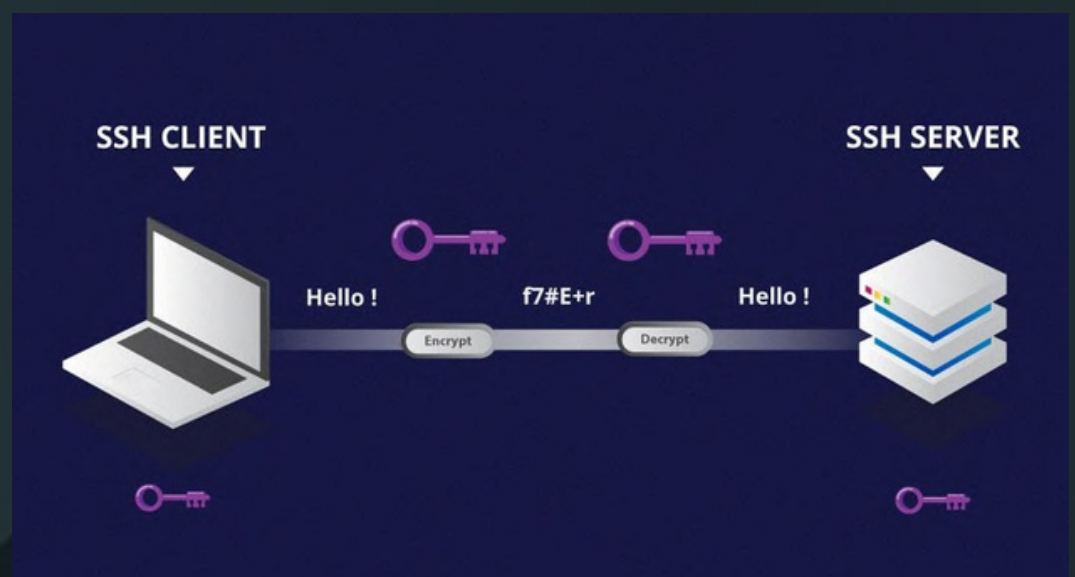


Installation et Configuration de **SSH** sur **Debian**



1. OpenSSH est-il présent sur le serveur ?

- 1) Ouvrir le terminal sur Debian.
- 2) Se mettre en SuperUtilisateur avec la commande "su".
Puis taper le mot de passe.

```
su
```

```
ethan1s@debian-12-Ethan1s:~$ su
Mot de passe :
root@debian-12-Ethan1s:/home/ethan1s#
```

Pour savoir si notre serveur Linux a déjà OpenSSH d'installé, voici la commande à saisir sous Debian :

```
apt-cache policy openssh-server
```

Voici ce que le serveur renvoi :

```
root@debian-12-Ethan1s:/home/ethan1s# apt-cache policy openssh-server
openssh-server:
  Installé : (aucun)
  Candidat : 1:9.2p1-2+deb12u4
  Table de version :
    1:9.2p1-2+deb12u4 500
      500 http://deb.debian.org/debian bookworm/main amd64 Packages
    1:9.2p1-2+deb12u3 500
      500 http://security.debian.org/debian-security bookworm-security/main amd64 Packages
root@debian-12-Ethan1s:/home/ethan1s#
```

Ici, nous pouvons voir "Installé : " puis une inscription "aucun", Cela signifie qu'il n'y a aucune version de SSH installée (en rouge).

En dessous, on constate que le système propose plusieurs versions d'OpenSSH pour l'installation sur notre machine (en vert).

Une autre façon de vérifier l'installation d'OpenSSH aurait été de vérifier que le fichier /etc/ssh/sshd_config est bien présent sur le système. Il s'agit du fichier de configuration du serveur SSH sur les distributions Linux :

```
ls -al /etc/ssh/sshd_config
```

Si le serveur répond cela, c'est que la configuration n'est pas présente, et donc que le serveur SSH n'est pas installé :

```
root@debian-12-Ethan1s:/home/ethan1s# ls -al /etc/ssh/sshd_config
ls: impossible d'accéder à '/etc/ssh/sshd_config': Aucun fichier ou dossier de ce type
root@debian-12-Ethan1s:/home/ethan1s#
```

2.Installation de OpenSSH

Si ce n'est pas le cas, passer en mode super utilisateur avec la commande "su", puis tapez le mot de passe.

```
su
```

