HAUTE ECOLE EN HAINAUT
Campus technique
8a, avenue V. Maistriau B-7000 Mons
Tél : +32 (0)65 33 81 54
Fax : +32 (0)65 31 30 51
E-mail : tech-mons@heh.be
www.heh.be

Projet Linux

Aout 2019
Version 1.0

Pittonet Benjamin

Table des matières

1. Introduction

Ce projet à pour but de configurer un serveur Linux from scratch en y incluant les services suivant :

- SSH
- NTP
- ClamAV (Antivirus)
- Bind (DNS)
- FTP
- Samba
- Mysql
- Apache
- PHP
- Firewall

J'utiliserai VMWare pour créer ma machine virtuelle et CentOS 7 comme distribution.

Ce choix est principalement motivé par le fait que CentOS est une distribution très populaire et bénéficie donc d'un soutien énorme de la communauté open-source.

En outre, elle est aussi dans le top 3 des meilleures distributions server.

Je documenterai au long de ce rapport comment j'ai configuré CentOS, ces services, les problèmes que j'ai rencontrés et les solutions trouvées.

Afin de réaliser chacune des taches, j'ai choisis de réaliser des scripts afin que chaque service puisse être installé indépendamment mais rapidement.

Tous ces scripts auront comme point commun :

- L'installation des services
- la copie en backup des fichiers de configuration d'origine
- la configuration automatisée du service
- la copie en backup des fichiers de configuration personnalisé
- les éventuels redémarrages de services requis
- un echo « EOF » signifiant la fin d'exécution de script

2. Installation

2.1. Distribution

Distribution stable, cohérente au projet et sécurisé, CentOS se base sur une solution commerciale de haute qualité. C'est donc tout naturellement que j'ai choisis cette distribution dans le cadre du projet.
J'avais choisi initialement une installation minimale, mais cette dernière posa des soucis de configuration à plus d'un titre, la plus problématique étant que la plupart des commandes de maintenance de base étaient manquantes.
J4ai donc choisis de relancer mon projet sur une version graphique basique de CentOS mais en désactivant l'interface graphique.

2.2. Partitions

Pour le partitionnement, j'ai adapté au projet les consignes que nous avions eu par le passé. Cela m'a permis d'avoir une bonne base sur laquel je suis venu implementer le reste des pré-requis du projet :

Point de montage	Type	Système de fichier	Taile
/boot	Partition Standard	Ext4	500 Mio
/home	LVM	Ext4	500 Mio
/	LVM	Ext4	7000 Mio
swap	Partition Standard	SWAP	2000 Mio
/var	LVM	Ext4	2000 Mio
/share	LVM	Ext4	7000 Mio

2.3. Config générale

En outre du partitionnement lors de l'installation, et lorsque le sytéme fut operationnel, j'ai également configuré les options suivantes :

Option	Configuration
Mot de passe root	toor
Utilisateur de base et Mot de passe	Support ; 404notfound
IP	Attribution d'un IP static (192.168.1.42)
Nom d'hote réseau	Plbp.localdomain
SELinux	Désactiver
Bash prompt	Colorer le shell pour améliorer la lisibilité
Interface graphique Gnome	Désactiver

J'ai également choisis de désactiver mon firewall jusqu'à complétion des taches afin de ne pas devoir constamment le réajuster. Sa configuration sera donc en fin de taches.

3. SSH

SSH permet de se connecter à distance sur la machine, ce qui le rend ce service l'un des plus pratique lors de la configuration et de la maintenance d'un serveur.

3.1. Script:

```
#!/bin/sh

# Install
yum install -y openssh openssh-server openssh-clients openssl-libs

# Backing up original conf file
cp /etc/ssh/sshd_config /root/backup_conf/original/

# Setting up home conf
echo "HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
SyslogFacility AUTHPRIV
PermitRootLogin no
AuthorizedKeysFile     .ssh/authorized_keys
PasswordAuthentication yes
ChallengeResponseAuthentication no
GSSAPIAuthentication yes
GSSAPICleanupCredentials no
UsePAM yes
X11Forwarding yes
Banner /etc/banner
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY
LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS
Subsystem      sftp      /usr/libexec/openssh/sftp-server" > /etc/ssh/sshd_config

# Setting up the banner
echo "Unauthorized Acces is Prohibited" > /etc/banner
```

```
# Backing up home conf file
cp /etc/ssh/sshd_config /root/backup_conf/home/
cp /etc/banner /root/backup_conf/home/

# Enable, reload and restart everything
systemctl enable sshd.service
systemctl restart sshd.service

# EOF
echo "EOF - SSH"
```

La majeure partie de la configuration SSH proviens du fichier d'origine.
Les principales lignes modifiées sont :

- Banner /etc/banner
- PasswordAuthentication yes
- PermitRootLogin no

Ces paramètres ont pour but d'empêcher de se connecter en tant que root et sans mot de passe. Nous en profitons aussi pour définir le fihcier dans lequel se trouvera un message de sécurité avertissant les contrevenants.

