

TP SSH

Configuration et gestion des clés OpenSSH

Version 0.4 - Michel SYSKA

À la fin du TP, déposez sur la classroom github votre rendu TP_SSH_Prenom_NOM.tar.gz contenant:

- ❑ le fichier réponse TP_SSH_Prenom_NOM.txt

Ce fichier doit contenir la liste des commandes que vous avez dû taper pour réaliser les exercices avec les commentaires utiles à leur compréhension.

- ❑ une copie des fichiers de configuration que vous avez modifiés, avec des commentaires #miage avant les lignes que vous avez modifiées

[1 Objectifs](#)

[2 Configuration et gestion des clés OpenSSH](#)

[2.1 Configuration du service sshd sur le guest](#)

[2.2 Configuration du client ssh](#)

[2.3 Cas de Fedora / RedHat / CentOS :](#)

[3 Configuration de PuTTY \(uniquement Windows\)](#)

[3.1 Génération des clés](#)

[3.2 Configuration de la session](#)

[4 Transfert de fichiers](#)

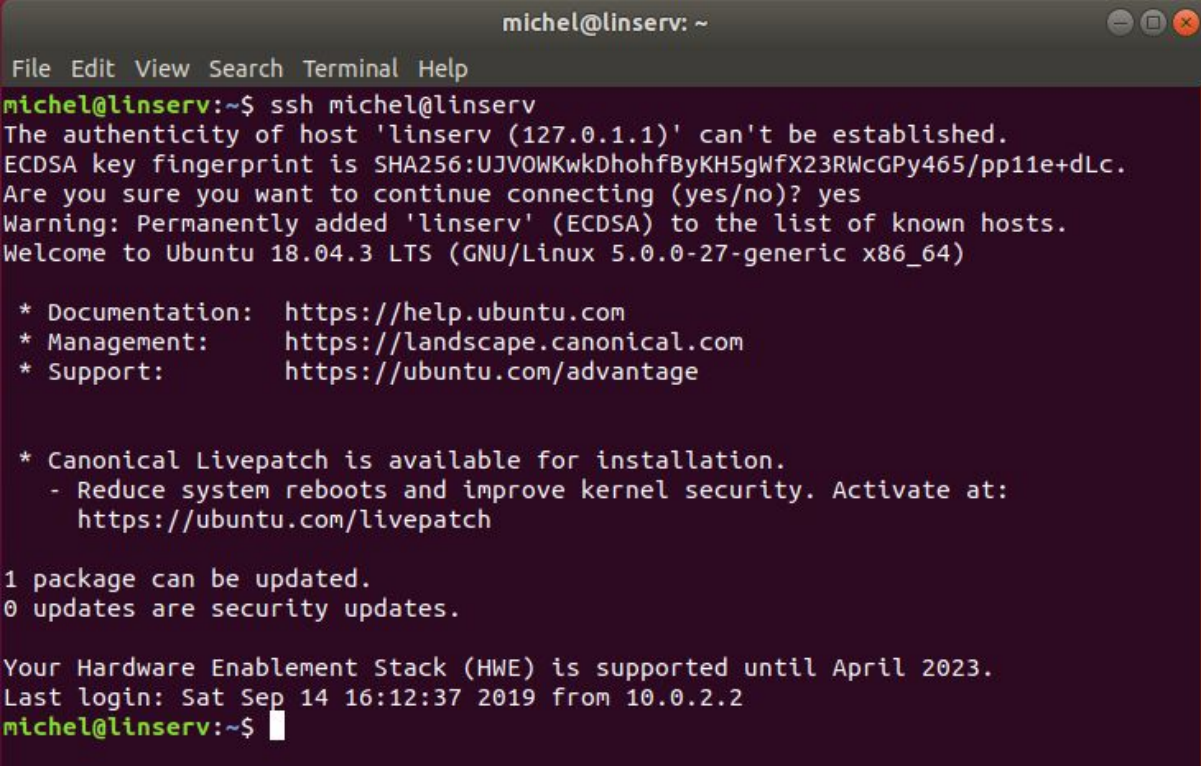
[4.1 scp](#)

