4XZOOomlMOCe+YF1AMY1HA8NOTt
l0XCB8JMsNJ7gTUMaCLvxJWoox29OYmd6cJZbYVL4hOU1S/ZLObWsaKoZRKnjYv7VBq9uOOz1k/bJnkL
sKSJiN1urRMq9OZXb3N4GUAN+g== root@nicolasserver
root@nicolasserver:~#
```

```
root@nicolasserver:~# cat /home/user1/.ssh/id_dsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXBOAAAAGAAAABCR4Ppxzt
4EQREhA/cmTK/UAAAAEAAAAAEAAAGxAAAAB3NzaC1kc3MAAACBANC3ntagwUtzmLfnoO5p
SkaBVDzDfAcaLVdcJLKxnsc67PaiJiem3p5HGzFQ6lHPQpOmGqZtLOpUnVYen9q90A64Mr
jK4MOzQsqGY1w1DIsz2GbMlyXTcIecvf2JxAbkvkOesCSixJ8Yl/XEP46311RqUWAnhLUP
8IK7fvbYeLRHAAAAFQDfypej5ops8Mhu5s1tyCAJYUtECwAAAIApCw2+I1e5ySvZvL/Z3N
MpcBE1ZhJFfXbF3Oy7FoweEKBvggjKHpVkkTJjcGP9e+u5XONaDsCX99GkMpbwMB6o3eya
lv8WB+CEGmzy/FD/D+R2KukkfP8QPD1PHGffzfWkKXGoFzgrDal54V6O/3SNPx7/DGXPdW
2M1gD9tEx2ZQAAAIApRbaBN+S8t4puDtNEJB5y4FxIiv5aDvKncmJco4XZOOomlMOCe+YF
1AMY1HA8NOTtlOXCB8JMsNJ7gTUMaCLvxJWoox29OYmd6cJZbYVL4hOU1S/ZLObWsaKoZR
KnjYv7VBq9uOOz1k/bJnkLsKSJiN1urRMq9OZXb3N4GUAN+gAAAfBRVxYS257XGRY8eXnb
3059H2JcsEkUz7zvW47ObsNlqTaVkqKNIkUSzCUYM1EFLOZDOyg5rONPvlYsPngNkfL5xs
2X3ICeTpqHGMbKi5+YAruGjIIS5Il0IdOHUMMlNFNj85QD3dKHQc5I3JPFO9p5hwKNerV2
cGriHT6PUDTpGLUVODrJU3UPcC8D4dOCTi9f3Ug5ZPOTuiNOKGhD495RijSOWNJd9fczSm
P+bAodLvCjBYx15UkimvwW7SlSoByHPDDDsSOzZ21S5cNS4YfCItkEo2UoKGQ6dHeKVAsM
Qu8dU2n++MAoO8QH5R1vn9vrBatjbVaO4SMFFN7yYg6ppR3OLN+UzRcLTqRZmGKim2MvZP
DQqITDyy8VQTe7Rs7Pd3hwJggDgUvKprRaW98s65CrIGzyQxU1mH8slfphMj8eeK31odzq
XkYfxK4X/J9pki1Skb/GfY3sX4tyF9/QAjwJDid94kwNtnpZZLYYDQt92KzCbGrzXO8fdi
8Fgon8HbCEUs4FTPO+Vhbd43DCQ8Y1bA/5kY1KOJ/8DOHJ+7nBYhiBZ2fMmjwFvUM8851C
5c62bL/yatEZXcq2bfDyKHe76Ny9VZaHeVE3ikh5ixZ2UqZWESvYNNZ+eGEOPmYpmQAoQ+
XCy+tmlQschaOO
-----END OPENSSH PRIVATE KEY-----
root@nicolasserver:~#
```

Comme vous pouvez le voir les deux clés sont différentes ce qui est normal la première est la clé publique qui sera partagée avec le serveur alors que la deuxième est une clé privée qui est une clé qui ne sera connue que par la machine qui l'a générée.

```
root@nicolasserver:/home/user1# cd
root@nicolasserver:~# ssh-copy-id -i /home/user1/.ssh/id_dsa.pub root@192.168.20.59
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user1/.ssh/id_dsa.pub"
The authenticity of host '192.168.20.59 (192.168.20.59)' can't be established.
ED25519 key fingerprint is SHA256:eOCsetMLtOIeKKhFSEIuXeKG9118OgVteoVLKDhrc68.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.20.59's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@192.168.20.59'"
and check to make sure that only the key(s) you wanted were added.

root@nicolasserver:~# ssh-copy-id -i /home/user2/.ssh/id_dsa.pub root@192.168.20.59
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/user2/.ssh/id_dsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.20.59's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@192.168.20.59'"
and check to make sure that only the key(s) you wanted were added.

root@nicolasserver:~#
```

```
root@nicolasserver:~# ssh root@192.168.20.59
root@192.168.20.59's password:
Linux nicolasserver 6.1.0-17-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.69-1 (2023-
12-30) 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 Feb  7 15:37:23 2024 from 192.168.60.75
```

Nous allons ensuite pour les utilisateurs concernés envoyés une copie de la clé publique au serveur ssh afin qu'il puisse nous identifier avec la commande:

ssh-copy-id -i /home/utilisateur/.ssh/id_dsa.pub

Pour finaliser cette étape on vous demandera le mot de passe root de la machine.

Pour tester que vous avez bien une connexion entre votre client et votre ssh vous pourrez utiliser la commande ssh utilisateur@adress_ip_serveur (-p port (pas utile si laisser par défaut))

```
  GNU nano 7.2                    /etc/ssh/sshd_config *

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:
AllowGroups root ssh
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Pour finir nous allons sécuriser un peu plus notre connexion en autorisant uniquement les groupes root et ssh à la connexion ainsi tout utilisateur ne faisant pas partis de ces groupes ne pourrons pas se connecter.

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

Ensuite si vous souhaitez faire en sorte que l'authentification ne se fasse plus par mot de passe mais grâce à une clé alors modifier ce paramètre en no.

```
root@nicolasserver:~# ssh user2@192.168.20.59
user2@192.168.20.59's password:
Linux nicolasserver 6.1.0-17-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.69-1 (2023-
12-30) 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.
user2@nicolasserver:~$
```

Pour ensuite tester votre paramétrage essayez de vous connecter avec un utilisateur faisant partie d'un des deux groupes(ici user2 du groupe ssh) et un autre ne faisant partie d'aucun des deux groupes (ici user3 du groupe etudiant) et vous verrez que pour le deuxième la connexion est impossible.

```
user2@nicolasserver:~$ ssh user3@192.168.20.59
hostkeys_find_by_key_hostfile: hostkeys_foreach failed for /home/user2/.ssh/know
n_hosts: Permission denied
The authenticity of host '192.168.20.59 (192.168.20.59)' can't be established.
ED25519 key fingerprint is SHA256:e0CsetMLtOIeKKhFSEIuXeKG91180gVteoVLKDhrc68.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/home/user2/.ssh/known_hosts).
user3@192.168.20.59's password:
Permission denied, please try again.
user3@192.168.20.59's password:
```

Voici votre ssh configuré !