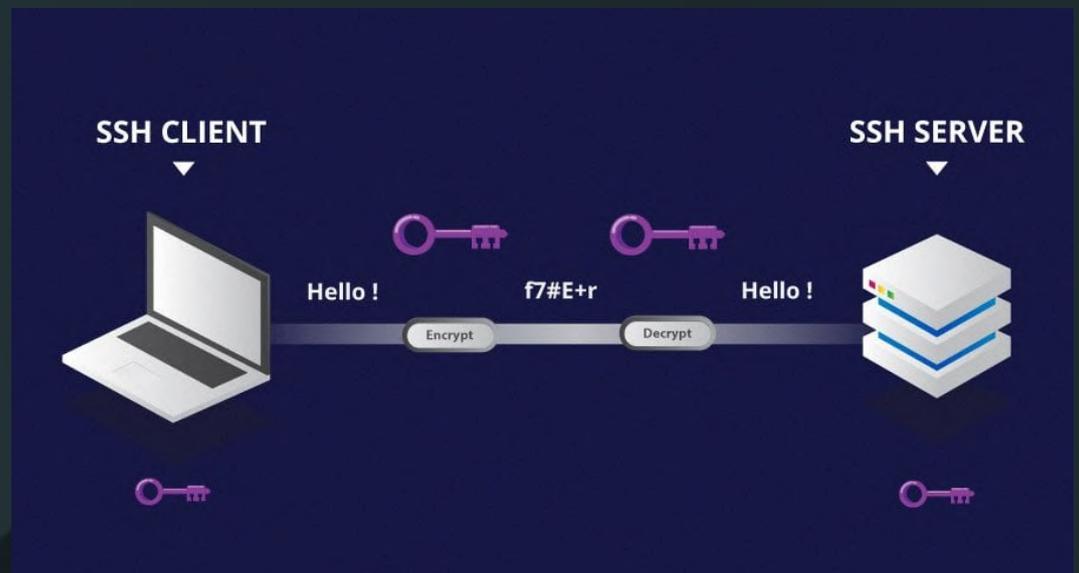


# Installation et Configuration de SSH sur Debian



# 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
```

```
nathan@debian:~$ su
Mot de passe :
root@debian:/home/nathan#
```

Pour savoir si notre serveur Linux a déjà *OpenSSH* d'installé, ce qui est le cas normalement. Sous *Debian*, voici la commande à saisir :

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

Voici ce que le serveur renvoi :

```
root@debian:/home/nathan# apt-cache policy openssh-server
openssh-server:
  Installé : 1:9.2p1-2 ←
  Candidat : 1:9.2p1-2
  Table de version :
  *** 1:9.2p1-2 500
      500 http://deb.debian.org/debian bookworm/main amd64 Packages
      100 /var/lib/dpkg/status
root@debian:/home/nathan#
```

Ici, nous pouvons voir "Installé : " puis une version, nous voyons alors que la version installée d'OpenSSH est la version 9.2 sur notre serveur, et **qu'OpenSSH est bien installé**.

Une autre façon de vérifier l'installation d'OpenSSH aurait été de vérifier quel 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 est bien présente, et donc que le serveur est bien installé :

```
root@debian:/home/nathan# ls -al /etc/ssh/sshd_config
-rw-r--r-- 1 root root 3223  8 févr. 2023 /etc/ssh/sshd_config
```

## 2. Installation de OpenSSH

---

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

```
su
```

```
nathan@debian:~$ su
Mot de passe :
root@debian:/home/nathan#
```

Commencer l'installation ensuite :

```
apt-get update
apt-get install openssh-server
```

Voilà OpenSSH est maintenant installé.

## 3. Gestion basiques d'OpenSSH

---

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

```
systemctl status sshd
```

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

```
root@debian:/home/nathan# systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Fri 2023-10-06 17:58:09 CEST; 29min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 573 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 594 (sshd)
     Tasks: 1 (limit: 2285)
    Memory: 9.0M
       CPU: 138ms
   CGroup: /system.slice/ssh.service
           └─594 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

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

```
systemctl start sshd
```

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

```
systemctl stop sshd
```

Et pour le redémarrer :

```
systemctl restart sshd
```

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 :

```
ss -lntp |grep "22"
```

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 :

```
State Recv-Q Send-Q Local Address:Port Peer
LISTEN 0      128  *:22  *:*      users:(("sshd",1311,3))
LISTEN 0      128  :::22 :::*     users:(("sshd",1311,4))
```

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 simple 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 servons 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 a".

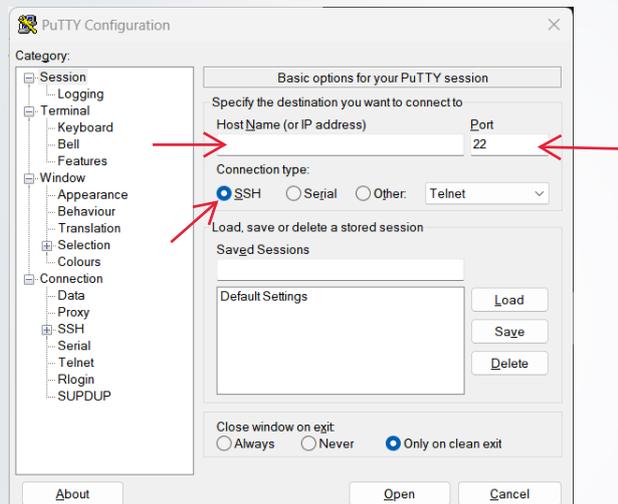
Dans l'exemple l'ip est 192.168.0.43

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

Et sur VirtualBox, dans les paramètres réseau de la machine virtuelle Debian, choisir "**accès par pont**".

```
root@debian:/home/nathan# 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:11:4d:93 brd ff:ff:ff:ff:ff:ff
    → inet 192.168.0.43/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
        valid_lft 39737sec preferred_lft 39737sec
    inet6 2a01:e0a:1e6:4040:94fc:21ea:416a:6915/64 scope global dynamic noprefixroute
        valid_lft 86141sec preferred_lft 86141sec
    inet6 fe80::b93f:6580:84e4:2ca6/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

- 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 connexions 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 ce 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.

```
nathan@debian: ~  
login as: nathan  
nathan@192.168.0.43's password:  
Linux debian 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1 (2023-09-07)  
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.  
Last login: Fri Oct 6 19:07:49 2023 from 192.168.0.27  
nathan@debian:~$
```

L'installation de SSH sur Debian est maintenant terminée !