3.2. Utilisation

Pour l'utiliser, nous pouvons utiliser PuTTY.
Nous devons simplement rentrer l'ip de la machine distante en spécifiant le port utiliser (SSH utilise par défaut le port 22).

4. NTP

Le service NTP permettra à notre serveur de fournir le temps aux machines qui le lui demanderont afin de les synchroniser.

4.1. Script:

```
#!/bin/sh

# Install
yum install ntp -y
systemctl enable ntpd

# Backing up original conf file
cp /etc/ntp.conf /root/backup_conf/original/

# Setting up log file
echo "# Logfile for NTP service" > /var/log/ntp.log

# Sed will comment the default line, echo will add our version of it
sed -i -e "s/^logfile*/#logfile*/" /etc/ntp.conf
echo "logfile /var/log/ntp.log" >> /etc/ntp.conf

# Backing up home conf file
cp /etc/ntp.conf /root/backup_conf/home/

# Restart and enable
systemctl enable ntpd
systemctl restart ntpd

# EOF
echo "EOF - NTP"
```

Rien de bien particulier ici si c n'est l'utilisation de la commande « sed » qui va permettre d'automatiser notre script.
En effet contrairement à SSH ou le nombre de lignes de configuration n'était pas trop élevé, ce ne fut pas le cas ici. J'ai donc choisis d'utiliser la commande « sed » afin de commenter la ligne que je voulais remplacer et l'ai simplement echo juste après en fonction de la configuration voulues.

5. ClamAV

ClamAV sera notre antivirus pour ce projet et permettra de fournir une protection à notre serveur afin que ce dernier puisse détecter les menaces éventuelles.

5.1. Script:

```
#!/bin/sh

# Install
yum -y install epel-release
yum clean all
yum -y install clamav-server clamav-data clamav-update clamav-filesystem clamav clamav-scanner-systemd clamav-devel clamav-lib clamav-server-systemd

# Backing up original conf file
cp /etc/clamd.d/scan.conf /root/backup_conf/original/
cp /etc/freshclam.conf /root/backup_conf/original/

# Setting up home conf
sed -i -e "s/^Example/#Example/" /etc/clamd.d/scan.conf
sed -i -e "s/^Example/#Example/" /etc/freshclam.conf
sed -i -e "s/^User/#User/" /etc/clamd.d/scan.conf
echo "User root" >> /etc/clamd.d/scan.conf
echo "LocalSocket /var/run/clamd.scan/clamd.sock" >> /etc/clamd.d/scan.conf

# Backing up home conf file
cp /etc/clamd.d/scan.conf /root/backup_conf/home/
cp /etc/freshclam.conf /root/backup_conf/home/

# Enable and restart
freshclam
systemctl enable clamd@scan
systemctl restart clamd@scan

# Adding cron job for updates and scan
# DB update twice a day, AV update and sys scan everynight
crontab -l | { cat; echo "00 01,13 * * *  /usr/bin/freshclam --quiet";  } | crontab -
```

```
crontab -l | { cat; echo "0 3 * * * /bin/freshclam ; /bin/clamscan / --recursive=yes -i> /tmp/clamav.log ; mail -s clamav_log_`hostname` helpdesk@agix.local< /tmp/clamav.log"; } | crontab –

# EOF
echo "EOF - ClamAV"
```

Les deux points intéressant dans ce script sont dans l'utilisation de la commande « sed » qui encore une fois nous permet de commenter les partie de la configuration initiale non désiré pour le projet mais aussi et surtout dans l'utilisation de la commande crontab.

```
$: crontab -l | { cat; echo "<cronstring>";  } | crontab –
```

- Crontab –l : Cette partie de la commande va lister les jobs cron déjà présent
- | : va passer le resultat de crontab –l à la commande suivante
- { } : vont permettre d'encapsuler la commande « cat » ainsi que l'echo
- Cat ; echo «cronjob » ; : va permettre de concatener la liste de job actuelle avec le job que nous voulons ajouter.
- Crontab - : va ajouter le resultat de notre cron customiser dans les jobs cron

Cette méthode permet d'ajouter à cron sans effacer les jobs précédents. Cela nous sera pratique lorsque nous devrons rajouter via un autres script des cronjob de backup de fichier.