[4.2 rsync](#)

1 Objectifs

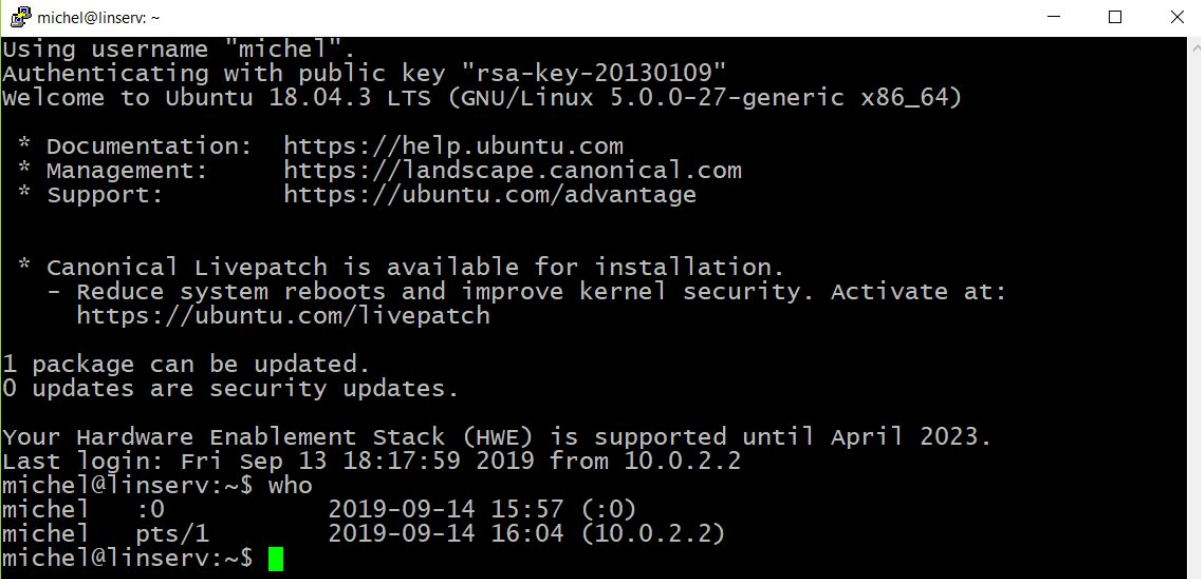
On veut pouvoir se connecter par ssh en tant que simple utilisateur sur la machine virtuelle Ubuntu que vous avez installée sur un hôte Windows, Linux ou Mac OS. Dans la suite les exemples sont fait à partir de Windows. On veut se connecter sans taper de mot de passe ni de passphrase. **Une fois que vous aurez tout configuré (section 2), voici ce que vous pourrez faire (juste pour avoir une idée de là où on va dans ce TP).**

On peut d'abord essayer de paramétrer cette procédure en local sur la machine guest Ubuntu installée dans VirtualBox en tapant simplement une commande ssh dans un terminal comme sur l'exemple suivant avec l'utilisateur michel sur le guest linserv. La première fois, le host auquel on se connecte n'est pas enregistré dans le fichier ~/.ssh/known_hosts, d'où la question à propos de l'empreinte (fingerprint).



