



# SERVEUR LINUX

## INSTALLATION ET CONFIGURATION

2<sup>ÈME</sup> BACHELIER EN INFORMATIQUE

---

# Projet Linux

---

*Auteur :*

Terencio AGOZZINO

*Auteur :*

Alexandre DUCOBU

*Enseignants :*

Antoine MALAISE

Julien DE BODT



Campus  
technique

Année académique 2016 - 2017





# SERVEUR LINUX

## INSTALLATION ET CONFIGURATION

2<sup>ÈME</sup> BACHELIER EN INFORMATIQUE

---

# Projet Linux

---

*Auteur :*

Terencio AGOZZINO

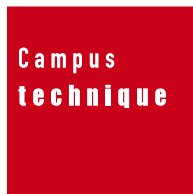
*Auteur :*

Alexandre DUCOBU

*Enseignants :*

Antoine MALAISE

Julien DE BODT



Année académique 2016 - 2017



Ce document est mis à disposition selon les termes de la licence Creative Commons  
“Attribution - Pas d’utilisation commerciale 4.0 International”.



# Table des matières

<b>1</b>	<b>Présentation générale du projet</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	Déontologie . . . . .	3
1.3	Sécurité . . . . .	3
<b>2</b>	<b>Choix</b>	<b>4</b>
2.1	Distribution . . . . .	4
2.2	Langue . . . . .	6
2.3	Noyau . . . . .	6
2.4	Partitionnement . . . . .	6
2.5	Sauvegardes . . . . .	7
2.6	Sécurité des mots de passe . . . . .	8
<b>3</b>	<b>Services</b>	<b>9</b>
3.1	Base de données . . . . .	9
3.2	DNS . . . . .	10
3.3	FTP . . . . .	12
3.4	iptables . . . . .	14
3.5	NFS . . . . .	18
3.6	NTP . . . . .	20
3.7	Quotas . . . . .	23
3.8	Samba . . . . .	24
3.9	SELinux . . . . .	26
3.10	Serveur Web . . . . .	27
3.11	SSH . . . . .	29
3.12	Utilisateurs . . . . .	31
<b>4</b>	<b>Conclusion du projet</b>	<b>33</b>
	<b>Références</b>	<b>0</b>

# 1 Présentation générale du projet

## 1.1 Introduction

Dans le cadre de ce projet, il nous a été demandé d'administrer un serveur sous Linux.

Le choix de la distribution ainsi que la gestion des sauvegardes est libre et devra être justifié.

Le serveur devra contenir :

- un partage NFS qui permettra aux utilisateurs du réseaux local d'y stocker des fichiers ;
- un partage Samba qui permettra aux utilisateurs de Windows d'accéder à ce même partage ;
- un serveur Web, FTP, MySQL et DNS qui permettra un hébergement multi-utilisateurs ;
  - le serveur FTP permettra à chaque utilisateur d'accéder à son dossier Web ;
  - le serveur DNS contiendra une zone qui sera indispensable pour les sites Web de l'utilisateur ;
  - le serveur DNS fera également office de DNS cache pour le réseau local.
- un serveur NTP pour que les machines du réseau local puissent se synchroniser ;
- le support du module SSH.

## 1.2 Déontologie

En tant qu'administrateurs du serveur, nous serons tenus de suivre de nombreuses règles telles que :

- la documentation des actions entreprises sur le serveur ;
- l'automatisation des installations et configurations au travers de scripts ;
- la sécurité : mise en place de mots de passe forts, du SSH, etc. ;
- la vigilance et la prévoyance, par exemple par la mise en place de sauvegardes avant et après chaque changement sur le serveur ;
- le contrôle du bon fonctionnement de chaque élément.

## 1.3 Sécurité

Du côté de la sécurité, quelques contraintes seront prises en compte :

- mise en place d'une politique utilisateur ;
- mise en place de quotas ;
- partitionnement et gestion du disque (LVM et RAID) ;
- mise en place d'une stratégie de sauvegarde ;
- désactivation des éléments inutiles et des mises à jours ;
- mise en place d'un antivirus, d'un firewall, etc.



## 2 Choix

### 2.1 Distribution

Le choix de la distribution s'est naturellement porté sur Debian, pour ses nombreux avantages. En voici quelques exemples :

- Large communauté : grâce à cela, les erreurs et problèmes rencontrés ont souvent plusieurs solutions connues et éprouvées.
- Plusieurs architectures et noyaux : Debian supporte la majorité des architectures de processeurs comme AMD, Armel, i386, MIPS, etc. Elle supporte aussi de nombreux noyaux tels que FreeBSD et GNU Hurd.
- Sécurité : vu que la distribution est open-source, cela signifie que les *backdoors* sont presque inexistantes. De surcroît, lorsqu'une faille de sécurité est détectée, celle-ci est rapidement corrigée par la communauté.

En outre, Debian comprend de nombreux logiciels de sécurité tels que GPG (et PGP), SSH et autres.

- Stabilité : les serveurs doivent avoir le plus grand temps d'accessibilité ( $\approx 99.999\%$ ). Sous Debian, il existe de nombreux exemples de machines qui tournent sans arrêt pendant des années, mis à part lors de pannes ou de mises à jour liées au matériel.
- Système de paquets : grâce au système de paquets, les distributions Linux ont la possibilité d'installer de nombreux logiciels par une seule ligne de commande.

