Compte Rendu Tp 4 / Administration Réseaux Informatiques

1 - Pour lister la liste des interface réseaux actives , on utilise la commande :

- ifconfig :

```
sabriserver@sabriserver:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe77:38a0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:77:38:a0  txqueuelen 1000  (Ethernet)
        RX packets 74810  bytes 108491362 (108.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9180  bytes 583508 (583.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 118  bytes 10710 (10.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 118  bytes 10710 (10.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Pour lister aussi les interfaces réseaux désactivés , on utilise la commande :

- ifconfig -a

```
sabriserver@sabriserver:~$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe77:38a0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:77:38:a0  txqueuelen 1000  (Ethernet)
        RX packets 74812  bytes 108491482 (108.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9186  bytes 583988 (583.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 118  bytes 10710 (10.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 118  bytes 10710 (10.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2 - Pou savoir le mode d’adressage sur notre machine en s’adresse au fichier 00-installer-config.yaml , On utilisant la commande :

- sudo vim /etc/netplan/00-installer-config.yaml

```
sabriserver@sabriserver:~$ sudo vim /etc/netplan/00-installer-config.yaml
```

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
            dhcp4 : true
            dhcp6 : true
            #addresses:
            #- 192.168.1.2/24
            #gateway4: 192.168.1.1
            #nameservers:
            #addresses:
            #- 192.168.1.0
            #- 192.168.1.1
            #search: []
  version: 2
```

```
 enp0s3:
         dhcp4 : true
         dhcp6 : true
```

- Donc Le mode D'adressage est Automatique .

3 - En se basant sur la question précédente , l'adressage ip de ma machine est configure manuellement .

4 - Pour sauvegarder le fichier de configuration réseau on utilise la commande .

- $sudo netplan apply , Pour Appliques les nouveaux changements .

```
 sabriserver@sabriserver:~$ sudo netplan apply
```

5 - Apres avoir faire une copie du fichier de configuration , on applique la nouvelle configuration :

```
sabriserver@sabriserver:~$ sudo vim /etc/netplan/00-installer-config.yaml _
```

```
ethernets:
  enp0s3:
          # dhcp4 : true
          # dhcp6 : true
          addresses:
          - 192.168.1.2/24
          gateway4: 192.168.1.1
          nameservers:
          addresses:
          - 192.168.1.1
          - 8.8.8.8
          search: []
version: 2_
```

- On Doit enregistrer les changement :

```
:wq!
```

```
"/etc/netplan/00-installer-config.yaml" 15L, 358C written
```

6 -

- Verifiont si il ya des erreurs dans la nouvelle configuration :

  - $ sudo netplan –d apply

```
configuration accepted_
```

- On doit appliquer les nouvelles changement on utilisant la commande :

```
sabriserver@sabriserver:~$ sudo netplan apply
```

-

- On doit relancer le service on utilisant la commande suivantes :

  - systemctl restart systemd-networkd :

```
sabriserver@sabriserver:~$ sudo systemctl restart systemd-networkd
[sudo] password for sabriserver:
```

7 - Oui Il Exesiste une autres methode qui est :

- Tapte la commande :
  - netplan apply

8 -

- route -n :

```
sabriserver@sabriserver:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
10.0.2.2        0.0.0.0         255.255.255.255 UH    100    0        0 enp0s3
sabriserver@sabriserver:~$ _
```

- Cette Comamnde Affiche les diferrentes routes vers les differents reseaux .

- netstat -nr :

