Utiliser SSH pour se connecter/travailler au CREMI

The UF Info Team

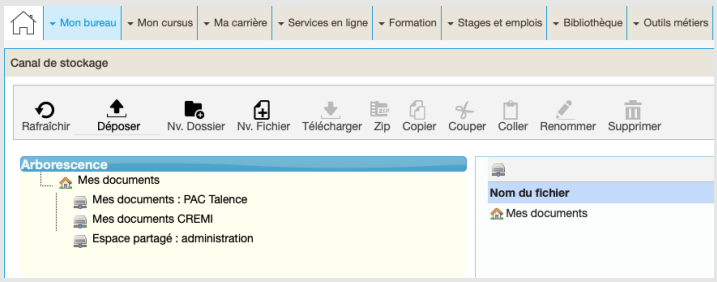
18 novembre 2020

Ce petit guide a pour objectif de vous simplifier les sessions de travail à distance au CREMI, ou simplement le rapatriement de fichiers de travail. Ce n'est aucunement un cours sur le chiffrement (a)symétrique d'échanges de données, mais plutôt un guide très pragmatique de commandes utiles. Dans l'intranet du CREMI, vous pourrez trouver de précieuses informations complémentaires à cette petite note :

https://services.emi.u-bordeaux.fr/intranet/spip.php?article21

Note

Notez que l'interface web de votre ENT vous permet de faire des transferts de fichiers vers/depuis le CREMI. Cliquez sur l'onglet « Mon bureau », puis choisissez « Canal de stockage » :



1 Cas particulier Windows

Si vous n'avez pas de Linux ou MacOS, sous Windows vous devez installer un client SSH.

— À partir de Windows 10, il suffit d'installer le composant OpenSSH, il se trouve dans "Applications et fonctionnalités", "Gérer les fonctionnalités facultatives", "OpenSSH Client". Cela suffira pour les parties non graphiques.

— Pour les parties graphiques (ou si vous ne pouvez pas installer OpenSSH, on peut utiliser MobaXterm et cliquer sur le bouton "Start local terminal", cela vous lance un shell où vous pourrez lancer toutes les commandes documentées ci-dessous. Il n'y a que la gestion d'agent ssh qui n'est pas supportée, donc au moment de la génération de clé, mettez une *pass phrase* vide.

— vous pouvez aussi utiliser WSL (Windows Subsystem for Linux) qui permet d'intégrer par exemple un Linux/Ubuntu dans Windows 10.

Attention, chacune de ces 3 options vit dans son propre monde, si vous en utilisez plusieurs, il faut suivre ce tutoriel pour chaque pour générer les clés. Si vous utilisez VSCode, il faut utiliser le composant OpenSSH pour pouvoir éditer facilement les fichiers à distance.

2 Générer des clés SSH

Si vous disposez déjà d'une paire de clés privée/publique, vous pouvez directement passer à la section 3.

SSH permet d'éviter l'authentification par mot de passe en utilisant un protocole cryptographique asymétrique s'appuyant sur une paire (clé privée, clé publique). La clé privée restera toujours stockée sur votre ordinateur, tandis que vous déposerez une copie de votre clé publique sur les serveurs auxquels vous souhaitez vous connecter sans taper de mot de passe.

Pour générer votre couple de clés (ici des clés ECDSA), utilisez la commande `ssh-keygen` :

```
[pbismuth@mymachine] ssh-keygen -t ecdsa -b 521
```

La commande vous demande alors d'entrer une « *pass phrase* », c'est-à-dire un mot de passe qui protègera votre clé privée. Nous verrons dans la section suivante comment faire pour ne pas entrer ce mot de passe à chaque connexion SSH.

Une fois la *pass phrase* entrée, la commande `ssh-keygen` génère deux fichiers, `id_rsa` (clé privée) et `id_rsa.pub` (clé publique), dans le sous-répertoire `${HOME}/.ssh/`

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pbismuth/.ssh/id_rsa):

Your identification has been saved in /home/pbismuth/.ssh/id_rsa.
Your public key has been saved in /home/pbismuth/.ssh/id_rsa.pub.
The key fingerprint is:
...
```

3 Faire confiance à son agent

Par défaut, SSH vous demandera d'entrer votre *pass phrase* à chaque fois qu'il aura besoin d'utiliser votre clé privée. Heureusement [1], votre agent SSH est là pour ça : ce processus vous demandera votre « *pass phrase* » en début de session, puis se chargera de l'accès à votre clé privée à chaque fois que le protocole SSH l'exigera.

Pour autoriser votre agent à récupérer votre clé privée, utilisez la commande `ssh-add` :