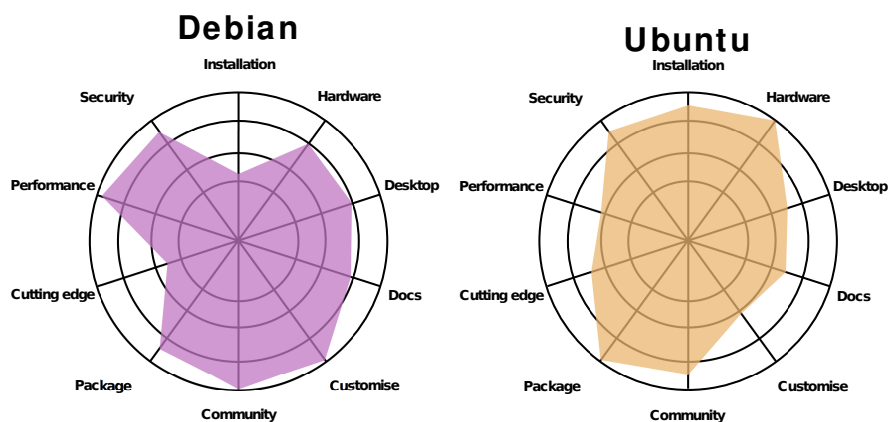


FIGURE 1 – Différences entre Debian et Ubuntu

De même, la distribution Debian est plus professionnelle et celle-ci possède le *leadership* depuis des années.

Remarque : depuis mai 2016, Ubuntu a les mêmes parts de marché que Debian.

La distribution Ubuntu n'a pas été choisie pour les raisons suivantes :

- Dérivée de Debian : de ce fait, un administrateur sachant configurer un serveur sous Debian pourra facilement s'adapter aux serveurs sous Ubuntu.
- Vise le grand public. Par conséquent, est beaucoup moins utilisée dans le milieu professionnel.
- Assez récente sur le marché du serveur.
- Moins performante que Debian.

Concernant les autres distributions, CentOS est en baisse, mais reste au-dessus de Red Hat et de Fedora qui sont en chute libre.

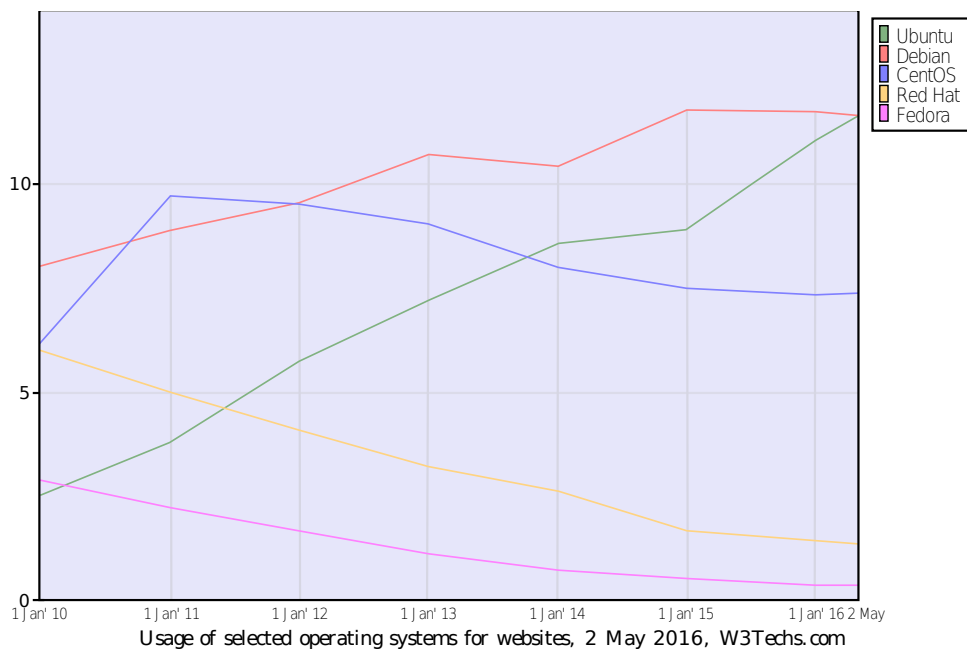


FIGURE 2 – Parts de marché des distributions Linux

## 2.2 Langue

Pour le choix de la langue lors de l'installation, il a été préféré d'utiliser l'anglais vu que la majorité des documentations et forums sont disponibles dans cette langue. De plus, cela permet d'éviter une mauvaise traduction concernant les nouvelles mises à jour et de toucher un public le plus large possible.

## 2.3 Noyau

Un noyau (monolithique<sup>1</sup>) modulaire a été choisi afin de gérer les modules. En effet, celui-ci facilite l'ajout et la suppression de modules à chaud. Ces modules, pas toujours indispensables, peuvent être la source de failles et de bugs.

## 2.4 Partitionnement

Le partitionnement du disque a été réalisé avec une partition racine, */boot*, et un groupe de volume LVM nommé *VolGroup*. De surcroît, deux disques de 20 Go en RAID 1 ont été utilisés.

Label	Type	Taille (Mo)	Format
<i>/boot</i>	primary	512	ext4
VolGroup	logical	20958	lvm

LVsrv ( <i>/srv</i> )	lvm	6144	ext4
LVswap ( <i>/swap</i> )	lvm	4096	swap
LVhome ( <i>/home</i> )	lvm	2048	ext4
LVroot ( <i>/root</i> )	lvm	2048	ext4
LVusr ( <i>/usr</i> )	lvm	2048	ext4
LVopt ( <i>/opt</i> )	lvm	2048	ext4
LVvar ( <i>/var</i> )	lvm	1024	ext4
LVtmp ( <i>/tmp</i> )	lvm	1024	ext4

TABLE 1 – Tableau du partitionnement (*avec LVM*)

---

1. Les fonctions du système et pilotes sont regroupés dans le kernel space généré à la compilation.

## 2.5 Sauvegardes

Dans le milieu de l'entreprise, deux types de sauvegarde sont principalement utilisés : incrémentielle<sup>2</sup> et différentielle<sup>3</sup>.