```
sabriserver@sabriserver:~$ sudo netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.2.2        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
10.0.2.2        0.0.0.0         255.255.255.255 UH        0 0          0 enp0s3
sabriserver@sabriserver:~$ _
```

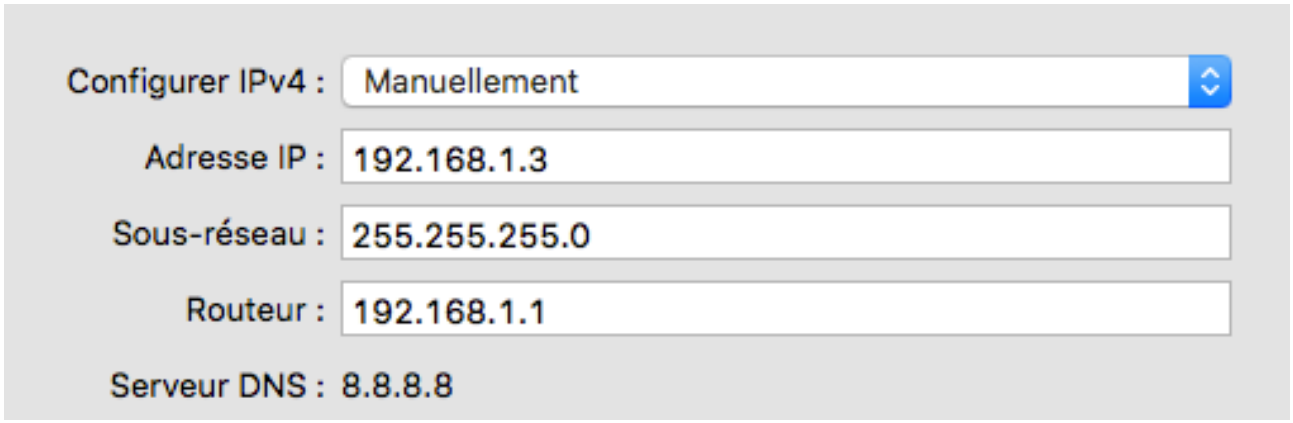
- **netstat** , pour « network statistics », **est** une ligne de commande affichant des informations sur les connexions réseau, les tables de routage et un certain nombre de statistiques dont ceux des interfaces

9 - On tape la commande :

- sudo vim /etc/hosts .

```
127.0.0.1 localhost
127.0.1.1 sabriserver
192.168.1.3 clientphysique

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```



10 -

```
sabriserver@sabriserver:~$ ping 192.168.8.102
PING 192.168.8.102 (192.168.8.102) 56(84) bytes of data.
64 bytes from 192.168.8.102: icmp_seq=1 ttl=64 time=0.987 ms
64 bytes from 192.168.8.102: icmp_seq=2 ttl=64 time=0.409 ms
64 bytes from 192.168.8.102: icmp_seq=3 ttl=64 time=0.410 ms
64 bytes from 192.168.8.102: icmp_seq=4 ttl=64 time=0.396 ms
64 bytes from 192.168.8.102: icmp_seq=5 ttl=64 time=0.404 ms
64 bytes from 192.168.8.102: icmp_seq=6 ttl=64 time=0.539 ms
64 bytes from 192.168.8.102: icmp_seq=7 ttl=64 time=0.406 ms
64 bytes from 192.168.8.102: icmp_seq=8 ttl=64 time=0.409 ms
64 bytes from 192.168.8.102: icmp_seq=9 ttl=64 time=0.401 ms
64 bytes from 192.168.8.102: icmp_seq=10 ttl=64 time=0.410 ms
64 bytes from 192.168.8.102: icmp_seq=11 ttl=64 time=0.321 ms
64 bytes from 192.168.8.102: icmp_seq=12 ttl=64 time=0.214 ms
64 bytes from 192.168.8.102: icmp_seq=13 ttl=64 time=0.508 ms
64 bytes from 192.168.8.102: icmp_seq=14 ttl=64 time=0.384 ms
64 bytes from 192.168.8.102: icmp_seq=15 ttl=64 time=0.418 ms
64 bytes from 192.168.8.102: icmp_seq=16 ttl=64 time=0.357 ms
64 bytes from 192.168.8.102: icmp_seq=17 ttl=64 time=0.255 ms
64 bytes from 192.168.8.102: icmp_seq=18 ttl=64 time=0.347 ms
64 bytes from 192.168.8.102: icmp_seq=19 ttl=64 time=0.234 ms
```

```
sabriserver@sabriserver:~$ ping client-physique
PING client-physique (192.168.8.102) 56(84) bytes of data.
64 bytes from client-physique (192.168.8.102): icmp_seq=1 ttl=64 time=0.249 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=2 ttl=64 time=0.283 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=3 ttl=64 time=0.549 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=4 ttl=64 time=0.401 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=5 ttl=64 time=0.332 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=6 ttl=64 time=0.302 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=7 ttl=64 time=0.301 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=8 ttl=64 time=0.405 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=9 ttl=64 time=0.740 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=10 ttl=64 time=0.325 ms
64 bytes from client-physique (192.168.8.102): icmp_seq=11 ttl=64 time=0.194 ms
-
```

