

TP ADMIN A
DISTANCE_SSH

ROMAIN LHEUREUX



La commande "which ssh" est utilisée pour afficher le chemin absolu de l'exécutable SSH sur votre système.

```
root@debiansio:~# which ssh
/usr/bin/ssh
root@debiansio:~# _
```

LA COMMANDE "WHICH SSH"

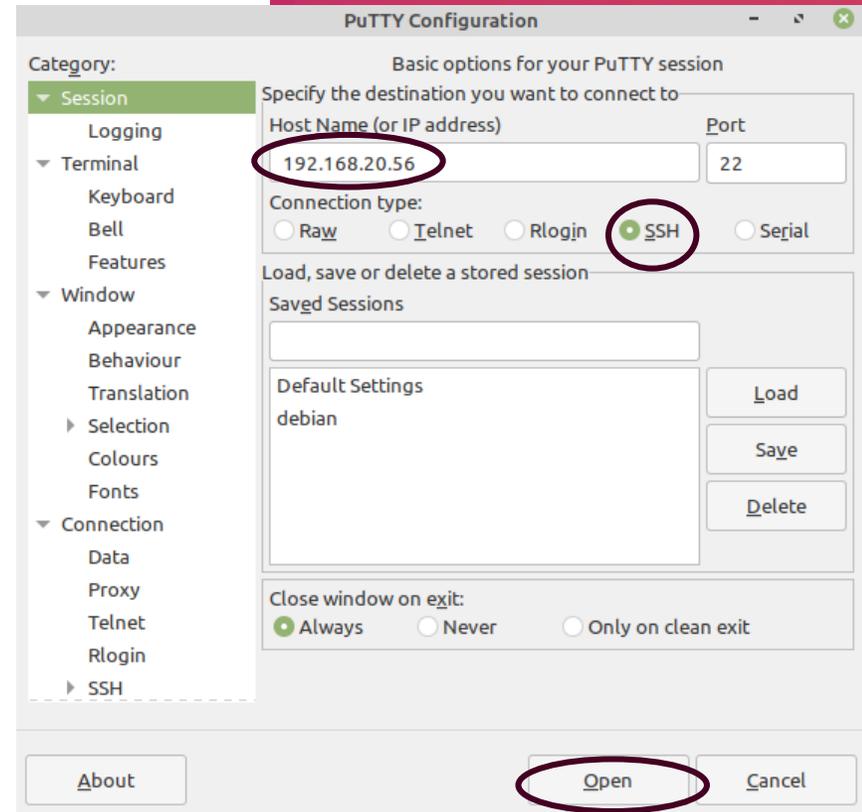
INSTALLER SSH

Commande:

```
« apt install  
openssh-  
server »»
```

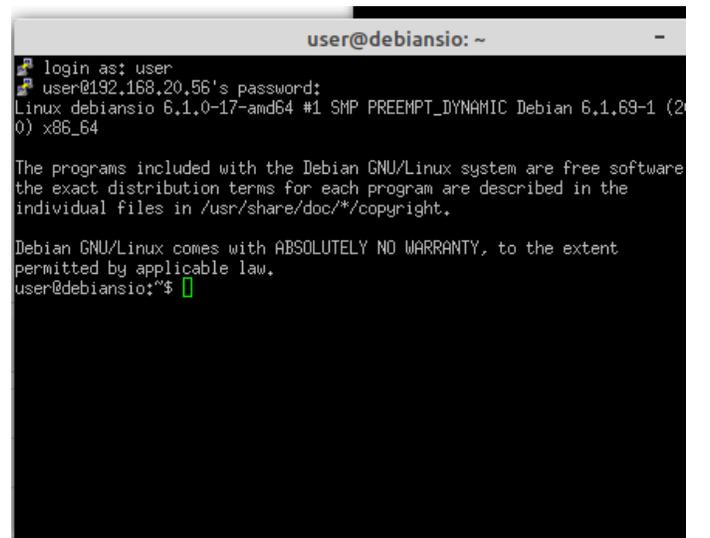
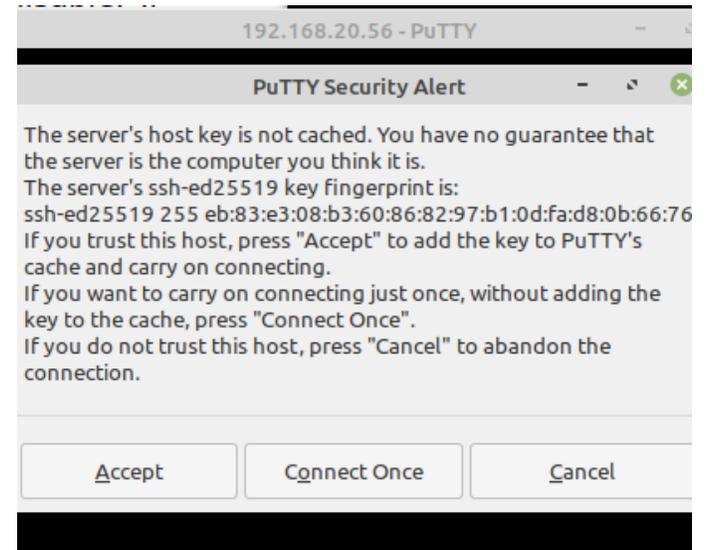
CONNECTION AU SSH