Afin de trouver un compromis, ces deux types de sauvegardes ont été utilisés :

1. différentielle : afin de restaurer les données plus rapidement par rapport à la sauvegarde incrémentielle ;
2. incrémentielle : pour une rapidité de sauvegarde et un stockage en mémoire plus économe que la sauvegarde différentielle.

Dans le but d'éviter de perturber l'accès au serveur, tous les jours à deux heures du matin, une sauvegarde incrémentielle est lancée à l'aide d'un *cron*<sup>4</sup>, dans le but d'enregistrer les données modifiées et créées en cette journée.

```
1 crontab -e 0 2 * * * /usr/bin/backup-make.sh -i
```

Quant à la sauvegarde différentielle, celle-ci s'opère uniquement le dimanche à 2 heures du matin. Le dimanche étant un jour de congé pour la majorité du monde, ce qui aura un impact mineur sur les performances du serveur.

```
1 crontab -e 0 2 * * 0 /usr/bin/backup-make.sh -d
```

Remarque : à défaut d'utiliser *fcron* n'étant plus disponible sur Debian, le serveur doit être alimenté à l'heure de l'exécution du *cron*. Néanmoins, cela ne pose pas de difficultés, vu que le rôle du serveur est de fournir une disponibilité permanente.

---

2. Sauvegarde exclusivement les données modifiées ou ajoutées depuis la précédente sauvegarde.

3. Sauvegarde les données modifiées ou ajoutées en référence à la dernière sauvegarde complète.

4. Gestionnaire des tâches sous Linux devant être exécutées à un moment précis.

## 2.6 Sécurité des mots de passe

Afin de sécuriser l'accès au serveur, le mot de passe de l'administrateur est « **4c,4cs,1lM&6lm.** ». Il a une longueur de 15 caractères et est composé de 4 chiffres, 4 caractères spéciaux, 1 lettre majuscule et 6 lettres minuscules.

Dans le meilleur des cas, tous les mots de passes seraient semblables, dans leur force, mais différents dans leur forme.

Afin de nous simplifier la tâche lors de la présentation, nous avons décidé que tous les mots de passe, *autres que celui de l'administrateur*, seraient « **test** ».

Les différents logins sont donc :

- administrateur : root - **4c,4cs,1lM&6lm.** ;
- utilisateur 1 : alexandre - **test** ;
- utilisateur 2 : terencio - **test** ;
- administrateur de la base de données : root - **test**.

Remarque : le mot de passe de l'administrateur a été construit directement depuis cette phrase : « *4 chiffres, 4 caractères spéciaux, 1 lettre majuscule et 6 lettres minuscules.* » en prenant la première lettre de chaque mot et en remplaçant **et** par **&**.

## 3 Services

### 3.1 Base de données

Le serveur utilise MariaDB, un fork communautaire de MySQL édité sous licence GPL.

MySQL est un système de gestion de bases de données relationnelles. Il fait partie des logiciels de gestion de bases de données les plus utilisés au monde.

#### 3.1.1 Mise en place

La totalité de l'implémentation se trouve dans le fichier `/usr/bin/deepblue.sql`

La base de données a été installée et configurée sur le serveur en différentes étapes :

- installation de MariaDB par APT (*Advanced Package Tool*) ;

```
1 apt-get install -y mariadb-server
```

- configuration sécurisée de l'installation de MariaDB ;

```
1 mysql_secure_installation
```

- création de la base de données, nommée *deepblue* ;

```
1 CREATE DATABASE IF NOT EXISTS deepblue
2 CHARACTER SET 'utf8'
3 COLLATE utf8_general_ci;
```

- création de la table « *users* » contenant les différents utilisateurs.

```
1 USE deepblue;
2
3 CREATE TABLE IF NOT EXISTS users (
4 id INT NOT NULL AUTO_INCREMENT,
5 username VARCHAR(30) NOT NULL,
6 created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
7 updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
8 CONSTRAINT pk_users PRIMARY KEY(id)
9 ) ENGINE=InnoDB;
```

## 3.2 DNS

Le DNS (*Domain Name System*), est un service permettant de résoudre un nom de domaine.

De fait, les serveurs étant identifiés par leur adresse IP, il a fallu créer un processus afin d'associer leur adresse à un nom plus simple à retenir, le « nom de domaine ».

### 3.2.1 Sélection du DNS

Il a été choisi d'utiliser BIND9 (*Berkeley Internet Name Daemon*).

Celui-ci est le serveur DNS le plus utilisé sur Internet, spécialement sur les systèmes de type UNIX et est devenu de facto un standard.

### 3.2.2 Mise en place

La majorité de l'implémentation se trouve dans le fichier */etc/bind/named.conf.options*.

