

Projet Linux

FOFANAH Mankoulani VANGEEBERGEN Augustin

May 16, 2024



Table des matières

1	Introduction	3
1.1	Consignes professeur	4
1.2	Roadmap	5
1.3	Répartition des tâches	6
2	Code	6
2.1	Intro au Bash	6
2.2	Menu TUI	7
2.3	Partage SAMBA	7
2.4	Partage NFS	7
2.5	Serveur Web	8
2.6	FTP server	10
2.7	Backup	11
2.8	Partitionnement	12
3	Conclusion	13

1 Introduction

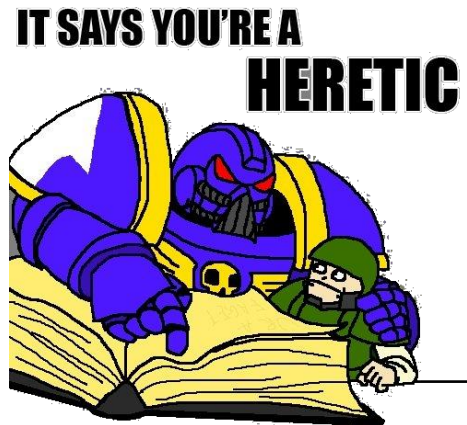
Dans le cadre de ce projet, nous avons pour objectif de configurer un serveur GNU/Linux. Nous avons le choix d'utiliser n'importe quelle distributon RedHat-like, par exemple Fedora, ou bien Alma, sur laquelle nous avons travaillé en cours. Ce point est important, pour pouvoir poser une question précise au professeur en cas de souci.

Notre choix se porte donc sur Alma Linux, mais les scripts fonctionnant sous Alma fonctionnent théoriquement exactement de la même manière sous Fedora Linux.

L'OS fonctionnera dans une machine virtuelle, hébergée sur un HYPER-V de Windows. De cette manière, aussi bien un utilisateur Windows que linux pourra accéder aux différents services.

Le but est de pouvoir gérer en ssh un serveur avec plein d'outils, et de pouvoir installer/désinstaller les services à souhait, diminuant ainsi la surface d'attaque.

Me when Linux on Windows :



1.1 Consignes professeur

Les consignes sont les suivantes :

- Le serveur devra contenir un partage NFS qui permettra aux utilisateurs du réseaux local d'y stocker des fichiers.
- Un partage Samba permettra aux utilisateurs Windows d'accéder à ce même partage.
- Il faudra mettre en place un serveur Web, FTP, MySQL et DNS qui permettra un hébergement multiutilisateurs. Le FTP permettra à chaque utilisateur d'accéder à son dossier Web. Il faudra créer une zone dans le DNS pour nos sites. (Automatisation)
- Le DNS fera également office de DNS cache pour le réseau local.
- Vous mettrez en place un serveur de temps pour que les machines du réseau local puissent se synchroniser.
- Vous devrez pouvoir vous connecter en SSH au serveur et y effectuer des configurations. (sécurité !!!)
- Pour la sécurité :
 - politique utilisateur
 - quotas
 - partitionnement et gestion disque (LVM et raid)
 - Gestion backup(ou, comment,...)
 - Maj, désactiver l'inutile.
 - Antivirus, firewall,

- Vous êtes l'administrateur de ce serveur agissez en conséquence :
 - Documenter
 - Automatiser (script)
 - Soyez proactif
 - Sécurisez
 - Tester
- Le travail sera réalisé par groupe de deux et évalué par moi pendant la session de juin
- Il faudra une machine cliente pour effectuer les tests (Windows et Linux)
- Vous devrez me démontrer le fonctionnement des services en 15 minutes. Donc connaissez votre machine et vos configurations sur le bout des doigts. Vous aurez 10 minutes pour installer le tout.
- L'évaluation comptera pour 100% des points.

1.2 Roadmap

L'ensemble sera scripté pour coller à l'ensemble des cas d'utilisation. Voici donc la liste prévue de ces scripts (pour installer/désinstaller):

- Menu de sélection
- SSH config wizard
- File sharing install wizard
- Web server install wizard
- FTP server install wizard
- MySQL server
- DNS server (+ cache + création de zone + serveur cache)
- Time server
- Sécurisation
- Backup
- Updates
- Partitionnement

Il est donc indispensable de se former au BASH, afin de savoir faire un bon TUI, ainsi que des commandes conditionnelles, selon les features qui sont/ne sont pas déjà installées.

