

Ssh, VM et client/serveur Jami

Ensimag 1A

Grégory Mounié (CC-BY-SA )

2024-2025

L'exécution de Jami et son utilisation pour communiquer avec vos camarades n'est pas le but. Vous offrir une meilleure compréhension des processus, mémoire et fichiers, est le but. N'hésitez pas à poser des questions.

Un serveur est un programme qui s'exécute de manière continue pour répondre à des requêtes de clients. Ce programme est exposé aux attaques et doit rester disponible. Il faut donc maintenir le programme du serveur à jour. De plus, il faut aussi maintenir les dépendances du programme à jour. Et le tout doit rester constamment fonctionnel.

Une méthode ancienne, les machines virtuelles,¹ consiste à simuler une machine complète (un PC, avec CPU, RAM, disques, écran, carte son, carte réseaux, etc.).

Pour communiquer avec cette autre machine, il faudra passer le réseau (simulé). SSH devient alors l'outil de choix de cette communication. Il va nous permettre des communications et la configuration notre machine virtuelle.

Le TP est calibré pour occuper les plus rapides d'entre vous jusqu'à la fin du TP. Le sujet donne des indications générales avec peu d'aide. Une partie de l'exercice est que les plus rapides entraînent les autres pour que l'ensemble du groupe termine le sujet. Si vous êtes bloqué, vous trouverez à la fin du sujet la liste des commandes.

Ce document peut être téléchargé depuis l'adresse suivante :
<http://systemes.pages.ensimag.fr/unix-gitlab/atelier/ssh-vm.pdf>.

1. les mainframes exécutant des machines virtuelles sont apparus dans les années 1950, mais disons que leur implémentations moderne va de 1970 à nos jours.

1. Penser à effacer l'image de disque que vous allez créer

Nous allons fabriquer un fichier de 5 Gio dans votre HOME. N'oubliez pas que vous ne pouvez pas dépasser 10 Gio au total (votre quota). Ce TP étant en début d'année, donc cela ne posera aucun problème si **vous pensez bien à effacer votre image lorsqu'elle sera devenue inutile!**.

2. Fabriquer sa machine virtuelle : installer une Debian Linux en mode texte dans Qemu

Nous allons installer GNU/Linux Debian. C'est notre compromis entre facilité d'installation, taille raisonnable, complétude et choix importants explicites à l'installation, pour le côté pédagogique.

2.1. Installation d'un système d'exploitation dans un simulateur de PC

Nous utiliserons le simulateur de PC Qemu, avec son interface en ligne de commande (CLI). L'utilisation d'un simulateur permet de contrôler facilement le comportement de l'installation. Par contre cela oblige à définir, le nombre processeur simulé, la taille de la mémoire simulée, l'organisation du stockage, le réseau simulé, etc. Il faut aussi configurer les relations réseaux entre le système d'exploitation hôte (host) qui s'exécute sur votre machine physique, et le système d'exploitation invité (guest) qui s'exécute dans le simulateur.

Votre compte Ensimag a un quota maximal de 10 Gio. Pour faire ce TP nous allons utiliser un "disque" simulé de 5 Gio. Il faudra penser à l'effacer après le TP.

2.2. Démarrage d'un PC

Le démarrage et la gestion d'un ordinateur fait intervenir trois logiciels qui collaborent : l'UEFI/BIOS, le chargeur/amorce/bootstrap (« bootloader ») et le système d'exploitation. Les liens entre les trois ne sont vraiment visibles que lors des installations ou certains bugs autour la mise en veille.

L'UEFI (souvent encore appelé BIOS) est le logiciel écrit « en dur » dans une mémoire permanente et qui est le premier code exécuté par le processeur lorsque vous allumez la machine. Il est responsable de la détection et de l'allumage du matériel de base : taille de la mémoire, contrôle de l'alimentation électrique des périphériques, etc. L'UEFI/BIOS termine en trouvant un *chargeur* et en lui passant la main.

