



# RAPPORT DE PROJET

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

---

## Linux

---

*Auteur :*  
Timothée SIMON  
Florian GIARRUSSO  
Fabio CUMBO

*Enseignant :*  
Antoine MALAISE

The logo for Campus technique is a red square with the words 'Campus' and 'technique' in white, stacked vertically.

Année académique 2017 - 2018



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

1	Introduction . . . . .	2
1.1	Choix de la distribution . . . . .	2
1.2	Mode d'installation . . . . .	2
1.3	Organisation du groupe . . . . .	2
1.4	Machine physique . . . . .	3
2	Installation . . . . .	4
3	Serveur NTP . . . . .	5
4	Connexion SSH . . . . .	6
5	Serveur NTP . . . . .	7
6	Serveurs de fichiers . . . . .	8
6.1	Serveur NFS . . . . .	8
6.2	Serveur SMB . . . . .	9
7	Serveur FTP . . . . .	11
8	Remerciement . . . . .	12
	Références . . . . .	13

# 1 Introduction

## 1.1 Choix de la distribution

Pour le choix de la distribution nous avons commencé par mettre en place certains critères de recherche :

- La gratuité
- La stabilité
- La légèreté

Nos recherches nous montrent plusieurs choix possibles :

- RHEL : Payante
- Arch Linux : Pas la plus stable car elle fonctionne en rolling release
- Ubuntu Server et Debian : Ne possèdent pas de version core ce qui les rend plus lourdes
- CentOS : Répond au mieux à tous nos critères

Nous allons donc faire notre serveur sous CentOS car celui-ci répond à tous nos critères dans sa version Core.

## 1.2 Mode d'installation

Pour l'installation nous avons opté pour l'écriture de scripts pour chacune des fonctionnalités de notre serveur. Cela nous permet de nous rappeler de nos procédures et de toujours les comprendre dans quelques années. Il nous suffit maintenant de copier nos scripts sur une machine réelle ou virtuelle pour pouvoir immédiatement commencer à configurer Linux.

Bien évidemment nos scripts ne gèrent pas beaucoup d'erreurs et ne vérifient pas ce que l'utilisateur a entré, ils ne sont donc pas vraiment prêts pour tout usage mais sont très utiles comme notes.

## 1.3 Organisation du groupe

Pour nous organiser nous avons utilisé les outils de GitHub. Nous avons donc commencé par créer un repo (privé pour le moment mais nous le passerons sûrement en public après les examens) nous permettant de travailler ensemble sur notre code. Nous avons aussi utilisé la ToDo list de GitHub pour nous organiser dans notre développement.

## 1.4 Machine physique

Nous avons eu l'occasion d'utiliser une machine physique pour mettre en pratique nos scripts. Celle-ci est bien évidemment une machine de récupération, elle est composée de Intel Pentium 4, de 1Go de DDR2 et de deux disques dur en RAID 1.

## 2 Installation

Pour l'installation nous avons suivis les étapes suivantes :

- Mise en place de la machine avec l'iso bootable (sur machine virtuelle ou physique)
- Sélection des locale (English US)
- Modification de l'heure (Bruxelles)
- Partitionnement manuel (LVM avec encryption AKA LUKS) :
  - /dev/sda1 monté sur /boot : 1024 Mo en XFS
  - LVM (encryptée)
    - centos-root monté sur / : 5320 Mo en XFS
    - centos-home monté sur /home : 1023 Mo en XFS
    - centos-swap monté comme partition de swap : 819 Mo
- Activation de la NIC (enp0s3)
- Nom de domaine
- Lancement de l'installation
- Mise en place du mot de passe root

### 3 Serveur NTP

Le protocole NTP (Network Time Protocol) va permettre de synchroniser les horloges des ordinateurs connectés au même réseau local que celle du serveur de temps. Celui-ci devra synchroniser sa propre horloge en contactant un serveur de temps de référence à distance donc par internet. Cette synchronisation des heures permettra entre autres de ne pas perturber certaines applications utilisant l'horloge du système mais aussi pour donner plus de cohérence en cas de comparaison des messages de « logs » de plusieurs ordinateurs sur le réseau.

```
1 #!/bin/bash
3 source ../Common.sh
5 RootCheck
7 #Installation de NTP
  Installe ntp ntpdate ntp-doc
9
  #On stop le service ntpd
11 systemctl stop ntpd
13
  #On met le serveur à la bonne heure au préalable
  ntpdate be.pool.ntp.org
15
  #Configuration du serveur ntp
17 cp ntp.conf /etc/ntp.conf
19
  #Démarrage et activation du service
  Service ntpd
21
  echo "Le service NTP est maintenant installé et activé."
```

../scripts/ntp/NTP.sh

On déclare d'abord le fichier de dérive « driftfile ». Il va permettre de corriger les dérives de l'horloge système en l'absence de connexion réseau au serveur de référence. On déclare ensuite le répertoire et le fichier pour stocker les « logs » du service ntpd. On permet la synchronisation avec notre source de temps mais on interdit à la source de modifier ou d'interroger le service sur ce système. On autorise les accès sur l'interface de bouclage Ensuite, on se synchronise avec les serveurs NTP belge de référence. Pour résoudre les problèmes de charge on entre plusieurs adresse. Il s'agit d'un groupement de serveur et la redistribution se fait à l'aide du Round Robin DNS (association de plusieurs adresses IP à un FQDN) Enfin, on précise au serveur de se synchroniser sur l' « Undisciplined Local Clock » et on indique un faux « pilote » destiné à la sauvegarde de l'heure dans le cas où aucune source externe d'heure synchronisée n'est disponible.



