

# Déploiement

## Job 1 - Virtualisation

Pour pouvoir visualiser un système d'exploitation il est nécessaire d'installer un logiciel d'hypervision qui servira de visualiser d'autres systèmes d'exploitation . Pour cela nous allons installer VirtualBox d' Oracle (distribué sous licence GPL). Mais tout d'abord qu'est ce que VirtualBox et comment l'installer ?

### 1. VirtualBox d'Oracle

Oracle VM VirtualBox (anciennement VirtualBox) est un logiciel libre de virtualisation publié par Oracle. VirtualBox est installé sur une machine hôte, cette machine représente la machine physique qui va "héberger" une ou plusieurs machines virtuelles. En anglais, cela se nomme "Host".

On installe sur la machine host le système hôte ou système d'exploitation OS tels que Linux, MacOS et Windows.

L'hyperviseur VirtualBox alloue et gère la machine invité (machine virtuelle). On peut parfois l'appeler "client " mais ce terme est moins adapté car il ne faut pas le confondre avec le client d'un système client/serveur. En anglais, cela se nomme "guest".

Ensuite, sur la machine virtuelle on va installer le système invité qui représente le système d'exploitation qui est installé sur la machine virtuelle (ex : Windows 98, Linux Debian, Linux Ubuntu).

Enfin, on peut aussi trouver des suppléments invités qui sont une sorte de pack logiciel à installer sur la machine virtuelle pour optimiser son fonctionnement et ajouter de nouvelles fonctionnalités (dossiers partagés, glisser-déposer, ajustement automatique de la résolution de l'écran, etc.). La licence interdit d'utiliser les add-on en entreprise ou en université.

VirtualBox est un hyperviseur de type 2, c'est-à-dire qu'il doit être installé sur un système d'exploitation, et non directement sur un ordinateur en tant que système d'exploitation.

Pour installer VirtualBox, il suffit de se rendre sur le site <https://www.virtualbox.org/> et suivre le lien d'installation qui est assez lisible et compréhensible.

## Job 2 - Installation de GNU/Linux

La première étape pour utiliser VirtualBox est d'y installer un système d'exploitation mais avant cela il faut le télécharger. Dans notre exemple d'aujourd'hui, nous installerons Debian pour travailler sous Linux. Debian est un système d'exploitation Linux composé exclusivement de logiciels libres.

Debian réunit autour d'un noyau de système d'exploitation de nombreux éléments développés indépendamment les uns des autres, pour plusieurs architectures matérielles. Ces éléments, programmes de base complétant le noyau et logiciels applicatifs, se présentent sous forme de « paquets » qui peuvent être installés en fonction des besoins.

Debian est utilisée comme base de nombreuses autres distributions, telles que Linux Mint et Ubuntu ou encore Q4OS.

Nous ajoutons ensuite une nouvelle VM que j'ai appelée "MyFirstVirtualMachine" j'y est défini mon administrateur ainsi qu'un utilisateur ensuite le choix des logiciels à installer. Un des logiciels les plus importants à installer est GNU GRUB.

Ce programme est un programme d'amorçage de micro-ordinateur. Il s'exécute à la mise sous tension de l'ordinateur, après les séquences de contrôle interne et avant le système d'exploitation proprement dit, puisque son rôle est justement d'en organiser le chargement. Lorsque l'ordinateur héberge plusieurs systèmes (on parle alors de multi-amorçage), il permet à l'utilisateur de choisir quel système démarrer et c'est un logiciel libre. Il est utilisé pour Linux ou pour Windows.

## Job 03 - La fenêtre noir

Cette fenêtre noir s'appelle le terminal cette interface est équivalente à l'invite de commande Windows. Mais qu'est ce que c'est un terminal ? Comment fait-il pour répondre à nos demandes ?

## 1. Le terminal

Un terminal est une application permettant de dialoguer avec son ordinateur via l'écriture de lignes de commandes. Une ligne de commande est une instruction écrite, qui envoyée à l'ordinateur via le terminal, va effectuer une action, comme par exemple créer un répertoire, lancer un programme ou afficher le contenu d'un fichier.

Le terminal est l'ancêtre de programme comme l'explorateur Windows ou Finder (Mac).

Le terminal sert à naviguer dans une arborescence de fichiers, à créer/supprimer et déplacer des dossiers et des fichiers et lancer des programmes.

## 2. Le shell

Le shell (ou interface système informatique) est un programme qui reçoit des commandes informatiques données par un utilisateur à partir de son clavier pour les envoyer au système d'exploitation qui se chargera de les exécuter.

Au début de l'informatique, le shell était la seule interface utilisateur disponible sur un système de type Unix tel que Linux. De nos jours et avec l'arrivée de la souris, tous les ordinateurs ont commencé à utiliser des interfaces graphiques en plus des interfaces de ligne de commandes comme le shell.