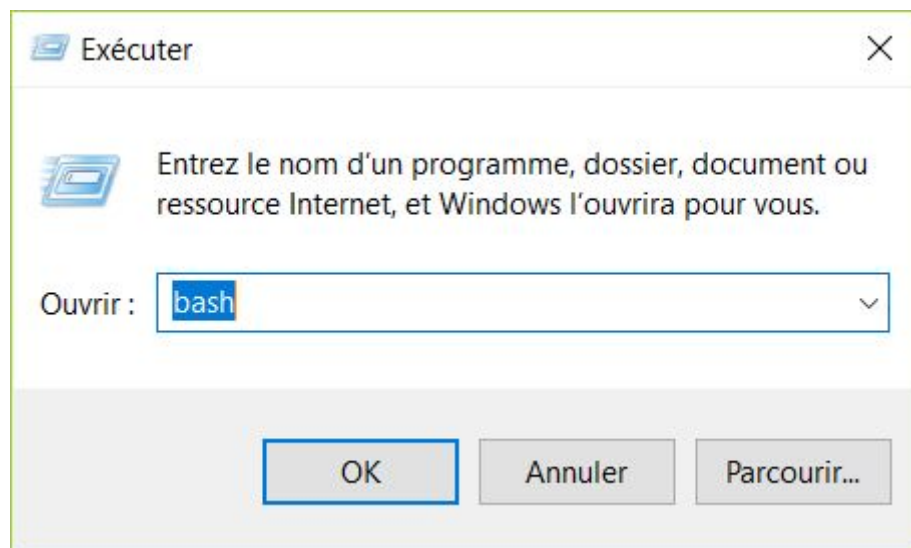
```
michel@linserv: ~  
File Edit View Search Terminal Help  
michel@linserv:~$ ssh michel@linserv  
The authenticity of host 'linserv (127.0.1.1)' can't be established.  
ECDSA key fingerprint is SHA256:UJVOWKwkDhohfByKH5gWfX23RWcGPY465/pp11e+dLc.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'linserv' (ECDSA) to the list of known hosts.  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-27-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
  https://ubuntu.com/livepatch  
  
1 package can be updated.  
0 updates are security updates.  
  
Your Hardware Enablement Stack (HWE) is supported until April 2023.  
Last login: Sat Sep 14 16:12:37 2019 from 10.0.2.2  
michel@linserv:~$
```

Si vous avez configuré convenablement PuTTY sur l'hôte Windows 10 (voir section 3) et la redirection de ports dans VirtualBox (voir la fiche VirtualBox.pdf), alors vous pouvez aussi vous connecter de la même façon avec un terminal PuTTY :



```
michel@linserv: ~  
Using username "michel".  
Authenticating with public key "rsa-key-20130109"  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-27-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
  https://ubuntu.com/livepatch  
  
1 package can be updated.  
0 updates are security updates.  
  
Your Hardware Enablement Stack (HWE) is supported until April 2023.  
Last login: Fri Sep 13 18:17:59 2019 from 10.0.2.2  
michel@linserv:~$ who  
michel    :0                2019-09-14 15:57 (:0)  
michel    pts/1            2019-09-14 16:04 (10.0.2.2)  
michel@linserv:~$
```

Si vos clés sont au bon endroit sous Windows, vous pouvez aussi utiliser le bash du sous-système Windows pour Linux :



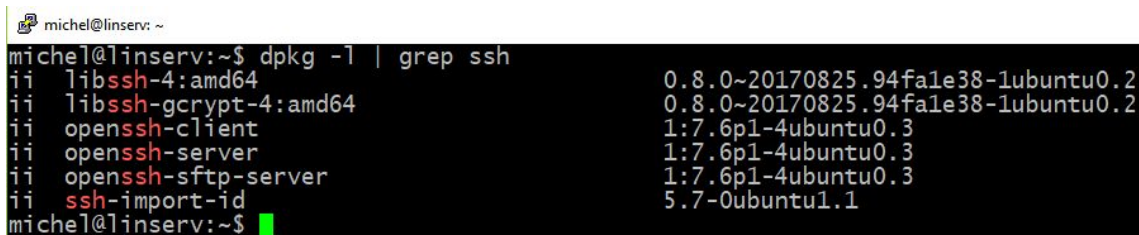
```
michel@linserv: ~  
msyska@syskoati:/mnt/c/Users/syska$ pwd  
/mnt/c/Users/syska  
msyska@syskoati:/mnt/c/Users/syska$ cd  
msyska@syskoati:~$ pwd  
/home/msyska  
msyska@syskoati:~$ ls .ssh/  
id_rsa id_rsa.pub known_hosts  
msyska@syskoati:~$ ssh michel@192.168.56.1 -p4096  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-27-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
  https://ubuntu.com/livepatch  
  
1 package can be updated.  
0 updates are security updates.  
  
Your Hardware Enablement Stack (HWE) is supported until April 2023.  
Last login: Sat Sep 14 16:04:51 2019 from 10.0.2.2  
michel@linserv:~$
```