Partie 2 :

- 1 :

```
sabriserver@sabriserver:~$ sudo apt-cache search -n nfs
libnfs-dev - NFS client library (development files)
libnfs13 - NFS client library (shared library)
libnfsidmap-dev - header files and docs for libnfsidmap
libnfsidmap2 - NFS idmapping library
nfs-common - NFS support files common to client and server
nfs-ganesha - NFS server in User Space
nfs-ganesha-ceph - nfs-ganesha fsal ceph libraries
nfs-ganesha-doc - Documentation for nfs-ganesha
nfs-kernel-server - support for NFS kernel server
argonaut-fai-nfsroot - Argonaut (tools, queues and status management)
fai-nfsroot - Fully Automatic Installation nfsroot package
libfile-nfslock-perl - perl module to do NFS (or not) locking
libnfs-utils - NFS client library (binaries)
libnfsidmap-regex - Plugin to map regex NFSv4 names to and from ids.
libyanfs-java - Yet Another NFS - a Java NFS library
nfs-ganesha-gluster - nfs-ganesha fsal gluster libraries
nfs-ganesha-gpfs - nfs-ganesha fsal gpfs libraries
nfs-ganesha-mem - nfs-ganesha fsal mem libraries
nfs-ganesha-mount-9p - nfs-ganesha mount.9P
nfs-ganesha-nullfs - nfs-ganesha fsal nullfs libraries
nfs-ganesha-proxy - nfs-ganesha fsal proxy libraries
nfs-ganesha-vfs - nfs-ganesha fsal vfs libraries
nfs-ganesha-xfs - nfs-ganesha fsal xfs libraries
nfs4-acl-tools - Commandline and GUI ACL utilities for the NFSv4 client
nfstrace - NFS tracing/monitoring/capturing/analyzing tool
nfstrace-doc - NFS tracing/monitoring/capturing/analyzing tool (documentation)
nfswatch - Program to monitor NFS traffic for the console
python3-nfs-ganesha - Python 3 bindings for nfs-ganesha
unionfs-fuse - Fuse implementation of unionfs
sabriserver@sabriserver:~$ _
```

12 -

```
sabriserver@sabriserver:~$ sudo apt-get install nfs-kernel-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
nfs-kernel-server is already the newest version (1:1.3.4-2.5ubuntu3.4).
0 upgraded, 0 newly installed, 0 to remove and 96 not upgraded.
sabriserver@sabriserver:~$
```

13 -

```
sabriserver@sabriserver:~$ su root
Password:
root@sabriserver:/home/sabriserver# cd
root@sabriserver:~# _
```

- Pour redemarrer le server on utilise la commande :

  - sudo systemctl start nfs-kernel-server

```
root@sabriserver:~# sudo systemctl start nfs-kernel-server
root@sabriserver:~# _
```

14 -

```
root@sabriserver:~/nfs# mkdir etudiant
root@sabriserver:~/nfs# cd etudiant
root@sabriserver:~/nfs/etudiant# mkdir public
root@sabriserver:~/nfs/etudiant# mkdir person
root@sabriserver:~/nfs/etudiant# _
```