Le DNS a été installé et configuré sur le serveur en différentes étapes :

— installation de BIND9;

```
1 # Installation of bind9.  
2 apt-get install bind9 bind9utils bind9-doc -y
```

— création des ACL (*Access Control List*) définissant le réseau local;

```
1 acl goodclients {  
2     10.1.0.0/16;  
3     localhost;  
4     localnets;  
5 };
```

— création et configuration du serveur DNS en lui-même :

- acceptation des requêtes uniquement pour le réseau interne;

```
1 recursion yes;  
2 allow-query { goodclients; };
```

- configuration des forwarders;

```
1 forwarders {  
2     8.8.8.8;  
3     8.8.4.4;  
4 };  
5 forward only;
```

- activation de *DNSSEC* qui sécurise les données envoyées par le DNS;

```
1 dnssec-enable yes;  
2 dnssec-validation yes;
```

- implémentation de la RFC1035<sup>5 6</sup>.

```
1 # Conform to RFC1035  
2 auth-nxdomain no;  
3 listen-on-v6 { any; };
```

---

5. <http://abcdrfc.free.fr/rfc-vf/rfc1035.html>

6. <http://www.bortzmeyer.org/1035.html>



## 3.3 FTP

Un serveur FTP (*File Transfer Protocol*), permet de transférer des fichiers par Internet ou par le biais d'un réseau informatique local (intranet). Pour ce serveur, il sera disponible au travers du réseau local.

### 3.3.1 Choix du serveur

Pour un maximum de sécurité, VsFTPD (*Very Secure FTP Daemon*) a été utilisé.

Ce serveur FTP est fortement axé sécurité : c'est l'un des premiers logiciels serveurs à mettre en œuvre la séparation des privilèges, minimisant ainsi les risques de piratage.

Dans sa configuration par défaut, VsFTPD est très restrictif :

- seul le compte anonyme est autorisé à se connecter au serveur, et en lecture seule ;
- les utilisateurs ne peuvent accéder qu'à leur compte.

### 3.3.2 Configuration

La totalité de l'implémentation se trouve dans les fichiers :

- `/etc/vsftpd.conf`
- `/etc/pam.d/vsftpd`

Voici les différentes étapes et options qui ont été effectuées :

- installation de vsFTPD

```
1 # Installation of vsftpd.  
2 apt-get install -y vsftpd
```

- choix du port et du monitoring

```
1 # Change the number of port to transmit.  
2 listen_port=52152  
3  
4 # Basic monitoring.  
5 setproctitle_enable=YES
```

— configuration de vsFTPD;

```
1 # Set vsftpd in standalone mode.
2 listen=YES
3 # Block the not allowed users.
4 anonymous_enable=NO
5 # Allow the local connections
6 local_enable=YES
7 write_enable=YES
8 local_umask=022
9 # Allow connection for guests users.
10 guest_enable=YES
11 # Default user for connections.
12 guest_username=apache
13 # Avoid local users go to /root.
14 chroot_local_user=YES
15 # Send virtual users into the default folder.
16 local_root=/srv/www/
17 # PAM manages the authentications of the system.
18 # We can set a login and a password to all the systems.
19 pam_service_name=vsftpd
20 # Create a default folder for users.
21 user_config_dir=/etc/vsftpd/vsftpd_conf_users
```

— configuration de PAM;

```
1 # Create the configuration file.
2 ##### PAM VSFTPD CONFIGURATION #####
3
4 # Authentication"
5 auth required /lib/x86_64-linux-gnu/security/pam_userdb.so
6     db=/etc/vsftpd/login
7 account required /lib64/security/pam_userdb.so
8     db=/etc/vsftpd/login
```

## 3.4 iptables

iptables permet de configurer les règles du pare-feu en IPv4.

### 3.4.1 Écriture du script

Le script est `/bin/script/iptables/iptables-conf.sh`.

Voici comment il est structuré :

— suppression des règles par défaut ;

```
1 # Flushing all rules from all tables.
2 iptables -F ; iptables -X
3 iptables -t nat -F ; iptables -t nat -X
4 iptables -t mangle -F ; iptables -t mangle -X
5 iptables -t raw -F ; iptables -t raw -X
```

— mise en place de règles par défaut ;

```
1 # Setting default filter policy.
2 iptables -P INPUT DROP
3 iptables -P OUTPUT DROP
4 iptables -P FORWARD DROP
```

— autorisation des paquets liés à l'adresse de *loopback* ;

```
1 # Allow loopback access.
2 iptables -A INPUT -i lo -j ACCEPT
3 iptables -A OUTPUT -o lo -j ACCEPT
```

— autorisation du service ICMP ;

```
1 # Ping from inside to outside.
2 iptables -A INPUT -p icmp --icmp-type \
3 echo-reply -j ACCEPT
4 iptables -A OUTPUT -p icmp --icmp-type \
5 echo-request -j ACCEPT
```

```

1 # Ping from outside to inside.
2 iptables -A OUTPUT -p icmp --icmp-type \
3 echo-reply -j ACCEPT
4 iptables -A INPUT -p icmp --icmp-type \
5 echo-request -j ACCEPT

```

— autorisation du service DNS;

```

1 # Allow DNS (53)
2 iptables -A OUTPUT -p udp -o eth0 --dport 53 -j ACCEPT
3 iptables -A INPUT -p udp -i eth0 --sport 53 -j ACCEPT

```

— autorisation des services HTTP et HTTPS;

```

1 # Allow incoming HTTP (80) and HTTPS (443).
2 iptables -A INPUT -i eth0 -p tcp -m multiport --dports
3 80,443 -m state --state NEW,ESTABLISHED -j ACCEPT
4 iptables -A OUTPUT -o eth0 -p tcp -m multiport --sports
5 80,443 -m state --state ESTABLISHED -j ACCEPT