Le mot shell signifie enveloppe ou coque en français : à l'inverse d'un noyau d'un ordinateur, le shell désigne la couche la plus haute de toutes les interfaces des systèmes Unix (Linux, macOS).

Le bash est le shell de Linux (qui signifie Bourne Again SHell) agit en tant que programme Shell. Sur macOS, le shell est accessible par l'application Terminal. Un terminal est un programme qui permet à l'utilisateur d'interagir avec le shell. Plusieurs émulateurs de terminaux existent sur Linux, comme gnome-terminal, Konsole, xterm, rxvt, kvt, nxterm et eterm.

## 3. xterm

Xterm est le terminal de Linux.

## Job 4 - Commandes système

Commandes Linux	A quoi ça sert
pwd = Print Working Directory	Afficher le répertoire en cours
cd	Changer de répertoire
cd..	Revenir au répertoire précédent
ls	Lister les fichiers présents dans un répertoire
ls -l	Lister les fichiers présents dans un répertoire avec leurs droits associés, sous forme de liste et en incluant les fichiers cachés
touch Nom_Du_Fichier	Créer un fichier
sudo	Permet à l'administrateur du système d'autoriser certains utilisateurs à exécuter des commandes en tant que superutilisateur
rm	Supprimer un ou plusieurs fichiers d'un répertoire
cat	Afficher le contenu d'un fichier
mkdir	Créer un répertoire
ln	Créer un lien symbolique

<code>rm -r + nom du répertoire</code>	Supprimer un répertoire
<code>cp</code>	Copier un répertoire
<code>cp_mv</code>	Renommer un répertoire
<code>mv</code>	Déplacer un répertoire
<code>find</code>	Afficher le manuel de la commande "find"
<code>find -name</code>	Chercher un fichier sur votre disque en se basant sur son nom
<code>grep</code>	Rechercher du texte dans un fichier

<code>echo</code>	Afficher le texte "Bonjour tout le monde"
<code>history</code>	Afficher l'historique des commandes qui ont été tapées
<code>cat /proc/version</code>	Afficher la version du système d'exploitation installée
<code>date</code>	afficher la date et l'heure
<code>uptime</code>	afficher la durée depuis laquelle le système d'exploitation est allumé
<code>aptitude update</code>	Recherche les mises à jour disponibles pour le système

<code>aptitude full-upgrade</code>	Installer les nouvelles mise à jour disponibles depuis la dernière recherche
<code>sudo passwd root</code>	se connecter en tant que superutilisateur
<code>sudo apt-get install emacs25-nox</code>	Installer l'éditeur de texte "emacs"
<code>ifconfig</code>	Connaître son/ses adresses ip

## Job 5 - Prise en main à distance

Pour se connecter à distance on va installer un serveur ssh. Mais d'abord à quoi ça sert réellement ?

### 1. Le protocole ssh

SSH est un protocole permettant d'établir une communication chiffrée, donc sécurisée (on parle parfois de *tunnel*), sur un réseau informatique (intranet ou Internet) entre une machine locale (le *client*) et une machine distante (le *serveur*).

La sécurité du chiffrement peut être assurée par différentes méthodes, entre autres par mot de passe ou par un système de clés publique / privée (mieux sécurisé, on parle alors de cryptographie asymétrique).

Il y a plusieurs usages de SSH. Dans notre cas, on va l'utiliser pour accéder à distance à la console en ligne de commande (shell) ce qui permet d'effectuer la totalité des opérations courantes et/ou d'administration sur la machine distante. OpenSSH est la solution la plus utilisée pour mettre en place une communication SSH via un ensemble d'outils libres.

### 2. Installation du serveur SSH

Il faut tout d'abord installer le paquet openssh-server sur son poste. Le serveur SSH fonctionne en tant que service automatiquement au démarrage de la machine. Pour vérifier si il est installé on utilise la commande `ssh -v` qui va nous retourner le nom du serveur ainsi que sa version. Les commandes utilisées pour ssh : activer : `sudo`

`systemctl start ssh,` , arrêter : `sudo systemctl stop ssh` et relance : `sudo systemctl restart ssh`