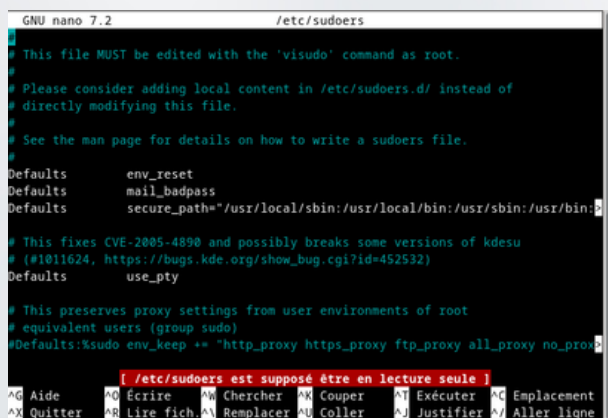
```
ethan1s@debian-12-Ethan1s:~$ su
Mot de passe :
root@debian-12-Ethan1s:/home/ethan1s#
```

Commencer l'installation en tapant: `sudo apt-get update`

```
root@debian-12-Ethan1s:/home/ethan1s# sudo apt-get update
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://deb.debian.org/debian bookworm-updates InRelease
Atteint :3 http://security.debian.org/debian-security bookworm-security InRelease
Lecture des listes de paquets... Fait
```

ensuite, rentrer: `nano /etc/sudoers`

On obtient ceci:



Scroller jusqu'à trouver cette ligne (flèche bleu)

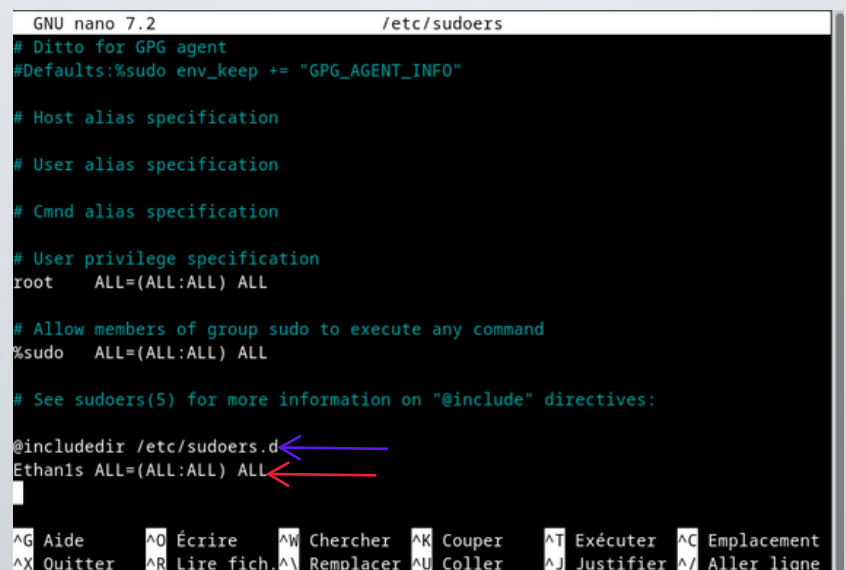
Rajouter cette ligne (flèche rouge):

```
Nom_de_l'utilisateur_machine ALL=(ALL:ALL) ALL
```

Le Nom_de_l'utilisateur_machine doit être remplacé par le nom complet de votre utilisateur, comme affiché au démarrage de votre machine Linux.

Dans notre cas,

Nom_de_l'utilisateur_machine = Ethan1s



Enregistrer ensuite les modifications en faisant Ctrl+X, puis valider en appuyant sur oui ou o.

Enfin, redémarrer votre VM. Pour le faire, rechercher le bouton power puis cliquer dessus.

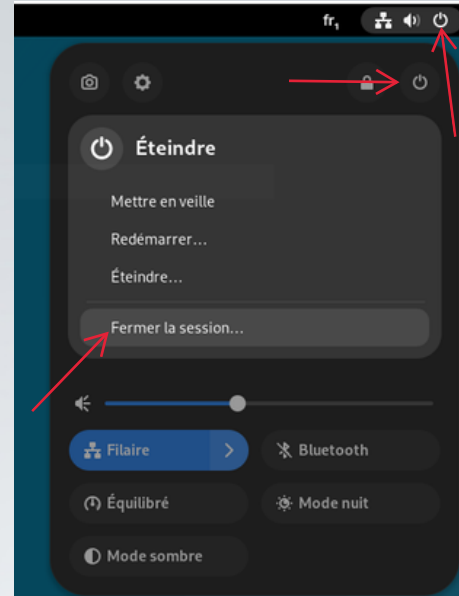
Un panneau déroulant vas s'afficher.

