# DEMO 8 : Connexion des machines via SSH

## Authentification SSH

*OpenSSH* pour la communication à distance. Ça permet l'authentification sans **mot de passe** et par **mot de passe** pour exécuter des commandes sur les nœuds gérés.

### Authentification par mot de passe

L'authentification par mot de passe peut être utilisée si nécessaire en fournissant l'option **--ask-pass**. Cette option nécessite **sshpass** sur la machine de contrôle.

```
### RHEL /CentOS  / Fedora ###
```

```
yum install -y sshpass
```

```
### Ubuntu / Debian  ###
```

```
sudo apt-get update
sudo apt-get install -y sshpass
```

Testez l'accès aux noeuds gérés à partir du noeud *rocky*:

```
ssh root@192.168.60.4
root@192.168.60.4's password:
```

Tapez le mot de passe de l'utilisateur *root :rocky*

Vous serez directement transféré sur la machine clone rocky, en remarquant votre nouveau prompt:

```
Last login: Sat Nov 13 09:15:15 2021 from 10.0.2.2
[root@clone rocky ~]$
```

Pour quitter la machine *app1* et revenir sur votre machine *master*, utilisez la commande `exit`, ou la combinaison des touches **Ctrl-D**.

### Remarque :

Selon la configuration par défaut de votre système, parfois l'accès en SSH clair est désactivé, vous aurez le message d'erreur suivant :

```
...

Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

Pour autoriser l'accès par le pair login/mot de passe, éditez le fichier de configuration "**/etc/ssh/sshd_config**"

```
[root@clone rocky ~]$ vi /etc/ssh/sshd_config
```

Puis change la directive **PasswordAuthentication** en **"Yes"**

```
...
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes
```

## Authentification par clé SSH (*authentification sans mot de passe*)

Lorsqu'il s'agit d'authentification **ssh**, par défaut, il utilise des clés ssh (authentification sans mot de passe) pour s'authentifier auprès de la machine distante.

### SSH Configurer l'accès par paire de clès RSA

L'authentification par clé est le plus sécurisé de plusieurs modes d'authentification utilisables avec OpenSSH, tels que le mot de passe simple et les tickets Kerberos. L'authentification par clé présente plusieurs avantages par rapport à l'authentification par mot de passe, par exemple, les valeurs de clé sont nettement plus difficiles à forcer ou à deviner que les mots de passe simples, à condition que la longueur de la clé soit suffisante. Les autres méthodes d'authentification ne sont utilisées que dans des situations très spécifiques.

SSH peut utiliser des clés "RSA" (Rivest-Shamir-Adleman) ou "DSA" ("Digital Signature Algorithm").

L'authentification par clé utilise deux clés, une clé "publique" que tout le monde est autorisé à voir et une autre clé "privée" que seul le propriétaire est autorisé à voir. Pour communiquer en toute sécurité à l'aide de l'authentification par clé, il faut créer une paire de clés, stocker en toute sécurité la clé privée sur l'ordinateur à partir duquel on veut se connecter et stocker la clé publique sur l'ordinateur auquel on veut se connecter.

L'utilisation de connexions basées sur des clés avec ssh est généralement considérée comme plus sécurisée que l'utilisation de connexions par mot de passe simples.

***Génération de clés RSA***

La première étape consiste à créer un ensemble de clés RSA à utiliser dans l'authentification.

Cela devrait être fait sur le *master* à partir duquel vous allez gérer les autres noeuds.

Pour créer vos clés SSH publiques et privées sur la ligne de commande :

```
ssh-keygen -t rsa
```

Vous serez invité à indiquer un emplacement pour enregistrer les clés et une phrase secrète pour les clés. Cette phrase secrète protégera votre clé privée pendant qu'elle est stockée sur le disque dur :

```
[root@rocky ~]$ ssh-keygen -t rsa

Generating public/private rsa key pair.
Enter file in which to save the key (/home/vagrant/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vagrant/.ssh/id_rsa.
Your public key has been saved in /home/vagrant/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tcrUXS2cOMwGMZWQ0xyUE4hAZAo56bBQKGFxci+Uq80 vagrant@master.dev
The key's randomart image is:
+---[RSA 2048]----+
|.B+*.o=. .=X==   |
|* Oo.o  . +=*o o |
|o+ oo.    ..*.= .|
|. ...    o + o . |
|        o        |
+----[SHA256]-----+
```

Toutes nos félicitations! Vous avez maintenant un jeu de clés. Il est maintenant temps de faire en sorte que vos systèmes vous permettent de vous connecter avec eux.

Votre phrase secrète (passphrase) de clé SSH est uniquement utilisée pour protéger votre clé privée contre les voleurs. Elle n'est jamais transmise sur Internet et la force de votre clé n'a rien à voir avec la force de votre phrase secrète.

*Niveau de cryptage de clé*

Remarque : la valeur par défaut est une clé de 2048 bits. Vous pouvez augmenter cela à 4096 bits avec l'indicateur -b (Augmenter les bits rend plus difficile le craquage de la clé par des méthodes de force brute).

```
ssh-keygen -t rsa -b 4096
```

***Transférer la clé publique vers l'hôte géré***

La clé que vous devez transférer à l'hôte est la clé publique. Si vous pouvez vous connecter à un ordinateur via SSH à l'aide d'un mot de passe, vous pouvez transférer votre clé RSA en procédant comme suit depuis votre propre ordinateur :

```
ssh-copy-id <nom d'utilisateur>@<hôte>
```

Où <username> et <host> doivent être remplacés par votre nom d'utilisateur et le nom de l'ordinateur vers lequel vous transférez votre clé.

```
[root@rocky ~]$ ssh-copy-id root@192.168.60.4

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/vagra
nt/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to f
```

```
ilter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are
prompted now it is to install the new keys
vagrant@192.168.60.4's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'vagrant@192.168.60.4'"
and check to make sure that only the key(s) you wanted were added.
```

Une autre alternative consiste à copier le fichier de clé publique sur le serveur et à le concaténer manuellement dans le fichier ~/.ssh/authorized_keys.

***Tester l'accès SSH sans mot de passe***

Vous pouvez vous assurer que cela a fonctionné en faisant :

```
ssh <nom d'utilisateur>@<hôte>
```

Une fois que vous avez configuré la communication sans mot de passe, vérifiez-la.

```
[root@rocky ~]$ ssh 192.168.60.4
Last login: Sat Nov 13 09:35:39 2021 from 192.168.60.1
[root@clone rocky ~]$
```