- Pour se connecter au ssh on peut utiliser PuTTY, il nous suffit de saisir l'adresse IP du serveur.



CONNECTION SSH

- Comme nous pouvons le voir, nous sommes connectés au serveur sous l'utilisateur "user".



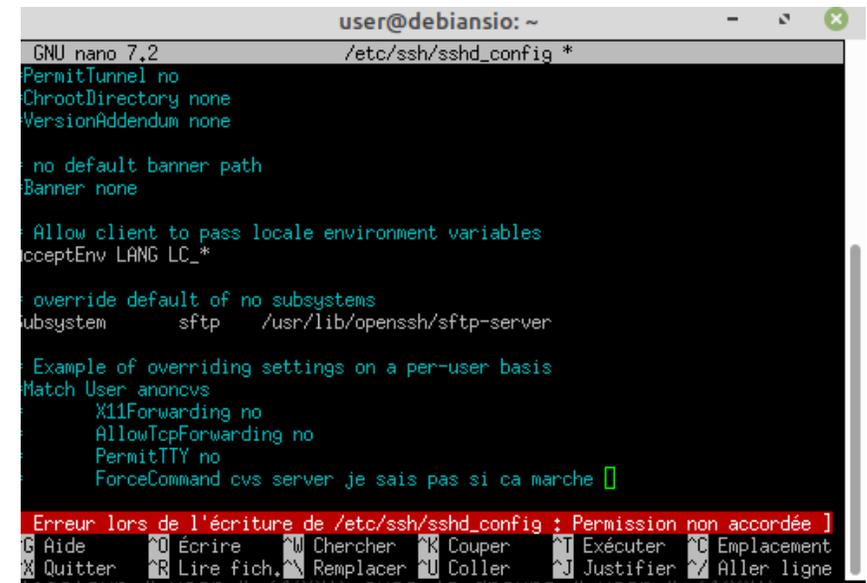
VÉRIFICATION DES DROITS

Commande : `nano /etc/ssh/sshd_config`

Avec cette commande nous pouvons accéder à la configuration du ssh.

Comme vu plus tôt avec nous sommes connectés sous l'utilisateur `user`. Et comme le montre le screen. Avec l'utilisateur `user` il n'est pas possible de modifier la configuration du ssh.

```
user@debiansio:~$ nano /etc/ssh/sshd_config
user@debiansio:~$ nano /etc/ssh/sshd_config
```



```
user@debiansio: ~
GNU nano 7.2 /etc/ssh/sshd_config *
PermitTunnel no
ChrootDirectory none
VersionAddendum none

# no default banner path
Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem sftp /usr/lib/openssh/sftp-server

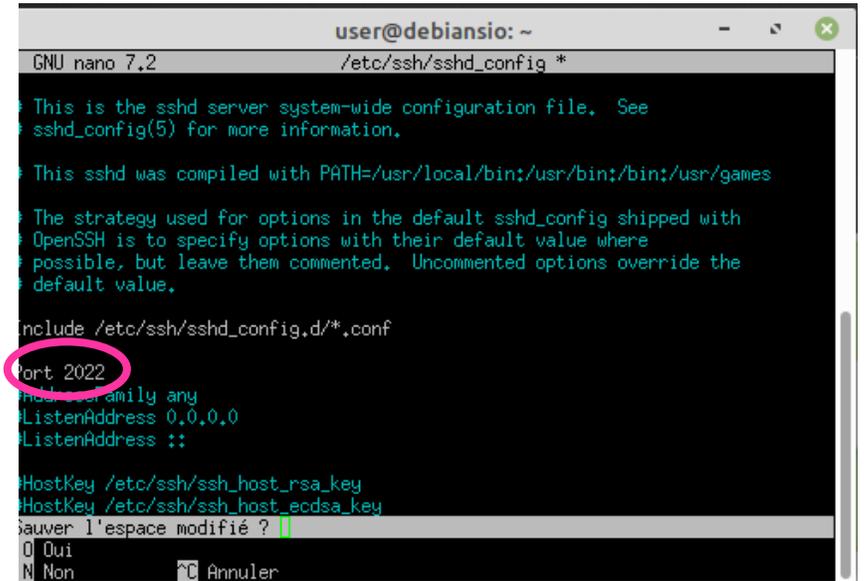
# Example of overriding settings on a per-user basis
Match User anoncvs
    X11Forwarding no
    AllowTcpForwarding no
    PermitTTY no
    ForceCommand cvs server je sais pas si ca marche []

Erreur lors de l'écriture de /etc/ssh/sshd_config : Permission non accordée
G Aide ^O Écrire ^W Chercher ^K Couper ^T Exécuter ^C Emplacement
X Quitter ^R Lire fich. ^M Remplacer ^U Coller ^J Justifier ^_ Aller ligne
```