- Pour atrribuer le repertoire au utilisateur etud101 on utilise la commande :

  - sudo chown -R etud101 nfs/etudiant

```
root@sabriserver:~# sudo chown -R etud101: nfs/etudiant
root@sabriserver:~#
```

15 -

-

```
root@sabriserver:~/nfs# mkdir etudiant
root@sabriserver:~/nfs# cd etudiant
root@sabriserver:~/nfs/etudiant# mkdir public
root@sabriserver:~/nfs/etudiant# mkdir person
root@sabriserver:~/nfs/etudiant# _
```

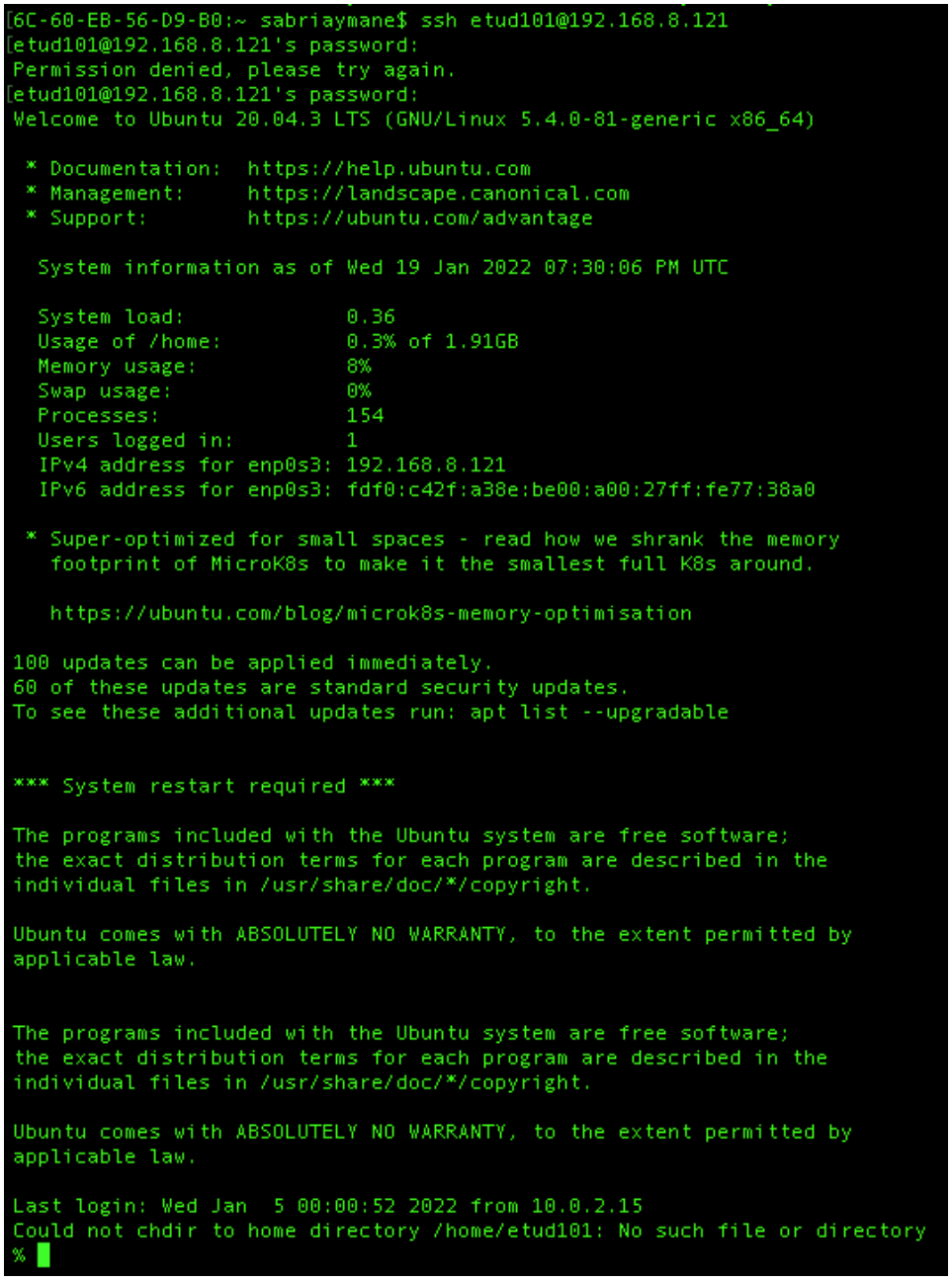
- Le Fichier /etc/exports :

```
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/root/nfs/etudiant/public *(ro)
/root/nfs/etudiant/perso 192.168.8.102(rw)
```

