## Les clés SSH

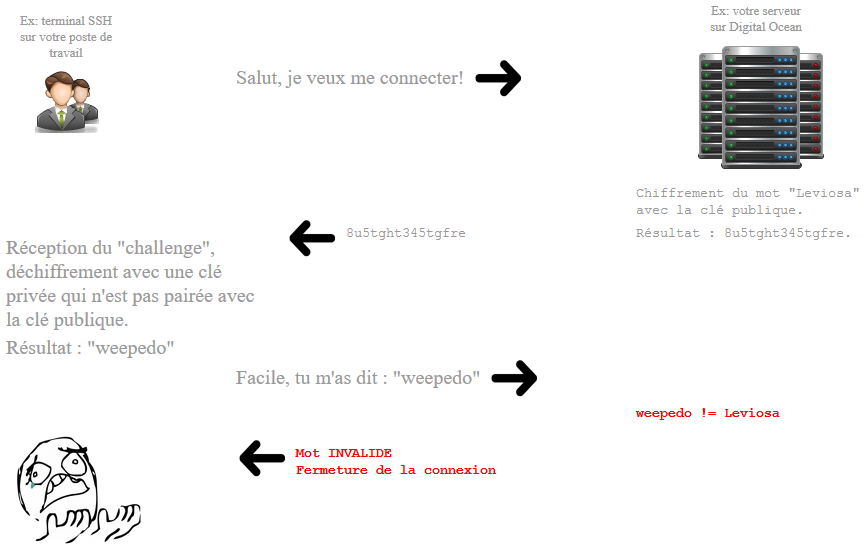
Permet la communication sécurisée d'une machine à une autre, basée sur une paire de clés publique/privée. L'utilisation de mot de passe n'est donc (habituellement) plus nécessaire pour établir la communication.

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:

Dans root

**[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.

La cle se trouve dans /root/.ssh/id\_rsa

On peut faire cat

b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn

NhAAAAAwEAAQAAAYEAtHhRxNJUkybm7JOaML+7gyYLFVFFwmHiHkNmoBO7FbsxQFEc98C3

2Pw4a5ZXtNXnlEkd/iJ4RPiAcQZtnlPOaRdb0lJoon7zko/ZlX5r0REAOwqfifC2dvNoxg

tQKC35HmoGkFNciMIqrM9NFY82a5XCLKYhNOMTMWE8wz5sPr2qKQvyy8paenA6x2a+na+Q

SxL06pfUXxaRpv0mkQJ/vEwghtluVjx29Arj7DGYJUofR8zCJa+9h+ow95mJnOT70mxyLS

BGTlbNu5GRexuRDBjRl/i0fMLiWzoUG1+ozhOeXM0QLwJullOh4UgLr+jOWSFCyUevOrY2

tS62lqPY0qI3ZesnDA/UAJfUPMgfJzICfoJAfv9Qj4GUZyoN480eHVoi2DfupDcumofak5

P0sSydY1FgBO6TPIHDt5ROKdZincOr5rK6F9CkPilYQi/MOqIdq6HUSDpMxkYdSUV/hw+V

e1pTOO1ooKCWc+k7q4ywciH+qlFlHSqtxuFyTz0xAAAFiH3OPkh9zj5IAAAAB3NzaC1yc2

EAAAGBALR4UcTSVJMm5uyTmjC/u4MmCxVRRcJh4h5DZqATuxW7MUBRHPfAt9j8OGuWV7TV

55RJHf4ieET4gHEGbZ5TzmkXW9JSaKJ+85KP2ZV+a9ERADsKn4nwtnbzaMYLUCgt+R5qBp

BTXIjCKqzPTRWPNmuVwiymITTjEzFhPMM+bD69qikL8svKWnpwOsdmvp2vkEsS9OqX1F8W

kab9JpECf7xMIIbZblY8dvQK4+wxmCVKH0fMwiWvvYfqMPeZiZzk+9Jsci0gRk5WzbuRkX

sbkQwY0Zf4tHzC4ls6FBtfqM4TnlzNEC8CbpZToeFIC6/ozlkhQslHrzq2NrUutpaj2NKi

N2XrJwwP1ACX1DzIHycyAn6CQH7/UI+BlGcqDePNHh1aItg37qQ3LpqH2pOT9LEsnWNRYA

TukzyBw7eUTinWYp3Dq+ayuhfQpD4pWEIvzDqiHauh1Eg6TMZGHUlFf4cPlXtaUzjtaKCg

lnPpO6uMsHIh/qpRZR0qrcbhck89MQAAAAMBAAEAAAGADWWVJFg8wQO1RsCaRJ4nDFBJqF

Z1KD3d7EMynhV44N8Zg93Bmb6rpybqBUqOzjTG0SIwYMosRWpxFnLUtlkgTHrYbR+C9vUX

nbyh+U/+LvHlI3zhZPsCgIgHMhRq6eCnypMyU4pspxZK1rM41K3h61CdrnTSqDtnFFZUjW

XX4TxxIkTpzCkEWGx2YJEG+2up+Xi5x6RntntVaqEGOOOHmfHIxFnlr+8jIXChhMoVskbz

VYAktWl7V1iSM/8o0QXKb97XDslRicwPB+JuWcPf+gUvwZu4PkF30o/VzK9j6/P4aT2DCj

IT8yKSkQcLqp7txpUcS937t1+IcQ9WXwsVAnLZCSiBgxwxVoGvyWC8z1gRg05REUYScfXj

X4XChcRs7OITJsJWE+uQ63CtPp6rF5X8sVF3l5NcmTIhJSUkbHiWLRkl8pRcG3arKpuwIm

/00273euGQ6Ylwdhd8BYPASp6YCj/jrGWfyJk8ux8tNTVZn4TZB0fhfgDlO8DhNXV1AAAA

wHXJQCN1a99SRov517Xg8IJt+d3E5Q6lgMrB4o7f+6isiZgT2drIfx76QnKiKTzHwI015K

5/ytNc+JengPn/DFhQXonY+ln/SzwMoqKFbkOVY/OUHUnQG84H18gp3TOBZlLB0oidzYfz

OAUt+AhFN5BGxRbQ3NWGSPDmx4sX+4IavmmULGCumKzdiIZrhgtyiwktoOg5b2vOOBccPI

HqGD5YcAwIwvAYmkvh5bZJCc2lfEAKcqaNxoE+WKw07J230AAAAMEA4Bb+AtjierqPcl5J

fxBE940r63brI0qs89Au6lui+qd/+0q3Bbpli172SYYu4W/yiFGpc/6Za8wm2y5eh9HNrp

i2mY2pipyAd6o4pn7zx70f6TmWXGUTYidwCl7Eyfs8JW6q+BFvCxFIqCMYV0EYuACmUyCa

QtcoZt1QoKIuOXXlQdnU+0+rltA4YseS8ulmBtLY/froryfNCVxHzbs1MXCn8yNWsTLhgk

3JhMg67IzrWMcwmdRIM0qPfIpyq1X7AAAAwQDOKzP8y3Ny6iGO2Iz1xThM5Mxp3TJURnpV

6LE6NuAGrj5qdXEDi82fbI5bUQHRBOoMip6YrY9lZYUcrvg3KCDYYD/PtFT0jDE98TkhVb

xkMuf7vtTHj61Pbs3wBR0HkCzv0PIR2fGdKnwS1EdZvkX9o+z/IgscLj7d6wx29qw/C14r

5/7SAe9qEh8RlsX8tU9YLna6iqyADxQnNc/1GypfyFu6azC3BN8WsTzJdRLTzrx545cRpr

H+iPdDpqsdDcMAAAARcm9vdEBiYjA0OTUyYmVjM2YBAg==

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.

Aller sur le ssh que nous avons créer

[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.

Dans benjamin

[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)

Le id est créer alors il fait le cat

touch .ssh/authorized\_keys

echo « La key » >> .ssh/authorized keys  
**[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

Avec git hub on prend la cle public settings, deploy key