

Annexe 2 – Accès aux machines virtuelles (VMs)

On peut noter que la VM n'a qu'un CPU (VCPU, pour Virtual CPU), même si elle tourne en réalité sur un serveur physique qui en a beaucoup plus que ça. Tout se passera donc comme si nous étions sur une machine mono-cœur, ce qui simplifie un peu les choses en terme d'ordonnancement.

1 Se connecter à sa VM

Q1. Connectez-vous en utilisant l'utilisateur étudiant. Depuis l'intérieur de l'université, on peut utiliser SSH normalement comme ceci :

```
ssh etudiant@IP_VM
```

Q2. Procédure

1. saisir le mot de passe "etudiant" ~> le prompt indique un changement de mot de passe obligatoire et demande de saisir le mot de passe actuel
2. saisir le mot de passe "etudiant" (une seconde fois donc)
3. saisir le nouveau mot de passe
4. Ressaisir le nouveau mot de passe ~> le serveur termine la connexion
5. `ssh etudiant@IP_VM`
6. saisir le nouveau mot de passe

Si vous faites le TP de l'extérieur de l'université, alors malheureusement la machine virtuelle n'est pas accessible directement. Sinon, utilisez un rebond SSH sur `linuxetu` avec l'une des méthodes indiquées ci-dessus (en utilisant l'IP de votre VM comme nom de machine au lieu de `b71010202.univ-lyon1.fr`, et le login `etudiant` au lieu de votre login Lyon 1).

Par exemple, la connexion avec rebond SSH manuel ressemble à ceci :

```
moy@moylip:~$$ ssh p1234567@linuxetu.univ-lyon1.fr
p1234567@linuxetu.univ-lyon1.fr's password: <mot-de-passe-Lyon1>
[...]
matthieu.moy@linuxetu:~$ ssh etudiant@IP_VM
etudiant@IP_VM's password: <mot-de-passe-de-etudiant>
[...]
etudiant@vm-26:~$
```

Q3. La commande SSH vous a donné un accès shell, en mode texte. Les outils graphiques dont vous avez probablement l'habitude ne sont donc pas disponibles. La commande SSH permet en général d'utiliser des commandes graphiques si on utilise `ssh -X` pour se connecter au serveur, mais cela ne marche pas sur `linuxetu`. Profitons-en pour nous familiariser avec des éditeurs de textes en mode texte : `nano` est un éditeur peu avancé mais simple à utiliser. `emacs` et `vim` sont deux éditeurs très avancés mais demandant une phase d'apprentissage relativement longue. Dans les trois cas, vous pouvez ouvrir un fichier simplement en appelant l'éditeur avec le nom du fichier en paramètre sur la ligne de commande, par exemple `nano lancement.cpp`.

Q4. Récupérez maintenant le fichier `sched.cpp` sur votre VM. Une manière de faire est d'utiliser la commande `wget` (qui télécharge un fichier) depuis la VM :

```
wget https://asr-lyon1.gitlabpages.inria.fr/prog-concurrente/tp5/sched.cpp
```

Une autre est de télécharger le fichier sur votre PC physique, puis de l'envoyer à la VM avec une commande comme :

```
rsync -av sched.cpp etudiant@IP_VM:
```

(malheureusement, cette commande ne fonctionnera pas directement depuis l'extérieur de l'université sans VPN ou ajout dans le `~/.ssh/config` comme indiqué ci-dessus)

Q5. Vous pouvez passer root via la commande `sudo su`. Il est conseillé de travailler comme utilisateur normal (`etudiant`), et de ne passer root que quand c'est nécessaire, c'est à dire pour lancer l'exécutable de votre TP pour le cas qui nous intéresse. Par exemple :

```
$ nano sched.cpp
$ g++ -std=c++11 -pthread sched.cpp -o sched
$ sudo ./sched
```

Pour le TP d'ordonnement, votre programme ne s'exécutera qu'en étant root. Vous pourrez exécuter les commandes `g++` et `nano` normalement avec l'utilisateur `etudiant`, alors que `./sched` s'exécutera avec l'utilisateur `root`, comme ceci :

```
$ nano sched.cpp
$ g++ -std=c++11 -pthread sched.cpp -o sched
$ sudo ./sched
```

Attention : la VM que vous avez créée pourra être supprimée sans avertissement. En fin de TP, récupérez vos données sur votre compte (par exemple avec la commande `rsync`).

2 Un peu de confort avec VSCode

Si vous avez vos habitudes avec un éditeur graphique avancé (comme VSCode que nous utiliserons ici), vous aurez peut-être envie de les conserver en travaillant à distance. Installer VSCode sur la VM et l'utiliser à distance serait pénible : beaucoup d'espace disque gaspillé sur la VM, et l'affichage graphique déporté est assez lent. Nous allons faire autrement : utiliser VSCode en local, mais lui demander d'accéder aux fichiers à distance via SSH.

- Installer l'extension *Remote - SSH*¹
- Configurez le rebond SSH automatique dans `~/.ssh/config` si ce n'est pas déjà fait (cf. section ??)
- Cliquer sur l'icône « Open a remote window » avec une icône ressemblant à `>>` en bas à gauche de l'écran, ou faites `Control+Shift+P` puis entrez la commande `>Remote-SSH: Connect to Host...`
- Entrer `etudiant@IP_VM` et validez. Entrer vos mots de passes quand VSCode vous les demande.
- Et voilà, vous avez une fenêtre VSCode connectée à votre VM. Vous pouvez l'utiliser presque comme si vous étiez en local sur la VM.

Plus d'informations : <https://code.visualstudio.com/docs/remote/ssh>.

1. <https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-ssh>

3 Créer sa VM

Cette partie est gardée pour mémoire, mais cette année la VM est déjà créée pour vous par votre serveur. L'adresse IP de votre VM est disponible dans TOMUSS, colonne IP_VM. Dans toutes les commandes qui suivent, IP_VM est à remplacer par l'adresse IP en question.

Pour cela, vous devez vous connecter à l'interface d'administration :

<http://cloud-info.univ-lyon1.fr/> (inaccessible depuis l'extérieur du réseau de l'université) et utiliser le domaine UNIV-LYON1.fr, avec vos identifiants habituels Lyon 1. Vous faites partie du projet OpenStack ASR7 (si besoin, sélectionnez-le depuis le menu déroulant en haut de l'écran. Vous devez bien faire attention à ne pas créer trop de machines, ni utiliser trop de ressources.

Remarque : L'URL précédente, tout comme les VMs que vous pourrez générer, n'est pas accessible depuis l'extérieur de l'Université, sauf en passant par une machine relais, c'est-à-dire un proxy. Des explications pour vous y connecter sont disponibles en section ??.

Vous devez créer une instance de votre machine grâce à
Projet->Calcul->Instances->Lancer une instance.

Faites attention à :

- donner un nom reconnaissable pour votre instance ;
- utiliser l'image Ubuntu Server 22.04.1 LTS - Login étudiant ;
- utiliser le gabarit (ensemble de ressources) m1.tiny ;
- ne créer qu'une seule instance ;
- Lancer la création.

La machine virtuelle va être créée (cela demande un peu de temps) et apparaître dans la liste des Instances. Elle obtiendra une **adresse IP** que vous noterez. Nous allons l'utiliser plus loin, sous le nom IP_VM. **Attendez** que l'état de l'alimentation soit En fonctionnement avant de pouvoir vous y connecter.