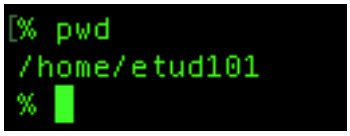
- Pour la redemaration :

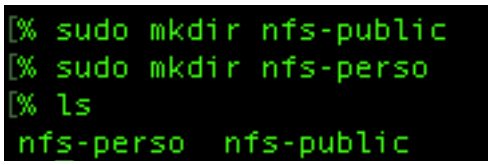
```
"/etc/exports" 13L, 465C written
root@sabriserver:~# sudo /etc/init.d/nfs-kernel-server reload
Reloading nfs-kernel-server configuration (via systemctl): nfs-kernel-server.service.
```

- On Se Connect via SSH du notre machine a notre serveur d'adress 192.168.8.121 :



- On Cree les deux repertoires : nfs-public et nfs-perso dans le repertoire /home/etud101 :





16 -

```
"/etc/hosts" 10L, 257C written
root@sabriserver:~# mount -t nfs 192.168.8.120:/root/nfs/etudiant/public /home/etud101/nfs-public -o
 ro
mount.nfs: Connection timed out
```

```
/swap.img       none    swap    sw      0       0
192.168.8.120:/root/nfs/etudiant/public media/NFS nfs defaults,user,auto,_netdev,bg00
```

17 - Pardon Professeur j'ai pas pu résoudre cette question a cause des problèmes de nfs

18 -

- Le Port 80 Correspond au trafic-web normal , non crypte

- Le Port 443 correspond au trafic-crypte TSL/SSL

RM : le serveur n'accepter que des requêtes que sur les ports spécifiés

19 -

- Installation du Apache 2 :

```
root@sabriserver:~# sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following packages will be upgraded:
  apache2 apache2-bin apache2-data apache2-utils
4 upgraded, 0 newly installed, 0 to remove and 91 not upgraded.
Need to get 0 B/1,518 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
(Reading database ... 98690 files and directories currently installed.)
Preparing to unpack .../apache2_2.4.41-4ubuntu3.9_amd64.deb ...
Unpacking apache2 (2.4.41-4ubuntu3.9) over (2.4.41-4ubuntu3.8) ...
Preparing to unpack .../apache2-bin_2.4.41-4ubuntu3.9_amd64.deb ...
Unpacking apache2-bin (2.4.41-4ubuntu3.9) over (2.4.41-4ubuntu3.8) ...
Preparing to unpack .../apache2-data_2.4.41-4ubuntu3.9_all.deb ...
Unpacking apache2-data (2.4.41-4ubuntu3.9) over (2.4.41-4ubuntu3.8) ...
```

- Verifions si le service apache2 est activer :

```
root@sabriserver:~# sudo systemctl is-enabled apache2.service
enabled
root@sabriserver:~# _
```

- Configurant le parfeu <ufw>

  - Pour afficher le status on execute la commande :

