Les bases du système Linux pour le calcul scientifique

https://pole-calcul-formation.gricad-pages.univ-grenoble-alpes.fr/ced/

Frédéric Audra, Glenn Cougoulat

Octobre 2023

Collège des Écoles Doctorales

Plan

- Protocoles de contrôle à distance et authentification.
- Première connexion SSH
- Transfert de fichier
- Utilisation avancées : les clés SSH
- Utilisation avancées : ProxyCommand
- Les clés SSH et GitLab

Protocoles de contrôle à distance et authentification.

- Il existe une multitude de protocoles de prise de commande à distance. Les plus connus sont : **Rlogin**, **RDP**, **SSH**, **VNC**, etc...
- Authentification via des identifiants (login / mot de passe)
- Le mot de passe est la protection la plus faible (ne jamais donner son mot de passe à un tiers), on lui préfère en général une clé ou un certificat électronique.
- En pratique : il est très improbable qu'un utilisateur puisse accéder physiquement à une machine de calcul. Il faut donc se connecter à distance.
- Le protocole **SSH** est très répandu dans notre communauté car il offre un excellent compromis entre facilité de mise en oeuvre et niveau de sécurité.

Première connexion SSH

Initialisation d'une connexion SSH :

```
jdoe@machine1:~$ ssh jdoe@machine2.imag.fr
The authenticity of host 'machine2.imag.fr (129.88.33.50)' can't be established.
ECDSA key fingerprint is 4f:2f:be:3b:63:5a:bb:09:53:c2:fe:3d:69:99:f3:5f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'machine2.imag.fr,129.88.33.50' (ECDSA) to the list of known hosts.
Password:
Last login: Mon Nov 23 16:24:04 2015
jdoe@machine2:~$
```

Transfert de fichier

- Transférer des fichiers sur une machine distante. Il existe plusieurs façons de transférer un/ des fichier(s) sur une machine distante.
- `scp` : équivalent d'un «cp» via «ssh». Très courant car on a généralement accès à ssh.

```
jdoe@machine1:~$ scp mon_fichier.txt jdoe@machine2:/chemin/
```

- `ftp` : plus performant (pas de cryptage), mais moins répandu pour des raisons de sécurité.
- `rsync` : algorithme performant de synchronisation de données (*man rsync*)

Utilisation avancées : les clés SSH

Il est généralement recommandé d'utiliser une **paire de clés SSH** afin de **sécuriser** et de **simplifier l'authentification**. La commande *ssh-keygen* permet cela :

```
jdoe@machine1:~/ $ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jdoe/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jdoe/.ssh/id_rsa.
Your public key has been saved in /home/jdoe/.ssh/id_rsa.pub.
The key fingerprint is:
fb:9d:8b:d5:74:25:5e:24:59:0c:63:74:93:2d:f7:13 jdoe@machine1
The key's randomart image is:
+---[RSA 2048]----+
| .***|
| .oE=|
| .o=|
| . +o|
| S o o|
| . o . |
| . . . |
| . + . |
| o +. |
+-----------------+
jdoe@machine1:~/ $
```

La commande *ssh-copy-id* permet ensuite de propager sa clé publique sur les machines où l'on souhaite se connecter.

```
toto@machine1:~$ ssh-copy-id toto@machine2.imag.fr
toto@machine2.imag.fr's password:
```

- Lors de l'utilisation des clés SSH pour les connexions vers des machines distantes, il est recommandé d'utiliser un agent SSH (ssh-agent) qui vous permettra de stocker votre passphrase pour ne pas avoir à la retaper à chaque connexion.

- Sur la majorité des systèmes Linux actuels un agent SSH tourne par défaut dans votre session.