2 Configuration et gestion des clés OpenSSH

2.1 Configuration du service sshd sur le guest

En premier lieu on va vérifier que le paquet openssh-server est bien installé sur la machine guest Ubuntu, par exemple avec la commande :

```
# dpkg -l | grep ssh
```



```
michel@linserv: ~  
michel@linserv:~$ dpkg -l | grep ssh  
ii  libssh-4:amd64                0.8.0~20170825.94fa1e38-1ubuntu0.2  
ii  libssh-gcrypt-4:amd64         0.8.0~20170825.94fa1e38-1ubuntu0.2  
ii  openssh-client                1:7.6p1-4ubuntu0.3  
ii  openssh-server                1:7.6p1-4ubuntu0.3  
ii  openssh-sftp-server           1:7.6p1-4ubuntu0.3  
ii  ssh-import-id                 5.7-0ubuntu1.1  
michel@linserv:~$
```

Si on ne trouve pas openssh-server dans la liste résultat, alors on installe le paquet ainsi :

```
# apt install openssh-server
```

Ensuite on change la configuration du serveur en éditant le fichier `/etc/ssh/sshd_config`

Si vous ne savez pas utiliser vi, vim ou emacs, nano est suffisamment simple pour être utilisé sans connaissance préalable (ou tout éditeur de texte).

Dans le fichier `sshd_config` on écrit sous la ligne

```
#PasswordAuthentication yes
```

```
PasswordAuthentication no
```

Désormais il faudra utiliser des clés pour se connecter avec un client ssh.

Après les modifications, pensez à relancer le service sshd :

```
# systemctl restart sshd.service
```

2.2 Configuration du client ssh

Ici, en tant que simple utilisateur, on va créer un répertoire `.ssh`, un couple de clés publique et privée, et copier la clé publique (`.pub`) dans le fichier `authorized_keys` de la machine sur laquelle on veut se connecter directement (ici sans passphrase). Pour commencer, on teste sur la machine locale.

```
$ cd
```

```
$ mkdir .ssh
```

```
$ chmod 700 .ssh
```

```
$ cd .ssh
$ ssh-keygen -t rsa
$ cat id_rsa      # juste pour voir
$ cat id_rsa.pub  # juste pour voir
$ cat id_rsa.pub >> authorized_keys
```

Ensuite on peut tester en local (l'adresse peut être différente sur votre VM) :

```
$ ssh michel@10.0.2.15
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-27-generic x86_64)
...
```

Sur une machine distante, avant de désactiver la connexion par mot de passe on peut utiliser la commande :

```
ssh-copy-id
```

Pour observer le journal des connexions :

```
# journalctl -f $(which sshd)
```

2.3 Cas de Fedora / RedHat / CentOS :

Les permissions sur ces fichiers et répertoires doivent être restreintes au propriétaire. En cas de problème avec selinux:

```
restorecon -Rv ~/.ssh
```

ou bien

Désactiver selinux:

dans /etc/selinux/config

remplacer la ligne

```
SELINUX=enforcing
```

par

```
SELINUX=disabled
```

puis rebooter le système

ou

```
# /usr/sbin/setenforce 0
```

Aide en ligne:

[Support > Product Documentation > Red Hat Enterprise Linux > 7 > System Administrator's Guide - Chapter 9. OpenSSH](#) - Red Hat, Inc.

