

TP1 - ASR2 - ENS Lyon - L3IF - 2019-2020

Intervenants:

Stella Bitchebe (celestine-stella.ndonga-bitchebe@ens-lyon.fr)

Lavoisier Wapet (patrick-lavoisier.wapet@ens-lyon.fr)

Alain Tchana (alain.tchana@ens-lyon.fr)

(Note: ce sujet a été rédigé par Stella et Lavoisier)

Partie 1: Installation d'Ubuntu 18.04 LTS

1. **Notion de distribution:** une distribution linux est un système (d'exploitation) conçu à partir du noyau Linux. Par exemple: Kali, Debian ou Ubuntu (qui est le système d'exploitation (SE) avec lequel nous allons travailler).
2. **Notion de netinstall:** pour installer ubuntu, nous devons télécharger un fichier image (extension *.iso*), qui contient tous les paquets (ensemble de fichiers) nécessaires à l'installation du SE sur notre machine. Ce fichier image est généralement très lourd (~2Go) en fonction de la version du SE et du type d'installation souhaitée (bureau ou serveur).

Cependant, il existe des images plus légères, dites *mini.iso*: on parle alors de *netinstall*. Netinstall est un type d'installation utilisant une image minimale qui va télécharger tous les paquets nécessaires au cours de l'installation. Ceci nécessite d'avoir une connexion haut débit pour permettre le téléchargement de ces paquets. L'utilisateur est donc libre de choisir les paquets qu'il souhaite installer, et de configurer le réseau à sa guise au cours de la procédure d'installation.

3. **Notion de partition:** une partie est juste une partie du disque. Lorsqu'on installe un système d'exploitation on doit définir des régions précises dont la plus importante est celle sur laquelle sera installé le système lui même (partition C pour Windows par exemple). Les partitions importantes à définir lors de l'installation d'Ubuntu sont:
 - a) Le **swap**: c'est un espace d'échange qui permet d'étendre la mémoire (RAM) de notre machine. Afin d'éviter un blocage de votre ordinateur lorsque sa RAM est pleine, Ubuntu se sert de cette partition pour décharger temporairement la RAM.

La taille du swap se calcule en fonction de celle de la RAM: si nous avons moins d'1 Go de RAM, le swap devra être entre 1,5× et 2× la taille de la RAM; si nous avons plus d'1 Go de RAM, de 1× à 1,5× la taille de la RAM. Cette partition est de type *SWAP*.

- b) La **racine (/)**: la partition racine est la base de l'arborescence du système Ubuntu. Par défaut, si aucun réglage n'est changé, c'est dans celle-ci que tous les fichiers

vont être placés : fichiers de configuration, programmes, documents personnels, etc.

Cette partition est de type *EXT4* et sa taille doit être au minimum de 8Go.

4. **Installation proprement dite** (suivre les indications sur virtualbox)

5. Une fois l'installation terminée:

a) La première chose à faire est de mettre à jour les paquets. Pour cela:

i. Ouvrir le fichier *sources.list* et décommenter certaines lignes (voir en TP):

```
stella@tpasr2:~$ sudo vi /etc/apt/sources.list
```

ii. Faire la mise à jour:

```
stella@tpasr2:~$ sudo apt update
```

iii. **APT** signifie *Advanced Package Tool*. Cette interface fonctionne avec les dépôts (distants) de base pour gérer l'installation et la suppression de logiciels sur Debian, Ubuntu et les distributions Linux associées. Pour rechercher un paquet dans la liste des paquets connus par APT, on utilise la commande *apt-cache search unMotif*.

6. **Installation d'Apache:** Apache est un serveur HTTP, i.e. il permet à un site web de communiquer avec un navigateur en utilisant le protocole HyperText Transfer Protocol (HTTP).

a) Installation d'apache2:

```
stella@tpasr2:~$ sudo apt install apache2
```

b) Vérification de l'installation:

```
stella@tpasr2:~$ apache2 -version
Server version: Apache/2.4.29 (Ubuntu)
Server built: 2019-09-16T12:58:48
```

c) Vérifions que le service apache est actif et en train de s'exécuter:

```

stella@tpasr2:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Mon 2020-02-03 16:18:16 CET; 55min ago
   Main PID: 5042 (apache2)
     Tasks: 55 (limit: 4675)
    CGroup: /system.slice/apache2.service
            └─5042 /usr/sbin/apache2 -k start
              └─5045 /usr/sbin/apache2 -k start
                └─5046 /usr/sbin/apache2 -k start

févr. 03 16:18:16 tpassr2 systemd[1]: Starting The Apache HTTP Server...
févr. 03 16:18:16 tpassr2 apachectl[5031]: AH00558: apache2: Could not reliably d
févr. 03 16:18:16 tpassr2 systemd[1]: Started The Apache HTTP Server.