Le chargeur (Grub) est le logiciel qui va copier le code du système d'exploitation dans la mémoire. Son code doit être à un endroit accessible par l'UEFI/BIOS : les 512 premiers octets d'un disque, ou dans une partition (morceau) d'un disque géré par UEFI.

2.3. Préparer un fichier creux comme disque

Avant de lancer le simulateur, nous allons préparer un fichier pour servir de stockage pérenne (équivalant à un disque dur ou un ssd) de 5 Gio. Il est possible de créer un fichier de 5 Gio mais qui restera creux (de taille nulle) tant qu'il ne sera pas utilisé.

Créez le fichier «creux» (`3mmunix_vmdisk.img`) avec la commande

```
1 dd of=3mmunix_vmdisk.img bs=5G seek=1 count=0
```

Il est possible de vérifier que votre disque est bien un fichier de taille 5 Gio creux, occupant pour l'instant 0 octet sur le disque.

```
1 alice@laptop1% ls -lh 3mmunix_vmdisk.img
2 -rw-rw-r-- 1 gregory gregory 5,0G 23 juil. 19:28 3mmunix_vmdisk.img
3 alice@laptop1% du -h 3mmunix_vmdisk.img
4 0          3mmunix_vmdisk.img
```

Démarrer le simulateur en lui indiquant d'utiliser 8 Gio de mémoire, l'image ISO Debian DVD comme cdrom et le fichier que nous venons de créer comme disque dur. Nous ajoutons aussi un réseau, pour la partie ssh qui suivra l'installation.

Attention à bien lancer la commande complète, même si elle apparaît sur plusieurs lignes dans ce sujet :

```
1 qemu-system-x86_64 -machine q35,accel=kvm -m 8G -cdrom
  → ~jdpunix/iso/debian-12.7.0-amd64-DVD-1.iso -nic
  → user,hostfwd=tcp:127.0.0.1:2222-:22 3mmunix_vmdisk.img
```

L'installation consiste principalement à appuyer de nombreuses fois sur « Entrée ». Il faut aussi appuyer une ou deux fois sur « Tabulation » et donner les mots de passes de l'administrateur (root) et le login et le mot de passe d'un utilisateur. Vous devriez réussir.

2.4. Installer Debian sur notre pseudo-disque de 5 Gio (environ 10 min)

Comme nous avons un disque petit, **ne pas choisir l'installation graphique, "Graphical Install"** mais choisir de lancer l'installateur en mode texte : « Installer ».

En cas d'erreur arrêter Qemu et relancez-le. Vous n'avez pas encore manipulé le disque simulé donc l'arrêt brutal ne pose aucun problème.

Langue et utilisateurs Après avoir choisi la langue, vous devrez choisir le mot de passe de l'administrateur, ainsi que le login et le mot de passe de l'utilisateur standard. Vérifiez bien que la disposition de clavier est en azerty pour le choix du mot de passe...

Partition du disque Le choix suivant est un peu délicat. Il faut choisir quelle place donner à Linux sur le disque. Avec notre disque vide, c'est particulièrement simple. Debian vous propose un choix par défaut (partitionnement assisté), qui est le plus simple et qui convient très bien dans notre cas.

Miroir réseau Il faut choisir un miroir pour pouvoir installer les logiciels par le réseau dans la deuxième partie ssh.

Les « miroirs réseau » sont des serveurs web qui contiennent l'ensemble des logiciels intégrés dans Debian. Pour que l'on puisse installer des logiciels une fois le système installé, il est donc nécessaire de choisir un miroir. Prenez un miroir en France pour que ce soit plus rapide. Vous pouvez garder celui par défaut ou, à l'Ensimag, celui d'une université (le réseau d'interconnexion des universités européennes est très rapide et l'Ensimag fait partie de ce réseau).

La fin de l'installation jusqu'au le bootloader La suite ne devrait pas vous poser de problème. La copie des données sur le disque peut prendre jusqu'à plusieurs minutes. À la fin de l'installation, Debian vous demande sur quel disque vous souhaitez installer le bootloader. Il faut choisir le disque virtuel. Si vous n'installez pas le bootloader, alors votre machine virtuelle ne pourra pas redémarrer !