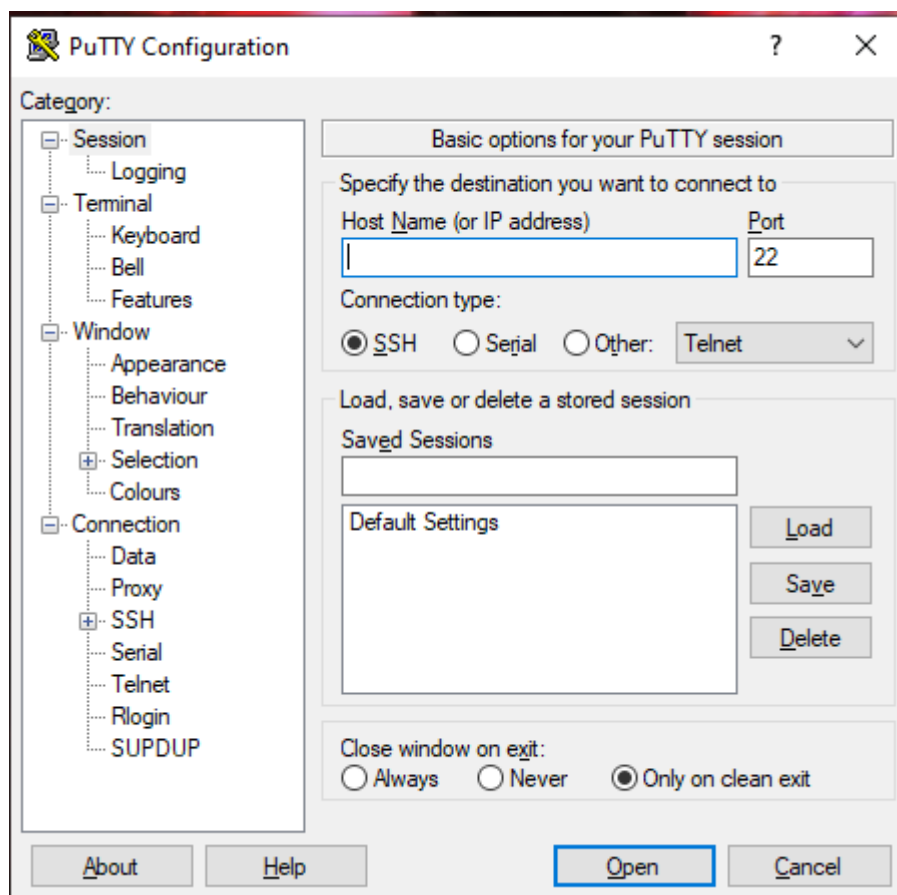
Après l'installation, on doit changer le port 22 en 2222. Pour cela, on va utiliser le terminal. On édite le fichier avec le chemin `/etc/ssh/sshd_config` et on modifie ensuite la ligne port 22 en port 2222. Quand on a fini ça on doit redémarrer le service pour cela on écrit ça dans le terminal : `/etc/init.d/sshd restart`. Après le redémarrage du service, le port a été changé.

Après la configuration de ssh, on va pouvoir installer putty qui va nous permettre d'accéder au terminal de la vm Linux.

### 3. Putty

PuTTY est un émulateur de terminal doublé d'un client pour les protocoles SSH, Telnet, rlogin, et TCP brut. Il permet également des connexions directes par liaison série RS-232.

Pour utiliser Putty, on a besoin de notre adresse ip. Pour cela, sur mon terminal je me connecte en tant qu'admin pour cela j'utilise la commande `su root` et j'y rentre mon mot de passe. Pour voir mon adresse ip il suffit d'utiliser la commande `ip address` et je récupère la deuxième.



Dans le host name, on y renseigne l'adresse ip et on choisit le port 2222.

## Job 6 - Installation du serveur Web

Maintenant qu'on a installé notre client et notre serveur pour une connexion à distance on va pouvoir installer notre serveur web.

### 1. Apache

On commence à installer Apache qui sera notre serveur HTTP Server. Pour pouvoir l'utiliser sous Linux, il nous faut l'installer par le terminal en utilisant la ligne de commande suivante :

```
apt-get install apache2
```

### 2. Php

Lorsque Apache est installé il faut ensuite installer la dernière version de PHP 7. Pour cela, on retourne dans le terminal et on y tape la commande suivante :

```
apt-get install libapache2-mod-php
```

### 3. Mariadb server

On a installé notre serveur http, notre langage PHP maintenant il nous faut un système de gestion de base de données. Dans notre cas, on utilise MariaDb pour MySQL.

Pour l'installation, on utilise le terminal Linux et on installe MariaDb avec la commande suivante :

```
apt install mariadb-server
```

Lorsque l'installation est terminée, il faut maintenant configurer pour cela, on utilise la commande suivante :

```
mysql_secure_installation
```

Plusieurs questions vont nous être posées. Pour ça j'ai suivi ce tuto complet qui explique comment répondre aux questions de config du terminal.

<https://www.digitalocean.com/community/tutorials/how-to-install-mariadb-on-debian-10>



#### 4. PhpMyAdmin

Après l'installation de notre gestionnaire de base de données. On installe une application web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB. Dans notre cas, nous allons utiliser PhpMyadmin.

Pour cette installation, il y a beaucoup d'étapes. Il faut d'abord installer la dernière version de phpmyadmin en local le unzipper grâce à une ligne de commande.

Pour la suite, la page d'apache ne s'affiche pas car il manque des extensions php qui sont présentes sur l'ordinateur (iconv et json) même après avoir utilisé les lignes de commandes apt install php-iconv et php-json. Je n'ai pas encore réussi à débloquer ça.