```

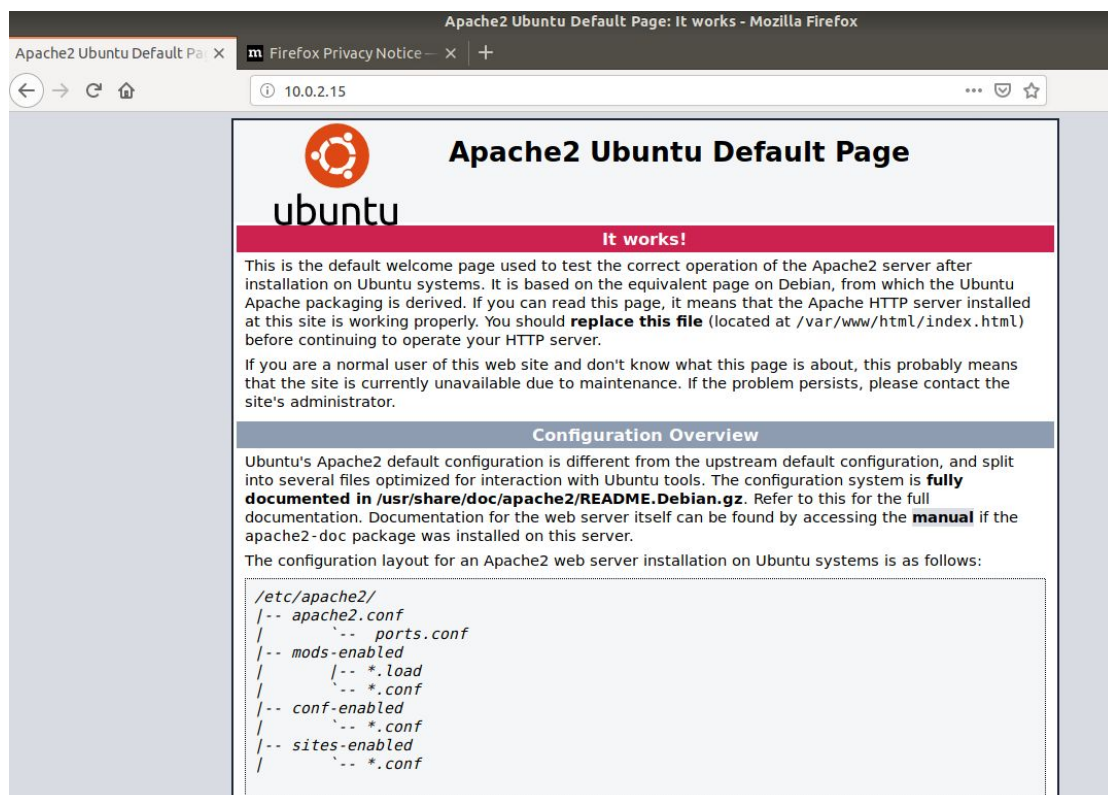
- d) Vérifions que le serveur écoute bien sur notre adresse IP. Pour connaître son adresse IP faire:

```

stella@tpasr2:~$ hostname -I
10.0.2.15

```

Ensuite depuis un navigateur rentrer cette adresse comme ceci: http://ip_address



Ensuite, essayer aussi avec localhost, et 127.0.0.1 (adresse du localhost): vous devriez obtenir le même résultat.

e) Création de notre site:

- i. Apache est installé avec une page web par défaut (voir le dossier /var/www/html/index.html: c'est le code de la page par défaut qui s'est affichée précédemment). Nous allons créer un nouveau dossier dans /var/www/html pour installer notre site:

```
stella@tpasr2:~$ sudo mkdir -p /var/www/html/tp1asr2_domain.com/
```

```
stella@tpasr2:~$ sudo chown $USER:$USER /var/www/tp1asr2_domain.com
stella@tpasr2:~$ sudo chmod -R 755 /var/www/tp1asr2_domain.com
```