    - ufw status

```
root@sabriserver:~# sudo ufw status
Status: inactive
```

  - Nous devons activer le parfeu ufw et activer aussi le apache avec les commande :

    - sudo ufw enable

```
Firewall is active and enabled on system startup
root@sabriserver:~#
```

```
root@sabriserver:~# ufw status
Status: active
```

    - sudo ufw allow ‘Apache’

```
root@sabriserver:~# sudo ufw allow 'Apache'
Rule added
Rule added (v6)
```

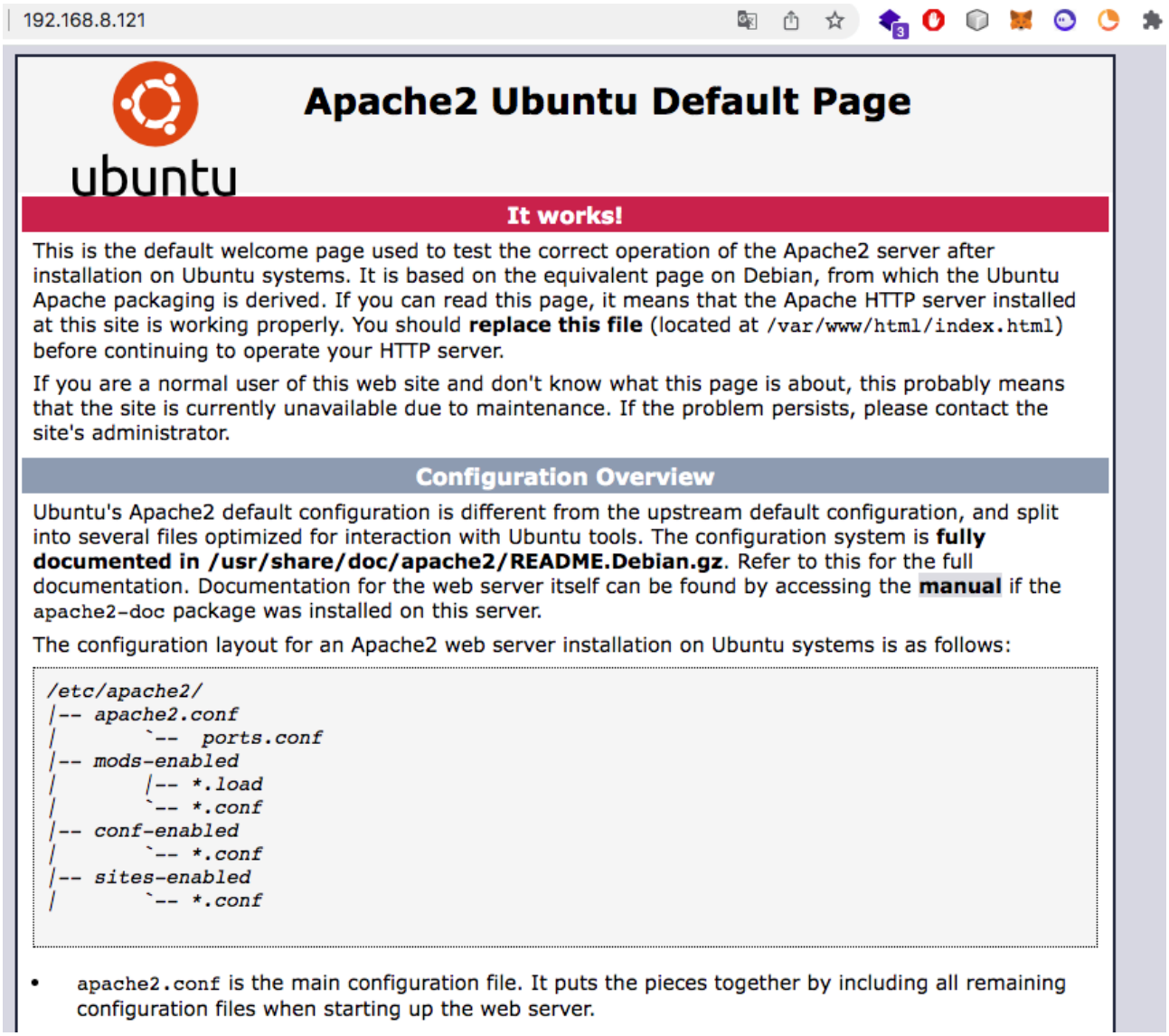
```
root@sabriserver:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)
```

```
root@sabriserver:~# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-01-19 21:52:17 UTC; 17min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 16491 (apache2)
      Tasks: 55 (limit: 5676)
     Memory: 4.9M
     CGroup: /system.slice/apache2.service
             ├─16491 /usr/sbin/apache2 -k start
             ├─16493 /usr/sbin/apache2 -k start
             └─16494 /usr/sbin/apache2 -k start

Jan 19 21:52:17 sabriserver systemd[1]: Starting The Apache HTTP Server...
Jan 19 21:52:17 sabriserver apachectl[16478]: AH00558: apache2: Could not reliably determine the
Jan 19 21:52:17 sabriserver systemd[1]: Started The Apache HTTP Server.
lines 1-15/15 (END)
```