5.2. Utilisation

Pour utiliser notrez antivirus, il nous suffit d'apeller la commande « clamscan » accompagner d'eventuelles options (-- infected – remove –recursive) qui va scanner notre système en recherche de fichier infecter.

6. DNS

Le service DNS fournis par « named » permettra à notre serveur de devenir serveur DNS pour les autres machines du réseau. Ce service utilise le port 53 que nous devrons configurer aussi dans notre firewall.

Le script de ce service sera plus conséquent que les précédents car il requiert la création complète de certains fichiers.

6.1. Script:

```
#!/bin/sh
# Install
yum install bind bind-utils -y

# Backing up original conf file
cp /etc/named.conf /root/backup_conf/original/
cp /etc/resolv.conf /root/backup_conf/original/

# Setting up home conf
# First the /etc/named.conf
echo "options {
	listen-on port 53 { 127.0.0.1; 192.168.1.42; };
	listen-on-v6 port 53	{ ::1; };
	directory		\"/var/named\";
	dump-file		\"/var/named/data/cache_dump.db\";
	statistics-file		\"/var/named/data/named_stats.txt\";
	memstatistics-file		\"/var/named/data/named_mem_stats.txt\";
	recursing-file		\"/var/named/data/named.recursing\";
	secroots-file		\"/var/named/data/named.secroots\";
	    allow-query		{ localhost; 192.168.1.0/24; };
	forwarders		{ 8.8.8.8; 8.8.4.4; };

	recursion yes;

	dnssec-enable yes;
	dnssec-validation yes;

	bindkeys-file \"/etc/named.iscdlv.key\";
```

```
	managed-keys-directory \"/var/named/dynamic\";

	pid-file \"/run/named/named.pid\";
	session-keyfile \"/run/named/session.key\";
};

logging {
	channel default_debug {
		file \"data/named.run\";
		severity dynamic;
	};
};

zone \".\" IN {
	type hint;
	file \"named.ca\";
};

zone \"projetlinux.local\" IN {
	type master;
	file \"forward.projetlinux\";
	allow-update { none; };
};

zone \"1.168.192.in-addr.arpa\" IN {
	type master;
	file \"reverse.projetlinux\";
	allow-update { none; };
};

include \"/etc/named.rfc1912.zones\";
include \"/etc/named.root.key\";" > /etc/named.conf

# Second the /etc/named/forward.projetlinux
echo "\$TTL 1D
@	IN SOA	masterdns.projetlinux.local. root.projetlinux.local. (
					504	; serial
					1D	; refresh
					1H	; retry
```

```
                1W      ; expire
                3H )    ; minimum

@       IN      NS      masterdns.projetlinux.local.
@       IN      A       192.168.1.42
masterdns   IN      A       192.168.1.42
www.toto        CNAME   projetlinux.heh.be.
www.tata        CNAME   projetlinux.heh.be." > /var/named/forward.projetlinux

# Third, the reverse /var/named/reverse.projetlinux
echo "\$TTL 1D
@       IN SOA  masterdns.projetlinux.local. root.projetlinux.local. (
                504     ; serial
                1D      ; refresh
                1H      ; retry
                1W      ; expire
                3H )    ; minimum

@       IN      NS      masterdns.projetlinux.local.
@       IN      A       192.168.1.42
@       IN      PTR     projetlinux.local.
masterdns   IN      A       192.168.1.42
42      IN      PTR     masterdns.projetlinux.local." > /var/named/reverse.projetlinux

# Fourth, resolv.conf and sysconfig
echo "nameserver=192.168.1.42" >> /etc/resolv.conf
# Commenting original DNS
sed -i -e "s/^DNS1=8.8.8.8/#DNS1=8.8.8.8/" /etc/sysconfig/network-scripts/ifcfg-ens33
# Uncomment local DNS (that we preped in earlier script)
sed -i -e "s/#DNS1=192.168.1.42/DNS1=192.168.1.42/" /etc/sysconfig/network-scripts/ifcfg-
ens33

# Backing up home conf file
cp /etc/named.conf /root/backup_conf/home/
cp /var/named/forward.projetlinux /root/backup_conf/home/
cp /var/named/reverse.projetlinux /root/backup_conf/home/
cp /etc/resolv.conf /root/backup_conf/home/
cp /etc/sysconfig/network-scripts/ifcfg-ens33 /root/backup_conf/home/ifcfg-
ens33.dnsready
```

```
# Enable and restart everything
systemctl enable named
systemctl restart named
systemctl restart network

# EOF
echo "EOF - Bind"
# EOF
echo "EOF - SSH"
```

Pour la configuration de ce service, le script doit complètement configurer la fichier named.conf. Par la suite il lui faut également crée les fichiers de zone.