CONFIGURATION SSH

- Pour pouvoir modifier la configurations du ssh il nous faut passer en admin. Pour se faire nous allons utiliser la commande “SU”
- Ensuite une fois dans le nano nous pouvons changer le port du serveur ce qui peut augmenter la sécurité en rendant notre serveur moins visible pour les attaquants qui ciblent généralement les ports standard.
- Pour se faire trouver dans le nano « Port » et enlever le # pour que la modifications soit pris en compte

```
user@debiansio:~$ su
Mot de passe :
su: Échec de l'authentification
user@debiansio:~$ su
Mot de passe :
root@debiansio:/home/user# nano /etc/ssh/sshd_config
```



```
user@debiansio: ~
GNU nano 7.2 /etc/ssh/sshd_config *
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
include /etc/ssh/sshd_config.d/*.conf
Port 2022
#Port 22
#ListenAddress 0.0.0.0
#ListenAddress ::
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
sauver l'espace modifié ?
O Oui
N Non Ctrl Annuler
```

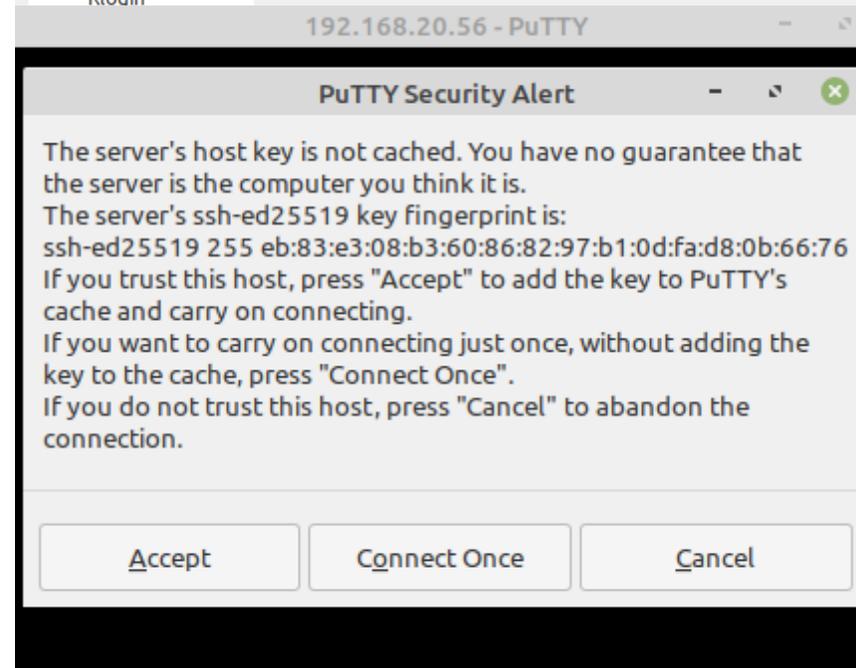
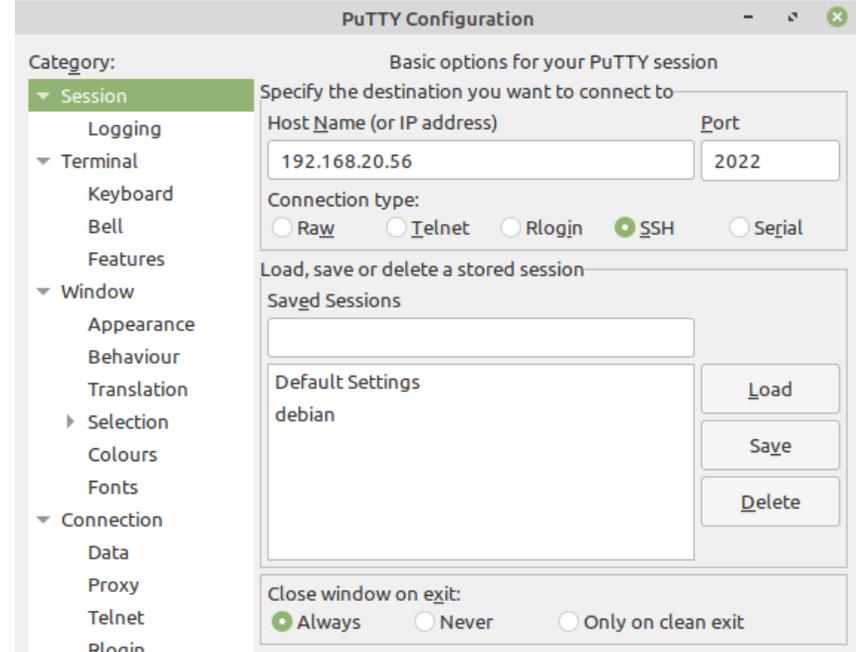
CONFIGURATION SSH