<http://wiki.centos.org/HowTos/Network/SecuringSSH>

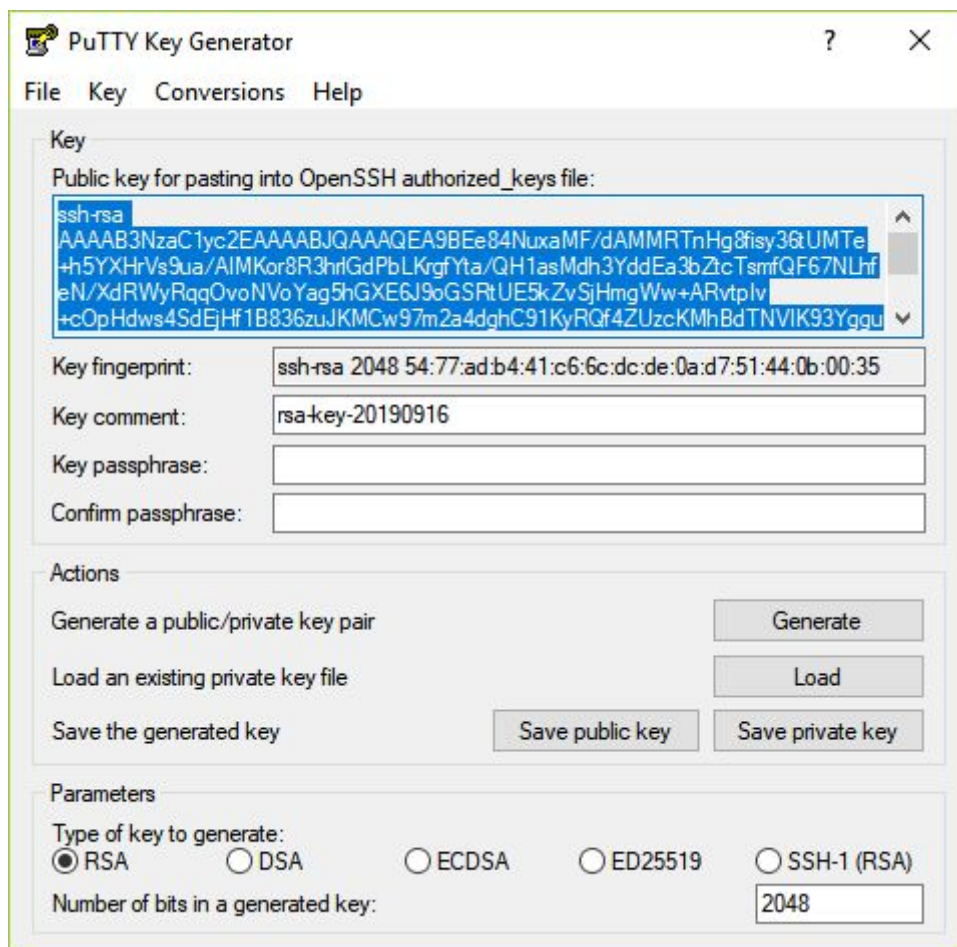
3 Configuration de PuTTY (uniquement Windows)

Si vous ne disposez pas des outils ssh PuTTY pour Windows : <https://www.putty.org/>

Si vous êtes sur un hôte Linux ou MacOS, vous pouvez faire la même procédure que 2.2 pour la génération des clés sur le client et adapter la procédure pour le serveur.

3.1 Génération des clés

Pour générer la clé publique et la clé privée on utilise PuTTYgen :



Dans l'ordre :

1. cliquer sur **Generate**
2. bouger la souris dans la fenêtre pour rendre la génération aléatoire
3. cliquer sur **Save public key** puis **Save private key** et enregistrer les deux clés
4. copier coller la clé au format OpenSSH (ici en bleu) pour l'ajouter au fichier `~/.ssh/authorized_keys` de votre session Linux.

Vous pouvez aussi copier le fichier contenant la clé publique (de nom rsa.pub par exemple) sous Linux dans le répertoire .ssh (à travers le partage VirtualBox) puis appliquer les commandes :

```
$ ssh-keygen -i -f rsa.pub > rsa-os.pub  
$ cat rsa-os.pub >> authorized_keys
```

