



RAPPORT DE PROJET

2^{ÈME} BACHELIER EN INFORMATIQUE

Linux

Auteur :
Timothée SIMON
Florian GIARUSSO
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 | Remerciement | 7 |
| | Références | 8 |

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 pouvoir la comprendre même dans quelques années. Il nous suffit maintenant de copier nos scripts sur une machine réelle pour pouvoir immédiatement les lancer. 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 To Do 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
  systemctl start ntpd
21 systemctl enable ntpd
23
  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.

```
1 #!/bin/bash
3 source ../Common.sh
5 RootCheck
7 #Installation du paquet open-ssh (vérification) permettant de partager
  un service ssh, par défaut il est normalement déjà installé
  Installe openssh-server
9
  # Utilisation du fichier de config
11 cp sshd_config /etc/ssh/sshd_config
  cp sshbanner /etc/ssh/banner
13
  #démarrage service sshd
15 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 Remerciement

Je remercie Terencio AGOZZINO pour avoir réalisé la mise en page de ce document en \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