

# TP ASR2

## Partie 1: Installation d'Ubuntu 18

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 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 système d'exploitation 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 partition 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, 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** (durant la séance)

##### 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 (bibliothèques) de base pour gérer l'installation et la suppression de logiciels sur Debian, Ubuntu et les distributions Linux associées. APT simplifie le processus de gestion des logiciels sur les systèmes informatiques de type Unix en automatisant la récupération, la configuration et l'installation des progiciels, à partir de fichiers précompilés ou

en compilant le code source.

Pour rechercher un paquet dans la liste des paquets connus par APT, on utilise la commande *apt-cache search*.

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) Configuration du firewall: afin de configurer Apache, nous devons d'abord autoriser l'accès extérieur à certains ports Web de notre système et autoriser Apache sur notre pare-feu UFW (Uncomplicated FireWall).

i. Lister les profils d'application ufw:

```
stella@tpasr2:~$ sudo ufw app list
[sudo] Mot de passe de stella :
Applications disponibles :
  Apache
  Apache Full
  Apache Secure
  CUPS
```

ii. Autoriser apache sur le pare-feu et vérifier le status:

```
stella@tpasr2:~$ sudo ufw allow 'Apache'
Les règles ont été mises à jour
Les règles ont été mises à jour (IPv6)
```

```

stella@tpasr2:~$ sudo ufw status
État : actif

Vers          Action      De
----          -
Apache        ALLOW       Anywhere
Apache (v6)    ALLOW       Anywhere (v6)

```

Si le résultat de *status* vous renvoie *Etat: Inactif*, alors tapez ceci:

```

stella@tpasr2:~$ sudo ufw enable
Le pare-feu est actif et lancé au démarrage du système

```

d) Configuration du serveur Apache:

i. 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 tpasr2 systemd[1]: Starting The Apache HTTP Server...
févr. 03 16:18:16 tpasr2 apachectl[5031]: AH00558: apache2: Could not reliably d
févr. 03 16:18:16 tpasr2 systemd[1]: Started The Apache HTTP Server.

```

ii. Vérifions que le server écoute bien sur notre adresse IP. Pour connaître son adresse IP faire:

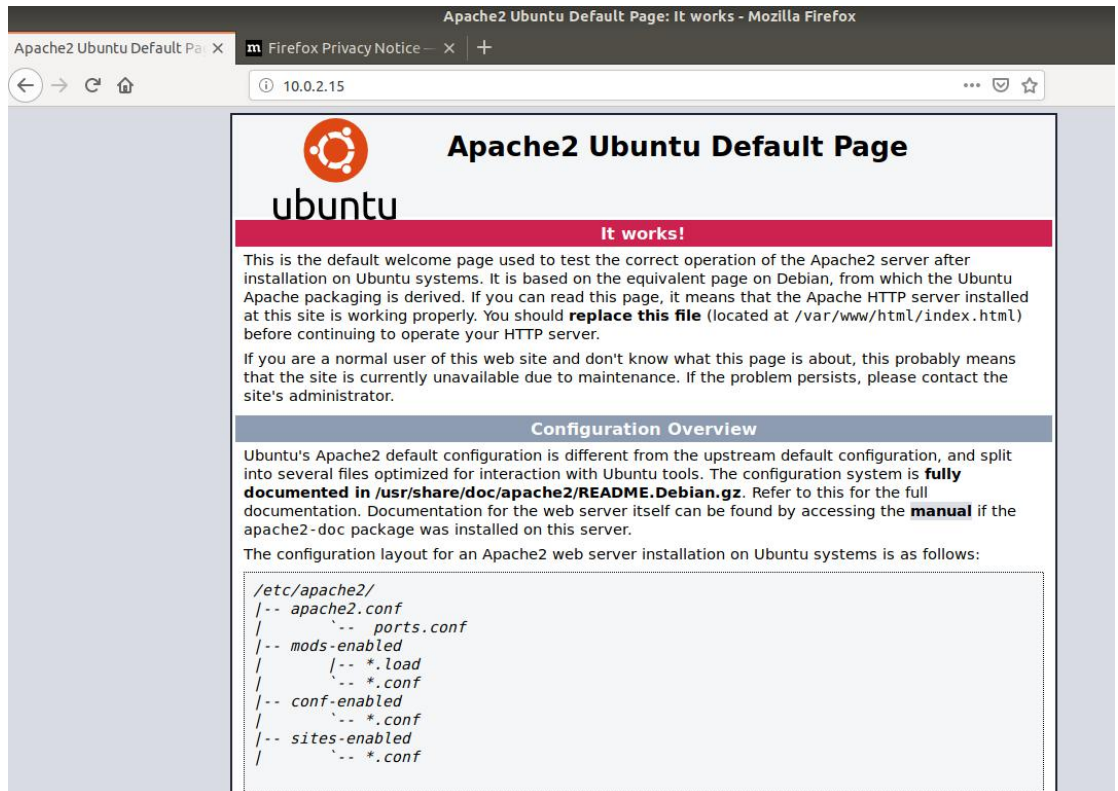
```

stella@tpasr2:~$ hostname -I
10.0.2.15

```

Ensuite depuis un navigateur rentrer cette adresse comme ceci:

[http://ip\\_address](http://ip_address)



e) Création de notre site:

- i. Installation d'un nom de domaine: Apache est installé avec un site 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 domaine:

```
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 disques. Tout d'abord visualiser les différent périphériques de stockage présents (ce sont les devices):

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

b/ Les fonctionnalités de fdisk s'appliquent sur un disque en mode interactif, par exemple créons une partition sur un disque ( par exemple /dev/sdb).