```
[pbismuth@mymachine] ssh-add
```

L'agent vous demande alors (une seule fois) votre « *pass phrase* », puis il travaillera discrètement pour vous en arrière plan.

1. Sauf sous Windows...

4 Déposer sa clé publique sur le serveur

Il nous reste une étape à réaliser pour autoriser l'authentification sans mot de passe : déposer votre **clé publique** `id_rsa.pub` sur le serveur. Il faut pour cela ajouter votre clé publique au fichier `${HOME}/.ssh/authorized_keys` sur la machine passerelle du CREMI, qui se nomme `jaguar`. Cela s'effectue très facilement au moyen de la commande `ssh-copy-id` :

```
[pbismuth@mymachine] ssh-copy-id mylogin@jaguar.emi.u-bordeaux.fr
```

Note : Si vous utilisez le composant OpenSSH sous Windows, `ssh-copy-id` n'est pas disponible. Il vous faut alors effectuer cette manipulation manuellement, en copiant d'abord votre fichier `id_rsa.pub` au CREMI, puis en le concaténant au fichier `~/.ssh/authorized_keys` :

```
[pbismuth@mymachine]$ scp ~/.ssh/id_rsa.pub mylogin@jaguar.emi.u-bordeaux.fr:
[pbismuth@mymachine]$ ssh mylogin@jaguar.emi.u-bordeaux.fr
mylogin@jaguar$ tee -a .ssh/authorized_keys < id_rsa.pub
```

Ce qui permet d'ajouter la clé sans supprimer celles que vous auriez déjà mises.

Notez que, s'il s'agit de votre première connexion sur `jaguar`, SSH va vous demander de confirmer qu'il s'agit bien d'une machine de confiance :

```
The authenticity of host 'jaguar.emi.u-bordeaux1.fr (2001:660:6101:800:252::4)'
cannot be established.
ECDSA key fingerprint is SHA256:ZLwJaxoK0SZf4G8dVq5SEYvzXJTC7HeD8Zf38rdmjFg.
Are you sure you want to continue connecting (yes/no)?
```

Répondez `yes` et, normalement, votre clé publique est ajoutée au fichier `authorized_keys`. Désormais, vous pouvez vous connecter sans taper de mot de passe sur la machine passerelle `jaguar` :

```
[pbismuth@mymachine] ssh mylogin@jaguar.emi.u-bordeaux.fr
mylogin@jaguar:~$ ls
Bureau
Desktop
Documents
espaces
Musique
Public
Téléchargements
mylogin@jaguar:~$ exit
[pbismuth@mymachine]
```

5 Configurer SSH pour se simplifier la vie

Il est possible de créer un fichier de configuration `${HOME}/.ssh/config` dans lequel on pourra définir des "alias" (i.e. pseudonymes pour les machines), son identifiant de connexion par serveur, ou encore les options d'affichage déporté.

Typiquement, la chaine `mylogin@jaguar.emi.u-bordeaux.fr` est assez longue à taper. Le préfixe `mylogin@` (qui indique votre identifiant de connexion au CREMI) peut être omis si cet identifiant est le même que sur votre machine locale. En général, ce n'est pas le cas.

Pour davantage de commodité, il est possible de créer le fichier `~/.ssh/config` suivant :

```
################ CREMI ################
Host cremi
Hostname jaguar.emi.u-bordeaux.fr
User mylogin
ForwardAgent yes
```

Attention, c'est bien dans `~/.ssh/config` et pas ailleurs. Sous Windows avec WSL il vous emmène par défaut ailleurs que dans ~, il faut donc bien préciser le ~

Il est désormais possible de vous connecter au CREMI de cette façon :

```
[pbismuth@mymachine] ssh cremi
mylogin@jaguar:~$
```

Ou d'y exécuter une commande distante :

```
[pbismuth@mymachine] ssh cremi hostname -f
jaguar.emi.u-bordeaux.fr
[pbismuth@mymachine]
```

Copier un fichier au CREMI est tout aussi facile :

```
[pbismuth@mymachine] scp mon_fichier_local cremi:sous_repertoire_distant/
```

Ou, dans l'autre sens :

