## Les clés SSH

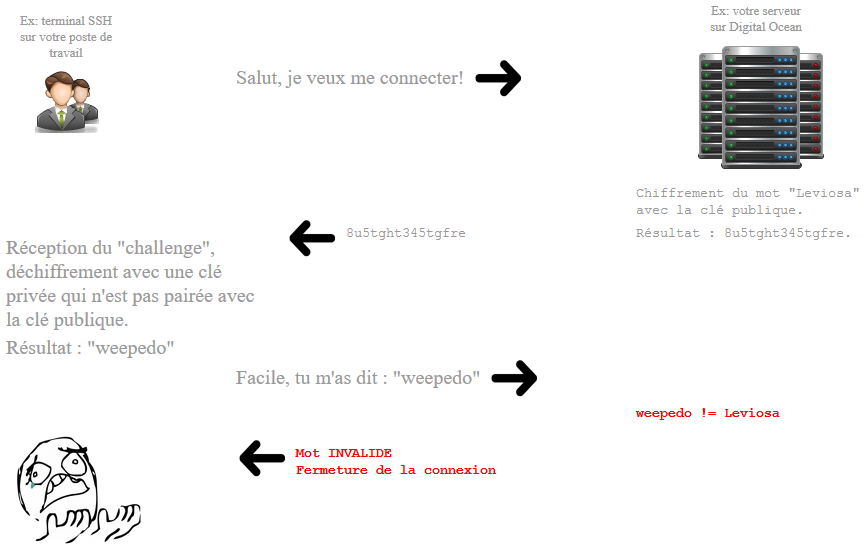
Permet la communication sécurisée d'une machine à une autre, basée sur une paire de clés publique/privée. L'utilisation de mot de passe n'est donc (habituellement) plus nécessaire pour établir la communication.

Visuellement, le processus d'authentification par clés fonctionne comme ceci (pour la version 1 du protocole)

Cas valide



Cas invalide…



## Mise en place d'une authentification par clés SSH

1- Création des clés  
Sur la machine locale, faire:

**[usr@machine]$ ssh-keygen -t rsa**

Generating public/private rsa key pair.   
Enter file in which to save the key (/home/usr/.ssh/id\_rsa): Appuyer sur <Enter>   
passphrase (empty for no passphrase): Appuyer sur <Enter> ou définir une passphrase   
Enter same passphrase again: Appuyer sur <Enter> ou réécrire une passphrase   
Your identification has been saved in id\_rsa.   
Your public key has been saved in id\_rsa.pub.   
The key fingerprint is: 3d:15:66:f0:02:31:ca:0d:c5:d6:98:8b:64:07:ed:94 usr@machine

La clé publique et la clé privée sont maintenant créées.

2- Envoyer la clé publique sur le serveur distant  
En étant connecté sur la machine locale, il faut aller dans le répertoire où se trouvent les clés.

[usr@machine]$ cd ~/.ssh

Il faut par la suite copier la clé publique sur le serveur distant. Finalement, sur le serveur distant, il reste à mettre la clé publique comme étant autorisée.

[usr@machine]$ ssh usr2@serveur.com  
Fournir le mot de passe habituel  
[usr2@serveur]$ cd ~  
[usr2@serveur]$ mkdir .ssh (si le dossier n’existe pas)  
**[usr2@serveur]$ cat id\_rsa.pub >> .ssh/authorized\_keys  
[usr2@serveur]$ chmod 700 ~/.ssh  
[usr2@serveur]$ chmod 600 ~/.ssh/authorized\_keys**  
Et pour les machines qui ont SELinux (ex: CentOS, RHEL)  
[usr2@serveur]$ restorecon -Rv ~/.ssh

La clé est maintenant autorisée par le serveur.

3- Tester le tout  
Il ne reste qu'à essayer la connexion SSH de nouveau. Si tout fonctionne, le mot de passe ne sera plus demandé. Cependant, si une "passphrase" a été définie, celle-ci sera demandée.

[usr@machine ~]$ ssh usr2@serveur

## Autres informations

L'utilisation de passphrase  
Dans le cas où il n'y aurait pas de passphrase de définie, la connexion est instantanée. Cependant, si quelqu'un met la main sur la clé privée de la machine locale, il pourra s'en servir pour se connecter au serveur distant. Pour prévenir cela, c'est une bonne pratique de définir une passphrase. Ainsi, même si quelqu'un réussit à avoir la clé privée, il lui sera impossible de l'utiliser.  
  
Si vous avez spécifié une passephrase, et pour éviter que la passphrase soit demandée à chaque utilisation de la clé (connexion SSH), il faut utiliser l'utilitaire ssh-agent.

[usr@machine ~]$ eval `ssh-agent`  
Agent pid 8559  
[usr@machine ~]$ ssh-add  
Entrer la passphrase de la clé privée.

Empêcher les connexions SSH traditionnelles (n'accepter que les connexions par clés)  
Si vous voulez empêcher les connexions par nom d'usager/mot de passe, pour n'accepter que les connexions qui se font par clés SSH, il faut simplement modifier une ligne du fichier /etc/ssh/sshd\_config.

PasswordAuthentication no

Redémarrez ensuite le service SSH :

[usr2@serveur]$ systemctl restart ssh