2.5. Premier redémarrage et connexions

Après un redémarrage, vous obtenez à nouveau le menu de Grub, mais cette fois il démarre votre OS nouvellement installé.

Connectez-vous comme l'utilisateur standard, avec le login et le mot de passe que vous avez choisi à l'installation. Puis de cet utilisateur, devenez « root » (l'administrateur), soit avec `su -`. Nous installerons sudo dans la partie suivante.

```
1  alice@vm$ su - # votre mot de passe de root
2  # exit
3  alice@vm$
```

2.5.1. Se connecter à l'hôte depuis votre machine virtuelle

Il est possible de se connecter, par défaut, de la machine virtuelle vers l'hôte, même si vous n'avez pas mis l'option `-nic ...` à la commande Qemu. Par défaut, votre machine a comme adresse `10.0.2.15`. La machine hôte est sur le même réseau à l'adresse `10.0.2.2`. En utilisant votre login Ensimag, vous devriez pouvoir vous connecter avec ssh depuis votre Debian vers votre Ensipc.

Connectez-vous par ssh depuis votre machine virtuelle vers votre hôte pour vérifier que cela fonctionne, puis déconnectez-vous.

```
1  alice@vm$ ssh arecoque@10.0.2.2
2  arecoque@ensipc$ exit
```

Mais dans la vraie vie, votre VM est souvent distante, et donc c'est de l'inverse dont vous avez besoin : vous connectez depuis l'hôte vers la VM.

C'est ce que nous utiliserons dans la partie suivante.

3. SSH

L'une des forces des UNIX est de pouvoir tout faire dans un terminal shell. Comme il est facile d'obtenir à distance un accès à un terminal avec SSH, il est donc facile travailler sur un ordinateur distant sous UNIX.

Ssh est un outil essentiel dans les systèmes distribués (Clouds), ou de grands systèmes distants (Datacentres, Supercalculateurs). Ssh est présent depuis sa création au millénaire dernier sous tous les Unix, dont Linux et MacOS X. Depuis le premier confinement Covid en 2020 aux US, Ssh est disponible par défaut aussi dans Windows.

3.1. Ce que nous ne verrons pas dans ce TP : les clefs publiques/clefs privées et autres sujets

Les clefs publiques et privées servent à se connecter sans avoir à taper votre mot de passe à chaque fois. L'idéal est de les utiliser avec un *agent SSH*. Ce sujet sera traité plusieurs fois dans votre scolarité, notamment en Réseaux 1A, en TP d'UNIX avancés 1A, pour Gitlab, etc. Nous ne proposons pas de faire, tout de suite, des exercices redondants sur ce sujet.

Pour en savoir un peu plus, nous vous conseillons de lire les différents manuels de SSH, très complets et avec de nombreux exemples

```
1 you@ensipc$ man ssh
2 you@ensipc$ man ssh_config
3 you@ensipc$ man ssh-agent
4 you@ensipc$ yelp man: ssh-add # viewer gnome
5 you@ensipc$ man sshd
6 you@ensipc$ emacs -q --eval='(progn (man "ssh-keygen")(delete-window))'
  ↪ # emacs sait construire la table des matières du man, très utile
  ↪ dans les très longs man
7
```

Si vous voulez vraiment tout savoir sur SSH, nous vous conseillons le wikibook sur OpenSSH (<https://en.wikibooks.org/wiki/OpenSSH>).

3.2. Les connexions SSH : obtenir un shell distant vers votre VM

Comme vous n'êtes pas administrateur des ensipc (sauf les bugs-busters qui liraient ce sujet), vous ne pouvez pas changer leurs tables de routages. Donc le Qemu que nous avons lancé avec l'option `-nic user,hostfwd=tcp:127.0.0.1:2222-:22` va s'accrocher au

port 2222 de votre ensipc, pour renvoyer le trafic vers le réseau simulé au port 22 (ssh) de votre machine virtuelle.