Il est à noter que la commande « sed » se trouve encore une fois bien utile puisque lors de notre configuration d'ip static de base de la machine nous avons ajouté le DNS de google afin de conserver une connexion stable pour l'installation des services.

Lors de l'installation, le script va donc aller commenter notre précédente configuration DNS et décommenter la ligne DNS du fihcier qui fput ajouter en prévision de cette tahces.

7. FTP

Le service FTP fournis par VSFTPD nous permettra de configurer le service de transfert de fichier pour permettre au utilisateur d'acceder à leurs fichier à distance. Ce service utilise comme SSH le port 22.

7.1. Script:

```
#!/bin/sh

# Install
yum install ftp vsftpd -y

# Backing up original conf file
cp /etc/vsftpd/vsftpd.conf /root/backup_conf/original/
touch /etc/vsftpd/vsftpd.userlist

# Generating RSA key
mkdir -p /etc/vsftpd/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout /etc/vsftpd/ssl/vsftpd.pem -
out /etc/vsftpd/ssl/vsftpd.pem

# Setting up home conf
echo "anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
xferlog_std_format=YES
connect_from_port_20=YES
listen=YES
listen_ipv6=NO
pam_service_name=vsftpd
tcp_wrappers=NO
check_shell=NO
ls_recurse_enable=YES

userlist_enable=YES
userlist_deny=NO
```

```
userlist_file=/etc/vsftpd/vsftpd.userlist

chroot_local_user=YES
user_sub_token=$USER
local_root=/home/$USER/ftp

pasv_enable=YES
pasv_min_port=50001
pasv_max_port=50010

ssl_enable=YES
allow_anon_ssl=NO
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO

require_ssl_reuse=YES
ssl_ciphers=HIGH

rsa_cert_file=/etc/vsftpd/ssl/vsftpd.pem
rsa_private_key_file=/etc/vsftpd/ssl/vsftpd.pem" > /etc/vsftpd/vsftpd.conf

# Backing up home conf file
cp /etc/vsftpd/vsftpd.conf /root/backup_conf/home/
cp /etc/vsftpd/vsftpd.userlist /root/backup_conf/home/

# Restart and enable
systemctl enable vsftpd
systemctl restart vsftpd

# EOF
echo "EOF - FTP"
```

Notre service est désormais installer et operationnel. Cependant nous devons autoriser et configurer les partages en fonction des utilisateur. Pour ce faire, nous allons également scripter cette cofniguration que nous pourrons lancer pour chaque utilisateur que nous voulons autoriser en FTP.

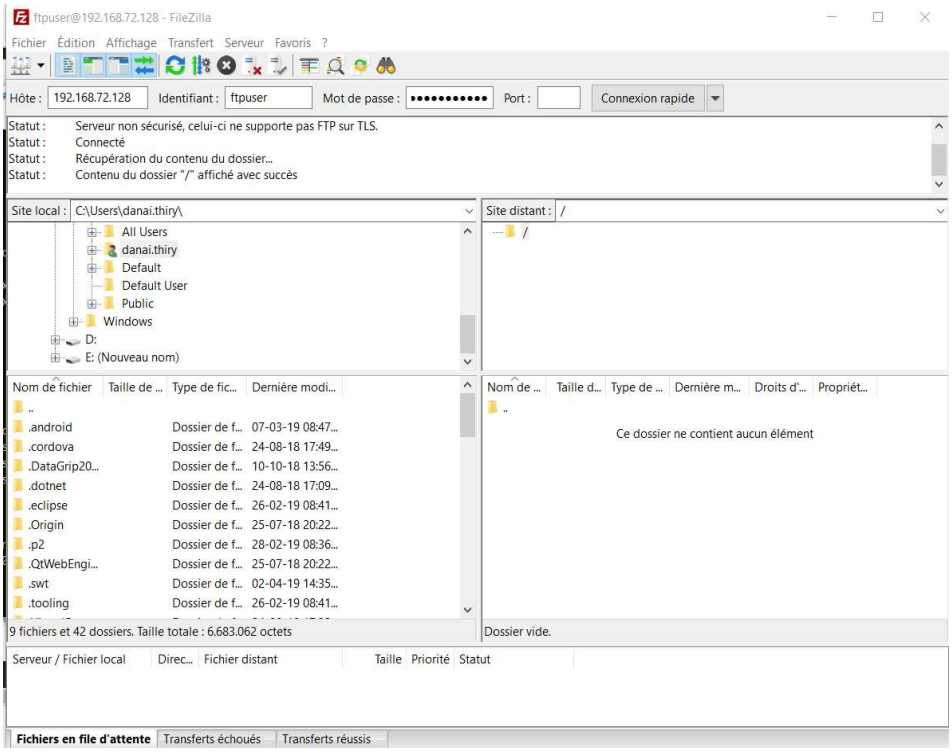
7.1. Ajout d'utilisateur:

```
#!/bin/sh
# User creation
# Will be skipped by the OS if user is already in the system
read -p "Enter username to create: " ftpuname
echo $ftpuname will be created
useradd -m $ftpuname
passwd $ftpuname
echo $ftpuname >> /etc/vsftpd/vsftpd.userlist

# Alt Local root FTP dir
mkdir /home/$ftpuname/ftp
chown nobody:nobody /home/$ftpuname/ftp
chmod a-w /home/$ftpuname/ftp

# Dir for user file
mkdir /home/$ftpuname/ftp/files
chown $ftpuname:$ftpuname  /home/$ftpuname/ftp/files
chmod 0700 /home/$ftpuname/ftp/files/

#Backing up and restart
cp /etc/vsftpd/vsftpd.userlist /root/backup_conf/home/
systemctl restart vsftpd

# EOF
echo "EOF - FTP Useradd"
```