```
[pbismuth@mymachine] scp cremi:sous_repertoire_distant/mon_fichier_distant .
```

6 Travailler au CREMI

Attention!

La machine `jaguar` est une passerelle qui accueille toutes les connexions entrantes au CREMI. Pour cette raison, elle ne doit pas être surchargée et elle ne peut donc pas être utilisée pour des compilations ou pour lancer des applications.

Il faut donc n'utiliser la passerelle `jaguar` qu'à titre *transitoire* pour se connecter sur une machine banalisée (salle TP ou serveur). Avant de vous connecter sur une machine, vérifiez que personne ne s'y trouve déjà et allumez-là à distance au besoin, via l'interface web du CREMI :

https://services.emi.u-bordeaux.fr/exam/?page=wol

(rubrique « nos services numériques », puis « démarrage à distance »)

Supposons que vous vouliez travailler sur la machine `hautbrion` du CREMI, vous devez donc procéder ainsi :

```
[pbismuth@mymachine] ssh cremi
mylogin@jaguar:~$ ssh hautbrion
mylogin@hautbrion:~$
```

Note

En temps normal, vous seriez invité à entrer votre mot de passe lors du second `ssh` (et non pas une *pass phrase*) car il n'y a pas de couple de clé privée/publique sur votre compte CREMI… Toutefois, ici on exploite une fonctionnalité très pratique de SSH qui permet de faire suivre les requêtes d'authentification sur `jaguar` à l'agent qui s'exécute sur votre machine locale : c'est l'objet de la ligne `ForwardAgent yes` dans le fichier `~/.ssh/config` établi en section 5.

7 Rebondir automatiquement

Pour simplifier encore la connexion avec la machine choisie au CREMI, il est possible d'automatiser le « rebond SSH » effectué précédemment. Voici le nouveau fichier `~/.ssh/config` permettant de se connecter sur `hautbrion` sans chichi :

```
################ CREMI ################
Host cremi
Hostname jaguar.emi.u-bordeaux.fr
User mylogin
ForwardAgent yes

Host hb
Hostname hautbrion.emi.u-bordeaux.fr
User mylogin
ProxyJump cremi
ForwardX11 yes
```

Note : si votre (vieux) client râle qu'il ne connait pas l'option ProxyJump, vous pouvez remplacer la ligne `ProxyJump cremi` par une version un peu moins efficace :

```
ProxyCommand ssh cremi nc -w 600 %h %p
```

Désormais, vous pouvez vous connecter sur `hautbrion` comme ceci :

```
[pbismuth@mymachine] ssh hb
mylogin@hautbrion:~$
```

Les clients SSH récents proposent aussi une option directement en ligne de commande pour faire un saut intermédiaire sur une machine :

```
[pbismuth@mymachine] ssh -J cremi hautbrion
mylogin@hautbrion:~$
```

8 Afficher des fenêtres graphiques à distance

Lorsque les applications ne sont pas gourmandes en bande passante, vous pouvez déporter l'affichage d'application s'exécutant au CREMI sur votre ordinateur local. Pour cela, vous devez disposer d'un serveur X11 (standard sous Linux ou Mac, installable facilement sous Windows).

Avant de lancer une application graphique, il faut s'être connecté via `ssh -Y` pour spécifier que vous souhaitez rediriger les requêtes d'affichage graphique vers votre poste. Ou bien, comme nous l'avons fait dans le fichier de configuration précédent, utiliser l'option `ForwardX11 yes`.

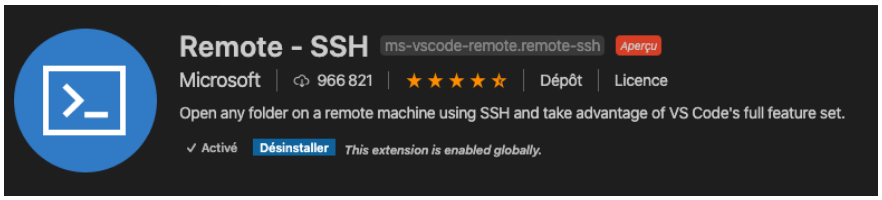
Par exemple, il est possible d'exécuter la commande Unix `xterm` sur la machine `hautbrion` avec un affichage sur votre machine locale :