Se connecter en SSH à votre machine virtuelle consiste donc à se connecter en SSH au port 2222 de votre ensipc.

Connectez-vous !

3.3. Installation d'autres logiciels dans votre VM

La grande force de Debian est son immense catalogue de logiciels (environ 50 000), bien testés et cohérents les uns avec les autres. Debian est une distribution communautaire, avec plus d'un millier de développeurs à travers le monde qui s'occupent de cette base de logiciels et de leur intégration dans Debian.

3.3.1. Su ou sudo

Au début, l'utilisateur de base n'a aucun droit. L'administrateur peut lui donner les droits d'utiliser sudo, pour faire ce que l'administrateur peut faire.

Pour cela, il faut se connecter comme administrateur dans un terminal et ajouter le logiciel sudo et l'utilisateur au groupe sudo.

```
1 su - # pour devenir administrateur
2 apt update
3 apt install sudo
4 adduser alice sudo # pour ajouter l'utilisatrice alice au group sudo
5 groups alice # devrait afficher plein de groupes, dont sudo
```

L'appartenance au groupe ne sera valide qu'à la prochaine connexion de l'utilisateur. Déconnectez-vous et reconnectez-vous.

Si sudo ne marche pas et que l'utilisateur est bien dans le groupe sudo, fermer et redémarrer la machine virtuelle puis reconnectez-vous.

Si vous êtes impatient, vous pouvez vous reconnecter comme utilisateur alice dans un terminal :

```
1 id # devrait afficher alice
2 su - alice # pour devenir alice, mais avec les nouveaux groupes
3 sudo ls /root # devrait maintenant fonctionner
```

3.4. Installer tmux

Le réseau entre une machine virtuelle et vous n'est pas toujours stable. Sur les ensipc, ce n'est pas un problème, mais ce n'est pas une raison pour ne pas faire les choses bien.

Parfois, lorsque l'on travaille à distance sur une machine, on peut avoir besoin de couper la connexion avec cette machine (le TGV arrive en gare et vous allez perdre son wifi, la batterie de votre ordinateur portable est presque vide, vous êtes en train de vous endormir

sur votre clavier, etc.). La loi de Murphy impose que cela arrive toujours au mauvais moment.

Vous aimeriez aussi pouvoir lancer facilement à distance un autre shell pendant que votre commande préférée s'exécute. Mais faut-il alors lancer une nouvelle connexion SSH ?

Il existe deux outils très similaires pour vous aider `screen` et `tmux`. Dans le reste de l'exercice, nous allons utiliser `tmux` qui est plus explicite dans ses commandes.

Pour mettre à jour la liste des logiciels disponibles :

```
1 sudo apt update
```

Installer tmux dans la machine virtuelle (vous pouvez utiliser sudo et apt)

```
1 alice@vm$ sudo apt install tmux
2 [sudo] Mot de passe de alice :
3 Lecture des listes de paquets... Fait
4 ... < penser à dire (O)ui > ...
5 alice@vm$
```

Lancez la commande tmux !

Lancez un petit compte à rebours dedans

```
1 TICS=300; while [ $TICS -gt 0 ]; do printf "$TICS ... "; sleep 5;
↪ TICS=$(( TICS - 5 )); done
```

3.4.1. Les commandes de tmux

Toutes les commandes de `tmux` commencent par `C-b` (Control b). Si vous connaissez le nom de la commande mais pas son raccourci, vous pouvez taper `C-b :` (Control b puis deux-points, ce qui déclenche la commande *command-prompt*) auquel vous donnez le nom de la commande. La liste des commandes, et des raccourcis, est disponible avec `C-b ?` (*list-keys*).

3.4.2. détachement et attachement

Tapez la commande `C-b d` (*detach-client*) pour revenir dans le shell qui a lancé tmux, puis quittez la session SSH en tapant :

```
1 exit
```

Reconnectez-vous avec SSH, puis lancez la commande

```
1 tmux attach
```

Vérifiez que le compte à rebours est toujours en train de tourner.