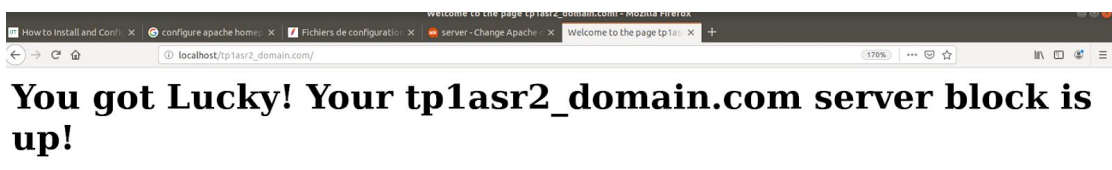
- ii. Création de la page d'accueil de notre site (index.html):

```
stella@tpasr2:~$ sudo vim /var/www/html/tp1asr2_domain.com/index.html
```

```
<html>
  <head>
    <title>Welcome to the page tp1asr2_domain.com!</title>
  </head>
  <body>
    <h1>You got Lucky! Your tp1asr2_domain.com server block is
up!</h1>
  </body>
</html>
```

- iii. Redémarrage du serveur et accès à notre site depuis le navigateur:

```
stella@tpasr2:~$ sudo systemctl restart apache2
```



Partie 2: Exploration du disque

1. Explorer les partitions avec fdisk

- a) **fdisk** est un utilitaire permettant de gérer les partitions des disques. Tout d'abord visualisons les différents disques présents sur notre machine :

```
avoisiertpvm@lavoisiertpvm-VirtualBox:~$ sudo fdisk -l
```

- b) **Création d'un nouveau disque:** nous allons le faire dans notre machine virtuelle, pour cela il est préférable de lui associer un nouveau disque virtuel via l'interface de virtualbox. Ce nouveau disque servira de cobaye pour la suite. Suivez les instructions suivantes:

- i. Arrêtez votre machine virtuelle
- ii. Allez dans le panneau de configuration de votre machine virtuelle pour y ajouter un disque virtuel de 5 GB.
- iii. Une fois le disque virtuel ajouté, démarrez votre machine virtuelle
- iv. Sélectionnez le nouveau disque créé via fdisk

```
lavoisiertpvm@lavoisiertpvm-VirtualBox:~$ sudo fdisk /dev/sdb
```

- v. Ensuite répondez par un "n" pour créer la nouvelle partition, puis par "p" pour spécifier qu'il s'agit d'une partition principale:
- vi. Faire enfin un "p" pour afficher la nouvelle partition créée puis un "q" pour quitter .

```
Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-10485759, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-10485759, default 10485759):

Created a new partition 1 of type 'Linux' and of size 5 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

- c) Formater la nouvelle partition à l'aide de la commande mkfs.ext3

```
tpvmlavoisier-VirtualBox:~$ sudo mkfs.ext3 /dev/sdb1
```

2. Cette nouvelle partition dev/sdb1 sur le disque /dev/sdb est prête à l'emploi. On peut donc la monter sur un dossier du système de fichier pour l'utiliser.

a) Créer le dossier sur lequel monter la partition

```
lavoisiertpvm@lavoisiertpvm-VirtualBox:~$ mkdir new_partition
```

b) Monter la partition sur ce dossier

```
lavoisiertpvm@lavoisiertpvm-VirtualBox:~$ sudo mount /dev/sdb1 new_partition/
```

Note: C'est une partition que l'on monte (sdb1, par exemple), **JAMAIS** un disque (sdb, par exemple) : cela ne fonctionne pas.

3. **Le fichier /etc/fstab** : le fichier /etc/fstab liste les partitions qui seront montées automatiquement au démarrage ou à la connexion du périphérique, avec toujours les mêmes options. Il est composé de plusieurs lignes décrivant chacune les conditions de montage de chaque partition (ou système de fichier). Ce fichier est créé lors de l'installation du système, avec les lignes correspondant au montage de la partition principale, et du swap. Mais on peut y ajouter manuellement des partitions supplémentaires, qui seront alors automatiquement lues par le système au démarrage.

4. **Utilisation de dd pour la sauvegarde du disque dur de la VM.**

a) Arrêtez la VM

b) Déplacez vous dans le dossier où se trouve le disque de la VM

c) sauvegardez le disque principal de la VM.

```
test_mount/virtualBoxDisks$ sudo dd if=tp_vm_principal_disk.vdi of=tp_vm_principal_disk_saved.vdi
```

d) Utilisez dd pour écrire des zéros sur les 512 premiers octets de ce disque.