Cliquer sur le bouton power situé sur ce anneau déroulant.

Un panneau déroulant avec la mention "Eteindre" ou "shutdown" vas s'afficher.

Cliquer sur le bouton "Fermer la session" ou "logout"

Dès que la machine à redémarrer, loguez-vous à votre session et passer à la suite.



Dès que vous êtes dans votre session et que vous avez ouvert votre terminal, tapez cette commande:

```
sudo apt install openssh-server -y
```

Voilà OpenSSH est maintenant installé.

3. Gestion basiques d'OpenSSH

Nous pouvons voir l'état du service SSH avec la commande "systemctl", le serveur est-il en fonctionnement (actif, activé) ou non ? :

```
sudo systemctl status ssh
```

Voici ce que l'on pourra voir si le serveur est effectivement déjà en route :

```
ethan1s@debian-12-Ethan1s:~$ sudo systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-02-17 23:45:18 CET; 25min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 9597 (sshd)
     Tasks: 1 (limit: 2243)
    Memory: 1.7M
       CPU: 14ms
   CGroup: /system.slice/ssh.service
           └─9597 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Sinon, il suffira de le démarrer avec la commande suivante :

```
sudo systemctl start ssh
```

Par extension, pour arrêter le service il faudra utiliser la commande suivante :

```
sudo systemctl stop ssh
```

Et pour le redémarrer :

```
sudo systemctl restart ssh
```

Si vous le voulez, il est possible de faire démarrer automatiquement SSH au boot :

```
sudo systemctl enable ssh
```

Si vous le voulez, désactivez le démarrage automatiquement SSH au boot :

```
sudo systemctl disable ssh
```

Je ne l'ai pas encore mentionné, mais le service SSH écoute par défaut sur le port 22 en TCP. Nous pouvons alors voir, sur notre serveur Linux, que le service est bien en écoute sur ce port :

```
sudo ss -tlnp | grep ssh
```

Ici, la commande "ss" (anciennement connue sous le nom de "netstat") permet de visualiser l'état des ports et des connexions du serveur sous les systèmes Unix.

L'option "-l" permet de ne lister que les ports en écoute, l'option "-n" permet d'afficher les ports de manière numérique, et non leur translation habituelle (exemple "telnet" à la place de "23") et l'option "-t" permet de ne lister que les ports TCP, l'option "-p" permet enfin de lister les processus derrière chaque ports :