- Pour appliquer les modifications il faut effectuer la commande « service ssh stop » puis « service ssh start ».



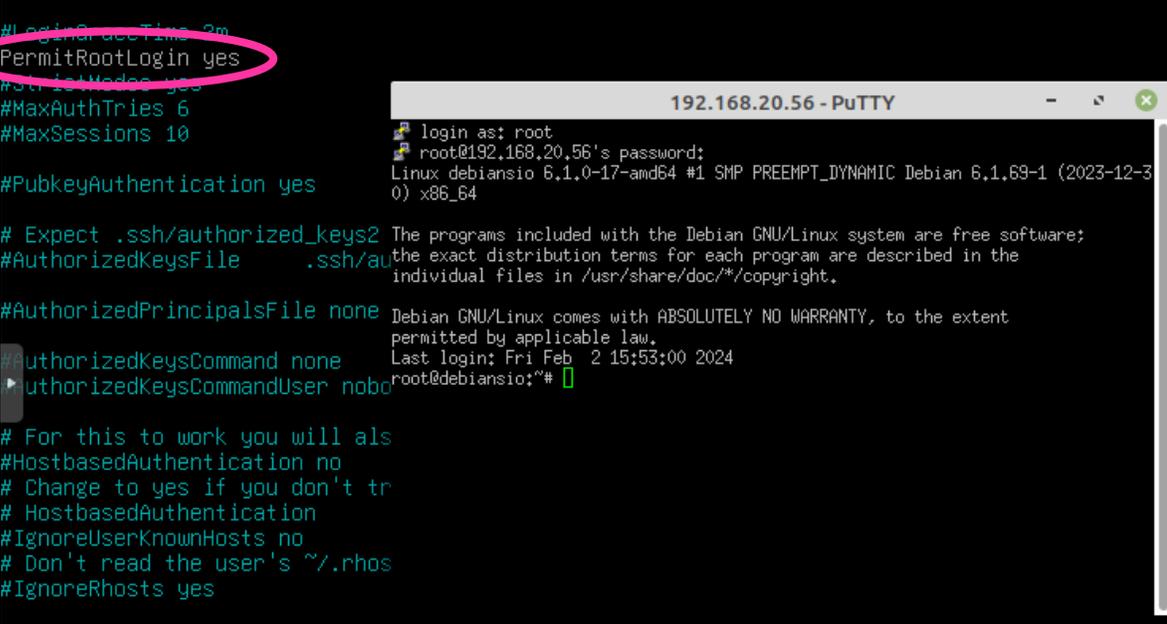
CONNECTION AU SERVEUR SSH

- Pour vérifier que nos modifications sont bien pris en compte nous allons une nouvelle fois nous connecter avec le port modifier,
- Pour revenir au port 22 il nous suffit de retourner dans le nano et mettre un # devant port



CONFIGURATION SSH

- Maintenant nous allons retourner dans le “nano” de notre serveur pour permettre de se connecter directement en root.
- Pour se faire il nous faut trouver “PermitRootLogin”, Une fois trouver il faut retirer le # et mettre yes.