- Nous Testons :



- Maintenant nous allons cree le domaine estfbs :

-

```
root@sabriserver:~# sudo mkdir /var/www/estfbs
root@sabriserver:~# _
```

```
root@sabriserver:~# sudo chown -R $USER:$USER /var/www/estfbs
root@sabriserver:~#
```

```
root@sabriserver:~# sudo chmod -R 755 /var/www/estfbs
root@sabriserver:~#
```

```
root@sabriserver:~# sudo mkdir /var/www/estfbs
root@sabriserver:~# _
```

```
root@sabriserver:~# sudo chown -R $USER:$USER /var/www/estfbs
root@sabriserver:~#
```

```
root@sabriserver:~# sudo chmod -R 755 /var/www/estfbs
root@sabriserver:~#
```

```
 GNU nano 4.8                        /var/www/estfbs/index.html
<html>
<head>
<title> Welcome To Est Fbs </title>
</head>
<body>Merci Pour Votre Attention Professeur RaChid Ait-Daoud  </body>
</html>
```

```
sabriserver@sabriserver:~$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-
available/estfbs.conf
sabriserver@sabriserver:~$ _
```

```
 ServerAdmin webmaster@localhost
 ServerName estfbs
 ServerAlias www.estfbs
 DocumentRoot /var/www/estfbs


 # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
 # error, crit, alert, emerg.
 # It is also possible to configure the loglevel for particular
 # modules, e.g.
 #LogLevel info ssl:warn

 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
sabriserver@sabriserver:~$ sudo a2ensite estfbs.conf
Enabling site estfbs.
To activate the new configuration, you need to run:
  systemctl reload apache2
sabriserver@sabriserver:~$ systemctl reload apache2
```

```
sabriserver@sabriserver:~$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to reload 'apache2.service'.
Multiple identities can be used for authentication:
 1.  Aymane Sabri (sabriserver)
 2.  Etudiant 1er annee genie Informatique  (etud101)
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ===
sabriserver@sabriserver:~$
```

```
sabriserver@sabriserver:~$ ls -l /etc/apache2/sites-enabled/
total 0
lrwxrwxrwx 1 root root 35 Jan  4 23:00 000-default.conf -> ../sites-available/000-default.conf
lrwxrwxrwx 1 root root 30 Jan 19 23:19 estfbs.conf -> ../sites-available/estfbs.conf
sabriserver@sabriserver:~$ _
```

```
sabriserver@sabriserver:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
sabriserver@sabriserver:~$ _
```

```
Syntax OK
```

```
Success! The your domain virtual host is working!
```

- Merci Pour Votre Attention / Aymane Sabri / Groupe 2