```

— autorisation du service SSH;

```

1 # Allow incoming SSH (62000).
2 iptables -A INPUT -i eth0 -p tcp --dport 62000
3 -m state --state NEW,ESTABLISHED -j ACCEPT
4 iptables -A OUTPUT -o eth0 -p tcp --sport 62000
5 -m state --state ESTABLISHED -j ACCEPT
6
7 # Allow outgoing SSH (62000).
8 iptables -A OUTPUT -o eth0 -p tcp --dport 62000
9 -m state --state NEW,ESTABLISHED -j ACCEPT
10 iptables -A INPUT -i eth0 -p tcp --sport 62000
11 -m state --state ESTABLISHED -j ACCEPT

```

— autorisation du service NTP ;

```
1 # Allow outgoing NTP (123).
2 iptables -A OUTPUT -p udp --dport 123 -j ACCEPT
3 iptables -A INPUT -p udp --sport 123 -j ACCEPT
4
5 # Allow incoming NTP (123).
6 iptables -A INPUT -p udp --dport 123 -j ACCEPT
7 iptables -A OUTPUT -p udp --sport 123 -j ACCEPT
```

— autorisation du service NFS ;

```
1 # Allow incoming NFS.
2 iptables -A INPUT -s 10.1.0.0/16 -d 10.1.0.0/16
3 -p tcp -m multiport --dports 111,2049,36089,
4 43008,43301,48232,50277 -m state --state
5 NEW,ESTABLISHED -j ACCEPT
6 iptables -A INPUT -s 10.1.0.0/16 -d 10.1.0.0/16
7 -p udp -m multiport --dports 111,2049,33111,
8 42714,43880,46765,55770 -m state --state
9 NEW,ESTABLISHED -j ACCEPT
10
11 # Allow outgoing NFS.
12 iptables -A OUTPUT -s 10.1.0.0/16 -d 10.1.0.0/16
13 -p tcp -m multiport --sports 111,2049,36089,
14 43008,43301,48232,50277 -m state --state
15 ESTABLISHED -j ACCEPT
16 iptables -A OUTPUT -s 10.1.0.0/16 -d 10.1.0.0/16
17 -p udp -m multiport --sports 111,2049,33111,
18 42714,43880,46765,55770 -m state --state
19 ESTABLISHED -j ACCEPT
```

— autorisation du service Samba.

```
1 # Allow incoming Samba (UDP: 137,138 | TCP: 139,445)
2 iptables -A INPUT -s 10.1.0.0/16 -d 10.1.0.0/16
3 -p udp -m multiport --dport 137,138 -m state
4 --state NEW,ESTABLISHED -j ACCEPT
5 iptables -A INPUT -s 10.1.0.0/16 -d 10.1.0.0/16
6 -p tcp -m multiport --dport 139,445 -m state
7 --state NEW,ESTABLISHED -j ACCEPT
8
9 # Allow outgoing Samba (UDP: 137,138 | TCP: 139,445)
10 iptables -A OUTPUT -d 10.1.0.0/16 -p udp -m multiport
11 --dport 137,138 -m state --state ESTABLISHED -j ACCEPT
12 iptables -A OUTPUT -d 10.1.0.0/16 -p tcp -m multiport
13 --dport 139,445 -m state --state ESTABLISHED -j ACCEPT
```

## 3.5 NFS

Le NFS (*Network File System*), est un protocole qui permet à un ordinateur d'accéder à des fichiers distants via un réseau.

Ce système de fichiers en réseau permet de partager des données principalement entre systèmes UNIX.

### 3.5.1 Constatation

Avant de commencer, il est à remarquer que, quelle que soit sa version, NFS est à déployer dans un réseau local, celui-ci n'a pas de vocation à être ouvert sur Internet.

En effet, les données qui circulent sur le réseau ne sont pas chiffrées et les droits d'accès sont accordés en fonction de l'adresse IP du client qui peut être usurpée.

### 3.5.2 Configuration côté serveur

La totalité de l'implémentation se trouve dans le fichier */etc/exports*

Voici les différentes étapes et options qui ont été effectuées :

- installation des différents services indispensables au NFS ;

```
1 # Installation of NFS.
2 apt-get install -y nfs-kernel-server nfs-common
```

- création du dossier de partage, et ajout de droits spécifiques ;

```
1 mkdir /srv/share
2 chmod 777 /srv/share
```

- activation du partage sur le réseau local et configuration dudit partage (autorise la lecture et l'écriture, retire les droit **root** à distance et désactivation de la vérification de sous-répertoires) ;

```
1 /srv/share 10.1.0.0/16(rw,no_subtree_check,root_squash)
```

- mises à jour de la tables des systèmes de fichiers partagés.

```
1 # Update the table of exported file systems.
2 exportfs -av
```

### 3.5.3 Configuration côté client

Sur le client, la configuration est similaire :

- installation des différents services indispensables au NFS ;

```
1 # Installation of NFS.
2 apt-get install -y nfs-common
```

- création du dossier de partage, et ajout de droits spécifiques ;

```
1 mkdir /mnt/share/users
2 chmod 777 /mnt/share/users
```

- installation d'AutoFS ;

```
1 # Installation of AutoFS.
2 apt-get install AutoFS
```

- configuration d'AutoFS

Contenu du fichier */etc/auto.master* :

```
1 /mnt/share    /etc/auto.nfs    --ghost,timeout=30
```

Contenu du fichier */etc/auto.nfs* :

```
1 users -noexec,nosuid,rw,ghost \
2 10.1.214.184:/srv/share/users
```

Remarque : l'adresse IP "10.1.214.184" étant celle du serveur NFS.