## 4 Connexion SSH

Pour l'administration à distance nous avons décidé d'utiliser SSH (*Secure **SH**ell*) car ce protocole est plus sécurisé que Telnet. Pour la connexion nous n'autorisons que l'authentification avec une clé RSA ou avec un mot de passe et un facteur de double authentification. Pour la double authentification nous avons décidé d'utiliser une clé Yubikey, ce facteur étant physique cela limite fortement l'accès aux personnes non autorisées. Nous avons choisi de permettre une autre méthode d'authentification car l'utilisation d'une clé RSA seule est un risque en cas de perte du fichier et nous voulions toujours pouvoir être capable d'administrer notre serveur même si ce fichier est perdu ou corrompu.

```
#!/bin/bash
2
source ../Common.sh
4
RootCheck
6
#Installation du paquet open-ssh (vérification) permettant de partager
  un service ssh, par défaut il est normalement déjà installé
8 Installe openssh-server
10
# Utilisation du fichier de config
cp sshd_config /etc/ssh/sshd_config
12 cp sshbanner /etc/ssh/banner
14
#démarrage service sshd
Service sshd
```

../scripts/ssh/SSH.sh

Après avoir vérifié que l'utilisateur qui lance le script est bien root et installé le serveur openSSH si celui-ci n'était pas déjà installé, nous copions simplement le fichier de configuration et la bannière au bon endroit. Pour finir nous lançons et activons au démarrage le daemon.

## 5 Serveur NTP

Le protocole NTP (Network Time Protocol) va permettre de synchroniser les horloges des ordinateurs connectés au même réseau local que celle du serveur de temps. Celui-ci devra synchroniser sa propre horloge en contactant un serveur de temps de référence à distance donc par internet. Cette synchronisation des heures permettra entre autres de ne pas perturber certaines applications utilisant l'horloge du système mais aussi pour donner plus de cohérence en cas de comparaison des messages de « logs » de plusieurs ordinateurs sur le réseau.

```
1 #!/bin/bash
3 source ../Common.sh
5 RootCheck
7 #Installation de NTP
  Installe ntp ntpdate ntp-doc
9
  #On stop le service ntpd
11 systemctl stop ntpd
13
  #On met le serveur à la bonne heure au préalable
  ntpdate be.pool.ntp.org
15
  #Configuration du serveur ntp
17 cp ntp.conf /etc/ntp.conf
19
  #Démarrage et activation du service
  Service ntpd
21
  echo "Le service NTP est maintenant installé et activé."
```

../scripts/ntp/NTP.sh

On déclare d'abord le fichier de dérive « driftfile ». Il va permettre de corriger les dérives de l'horloge système en l'absence de connexion réseau au serveur de référence. On déclare ensuite le répertoire et le fichier pour stocker les « logs » du service ntpd. On permet la synchronisation avec notre source de temps mais on interdit à la source de modifier ou d'interroger le service sur ce système. On autorise les accès sur l'interface de bouclage Ensuite, on se synchronise avec les serveurs NTP belge de référence. Pour résoudre les problèmes de charge on entre plusieurs adresse. Il s'agit d'un groupement de serveur et la redistribution se fait à l'aide du Round Robin DNS (association de plusieurs adresses IP à un FQDN) Enfin, on précise au serveur de se synchroniser sur l' « Undisciplined Local Clock » et on indique un faux « pilote » destiné à la sauvegarde de l'heure dans le cas où aucune source externe d'heure synchronisée n'est disponible.

## 6 Serveurs de fichiers

Nous avons mis en place deux serveurs de fichier, un serveur NFS et un serveur Samba.