3.4.3. Terminer brutalement une connexion SSH

Il arrive parfois de vouloir demander à SSH de terminer la connexion. Par exemple, si vous avez tué brutalement le serveur SSH distant par exemple en demandant le reboot, ou si une connexion graphique en tâche fond est toujours active.

Le raccourci est de taper «~.» en début de ligne (tilde et point). Il faut donc, de temps en temps, les précéder de la touche «Enter» pour passer en début de ligne.

Interrompez brutalement la connexion ssh !

C'est un arrêt brutal.

Reconnectez-vous avec SSH, puis relancez la commande `tmux attach`.

Vérifiez que le compte à rebours est toujours en train de tourner.

3.4.4. Remonter dans l'historique

Vous avez peut-être remarqué que vous ne pouvez pas vous servir de l'ascenseur de votre terminal pour voir ce qui c'est affiché avant.

Remontez dans l'historique en tapant `C-b [` (copy-mode) puis en utilisant les flèches ou page-up page down, puis `q` pour terminer. Vous pouvez aussi copier-coller le texte avec les raccourcis Emacs ou vim.

3.4.5. Ouvrir plusieurs shells

Pour obtenir plusieurs shells s'exécutant simultanément, il existe deux méthodes complémentaires utiles.

Ouvrir une nouvelle fenêtre (window) Créez un nouveau shell avec la commande `C-b c` (new-window). Naviguez entre les shells avec `C-b n` (next-window) et `C-b p` (previous-window).

Naviguez entre les shells 0 et 1 avec les commandes `C-b 1` (select-window) et `C-b 0`.

Scinder une fenêtre horizontalement et verticalement Créez un nouveau shell en scindant verticalement la fenêtre avec la commande `C-b "` (split-window).

Lancez un compte à rebours dedans.

Créez un nouveau shell en scindant horizontalement la fenêtre avec la commande `C-b %` (split-window -h).

Lancez une horloge dedans avec `C-b t` (clock-mode)

Vous avez maintenant 3 shells dans cette fenêtre. Vous pouvez naviguer entre ces shells avec `C-b o` (select-pane)

3.5. Installer «cowsay» «lolcat», «fortune-mod» et «fortune»

Pour vérifier que vous avez tout compris, nous allons ajouter quelques autres logiciels.

Bien qu'étant des programmes particulièrement indispensables, les programmes "lolcat", "cowsay" et "fortunes-debian-hints" ne sont pas installés par défaut. Installez-les.

Pour vérifier que tout fonctionne, vous pouvez les chaîner avec deux pipes (à prononcer à l'anglais (tuyau))

```
1  alice@vm$ fortunes | cowsay | lolcat
2  ... < un joli dessin informatif > ...
3  alice@vm$
```

4. Installer Jami (dernière version) à la main, dans la VM

Nous allons maintenant installer une application graphique dans la machine virtuelle, pour s'en servir à distance depuis ensipc. Mais nous n'avons pas installé d'environnement graphique, donc il va manquer des choses, mais tout installer dépasserait la place disponible!

Les commandes `df -h`, `rm`, `apt clean` devraient vous permettre de survivre.

4.1. SSH pour déplacer ses fichiers entre plusieurs machines

Nous avons choisi une application en cours de développement rapide, car sa dernière version n'est probablement pas celle disponible dans une Debian stable.

Nous pourrions utiliser snap (comme sur votre Ubuntu) ou flatpak, mais nous voulons aussi vous faire copier des fichiers entre l'extérieur et l'intérieur de la VM.

Sur votre ensipc, téléchargez la version debian (12, stable/bookworm) de Jami (<https://jami.net>), un logiciel de chat et communication sans serveur central.

Téléversez ce fichier dans la VM avec la commande scp (ou sftp si vous préférez).

Installez ce fichier dans votre VM Debian avec la commande `dpkg`. Mais il manquera des dépendances.

Effacez le .deb après utilisation, et les caches de apt pour maximiser la place disponible.