NB: Nous pouvons le faire dans notre machine virtuelle , pour ce faire il est préférable de lui associer un nouveau disque virtuel via l'interface de virtualbox. Ce nouveau disque servira de cobaye pour la suite. Suivre 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, démarrer votre machine
- iv/ sélectionner le nouveau disque créé via fdisk

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



iii/ Ensuite répondre par un n pour créer la nouvelle partition, puis par e pour spécifier qu'il s'agit d'une partition étendue:

```
Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): e
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 'Extended' and of size 5 GiB.
```

iv/ faire enfin un p pour afficher la nouvelle partition créée puis un q pour quitter .

```
Command (m for help): p
Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xb5b222cb

Device      Boot Start      End  Sectors  Size Id Type
help /dev/sdb1      2048 10485759 10483712    5G  5 Extended
```

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

```
lavoisierpvm@lavoisierpvm-VirtualBox:~$ sudo mkfs.ext3 /dev/sdb
```

d) cette nouvelle partition 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.

i) créer le dossier sur lequel monter la partition

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

ii) monter la partition sur ce dossier

```
lavoisierpvm@lavoisierpvm-VirtualBox:~$ sudo mount /dev/sdb new_partition/
```

2- Explications /etc/fstab : Il s'agit d'un fichier contenant tous les montages nécessaire au bon fonctionnement du système. Il est lu à chaque que fois

que la commande mount est utilisé par exemple lors du démarrage.  
D'autres informations comme le format se diront séance tenante.

3 - Utilisation de dd pour la sauvegarde du disque dur de la vm.

i/ Arrêter la VM

ii/ se déplacer dans le dossier où se trouve le disque de la vm (il est préférable d'utiliser le disque virtuel créé plus haut)

iii/ sauvegarder le disque de la vm

```
/tp_ens_système_reseaux$ dd if=disk_for_fdisk_manipulations.vdi of=disk_for_fdisk_manipulations_saved.vdi
```

iv/ Utiliser dd pour écrire des zéros sur ce disque

```
:p_ens_système_reseaux$ sudo dd if=/dev/zero of=disk_for_fdisk_manipulations.vdi bs=1G count=5
```

v/ Essayer de redémarrer la VM (Vous devez avoir une erreur liée à ce disque)

vi) Recouvrer le disque altéré

```
/tp_ens_système_reseaux$ sudo dd if=disk_for_fdisk_manipulations_saved.vdi of=disk_for_fdisk_manipulations.vdi
```

vii/ Redémarrer 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 le code source. C'est très utile pour modifier certaines fonctionnalités du noyau linux. Nous installeront 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>



## i / Télécharger le code source

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

## ii/ décompresser 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/
```

## iii/ se déplacer dans le dossier décompressé qui contient le code source de linux

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

## iv/importer le fichier de configuration par défaut de votre noyau courant

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

## v/ installer les utilitaires de compilation

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

## vi) réaliser la configuration

```
lavoisiertpvm@lavoisiertpvm-VirtualBox:~/linux-5.4.1$ make menuconfig
```

## vii) compiler le noyau (avec éventuellement un certain nombre de processeurs, option -j). Pour accélérer la compilation, arrêter la VM et configurer le nombre de processeurs.

```
lavoisiertpvm@lavoisiertpvm-VirtualBox:~/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.

viii) Installer les modules puis le noyau compilé

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

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

mettre à jour le grub pour qu'il prenne en compte le nouveau noyau lors du démarrage de la machine virtuelle

```
Box:~/linux-5.4.1$ sudo update-initramfs -c -k 5.4.1
```

```
ualBox:~/linux-5.4.1$ sudo update-grub
```

ix) redémarrer 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
```

x) supprimer le 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. Dézipper le kernel installé avec l'outil `extract-vmlinux` et retrouver dans le binaire obtenu à l'aide de l'outil `objdump` une fonction contenue dans le code source du noyau, par exemple "start\_readonly".

i/ installation de l'outil `kernel-extract`

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

ii/ copier le fichier `vmlinux` du répertoire `/boot` vers le répertoire courant

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

iii/ Extraire le binaire du noyau linux

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

iv) rechercher l'adresse de la fonction `start_readonly` dans le fichier `/boot/System.map-xxxxx`

```
~/linux-5.4.1$ sudo grep "start_readonly" /boot/System.map-$(uname -r)
```

v) Afficher son code assembleur dans le binaire à cette adresse.

```
~/linux-5.4.1$ objdump -D --start-address=0xffffffff82d3dd70 vmlinux | less
```

3. Explorer le mapping mémoire système-périphériques dans le fichier `/proc/iomem`

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