The image shows a terminal window with two panes. The left pane displays the configuration of the /etc/ssh/sshd_config file. The line `PermitRootLogin yes` is circled in pink. Other visible configuration lines include `#PubkeyAuthentication yes`, `# Expect .ssh/authorized_keys2`, `#AuthorizedKeysFile .ssh/au`, `#AuthorizedPrincipalsFile none`, `#AuthorizedKeysCommand none`, `#AuthorizedKeysCommandUser nobu`, `# For this to work you will als`, `#HostbasedAuthentication no`, `# Change to yes if you don't tr`, `# HostbasedAuthentication`, `#IgnoreUserKnownHosts no`, `# Don't read the user's ~/.rhos`, and `#IgnoreRhosts yes`. The right pane shows a PuTTY terminal session for IP 192.168.20.56. It displays the login prompt, password entry, system information (Linux debiansio 6.1.0-17-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.69-1 (2023-12-30) x86_64), a warning about Debian GNU/Linux warranty, the last login time (Fri Feb 2 15:53:00 2024), and the root shell prompt `root@debiansio:~#`.

UTILISATEUR SSH

- Nous allons créer des groupes et des utilisateurs pour notre serveur ssh
- Commande:
- Création d'utilisateur "adduser"
- Création de groupe "groupadd"
- Mettre des utilisateurs dans des groupes "usermod -a -G "nom du groupe" "nom de l'utilisateur".

```
192.168.20.56 - PuTTY
Autre []:
Cette information est-elle correcte ? [0/n]
Ajout du nouvel utilisateur « user2 » aux groupes supplémentaires « user
Ajout de l'utilisateur « user2 » au groupe « users » ...
root@debiansio:~# adduser user3
Ajout de l'utilisateur « user3 » ...
Ajout du nouveau groupe « user3 » (1005) ...
Ajout du nouvel utilisateur « user3 » (1005) avec le groupe « user3 » (1
Création du répertoire personnel « /home/user3 » ...
Copie des fichiers depuis « /etc/skel » ...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
Modifier les informations associées à un utilisateur pour user3
Entrer la nouvelle valeur, ou appuyer sur ENTER pour la valeur par défaut
NOM []:
Numéro de chambre []:
Téléphone professionnel []:
Téléphone personnel []:
Autre []:
Cette information est-elle correcte ? [0/n]
Ajout du nouvel utilisateur « user3 » aux groupes supplémentaires « user
Ajout de l'utilisateur « user3 » au groupe « users » ...
root@debiansio:~#
```

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

CONFIGURATION SSH

- Pour mettre les mots de passe vide il nous faut trouver “permitEmptyPasswords” puis changer le “no” par “yes”.

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

SERVEUR SSH

- la commande `chpasswd` permet de mettre à jour rapidement et efficacement les mots de passe des utilisateurs.

```
root@debiansio:~# chpasswd
chpasswd : ligne 1 : il manque le nouveau mot de passe
chpasswd : ligne 2 : il manque le nouveau mot de passe
azerty
chpasswd : ligne 3 : il manque le nouveau mot de passe
user1:Password1
user2:Password1
user3:Password1
```

SERVEUR SSH

Pour chaque utilisateur nous allons créer un dossier `.ssh` grâce à la commande `mkdir`. Il va principalement stocker les clés d'authentification SSH et les configurations spécifiques à SSH pour cet utilisateur,

Pour renforcer la sécurité nous allons utiliser la commande « `chmod 0770 ~/.ssh` » ce qui a pour effet de limiter l'accès aux seuls propriétaires et membres du groupe, renforçant ainsi la protection des clés d'authentification SSH stockées à l'intérieur.

```
-bash: user2 : commande introuvable
root@debiansio:/home# cd user2
root@debiansio:/home/user2# mkdir .ssh
```

```
root@debiansio:~# cd /home
root@debiansio:/home# cd user1
root@debiansio:/home/user1# chmod 0770 ~/.ssh
root@debiansio:/home/user1# cd ..
root@debiansio:/home# cd user2
root@debiansio:/home/user2# chmod 0770 ~/.ssh
root@debiansio:/home/user2# cd ..
root@debiansio:/home# cd user3
root@debiansio:/home/user3# chmod 0770 ~/.ssh
root@debiansio:/home/user3#
```