```
test_mount/virtualBoxDisks$ sudo dd if=/dev/zero of=tp_vm_principal_disk.vdi bs=512 count=1
```

e) Essayez de redémarrer la VM (impossible).

f) Recouvrez le disque altéré

```
test_mount/virtualBoxDisks$ sudo dd if=tp_vm_principal_disk_saved.vdi of=tp_vm_principal_disk.vdi
```

g) Redémarrez la VM. Tout fonctionne :)

Partie 3: Compilation du noyau linux

1. **Compilation du noyau linux** : Il s'agit de compiler un nouveau noyau en ayant sous la main son code source. C'est très utile pour modifier certaines fonctionnalités du noyau linux. Nous installerons la version 5.4.1 disponible en téléchargement via le lien <https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.4.1.tar.xz>

- a. Téléchargez le code source

```
lavoisiertpvm@lavoisiertpvm-VirtualBox:~$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.4.1.tar.xz
```

- b. Décompressez l'archive téléchargée

```
lavoisiertpvm@lavoisiertpvm-VirtualBox:~$ xz -d -v linux-5.4.1.tar.xz
```

```
lavoisiertpvm@lavoisiertpvm-VirtualBox:~$ tar xvf linux-5.4.1.tar linux-5.4.1/
```

- c. Déplacez vous dans le dossier décompressé (il contient le code source du noyau)

```
lavoisiertpvm@lavoisiertpvm-VirtualBox:~$ cd linux-5.4.1/
```

- d. Importez dans ce dossier le fichier de configuration par défaut de votre noyau courant.

```
ualBox:~/linux-5.4.1$ cp -v /boot/config-$(uname -r) .config
```

- e. Installez les utilitaires de compilation

```
ox:~/linux-5.4.1$ sudo apt-get install build-essential libncurses-dev bison flex libssl-dev libelf-dev
```

- f. Réalisez la configuration si nécessaire. (Assurez vous d'élargir l'interface virtualbox de votre machine virtuelle).

```
ox:~/linux-5.4.1$ make menuconfig
```

- g. Compilez le noyau (avec éventuellement un certain nombre de processeurs, option -j). Pour accélérer la compilation, arrêtez la VM, configurez le nombre de processeurs, puis redémarrez la VM.

```
alBox:~/linux-5.4.1$ make -j 4
```

NB : Avant d'installer le nouveau noyau, s'assurer

- que le fichier `/boot/grub/grub.cfg` ne contient aucune référence à la version du noyau que nous voulons installer.
- que le dossier `/boot` ne contient ni le binaire de notre nouveau noyau, ni son système de fichier.
- de noter la version actuelle du noyau installé (commande `uname -r`).

h. Installer les modules puis le noyau compilé

```
lBox:~/linux-5.4.1$ make modules_install
```

```
Box:~/linux-5.4.1$ make install
```

i. Redémarrez la VM puis vérifier que le nouveau noyau a bien été pris en compte. (Revoir tous les points du NB plus haut).

```
ualBox:~/linux-5.4.1$ reboot
```

j. Supprimez le fichier `vmlinuz` de l'ancien noyau (Désormais, il sera impossible de booter dessus à nouveau)

```
Box:~/linux-5.4.1$ rm /boot/vmlinuz-5.0.0-23-generic
```

2. **Extraction du binaire du noyau linux:** Dézippez le kernel installé avec l'outil `extract-vmlinux` et retrouvez dans le binaire obtenu à l'aide de l'outil `objdump` une fonction contenue dans le code source du noyau, par exemple `"tcp_v4_do_rcv"`.

a. Installez l'outil `kernel-extract`

```
ox:~/linux-5.4.1$ sudo apt-get install linux-headers-$(uname -r)
```

b. Copiez le fichier `vmlinuz` du répertoire `/boot` vers le répertoire courant

```
:~/linux-5.4.1$ sudo cp /boot/vmlinuz-$(uname -r) .
```

c. Extrayez le binaire du noyau linux de l'archive copiée

```
lBox:~/linux-5.4.1$ sudo /usr/src/linux-headers-$(uname -r)/scripts/extract-vmlinux vmlinuz-$(uname -r) > vmlinux
```

d. Pour pouvoir obtenir à l'aide du binaire d'une application, le code d'une fonction donnée, il faut au préalable avoir l'adresse de cette fonction. Dans le cas du noyau linux, elle se trouve dans le fichier `/boot/System.map-xxxxx`

où xxxxx représente la version du noyau. Recherchez donc l'adresse de la fonction `tcp_v4_do_rcv` dans le fichier `/boot/System.map-xxxxx`

```
pvmlavoisier-VirtualBox:~$ sudo grep "start_thread" /boot/System.map-5.4.1
```

e. Affichez son code assembleur dans le binaire à cette adresse.

```
pvmlavoisier-VirtualBox:~$ objdump --start-address=0xffffffff8102f1f0 -D vmlinux | less
```

3. Explorer le mapping mémoire de la RAM via le fichier `/proc/iomem`

```
ox:~/linux-5.4.1$ cat /proc/iomem
```