Ce qui donne au final la séquence suivante :

```
1  alice@vm$ sudo dpkg -i jami-all_amd64.deb
2  [sudo] Mot de passe de greg :
3  Sélection du paquet jami-all précédemment désélectionné.
4  ... < cela va rater et il manque plein de dépendances >
5  alice@vm$ rm jami-all_amd64.deb
6  alice@vm$ sudo apt clean
7  alice@vm$ sudo apt --fix-broken install # pour ajouter les
   ↪ dépendances qui manquent
8  alice@vm$ sudo apt clean
9  alice@vm$
```

4.2. SSH avec le forwarding X11 : se créer un compte jami

En utilisant `ssh -XC...` depuis votre ensipc, vous pouvez lancer `jami`. Vous pouvez vous créer un compte jami.

Jami est un service distribué. Il n'y a pas de serveur central qui stocke les données de votre compte. Il faut donc bien les conserver.

Pour permettre de vous retrouver facilement, vous pouvez quand même choisir un nom qui sera propagé dans l'annuaire de jami. Cela permettra à vos camarades de vous retrouver plus facilement que votre ID numérique (la longue séquence de lettres et de chiffres).

4.3. Sauvegardez votre compte jami que vous avez créé dans la VM

Si vous effacez la VM, vous allez perdre l'information du compte. Dans Jami, sauvegarder votre compte dans un fichier `compte_jami`.

Télécharger ce fichier de sauvegarde sur votre ensipc. Si vous reconstruisez la VM plus tard, ou si vous installez jami sur votre téléphone ou votre laptop, vous pourrez reprendre le même compte en repartant du fichier.

L'autre alternative est d'associer le même compte Jami sur votre téléphone Android, IOS, votre laptop, etc. pour perdre ce compte, il faut alors perdre ou effacer la totalité des machines ou il a été associé.

4.4. Utilisez Jami ! Discutez avec vos camarades !

Pour cela, il faut pouvoir les contacter (par l'annuaire, ou par l'ID très long). Pour la vidéo, cela ne marchera pas dans la VM. Vous n'avez pas simulé de webcam.

A. Solutions

```
1 dd of=3mmunix_vmdisk.img bs=4G seek=1 count=0
2 qemu-system-x86_64 -cdrom
   ↪ ~jdpunix/iso/debian-live-12.6.0-amd64-standard.iso -m 8G -machine
   ↪ q35,accel=kvm -nic user,hostfwd=tcp:127.0.0.1:2222-:22
3 # faire l'installation
4
5 # DANS LA VM, connexion SSH: VM -> ensipc
6 ssh alice@10.0.2.2 # puis exit
7
8 # depuis ensipc ouvrir SSH ensipc -> VM
9 ssh ssh://alice@localhost:2222
10
11 # dans la connexion ssh ensipc -> VM
12 su -
```

```
13 apt install sudo
14 groups alice
15 adduser alice sudo # si besoin
16 # fermer SSH
17 exit
18
19 # reouvrir la connexion ssh ensipc -> VM
20 ssh ssh://alice@localhost:2222
21
22 # dans la connexion ssh ensipc -> VM
23 sudo apt install tmux
24 tmux
25 TICS=300; while [ $TICS -gt 0 ]; do printf "$TICS ... "; sleep 5;
   ↪ TICS=$(( TICS - 5 )); done
26
27 sudo apt install cowsay lolcat fortunes-debian-hints
28 fortunes | cowsay | lolcat
29
30 # Sur ensipc, après y avoir téléchargé jami-all_amd64.deb
31 scp -P 2222 ~/Downloads/jami-all_amd64.deb alice@localhost:
32
33 # dans la connexion ssh ensipc -> VM
34 sudo apt clean
35 sudo dpkg -i jami-all_amd64.deb
36 rm jami-all_amd64.deb
37 sudo apt --fix-broken install
38 sudo apt clean
39
40 # Sur ensipc
41 ssh -XC ssh://alice@localhost:2222 jami
42
43 # sur ensipc
44 scp -P 2222 alice@localhost:compte_jami .
```