La configuration ci-dessus permet la création d'un point de montage lors de l'accès au répertoire.



## 3.6 NTP

Le NTP (*Network Time Protocol*), est le protocole utilisé afin de synchroniser les machines du réseau local en fonction de l'horloge du serveur.

### 3.6.1 Principe

Bien que tout équipement informatique dispose d'une horloge, celle-ci dérive comme toute montre ordinaire, ce qui peut amener à des erreurs de synchronisation.

La nécessité de synchroniser des équipements en réseau paraît alors évidente.

Chaque machine peut être à la fois serveur et cliente. Celle-ci sélectionnera un serveur de temps dans sa configuration, et récupérera l'heure, ainsi qu'un numéro de strate,  $n$ , et se déclarera de strate  $n + 1$ .

L'architecture du réseau est en arborescence, et divisée en trois couches :

1. les plus précises (horloges atomiques, récepteurs GPS, ...) sont de *strate 0* ;
2. les serveurs diffusant l'heure des sources sont de *strate 1* ;
3. les serveurs de *strate 2* sont généralement accessibles au public.

### 3.6.2 Configuration du serveur

La totalité de l'implémentation se trouve dans le fichier `/etc/ntp.conf`

Voici les différentes étapes et options qui ont été effectuées :

- activation des statistiques NTP ;
- ajout de trois serveurs (un belge et deux européens) ;
- activation de l'échange de l'heure avec tout le monde (aucune modification n'est acceptée) ;
- activation de la synchronisation avec les machines du réseau local.

```

1  ...
2
3  # Adjust time server.
4  ntpdate 1.be.pool.ntp.org
5
6  driftfile /var/lib/ntp/ntp.drift
7
8  # Statistics loopstats peerstats clockstats.
9  filegen loopstats file loopstats type day enable
10 filegen peerstats file peerstats type day enable
11 filegen clockstats file clockstats type day enable
12
13 # You do need to talk to an NTP server or two (or three).
14 server 1.be.pool.ntp.org iburst
15 server 3.europe.pool.ntp.org
16 server 2.europe.pool.ntp.org
17
18 # By default, exchange time with everybody, but don't
19 # allow configuration.
20 restrict -4 default kod notrap nomodify nopeer noquery
21 restrict -6 default kod notrap nomodify nopeer noquery
22
23 # Local users may interrogate the ntp server more closely.
24 restrict 127.0.0.1
25 restrict 10.1.0.0 mask 255.255.0.0 nomodify notrap nopeer
26 restrict ::1
27
28 # To provide time to the local subnet.
29 broadcast 10.1.255.255
30
31 ...

```

Remarque : l'adresse de diffusion (*broadcast*) a été adapté en fonction du réseau.

### 3.6.3 Configuration du client

Tout comme le serveur, la totalité de l'implémentation se trouve dans le fichier */etc/ntp.conf*

Sur le client, la configuration est beaucoup plus simple :

- activation des statistiques NTP ;
- ajout du serveur local.

```
1 ...
2
3 # Adjust time server.
4 ntpdate 1.be.pool.ntp.org
5
6 # File containing the average deviation.
7 driftfile /var/lib/ntp/ntp.drift
8
9 # Desired Statistics.
10 statistics loopstats peerstats clockstats
11 filegen loopstats file loopstats type day enable
12 filegen peerstats file peerstats type day enable
13 filegen clockstats file clockstats type day enable
14
15 # You do need to talk to an NTP server or two (or three).
16 server 10.1.214.184 prefer
17
18 ...
```

Remarque : l'adresse IP “10.1.214.184” étant celle du serveur NTP.

### 3.7 Quotas

Dans le but de mieux gérer l'espace personnel de chaque utilisateur, des quotas ont été mis en place sur la partition */home*.

À la création de chaque utilisateur, un quota avec une limite dure<sup>7</sup> de 500 Mo et une limite douce<sup>8</sup> de 400 Mo, lui sera attribué.

Remarque : lors du dépassement de la limite douce, l'utilisateur sera averti.

```
1 # Installation of quotatool, useful for scripts.
2 apt-get install -y quota quotatool
3
4 # Add this to /etc/fstab to the /home line.
5 usrquota,grpquota
6
7 fuser -k /dev/mapper/VolGroup-LVhome
8 # Create the file 'aquota.user' and aquota.group' and
9 # initialize all the partition that contains quotas.
10 quotacheck -cguvf /dev/mapper/VolGroup-LVhome
11 quotacheck -vagum
12
13 # Unmount the /home partition.
14 umount -l /dev/mapper/VolGroup-LVhome
15
16 # Activate quota.
17 quotaon -avug
18
19 # Mount the /home partition.
20 mount /dev/mapper/VolGroup-LVhome
```

---

7. Limite que nul ne peut dépasser.

8. Limite que l'utilisateur (ou groupe) peut dépasser pendant un certain laps de temps.

## 3.8 Samba

Samba est un outil permettant de partager des dossiers et des imprimantes à travers un réseau local.

Son utilisation est conseillée pour partager de manière simple des ressources entre plusieurs ordinateurs.

Celui-ci est compatible avec les systèmes d'exploitation suivants : Windows, macOS, ainsi que des systèmes GNU/Linux, \*BSD et Solaris dans lesquels une implémentation de Samba est installée.