Nous pouvons donc désormais executer ce script pour tout utilisateur ayant besoin du service FTP, ce qui ajoutera l'utilisateur dans la liste des utilisateur autoriser et créera par la même occasion les dossiers requis pour le fonctionnement du FTP.

7.2. Utilisation

Pour utiliser le service FTP nous pouvons utiliser FileZilla qui va nous permettre d'établir une connexion FTP depuis notre machines distante vers notre serveur.
Pour cela, il nous suffit de rentrer l'IP du serveur ainsi que le login et mot de passe utilisateur.



8. Samba

Samba permet de partager nos fichiers entre plusieurs machines Linux et Windows, ce qui évite d'avoir recours à des moyen de transfert externe et peux sécuriser (USB, CD, …). Samba utilise les ports UDP 137 et 138 et TCP 139 et 445 que nous devront donc ajouter dans notre firewall.

8.1. Script:

```
#!/bin/sh

# Install
yum install samba samba-client -y

# Backing up original conf file
cp /etc/samba/smb.conf /root/backup_conf/original/

# Setting up smbguest
useradd -g users -d /dev/null -s /sbin/nologin smbguest
passwd -l smbguest
smbpasswd -a smbguest -d

# Setting up home conf
echo "[Global]
        workgroup = WORKGROUP
        netbios name = srvprojetlinux
        server string = Srv Linux (VM)
        domain master = no
        invalid users = root
        encrypt passwords = yes
        guest account = smbguest
        log level = 0
        log file = /var/log/samba/log.%m
        max log size = 1000
        unix password sync = no
        security = user
        force group = users
        create mode = 0660
        directory mode = 0770
        passdb backend = tdbsam
```

```
[Public]
        path = /share/samba/public
        comment = Partage public
        public = yes
        only guest = yes
        read only = no
[Confidentiel]
        path = /share/samba/confidentiel
        comment = partage confidentiel
        read only = no
        invalid users = root nobody sbmguest" > /etc/samba/smb.conf

# Backing up home conf file
cp /etc/samba/smb.conf /root/backup_conf/home/

#Adding group for valid user
groupadd smbgrp

# Creating share folders
mkdir -pv -m 1777 /share/samba/{public,confidentiel}
chown -R nobody:nobody /share/samba/public
chown -R root:smbgrp /share/samba/confidentiel

# Enable and restart everyhting
systemctl enable smb
systemctl enable nmb
systemctl restart smb
systemctl restart nmb

# EOF
echo "EOF - SMB"
```

Le service installer et configurer, nous pouvons désormé utiliser samba. Sauf que, tout comme FTP, samba requiert également d'autoriser les utilisateurs au préalable.

8.2. Ajout d'utilisateur:

```
#!/bin/sh
# User creation
```

```
read -p "Enter username to create: " smbuname
echo $smbuname will be created
useradd -m $smbuname
passwd $smbuname
usermod -a -G smbgrp $smbuname
smbpasswd -a $smbuname -d

# EOF
echo "EOF - SMB Useradd”
```

Nous pouvons donc automatiser via ce script l’ajout des utilisateurs dans le groupe smbgrp afin de les autoriser à utiliser notre serveur samba.