```
[pbismuth@mymachine] ssh hb xterm
```

9 Éditer les fichiers à distance

Bien que vous puissiez lancer au CREMI un éditeur de texte s'affichant chez vous en suivant la méthode exposée en section 8, c'est fortement déconseillé en raison de la bande passante requise par le protocole X11.

Il est préférable[2] de lancer un éditeur de texte *localement* qui soit capable d'éditer les fichiers distants de manière transparente. C'est le cas d'Emacs (via l'extension `tramp`), ou de Visual Studio Code via l'extension « `remote - SSH` » :



Dans ce dernier cas, il suffit de cliquer dans la zone verte en bas à gauche de la fenêtre : vous serez invité à entrer le nom de la machine distante puis vous pourrez vous y connecter en toute transparence, y lancer des compilations, etc. Bien évidemment, pour les raisons mentionnées en section 6, **il ne faut pas utiliser la passerelle jaguar pour cible**. La figure 1 (page 7) montre une session de travail[3] ouverte depuis un poste hors université.

10 Ouvrir votre bureau de travail... comme si vous y étiez !

Dans la section 8, nous avons évoqué la possibilité d'afficher la fenêtre d'une application CREMI sur votre ordinateur. En fait, vous pouvez aller plus loin et afficher l'intégralité de votre bureau CREMI sur votre machine locale !

2. Bien sûr, il est également possible d'utiliser des éditeurs non-graphiques, tels que `vim` ou `emacs -nw`...
3. sur une machine nommée `scotty`

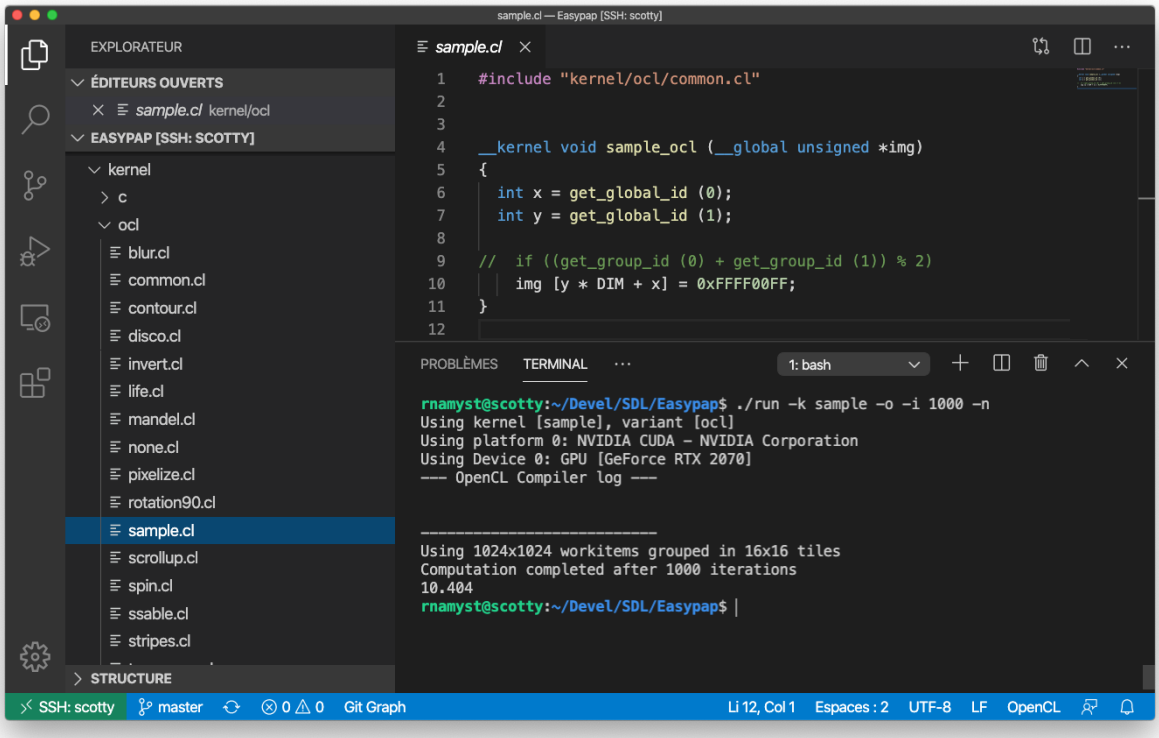


FIGURE 1 – L'extension « `remote – SSH` » de VS Code permet de travailler une machine distante de manière très pratique.

Pour cela, le plus simple est d'installer le client X2Go (qui s'appelle `x2goclient` sous Windows) sur votre machine. Vous trouverez une documentation sur l'intranet du CREMI ici :

`https://services.emi.u-bordeaux.fr/intranet/spip.php?article125`

Attention : Cette solution « bureau distant » sollicite durement le réseau en terme de traffic et risque de mener à des congestions !

11 Mettre en place des tunnels

Dans le cas ou vous où vous souhaiteriez accéder depuis chez vous (maison) à un serveur applicatif qui n'est accessible que depuis l'intérieur du CREMI, il est nécessaire de mettre en place ce qu'on appelle un tunnel. Le tunnel est composé de deux parties, une partie qui est une communication ssh entre votre machine et jaguar et une autre qui est une communication dans le protocole que vous avez choisi entre jaguar et le serveur applicatif ciblé. Un schéma descriptif de ce que l'on veut mettre en place est fourni en figure 2.

Dans cet exemple, nous souhaitons accéder au service web (protocole HTTP, serveur à l'écoute sur port 80) de la machine tesla qui n'est accessible que depuis l'intérieur du CREMI. Nous allons donc mettre en place un tunnel tel que décrit précédemment. Il s'agît d'utiliser la commande ssh

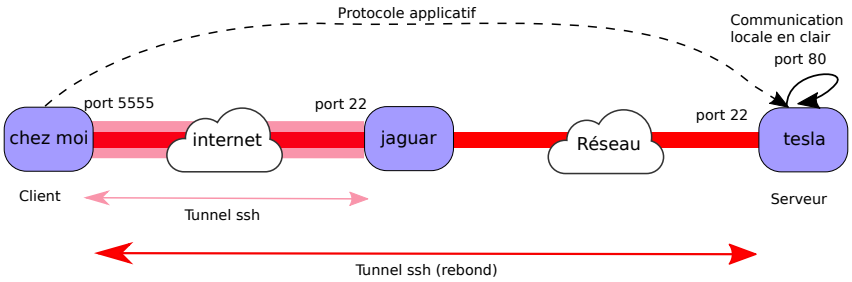


FIGURE 2 – Exemple de tunnel ssh.

suivante (à exécuter depuis votre machine personnelle, en remplaçant mylogin par votre login au CREMI) :