### 3.8.1 Configuration

La configuration du serveur Samba se déroule en trois parties, mais tout d'abord, il est nécessaire de créer le dossier de partage et de lui donner les droits appropriés.

```
1 mkdir -p /srv/share/users/
2 chown -R root:users /srv/share/users/
3 chmod -R 775 /srv/share/users/
```

La totalité de l'implémentation se trouve dans le fichier */etc/samba/smb.conf*.

1. configuration de Samba (désignation du *workgroup*, choix du nom de *netbios*, etc.) ;

```
1 # Installation of Samba.
2 apt-get -y install libcups2 samba samba-common cups
3 # If you don't know the name of the workgroup
4 # run this command on the Windows client to get
5 # the workgroup name: net config workstation.
6 [global]
7 workgroup = WORKGROUP
8 server string = Samba Server %v
9 netbios name = debian
10 security = user
11 map to guest = bad user
12 dns proxy = no
```

2. configuration du partage pour le groupe « *users* » (désignation du chemin, des droits, etc.);

```
1 [users]
2 comment = All Users
3 path = /srv/share/users
4 valid users = @users
5 force group = users
6 create mask = 0660
7 directory mask = 0771
8 writable = yes
```

3. configuration du partage du dossier « *home* » des utilisateurs (désignation des droits, vérification de l'identité, etc.).

```
1 [homes]
2 comment = Home Directories
3 browseable = no
4 valid users = %S
5 writable = yes
6 create mask = 0700
7 directory mask = 0700
```

### 3.9 SELinux

SELinux (*Security-Enhanced Linux*) permet de définir des politiques d'accès à différents éléments du système d'exploitation. Ces éléments peuvent être des processus (*démons*), ou encore des fichiers.

Dans le cadre de la mise en place d'un serveur conséquent, il aurait fallu implémenter ce type de service.

Pour ce projet, un début d'implémentation a été fait, mais vu que l'ensemble des services n'était pas encore répertorié, il a été décidé de faire abstraction de ce service afin d'éviter que certains services ne se retrouvent bloqués.

```
1 # Installation of SELinux.
2 apt-get install -y selinux-basics selinux-policy-default
3     auditd
4
5 # Configure GRUB and PAM and create /.autorelabel
6 selinux-activate
7 # Reboot the system.
8 reboot
9 # Check that everything has been setup correctly and
10 # catch common SELinux problems.
11 check-selinux-installation
12 # You can see all would-be denials since the last reboot.
13 audit2why -al
14 # SELinux, enable enforcing mode temporarily by running:
15 setenforce 1
16 # To enable enforcing mode permanently, you need to add it to
17 # the kernel command.
18 echo "enforcing=1" >> /etc/default/grub
19 # Reboot the system.
20 reboot
21 # Some of the PAM config files need to have "session required
22 # pam_selinux.so multiple"
```

## 3.10 Serveur Web

Pour la création du serveur Web, Apache a été utilisé.

### 3.10.1 Configuration côté serveur

La totalité de l'implémentation se trouve dans les fichiers

*/etc/apache2/mods-available/mpm\_prefork.conf*

*/etc/apache2/mods-available/mpm\_event.conf*

Voici les différentes étapes et options qui ont été effectuées :

- installation de la version 2 d'Apache ;

```
1 # Installation of Apache 2.
2 apt-get install -y apache2 apache2-doc apache2-utils
```

- configuration des modules multiprocessus (MPMs) ;

```
1 # prefork MPM.
2 <IfModule mpm_prefork_module>
3     StartServers           4
4     MinSpareServers        20
5     MaxSpareServers        40
6     MaxRequestWorkers      200
7     MaxConnectionsPerChild 4500
8 </IfModule>
```

- désactivation du module d'événement et activation de divers modules ;

```
1 # On Debian 8, the event module is enabled by default.
2 # This will need to be disabled, and the prefork
3 # module enabled.
4 a2dismod mpm_event
5 a2enmod mpm_prefork
6 a2enmod userdir
7 a2enmod rewrite
```



— configuration du module d'événement ;

```
1 # event MPM.
2 <IfModule mpm_event_module>
3     StartServers                2
4     MinSpareServers             25
5     MaxSpareServers             75
6     ThreadLimit                 64
7     ThreadsPerChild             25
8     MaxRequestWorkers           150
9     MaxConnectionsPerChild      3000
10 </IfModule>
```

— configuration du serveur Apache ;

```
1 # Disable the default Apache virtual host.
2 a2dissite 000-default.conf
3
4 # Creation of the Web folder.
5 mkdir -p /srv/www/example/www
```

— masquage de la version de PHP pour les utilisateurs ;

```
1 # Hide the version of PHP for users.
2 min_info
```

— installation du support PHP.

```
1 # Install PHP support.
2 apt-get install -y php5 php-pear
```

Remarque : les dossiers des sites Web ainsi que des logs sont créés lors de l'ajout d'un utilisateur.

## 3.11 SSH

Le SSH (*Secure Shell*), est un protocole de communication sécurisé. Celui-ci impose un échange de clés de chiffrement en début de connexion.

### 3.11.1 Type d'authentification

Il existe plusieurs manières de s'authentifier sur le serveur via SSH.

Les deux authentifications les plus utilisées sont :

1. par mot de passe ;
2. par clés publiques et privées du client.

L'identification automatique par clés a été mise en place pour ce serveur. De ce fait, il n'est plus nécessaire d'entrer le mot de passe à chaque connexion à distance.

Cette méthode est plus complexe à mettre en place, mais elle est surtout plus pratique.