9. MySQL

Le service MySQL une fois installer permettra à nos utilisateur d'administrer des base de données. Nous installeront d'ailleurs par la suite le service Apache et PHP qui viendront emplifier le sutilisation de MySQL.

9.1. Script:

```
#!/bin/sh

# Install
wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
rpm -ivh mysql-community-release-el7-5.noarch.rpm
yum update -y
yum install mysql-server -y
systemctl start mysqld

# Run the following to secure installation and follow the steps:
# Hit enter when asked for root mysql password
# Set root mysql password
# Remove anonymous? yes
# Disable root remote login? yes
# Remove test db? yes
# Reload tables? yes
mysql_secure_installation

# Enable and restart everyhting
systemctl restart mysqld
systemctl enable mysqld

# EOF
echo "EOF - MYSQL"
```

Malheureusement, ce script ne permet pas une automatisation total puisque la commande mysql_secure_installation va devoir demander à l'administrateur de faire quelques étapes manuellement.

Ces étapes sont renseigné dans les commentaires du script avec les parametre utiliser lors du projet.

10. Apache

Pour pouvoir activer les service Web et pouvoir crée nos propres sites, nous allons installer Apache sur la machine. Il posera la base du serveur Web sur laquel nous pourrons ajouter PHP afin de dynamiser nos sites.
Nous allons aussi le configurer afin que chaque utilisateur puisse crée son propre site.

10.1. Script:

```
#!/bin/sh

# Install
yum install httpd -y
systemctl enable httpd

# Backing up original conf file
# /etc/httpd/conf/httpd.conf is not used but backup just in case
cp /etc/httpd/conf/httpd.conf /root/backup_conf/original/
cp /etc/httpd/conf.d/userdir.conf /root/backup_conf/original/

# Setting up home conf
echo "<IfModule mod_userdir.c>
        UserDir enabled webuser
        UserDir public_html
</IfModule>
<Directory /home/*/public_html>
        Options Indexes Includes FollowSymlinks
        Require all granted
</Directory>" > /etc/httpd/conf.d/userdir.conf

# Backing up home conf file
cp /etc/httpd/conf/httpd.conf /root/backup_conf/home/

# Enable and restart everyhting
systemctl reload httpd
systemctl restart httpd

# EOF
echo "EOF - HTTPD"
```

La configuration d'Apache repose ici sur la directive :

```
<IfModule mod_userdir.c>
        UserDir enabled webuser
        UserDir public_html
</IfModule>
<Directory /home/*/public_html>
        Options Indexes Includes FollowSymlinks
        Require all granted
</Directory>
```

Cela à pour effet de permettre via notre second script d'ajouter des sites web en fonction des utilisateurs que nous ajoutons via le script suivant :

10.2. Ajout d'utilisateur:

```
#!/bin/sh

# User creation
read -p "Enter username to create: " webuname
echo $webuname will be created
useradd -s /sbin/nologin $webuname
passwd $webuname
usermod -a -G apache $webuname

# Basic userpage
mkdir -p /home/$webuname/public_html
chmod 711 /home/$webuname
chown $webuname:$webuname /home/$webuname/public_html
chmod 775 -R /home/$webuname/public_html
echo "Welcome to $webuname HTML web page." >
/home/$webuname/public_html/index.html
echo "Welcome to $webuname PHP web page.
<?php phpinfo(); ?>" > /home/$webuname/public_html/index.php

echo "You can now acces $webuname webpage in your browser"
echo "Just use: http://[thiserverip]/~$webuname"

# EOF
echo "EOF - HTTPD Useradd"
```

Lors de l'execution, ce script créera l'utilisateur voulus et preparera son architecture web dans sa propre home en y placant d'ailleurs deux fichiers index.html et index.php basique.

11. PHP

PHP va nous permettre de rendre nos sites dynamique par le biais des bases de données MySQL. Il s'agit ici d'une surcouche à apache et son service est donc le même (httpd).

11.1. Script:

```
#!/bin/sh

# Install
yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm -y
yum-config-manager --enable remi-php70   [Install PHP 7.0]
yum install php php-mcrypt php-cli php-gd php-curl php-mysql php-ldap php-zip php-fileinfo
-y
yum install php php-pear -y

# Enable and restart everyhting
systemctl reload httpd
systemctl restart httpd

# EOF
echo "EOF - PHP"
```

L'installation de PHP ne requierant pas de configuration additionnel le script est plus courts que les autres. Néanmoins il est a noter que PHP n'est plus disponible sur les repo de base de CentOS et qu'il à donc fallu l'ajouter en tête de script, faute de quoi l'installation se serait solder par un echec.

