SSH et CVS

Thomas Petazzoni

Atelier du Club LinUT du 13 décembre 2001

1 SSH

SSH (*Secure Shell Server*) permet plusieurs choses :

- de se connecter sur une machine distante pour y éxécuter des commandes
- de réaliser des connections X11
- de forwarder des ports

Le principal avantage de ssh est que ceci se fait de manière **sécurisée** (les données sont cryptées).

Pour se connecter sur une machine distante :

```
ssh login@nom_d_hote
```

Deux moyens pour se connecter :

- directement avec le mot de passe
- avec une pass-phrase

La connexion avec un mot de passe est simple, on tape la commande, on donne le mot de passe et on est connecté. La connexion via une pass-phrase met en jeu une paire de clé (clé publique/clé privée) générée par la commande `ssh-keygen`. Ensuite si la clé publique est dans `.ssh/authorized_keys`, alors vous pouvez vous connecter à l'hôte avec la pass-phrase, et non avec le mot de passe. Si votre clé n'est pas dans ce fichier alors il demandera le mot de passe. Pour copier votre clé dans le `.ssh/authorized_keys`, il faut utiliser la commande `ssh-copy-id`.

L'utilisation de `ssh-agent` (à lancer via `eval ‘ssh-agent‘`) puis de `ssh-add` permet tout au long d'une session de se dispenser de taper la pass-phrase.

D'autre part, ssh maintient une liste des hôtes auxquels vous vous connectez `.ssh/known_hosts`, avec une clé RSA. Si l'hôte change, la clé RSA change et donc ssh pourra le savoir.

ssh permet aussi de rediriger de réaliser des connections X11 dans un tunnel SSH (sécurisé) ou bien de sécuriser des connections (Pop par exemple), en les faisant passer dans un tunnel SSH.

Pour plus d'informations, consultez le `man` sur `ssh, ssh-add, ssh-agent, ssh-keygen, ssh-copy-id`.

2 CVS

2.1 Qu'est-ce que CVS ?

CVS *(Concurrent Versions System)* est un outil de gestion de versions. Il permet de tenir à jour l'historique de chacun des fichiers source d'un projet. Grâce à ce mécanisme, il est possible de récupérer n'importe quelle version d'un fichier source, de la modifier, et de reporter ces modifications sur les versions postérieures du projet. CVS s'occupe alors de gérer les éventuels conflits entre ces modifications et les évolutions qu'il aurait pu y avoir entre temps. CVS gère le développement simultané du même projet par plusieurs utilisateurs éventuellement distants (CVS s'articule alors autour d'un serveur). Dans les cas extrêmes où les développements se recoupent (travail sur le même fichier source par exemple), CVS est capable de gérer les conflits.

2.2 Principes

CVS s'articule autour d'un dépôt unique centralisé (le *repository*) des sources du projet. Ce dépôt renferme, pour chaque fichier source, tout l'historique de l'évolution de ces sources. CVS vous permet d'y accéder à distance. Notons que ce *repository* peut renfermer les sources de plusieurs projets : chaque projet est un module, selon la terminologie CVS.

2.3 Accéder à un *repository*

Trois méthodes existent pour accéder à un *repository* :

- La méthode locale : */var/cvs/repository*, par exemple
- La méthode distante : *:pserver :login@host :/var/cvs/repository*, par exemple
- Une autre méthode distante : *:ext :login@host :/var/cvs/repository*, par exemple.

La méthode locale est extrêmement simple, il suffit d'avoir un accès local sur la machine, et d'avoir les droits sur le *repository*.

La première méthode distante permet d'accéder depuis le réseau à un *repository*, mais les données

circulent de manière non sécurisée. La gestion des utilisateurs et des mots de passe se fait directement dans le *repository*.

La deuxième méthode distante permet d'accéder au *repository* en utilisant en shell externe, défini par la variable `CVS_RSH`. En utilisant `ssh`, ce système permet une utilisation sécurisée de CVS. Il faut alors avoir un compte sur la machine.

2.4 Authentification

Dans le premier et le troisième cas, il faut pour s'authentifier disposer d'un compte Unix sur la machine ou se trouve le *repository* et que ce compte ai les droits suffisants sur ce *repository*.

Dans le second cas, il faut créer trois fichiers dans le répertoire *CVSROOT* du *repository* :

- `passwd` qui contient des lignes de la forme user_cvs :password :user_unix. En effet, dans ce cas il existe des utilisateurs propres à CVS, mais il faut qu'ils soient "mappés" sur un utilisateur Unix.
- `readers` donnant la liste des utilisateurs ayant le droit de lecture.
- `writers` donnant la liste des utilisateurs ayant le droit d'écriture.

2.5 Créer un *repository*

La création d'un *repository* n'intervient qu'à la création du serveur, la toute première fois.

On utilise la commande

```
cvs -d chemin_d_acces init
```

Où `chemin_d_acces` spécifie l'emplacement du *repository* tel que nous l'avons vu précédemment.

2.6 Créer un module

On se place dans le répertoire des sources du module à importer, puis on utilise la commande :

```
cvs -d chemin_d_acces import nom_module
branche release
```

2.7 Télécharger un module

Télécharger un module n'a lieu qu'une seule fois, la toute première fois que vous téléchargez ce module. Par la suite, seules des mises à jour seront nécessaires.

```
cvs -d chemin_d_acces checkout
nom_module
```

2.8 Mettre à jour un module

Placez-vous dans le répertoire du module et tapez :

```
cvs update -dP
```

2.9 Soumettre les changements au *repository*

Il suffit de se placer dans le répertoire du module et de taper :

```
cvs commit
```

2.10 Ajouter ou supprimer un fichier

Pour ajouter un fichier non présent lors de l'import du module au *repository*, il faut utiliser la commande

```
cvs add nom_fichier
```

De même pour supprimer un fichier, on utilisera la commande

```
cvs remove nom_fichier
```

2.11 Variables d'environnements

La variable d'environnement *CVSROOT* désigne le *repository* à utiliser, ce qui évite de le donner à chaque fois.

La variable d'environnement *CVS_RSH* désigne le shell à utiliser dans le cas de l'utilisation de la méthode *ext*.

La variable d'environnement *EDITOR* désigne l'éditeur à utiliser pour rédiger les logs de commit.

Pour plus d'informations, vous pouvez consulter `man cvs`.

3 Liens

- OpenSSH : http ://www.openssh.org
- CVS : http ://www.cvshome.org
- Une doc simple, mais plus complète sur CVS : http ://kos.enix.org/cvs.php#doc