### 6.1 Serveur NFS

Le serveur NFS (*Network File System*) permettant à un utilisateur d'accéder à des fichiers sur un serveur de la même façon qu'il accède à des fichiers stockés en local.

```
#!/bin/bash
2 source ../Common.sh

4 RootCheck

6 s="./NFS.sh [DOSSIER] [ARGUMENTS] [IP]
DOSSIER: Dossier de partage (Default: /Partage)
8 ARGUMENTS: Arguments du partage nfs (Default: (rw,sync,no_root_squash,
no_subtree_check))
IP: Adresse ip du serveur (Default: adresse IP local de la machine)
10 "

12 Aide $1 $s
# Défaut du dossier de partage
14 DossierPartage='Argument $1 "/Partage"'
ARG='Argument $2 "(rw,sync,no_root_squash,no_subtree_check)'"
16 IP='Argument $3 \'ip addr | grep \'state UP\' -A2 | tail -n1 | awk '{
print $2}' | cut -f1 -d\'/\'\'\'

18 # Installation du serveur nfs
Installe nfs-utils

20
#création du dossier partagé si celui-ci n'existe pas encore
22 mkdir -p $DossierPartage

24 #Modification des permissions d'accès
chmod 755 $DossierPartage

26
echo "Le dossier $DossierPartage est maintenant créé"
28

30 #Activation et démarrage des services nfs au boot
Service rpcbind
32 Service nfs

34
#Configuration du fichier /etc/exports
36 echo "$DossierPartage $IP$ARG" >> /etc/exports

38 #On exporte le partage
exportfs -a
40 showmount -e
```

---

../scripts/nfs/NFS.sh

Après avoir vérifié que l'utilisateur qui lance le script est bien root et importé les variables en argument, nous installons le serveur nfs si celui-ci n'est pas déjà installé. Ensuite nous créons le dossier de partage avec les bons droits et activons les processus requis pour le serveur. Finalement nous ajoutons le dossier partagé ainsi que les arguments au fichier et partage et nous en informons le nfs.

## 6.2 Serveur SMB

Le serveur SMB (*Server Message Block*) permet à des ordinateurs de partager des fichiers et des imprimantes entre eux.

```
#!/bin/bash
2
souce ../Common.sh
4
RootCheck
6 s=" ./SAMBA.sh DOSSIER UTILISATEUR NOM GROUPE
DOSSIER: Dossier de partage (Default: /Partage)
8 UTILISATEUR: Utilisateur possédant les droits (Default: Utilisateur
courant)
NOM: Nom du partage (Default: DOSSIER)
10 GROUPE: Nom du groupe utilisé pour le partage (Default: GroupePartage)
"
12
Aide $1 $s
14
#On crée le dossier de partage
16 DossierPartage='Argument $1 "/Partage"'
UserP='Argument $2 $USER'
18 NameP='Argument $3 $DossierPartage'
GroupePartage='Argument $4 "GroupePartage"'
20
#Installation samba
22 Installe samba
24 #démarrage et activation du démon au démarrage
Service smb
26
#création du dossier partagé si celui-ci n'existe pas encore
28 mkdir -p $DossierPartage
30 #Ajout du groupe sharedFolder contenant user1 et user2
groupadd $GroupePartage
32 useradd -g sharedFolder $UserP
34 #On gère les permissions du dossier partagé
chmod 770 $DossierPartage
36 chown -R $UserP:$GroupePartage $DossierPartage
```

```

38 echo "Veuillez choisir un mot de passe pour l'utilisateur $UserP"
smbpasswd -a $UserP
40
42 # Copie des réglages de samba
echo "
[{$NameP}]
44 path=${DossierPartage}
comment=Partage crée par un script
46 public=yes
force directory mode=777
48 force create mode=777
writeable=yes
50 browseable=yes" >> smb.conf
cp smb.conf /etc/samba/smb.conf
52
# Désactivation de SELinux (car autrement impossible de se connecter
  depuis le client , il faudra configurer tout ça )
54 setenforce 0
56 #Redémarrage du service smb
sudo systemctl restart smb.service

```

../scripts/samba/SAMBA.sh

Tout d'abord nous intégrons un menu d'aide et nous importons les variables passées en arguments. Ensuite nous installons le serveur samba si celui-ci n'est pas encore installé et nous démarrons ses services. Nous créons le groupe relatif au partage et nous y ajoutons l'utilisateur. Maintenant il nous faut créer demander le mot de passe du partage à l'utilisateur et créer la configuration. Finalement nous redémarons le daemon.

## 7 Serveur FTP

Le protocole FTP (File Transfer Protocol) permet de transférer des fichiers d'un serveur à un client.

```
1 #!/bin/bash
  source ../Common.sh
3 RootCheck

5 s='./FTP.sh [DOSSIER]
  DOSSIER: Dossier de partage (Defaut : /Partage)
7 '

9 Aide $1 $s

11 Installe vsftpd

13 DossierPartage='Argument $1 "/Partage"'

15 systemctl start vsftpd.service
  systemctl enable vsftpd.service
17

17 mkdir -p $DossierPartage

19 echo "# Allow all connections \nvsftpd: ALL\n# IP address range\nvsftpd:
  10.0.0.0/255.255.255.0" > /etc/hosts.allow
```

../scripts/ftp/FTP.sh

## 8 Remerciement

Je remercie Terencio AGOZZINO pour avoir réalisé la mise en page de ce document en  $\text{\LaTeX}$ .

Configuration du ssh [https ://www.linux.com/learn/advanced-ssh-security-tips-and-](https://www.linux.com/learn/advanced-ssh-security-tips-and-tricks)■  
tricks