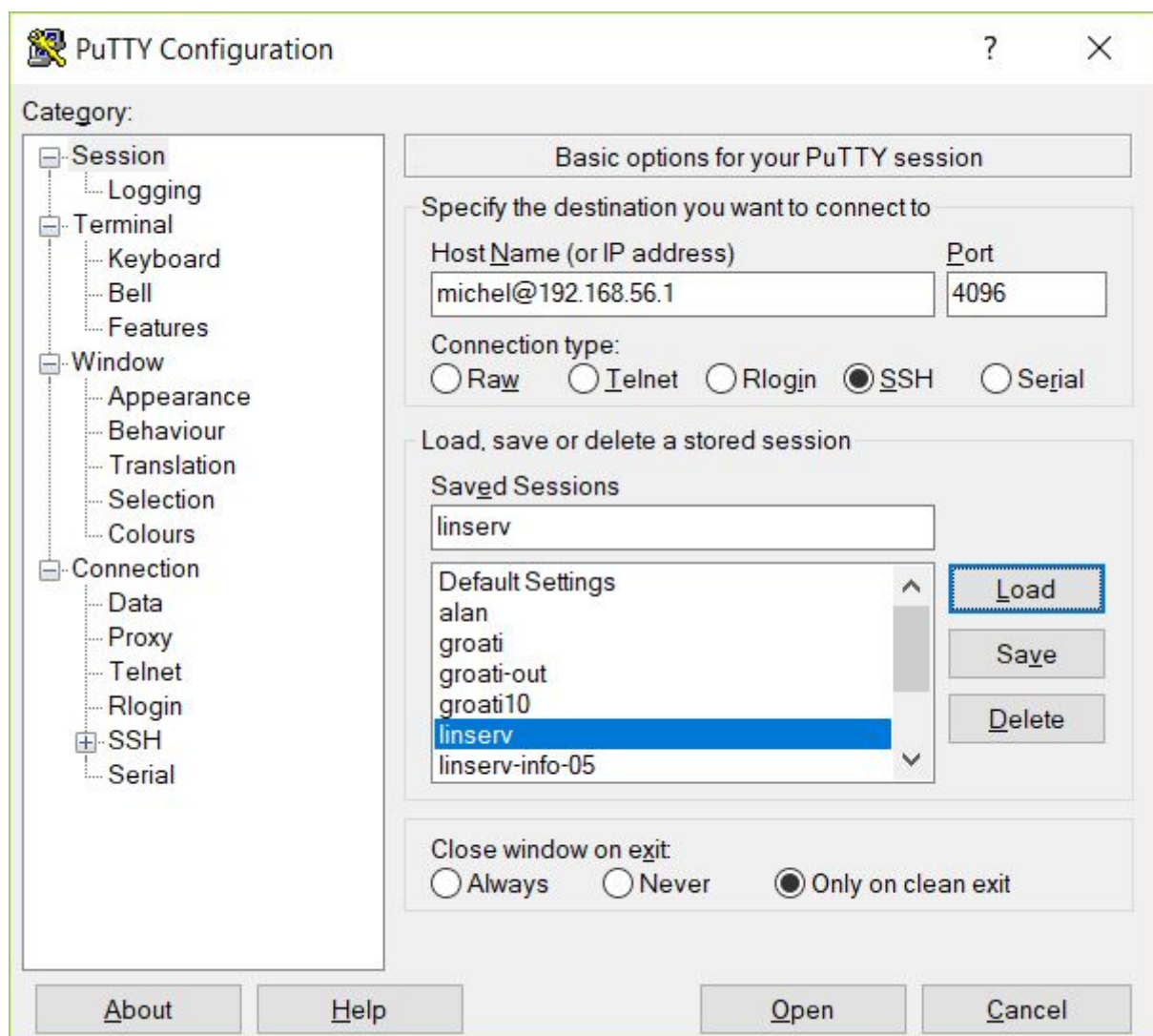
pour convertir une clé PuTTY en clé OpenSSH et l'ajouter au fichier authorized_keys

En une seule ligne :