On remarque rapidement son utilité si on se connecte fréquemment au serveur, car plus aucun mot de passe n'est demandé.

### 3.11.2 Implémentation

La majorité de l'implémentation se trouve dans le fichier `/etc/ssh/sshd_config`.

Le serveur a été configuré en respectant ces critères :

- installation de *openssh* ;

```
1 # Installation of OpenSSH.  
2 apt-get install openssh-server -y
```

- changement de port et passage à la version 2 de SSH pour plus de sécurité ;

```
1 # Using port number 62000  
2 Port 62000  
3 # Using Protocol 2 of SSH.  
4 Protocol 2
```

— ajout d’une bannière (disponible dans le fichier */etc/ssh-banner/banner*);

```
1 The debianThink server is for authorized personnel only.
2 WARNING! Access to this device is restricted to those
3 individuals with specific permissions. If you are not an
4 authorized user, disconnect now. Any attempts to gain
5 unauthorized access will be prosecuted to the fullest
6 extent of the law.
7
8 All access and use may (not will) be monitored
9 and/or recorded.
```

— connexion en tant que **root**;

```
1 # Privilege separation for security.
2 UsePrivilegeSeparation yes
```

— déconnexion après 120 secondes d’inactivité;

```
1 # Deactivation of the login in root and disconnection
2 # after 120 seconds if no connections.
3 LoginGraceTime 120
4 PermitRootLogin no
5 StrictModes yes
```

— désactivation de la connexion par mot de passe, vu que l’authentification passe par les clés RSA.

```
1 # We deny the authentication by password.
2 PasswordAuthentication no
```

Ensuite, une génération et un chiffrement d’une paire de clés publique / privée sur la machine cliente ont été nécessaires.

```
1 ssh-keygen -t rsa -b 4096 -C $email -f "$USER/.ssh/id_rsa" \
2 -N ""
```

Une fois cela fait, la clé publique de celle-ci a été enregistrée sur le serveur dans le fichier */etc/ssh/ssh\_host\_rsa\_key* afin d’accepter sa connexion au serveur.

## 3.12 Utilisateurs

Ce n'est pas un service en tant que tel, mais, lors de la création d'un nouvel utilisateur, celui-ci passe par plusieurs étapes :

- il est ajouté à la base de données ;
- un site est créé au nom de l'utilisateur ;
- il est ajouté à la liste des utilisateurs de Samba ;
- il possède désormais un espace de stockage.

Lors de la suppression de l'utilisateur, il est nécessaire de supprimer ses données.

### 3.12.1 Configuration

- création de l'utilisateur

```
1 # Creates an user .
2 useradd $username -m -G users -s /bin/bash
3 echo -e "$password\n$password" | (passwd $username)
4
5 quotatool -u $username -bq 400M -l 500M /home
6 smbpasswd -a $username
7 # Each user has access to the 'deepblue' database.
```

- création du site dans le fichier */etc/apache2/sites-enabled/\$username.lan.conf*

```
1 # Creates a website for a specific user.
2 <VirtualHost *:80>
3     ServerAdmin $username@$username.lan
4     ServerName $username.lan
5     ServerAlias www.$username.lan
6     DocumentRoot /srv/www/$username/www
7     ErrorLog /srv/www/$username.lan/logs/error.log
8     CustomLog /srv/www/$username.lan/logs/access.log
9         combined
```

```

10 </VirtualHost>
11 Alias /$username '/srv/www/$username/www'
12 <Directory '/srv/www/$username/www'>
13     Options Indexes FollowSymLinks MultiViews
14     AllowOverride All
15     Order deny,allow
16     Allow from all
17 </Directory>
18
19 mkdir -p "/srv/www/$username/www"
20 mkdir -p "/srv/www/$username.lan/logs/"
21 chmod 755 "/srv/www/$username/www"

```

— création de la page d'accueil du site */srv/www/\$username/www/index.html*

```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="utf-8">
5         <title>Server page</title>
6     </head>
7     <body>
8         <div id="greetings">
9             <h1>Welcome \$username!</h1>
10            <h2>This is your own server page.</h2>
11        </div>
12    </body>
13 </html>

```

## 4 Conclusion du projet

Dans le cadre du cours d'*Administration Linux*, il nous a été demandé d'administrer un serveur sous Linux.

Ce projet nous a permis de mettre en pratique la théorie vue au premier quadrimestre de manière réaliste. Nous avons libre choix quant à la distribution ainsi qu'au type de gestion des sauvegardes, à condition de les justifier.

Le serveur devait contenir un partage NFS et Samba, un serveur Web, FTP, MySQL, DNS ainsi que NTP, et le support du module SSH.

La mise en place de ces services nous a obligé à nous documenter et à prendre des précautions telles que :

- automatiser les installations et configurations au travers de scripts ;
- documenter ces scripts ;
- sauvegarder la machine virtuelle lors de chaque changement sur le serveur ;
- contrôler le bon fonctionnement de chaque élément.

De plus, nous avons dû nous répartir les tâches afin d'avancer efficacement dans le projet. Dans cette optique, le passage par GitHub<sup>9</sup>, un outils de gestion de développement et de version, nous a été bénéfique.

Celui-ci nous permettant de toujours travailler sur la dernière version des scripts sans rencontrer de problèmes de synchronisation.

Grâce à l'administration de ce serveur, nous avons pu approfondir nos connaissances dans la configuration de services et dans l'écriture de scripts.

---

9. <https://fr.wikipedia.org/wiki/GitHub>