12. Firewall

Par simplicité nous avions désactiver completement le Firewall en introduction du projet. Nos services etant maintenant tous opérationnel et fonctionnel nous pouvons désormais ajuster nos parametre et le redemmarer afin de valider que tout nos service sont bien autorisé et tjrs accessible.

12.1. Script:

```
#!/bin/sh

# empty all
iptables -F
iptables -X

# block everything by default
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# add loopback
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# add ICMP
iptables -A INPUT -p icmp -j ACCEPT
iptables -A OUTPUT -p icmp -j ACCEPT

# add SSH
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT

# add DNS
iptables -A INPUT -p tcp --dport 53 -j ACCEPT
iptables -A INPUT -p udp --dport 53 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 53 -j ACCEPT
iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
```

```
# add FTP
modprobe ip_conntrack_ftp
firewall-cmd --permanent --add-port=21/tcp
chkconfig vsftpd on

# add SMB
firewall-cmd --permanent --zone=public --add-service=samba
firewall-cmd --reload

# add HTTP
#firewall-cmd --add-service=http --permanent
#firewall-cmd --add-service=https --permanent

#Restart and reload everything
systemctl restart firewalld

# EOF
echo "EOF – Firewall”
```

13. Backup

Pour les backups, nous utiliseront cron afin d'exécuter notre script de backup qui effectuera un backup de tout les fichiers cibles.

13.1. Script:

```
#!/bin/sh

# Adding cron job for backups
crontab -l | { cat; echo "0 2 * * * /bin/sh /root/script/backup";  } | crontab -

# EOF
echo "EOF – Backup"
```

Nous lanceront l'exécution du script ci-dessous, chaque jour à 2h.

13.1. Backup:

```
#!/bin/sh

cp -avr /etc /backup
cp -avr /var /backup
cp -avr /home /backup
cp -avr /root /backup

# EOF
echo "EOF - BackupScript"
```

Notre script effectuera des copie des dossier voulus à l'emplacement cible.

14. Procédure de test

Pour tester chaque service :

- SSH : Utiliser PuTTY pour se connecter à la machine.
- NTP : utiliser les commandes :
  - ntpq –p
  - date –R
  - ntpdate -q IPDUSRV
- ClamAV : Recuperer un faux fichier infecter et scanner
  - wget http://www.eicar.org/download/eicar_com.zip
  - clamscan --infected --remove --recursive
- DNS:
  - dig masterdns.projetlinux.local
  - nslookup www.google.be
- FTP:
  - Connection via FileZilla
- SMB
  - Connecter vous depuis un client windows
- Apache : Dans un navigateur, navigué à l'adresse
  - IPDUSRV/~username/index.html
- PHP : Dans un navigateur, navigué à l'adresse
  - IPDUSRV/~username/index.php
- Firewall
  - iptables -L INPUT
  - iptables -L OUTPUT
- Cron
  - Crontab -l

15. Problèmes rencontrés

J'ai personellement rencontrer pas mal de soucis lors de ce projet :

- L'installation minimal qui n'avais pas toutes les commandes d'administration et qui donc requierais un surprlus de travail ce qui ma pousser à redemmarer le projet sur une autres version de CentOS
- Devoir prevoir le DNS de google comme base dans la configuration static sans quoi l'installationd 'autre service fut impossible
- L'ajout d'un yum clean pour l'antivirus sans quoi l'installation était impossible
- L'ajout du ssl_tls pour le FTP qui a detraquer mon service
- Les moults problemes de permission lors des création des scripts utilisateur ou il a fallu les adapter sur base d'essaie muliple.
- L'ajout de la commande mysql_secure_installation pour laquel je n'ai trouvé aucune documentation en ligne et dont la solution ma été transmise par un collegue
- La configuration du firewall que j'effectuais au fur et a mesure et qui ne marchais pas, ce qui est la raison pour laquel mon script de firewall contient deux type de commande différente en fonction des services.
- L'ajout des services web que j'ai du recommencer 3 fois chacun puisque la configuration se faisant initialement sans base d'utilisateur
- Les mauvais pathname dans le script d'utilisateur web (je tentais de rendre le path plus propre, mais ce fut choe impossible)
- J'ai egalement du trouver une solution pour la partie PHP qui refusais de fonctionner. La solution était l'installation de php-pear

J'ai egalement appliqué une politique strict pour la validation de mes scripts. Dés que la configuration de base de ma machine fut terminer, j'ai preparer tout mes scripts dans un dossier itnerne à la vm et réaliser un snapshot.
Chaque script fut ensuite éxecuter un a un et scrupuleusement analyser et tester afin de valider leurs bon fonctionnement.