SERVEUR SSH

- Commande: `ssh-keygen -t dsa -f ~/.ssh/id_dsa`.
- Cette commande nous permet de générer des clés. Ce qui est crucial pour sécuriser cela permet une authentifications sécurisée
- Nous allons faire cette commande pour chaque utilisateur.
- Il est normal que ces deux fichiers soient différents, car l'un est une clé privée et l'autre est une clé publique. La clé privée est gardée secrète et n'est pas partagée, tandis que la clé publique est partagée avec les serveurs distants. même si la clé publique est divulguée, la clé privée reste secrète, ce qui permet de vérifier l'identité de l'utilisateur.

```
root@debiansio:~# ssh-keygen -t dsa -f /home/user2/,.ssh/id_dsa
Generating public/private dsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user2/,.ssh/id_dsa
Your public key has been saved in /home/user2/,.ssh/id_dsa.pub
The key fingerprint is:
SHA256:gMYgU4y5kL/6Ko1mveh27zK7JX3CUUIocrlNpIdxtMA root@debiansio
The key's randomart image is:
+----[DSA 1024]-----+
|oB=+,                |
|*+EXoo              |
|o+++*+++           |
| . oo. o.          |
|   + + S           |
| . o .             |
| +. = +            |
|++o=o o            |
|B=o+Oo             |
+----[SHA256]-----+
root@debiansio:~#
```

```
root@debiansio:~# ssh-keygen -t dsa -f /home/user1/,.ssh/id_dsa
Generating public/private dsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user1/,.ssh/id_dsa
Your public key has been saved in /home/user1/,.ssh/id_dsa.pub
The key fingerprint is:
SHA256:+VhHuRCCes7Ik4gnAGX5pm2JtQW5YzzFaI+gjFKTSXU root@debiansio
The key's randomart image is:
+----[DSA 1024]-----+
|+.o.= .+.          |
|o+o=B E .+.        |
|++**o0 .o          |
|+oo,@ o .o .       |
|o X * S .o         |
| B = +.            |
| o .+.             |
|                    |
+----[SHA256]-----+
root@debiansio:~#
```

SERVEUR SSH

Avec la commande nano
~/.ssh/authorized_keys sert à voir que
les clés publiques car Les clés privées ne
sont pas comprises car leur présence
compromettrait la sécurité.

```
root@debiansio:~# nano ~/.ssh/authorized_keys
GNU nano 7.2 /root/.ssh/authorized_keys
ssh-dss AAAAB3NzaC1kc3MAAACBAOPCKuClwrIs5YJxq4AARs5EvaQjzxM8rAHAjVr3OxpPPhcmSRh>
ssh-dss AAAAB3NzaC1kc3MAAACBAI8/Q5eFwNSMTaHYaWF9fc7qjlYuIBzUrj0n5nHL6brEYehZvZ3>
ssh-dss AAAAB3NzaC1kc3MAAACBAIiHWc7JHFRBkHOv9Tee6HhuT0/QamCdAbpTDRWjwMLd7g9hk0s>
```

SERVEUR SSH

- Maintenant nous allons voir si nous avons une connection en se connectant sur un utilisateur pour se faire nous allons utiliser la commande `ssh user_name@adressesIPserveurSSH -p port`

```
root@debiansio:~# ssh user1@192.168.20.56 -p 22
user1@192.168.20.56's password:
Linux debiansio 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.
user1@debiansio:~$
```

SERVEUR SSH

Maintenant nous allons configurer les accès au serveur ssh
pour se faire nous avons les commandes:

AllowUsers nom_utilisateur1 ect

AllowGroups nom du groupe1 ect

```
# Authentication;  
AllowUsers user1 user2 user3  
AllowGroups root ssh
```

TEST

```
sisr-6@sisr-6:~$ ssh user1@192.168.20.56
user1@192.168.20.56's password:
Linux debiansio 6.1.0-17-amd64 #1 SMP PREEMPT_DYNAMIC Deb
0) x86_64

The programs included with the Debian GNU/Linux system ar
the exact distribution terms for each program are describ
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to th
permitted by applicable law.
Last login: Mon Feb  5 14:52:26 2024 from 192.168.20.56
user1@debiansio:~$
```

```
6@sisr-6:~$ ssh user3@192.168.20.56
@192.168.20.56's password:
t debiansio 6.1.0-17-amd64 #1 SMP PREEMPT_DYNAMIC
6_64

programs included with the Debian GNU/Linux syst
exact distribution terms for each program are de
ividual files in /usr/share/doc/*/copyright.

n GNU/Linux comes with ABSOLUTELY NO WARRANTY,
tted by applicable law.
```