Si tel n'est pas le cas vous pouvez lancer un agent comme ceci :

```
audraf@pomponette:~/ $> eval `ssh-agent`
Agent pid 10889
audraf@pomponette:~/ $>
```

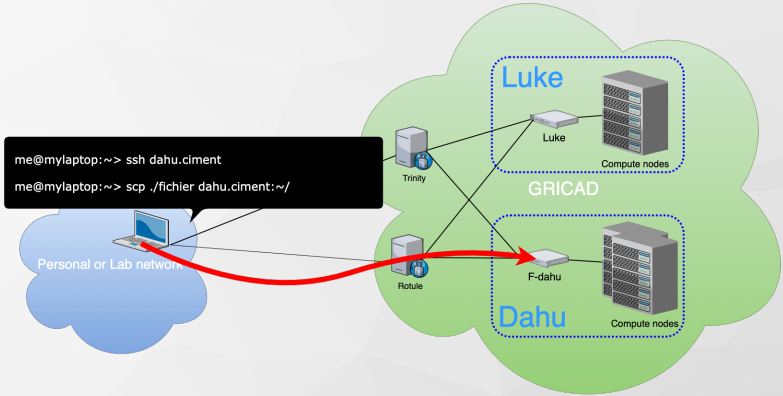
- Pour ajouter une passphrase à votre agent :

```
audraf@pomponette:~/ $> ssh-add
Enter passphrase for /home/audraf/.ssh/id_rsa:
```

Utilisation avancées : ProxyCommand

- L'accès à un cluster de calcul n'est jamais direct. Il est très souvent nécessaire de rebondir via une passerelle SSH.
- La directive ProxyCommand permet de rendre ce « rebond » transparent. Cela facilite, entre autre, les transferts de fichiers via scp, etc...

Utilisation avancées : ProxyCommand



Utilisation avancées : ProxyCommand

Mise en place :

Il faut créer un fichier de configuration : *~/.ssh/config*

Cette configuration propre à votre utilisateur sera lue à chaque connexion SSH.

```
# Fichier ~/.ssh/config
Host *.gricad
 User MONLOGIN
 ProxyCommand ssh -q MONLOGIN@access-gricad.u-ga.fr "nc -w 60 `basename %h .gricad` %p"
 ForwardAgent yes
```

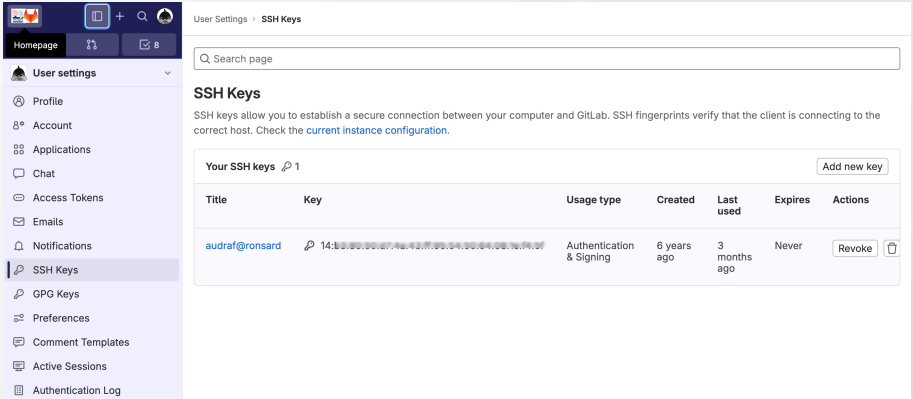
Utilisation :

```
# Connexion ssh avec rebond :
machine1:~$ ssh audraf@machine2.gricad

# Copie de fichier avec rebond :
machine1:~$ scp monfichier audraf@machine2.gricad:/tmp/
```

Les clés SSH et GitLab

- Les clés SSH sont aussi largement utilisées sur les plateformes de travail collaboratif de type GitLab.
- La clé publique doit alors être importer via l'interface web dans le menu adéquate. (Pour la plateforme GitLab de Gricad : Preferences / SSH Keys ou en cliquant ici)



Zoom : pré-requis

- Pour accéder aux ressources du méso-centre, il est nécessaire d'avoir un compte sur la plateforme perseus.
- La procédure détaillée de création de compte est décrite dans la documentation GRICAD.

Plateforme Perseus :

https://perseus.univ-grenoble-alpes.fr

Accès au TP

- Pour commencer le TP de cette session, veuillez suivre les indications depuis l'URL :

https://pole-calcul-formation.gricad-pages.univ-grenoble-alpes.fr/ced/plans_modules/#plan_unix

Merci !