# Travaux pratiques préparatoires
# SSH

Jonathan Lejeune

## Objectif

Ce sujet de travaux pratiques préparatoire est à faire au préalable de toute séance de TP. Il vous guidera dans la configuration du client ssh dans un environnement UNIX. Le service ssh est indispensable au déploiement de processus dans une infrastructure distribuée.

## Introduction

Le **S**ecure **SH**ell est un protocole de connexion et de communication sécurisé principalement utilisé dans les environnements UNIX. Il permet à un utilisateur de se connecter à distance sur une machine pour :

- soit y ouvrir un shell interactif
- soit y exécuter une commande particulière pour par exemple y déployer un démon. On parle alors de connexion non interactive.
- soit transférer des fichiers avec la commande `scp`.

L'avantage de ssh est qu'il assure la confidentialité des communications car les messages transitant sur le réseau sont cryptés. Il existe plusieurs implantations du protocole ssh, la version la plus utilisée étant openssh. Pour pouvoir se connecter depuis n'importe quelle machine vers n'importe quelle autre machine de votre infrastructure, il faut que chaque machine propose un service de connexion ssh. Pour que ce service soit accessible il faut que le démon `sshd` soit démarré pour pouvoir traiter les requêtes des clients ssh.

On notera qu'il est également possible et très courant dans la cadre de ce module de se connecter à distance sur la machine locale en passant par le réseau loopback (localhost) : le client ssh et son serveur associé sont par conséquent hébergés sur la même machine physique.

## Exercice 1 – Installation et connexion

**Question 1**

Pour savoir si un serveur ssh est en cours d'exécution sur votre machine, il suffit de savoir si le port 22 (port par défaut du serveur ssh) est occupé. Pour cela tapez netstat −na | grep LISTEN | grep tcp et vérifiez qu'il existe bien une ligne qui contienne :22. Si tel est le cas passez à la question suivante. Sinon, installez le package `openssh-server`.

**Question 2**

**IMPORTANT** : Sur certaines distributions, il se peut que la connexion via ssh ne prenne pas en compte les variables d'environnement définis dans le `.bashrc` si la connexion se fait de manière non interactive (ce qui sera le cas pour nous par la suite). Pour cela, assurez vous dans votre fichier `.bashrc` que les lignes suivantes soient commentées ou absentes (dans l'hypothèse où elles seraient présentes, elles doivent se trouver au début du fichier) :

```
case $- in
  *i*);;
  *) return;;
esac
```

**Question 3**

Pour se connecter à une machine et ouvrir un shell distant, le schéma de la commande à taper sur la machine cliente est :

```
ssh <votre_login_sur_la_machine_distante>@<adresseIP_ou_nomDNS_machine_distante>
```

Si vous n'avez pas configuré ssh pour qu'il s'authentifie par clé (cf. plus loin), il vous sera demandé de taper votre mot de passe de votre compte sur la machine distante. On notera que si le login est le même sur les deux machines il est alors possible de simplifier la commande ainsi :

```
ssh <adresseIP_ou_nomDNS_machine_distante>
```

Pour exécuter une commande quelconque en mode non interactif il suffit de l'ajouter en argument de la commande ssh.

```
ssh <adresseIP_ou_nomDNS_machine_distante> '<commande_a_executer>'
```

Si vous vous connectez pour la première fois à un hôte distant, il est possible que vous ayez à confirmer votre connexion en tapant `y`.

**Question 4**

Pour vous déconnecter de votre shell tapez la commande `exit`. On notera qu'il est possible de se connecter sur une machine distante à partir d'un shell issu d'une connexion distante, i.e. relancer la commande `ssh` dans un shell distant. Il est donc nécessaire de faire autant de `exit` que de `ssh` pour pouvoir revenir au shell local initial.

# Exercice 2 – Authentification par clé

SSH vous offre la possibilité de vous authentifier soit par mot de passe soit par clé. L'authentification par mot de passe peut être fastidieuse si l'on souhaite déployer automatiquement des démons sur l'infrastructure (ce qui sera le cas pour nous par la suite). En effet il est nécessaire de se connecter via ssh pour chaque démon (100 démons = 100 connexions = 100 mots de passe à taper).

**Question 1**

Pour se connecter par clé ssh, il est nécessaire de créer **si ce n'est pas déjà fait** un couple de clés ssh grâce à la commande `ssh-keygen`. Pour ceci tapez :

```
ssh-keygen -q -N '' -f ${HOME}/.ssh/id_rsa
```

Ceci a pour conséquence de vous créer les fichiers `id_rsa` et `id_rsa.pub` dans le répertoire `$HOME/.ssh` qui correspond respectivement à la clé privée et à la clé publique.

**Question 2**

Pour vous connecter par clé, il faut :

- que la machine source possède dans votre répertoire `.ssh` le fichier de la clé privée (id_rsa)
- déclarer à la machine destination la clé publique associée (ici id_rsa.pub). Pour ce faire, **sur votre compte de la machine destination**, il faut ajouter dans le fichier `$HOME/.ssh/authorized_keys` le contenu de la clé publique.

Ici nous allons considérer que la machine source et la machine destination sont les mêmes pour pour pouvoir s'authentifier par clé en local. Tapez :

```
cat ${HOME}/.ssh/id_rsa.pub >> ${HOME}/.ssh/authorized_keys
```

**Question 3**

Tapez la commande :

```
ssh localhost
```

- Si on vous demande de taper votre mot de passe recommencez depuis le début de cet exercice
- Si le message suivant s'affiche :
  ```
  WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!
  ```
  effacez le fichier
  `${HOME}/.ssh/known_hosts`
  et retapez la commande.