```
ssh -N -L 5555:localhost:80 -J cremi mylogin@tesla
```

Après l'authentification SSH, cette commande va créer le tunnel et va rester en attente sans rendre la main (*). Ainsi toute connexion sur le port 5555 de votre machine reviendra à une connexion à travers le tunnel sur le port 80 (HTTP) de la machine tesla. Il est ainsi possible d'accéder au serveur web de tesla via le tunnel (ce qui n'est pas possible autrement). Pour vérifier le bon fonctionnement de l'opération, il suffit de taper dans la barre de votre navigateur `http://localhost:5555` qui est le point d'entrée du tunnel mis en place avec la commande ci-dessus. Vous devriez obtenir la page illustrée en figure 3 (page 8).

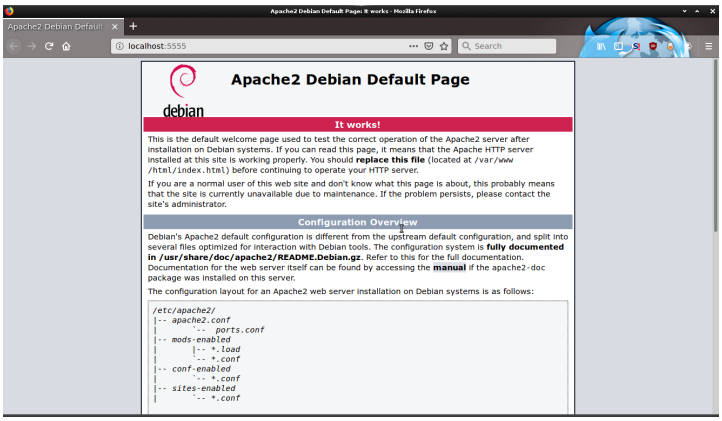


FIGURE 3 – Connexion au port 80 de la machine tesla au travers d'un tunnel SSH.

On peut appliquer cette technique dans de multiples contextes afin d'utiliser un service réseau, comme s'il était local à sa machine, alors que celui-ci n'est pas disponible sur Internet ;-)

Enfin pour supprimer le tunnel, il faut juste faire un ctrl + C à l'endroit où le tunnel est lancé… Et c'est tout !