Malgrer cela, apres avoir retenter une nouvelle VM et alors que tout fonctionnais précedement, certains services refusaient de fonctionner avec une configuration pourtant exactement identique. C'est d'ailleurs toujours le cas du service Samba qui même si il est bien installer et configurer refuse de fonctionner.

16. Conclusion

Ce projet bien que complexe de prime abord, s'avère au final être une bonne mise en situation du monde professionnelle.

Certaines procédures bien définies peuvent souvent être considérée comme acquise mais le projet, de part le nombre de services simultanés sur la même machine, force son administrateur à revoir et vérifier sa configuration afin de s'assurer que la machines répond aux exigences demandées.

Ma méthodes de test, bien que stricte, m'a aussi demontrer que malgrer toutes les autmatisation du monde, une machine sera tjrs différente de la précedente et devra recevoir un traitement en conséquences. En effet, bien que mes scripts fonctionne tous, chaque machines que j'ai configurer (4 au totals) a presenter des erreurs à chaque fois à different endroits des scripts.

Aucune des erreurs n'était similaire à la précedente et il ma donc fallu repenser plusieurs mes scripts afin de les rendre les plus universelles possible.

17. Sources

https://tecadmin.net/crontab-in-linux-with-20-examples-of-cron-schedule/
https://www.microlinux.fr/bind-centos-7/
https://linux-note.com/centos-7-serveur-dns-local/
http://denisrosenkranz.com/tuto-mettre-en-place-un-serveur-web-sous-centos-apache-mysql-php-vsftpd/
https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-centos-7
https://blog.microlinux.fr/samba-centos/
https://www.howtoforge.com/samba-server-installation-and-configuration-on-centos-7
https://www.microlinux.fr/centos-7-ntp/
https://www.microlinux.fr/client-ntp-centos-7/
https://www.tecmint.com/install-ftp-server-in-centos-7/

18. Annexes

En plus de tout les scripts de configuration, j'ai également réaliser un script me permettant de tester rapidement sou linux que tout mes services sont opérationnel.
Ce script effectue de simple commande et en affiche le résultat pour consultation.

18.1. Master Test:

```
#!/bin/sh

#Random config test
echo "----------------------------------------------SELinux"
getenforce
echo "----------------------------------------------Ifconfig"
ifconfig ens33 |grep inet
echo "------------------------------------------------NTP"
date -R
ntpq -p
echo "----------------------------------------------ClamAV"
crontab -l
wget http://www.eicar.org/download/eicar_com.zip
clamscan --infected --remove --recursive
echo "------------------------------------------------BIND"
dig masterdns.projetlinux.local
nslookup www.google.be
echo "------------------------------------------------SMB"
testparm
smbclient -L localhost -N
pdbedit -L
echo "------------------------------------------------PHP"
php -v
echo "------------------------------------------------CRON"
crontab -l
echo "----------------------------Firewall-INPUTandOUTPUT"
iptables -L INPUT
iptables -L OUTPUT
echo "-------------------------Configuration-Backup-Folder"
ll /root/backup_conf/home

#Process test
```

```
echo "-----------------------------------------------Process"
echo " SSH: systemctl status sshd |grep Active"
systemctl status sshd |grep Active
echo " NTP: systemctl status ntpd |grep Active"
systemctl status ntpd |grep Active
echo " ClamAV: systemctl status clamd@scan |grep Active"
systemctl status clamd@scan |grep Active
echo " BIND: systemctl status named |grep Active"
systemctl status named |grep Active
echo " VSFTPD: systemctl status vsftpd |grep Active"
systemctl status vsftpd |grep Active
echo " SMB: systemctl status smb|grep Active"
systemctl status smb|grep Active
echo " NMB: systemctl status nmb|grep Active"
systemctl status nmb|grep Active
echo " MYSQLD: systemctl status mysqld |grep Active"
systemctl status mysqld |grep Active
echo " HTTPD: systemctl status httpd |grep Active"
systemctl status httpd |grep Active
echo " Firewalld: systemctl status firewalld |grep Active"
systemctl status firewalld |grep Active
echo "-----------------------------------------------EOF"
```