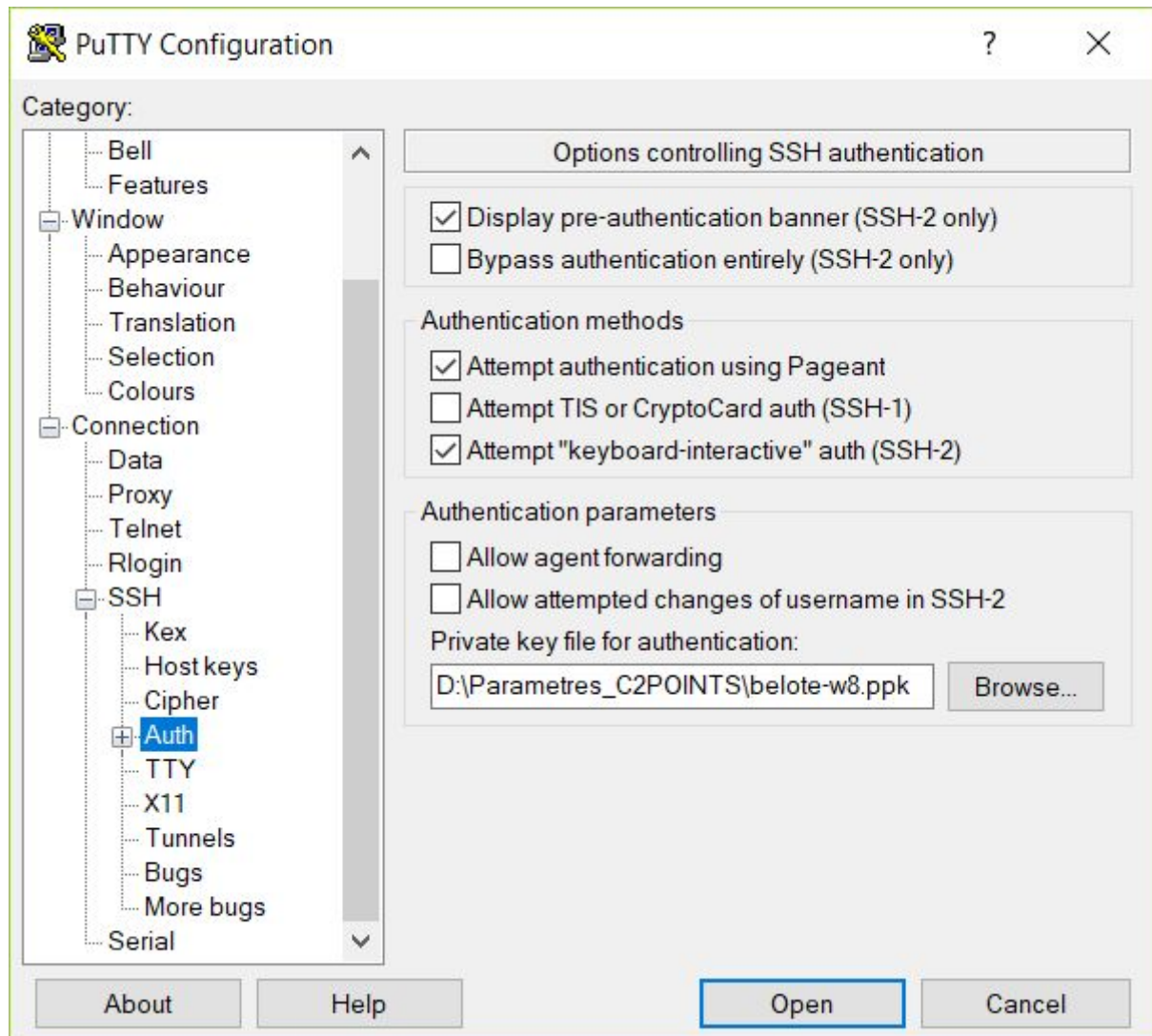
```
$ ssh-keygen -i -f rsa.pub >> authorized_keys
```

3.2 Configuration de la session

Dans les copies d'écran suivantes on suppose que VirtualBox est configuré comme dans la fiche du même nom (ici redirection ssh sur le port 4096).



Ici la clé privée correspond au fichier d'extension ppk



Pensez à enregistrer la session avant de la tester (Open ou double clic).

4 Transfert de fichiers

4.1 scp

4.2 rsync