```
ethan1s@debian-12-Ethan1s:~$ sudo ss -tlnp | grep ssh
[sudo] Mot de passe de ethan1s :
LISTEN 0      128          0.0.0.0:22      0.0.0.0:*      users:(("sshd",pid=9597,fd=3))
LISTEN 0      128          [::]:22        [::]:*         users:(("sshd",pid=9597,fd=4))
ethan1s@debian-12-Ethan1s:~$
```

Ici, on voit donc que le port 22 est bien écoute, en IPv4 ("*:22") et en IPv6 (":::22"). On voit également que c'est le processus "sshd" qui est derrière, aucun doute, il s'agit bien de notre serveur SSH.

4. Première connexion client-serveur SSH

Après avoir brièvement vu la partie serveur, nous allons passer à la partie client et plus précisément l'établissement de notre première connexion SSH client-serveur

Très bien, passons maintenant au côté client. Nous ferons simples dans un premier temps et allons établir une connexion SSH sur un serveur Linux depuis un client Windows avec le logiciel "Putty", nous nous en servirons beaucoup durant ce cours, vous pourrez le télécharger sur la page suivante : <http://www.putty.org/>

Pour connaître l'adresse IP de votre debian tapez la commande "ip address".

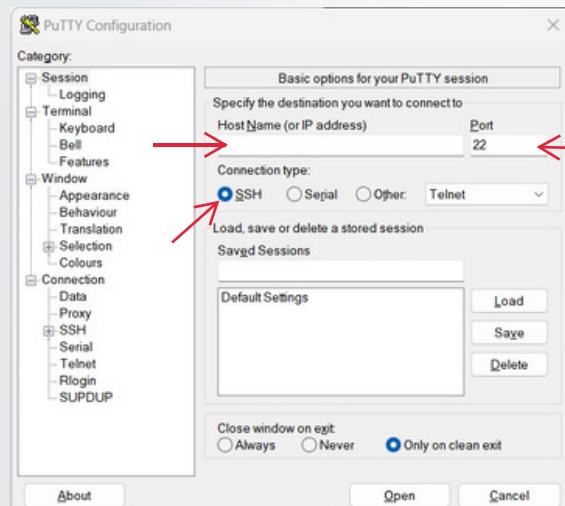
Dans l'exemple l'IP est 192.168.171.133

Vérifiez bien dans les paramètres de la Debian que le réseau est en DHCP et pas en statique !

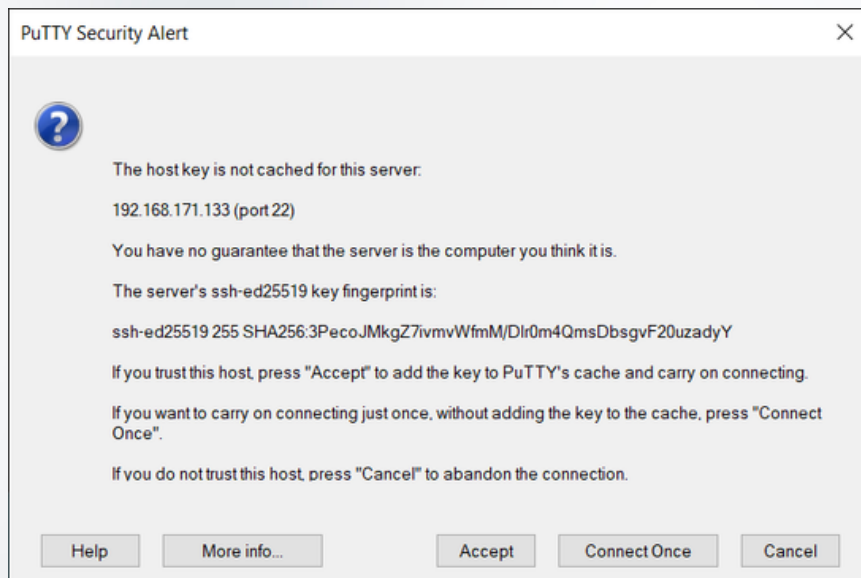
Et sur VM Ware, dans les paramètres réseau de la machine virtuelle Debian, choisir "NAT".

```
root@debian-12-Ethan1s:/home/ethan1s# ip address
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
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 00:0c:29:50:40:e4 brd ff:ff:ff:ff:ff:ff
   altname enp2s1
   → inet 192.168.171.133/24 brd 192.168.171.255 scope global dynamic noprefixroute ens33
       valid_lft 1168sec preferred_lft 1168sec
   inet6 fe80::20c:29ff:fe50:40e4/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
root@debian-12-Ethan1s:/home/ethan1s#
```

- Ouvrir Putty
- Choisir SSH
- Port 22
- Saisir l'adresse IP de la Debian



Comme vous pouvez le voir, Putty peut être utilisé pour établir des connexion en SSH, mais également en Serial, Telnet, etc.. Concentrons-nous sur le SSH. On va donc saisir l'adresse IP ou le nom DNS de notre serveur SSH (notre serveur Linux, configuré précédemment) dans le champ "Host Name", on laisse le port 22 en port cible, celui-ci étant le port par défaut de SSH. On s'assure ensuite que SSH est coché dans "Connection type" puis on clique sur "Open" pour initialiser la connexion, ce petit message va alors apparaître :



Ce message nous avertit du fait que la clé du serveur SSH n'est pas "cached in the registry", c'est-à-dire enregistrée sur notre ordinateur. Cela signifie qu'il faut valider le téléchargement de la clé pour établir la connexion, c'est-à-dire accepter la clé. Nous détaillerons se processus de chiffrement et d'échange de clés un peu plus loin. Pour valider l'établissement de la connexion, cliquez sur "Oui".

On arrive alors sur un Shell, dans lequel vous pourrez vous authentifier avec les identifiants... de votre serveur Linux ! Et oui, vous êtes à présent connecté en SSH sur votre serveur Linux distant.

```
ethan1s@debian-12-Ethan1s: ~/Bureau
login as: ethan1s
ethan1s@192.168.171.133's password:
Linux debian-12-Ethan1s 6.1.0-31-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.128-1 (
2025-02-07) 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.
ethan1s@debian-12-Ethan1s:~$
```

L'installation de SSH sur Debian est touche à sa fin !