1.3 Répartition des tâches

- Augustin
 - Rapport (en LaTeX, bien sûr)
 - Menu TUI (pas jetair)
- Mankou
 -

2 Code

2.1 Intro au Bash

Comme nous devons apprendre le Bash pour les scripts, voici une petite synthèse simplifiée.

Un code Bash doit toujours commencer par :

```
#!/bin/bash
echo "Hello , - world!"
```

Une variable va se déclarer comme ceci :

```
name="John"
age=30
```

On accède à sa valeur comme ceci :

```
# Access and print variables
echo "Name: - $name"
echo "Age: - $age"

# You can also use the curly braces syntax
echo "Name: - ${name}"
echo "Age: - ${age}"
```

Une condition va s'écrire comme ceci :

```
if [ condition ]; then
# Commandes exécutées si condition vraie
else
# Commandes exécutées si condition fausse
fi
```

2.2 Menu TUI

Le menu principal en TUI est un simple menu permettant de choisir le script concernant une manipulation spécifique d'un ou plusieurs services destiné(s) au serveur. Il affiche les options dans un ordre prédéfini, sélectionnables par un nombre à partir de 0.

2.3 Partage SAMBA

Premièrement le script va installer le package "samba", si ce n'est pas déjà fait. Il va ensuite donner le choix à l'utilisateur pour gérer son partage samba.

(afin de voir quels sont les utilisateurs existants, ainsi que les groupes et directories associés au(x) partage(s).) A noter que les utilisateurs samba ont par défaut besoin d'être associés à un utilisateur UNIX, même si les deux bases de données sont bien distinctes. Il va nous falloir un menu pour pouvoir éditer :

- Les utilisateurs :
 - ✓ Lister
 - ✓ utilisateurs UNIX
 - ✓ utilisateurs samba
 - ✓ Ajouter
 - ✓ utilisateur UNIX
 - ✓ utilisateur samba
 - Retirer
 - * utilisateur UNIX
 - ✓ utilisateur samba
 - ✓ tous les utilisateurs samba (à part root)
 - Désactiver utilisateur samba
 - Activer utilisateur samba
 - Changer le mot de passe

2.4 Partage NFS

Concu pour partager des fichiers entre OS de type Unix,

Montage d'un FS samba sous Unix : `mount -t smbfs -o` (voir annexes du cours de linux P78)

2.5 Serveur Web

Le déploiement d'un serveur web sous alma (ou fedora) est décrit dans cet article. Deux subdirectories sont utiles pour la configuration :

- `/etc/httpd/conf.d`
Pour stocker la configuration des différents sites web
- `/etc/httpd/conf.modules.d`
Pour les modules chargés dynamiquement

Historiquement, les données du site web sont par défaut stockées dans :

- `/var/www/`

Cependant, pour plusieurs sites, il existe deux méthodes.

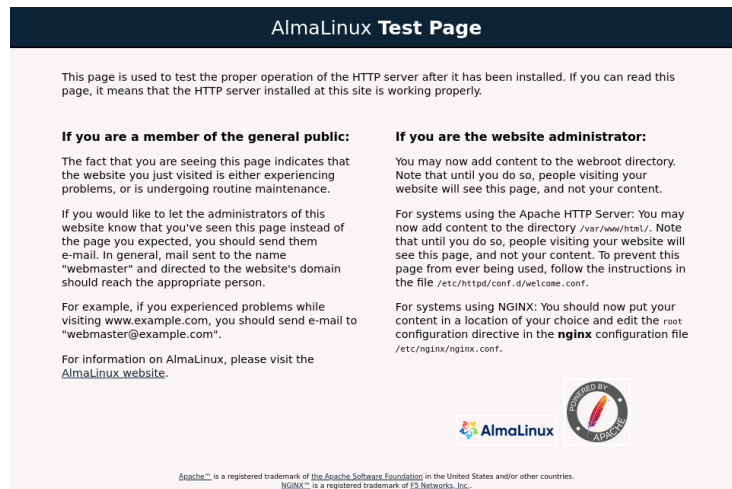
- utiliser le directory `/var/www/` et stocker les sites dans des subdirectories (facile pour SELinux, peu orthodoxe car modifie la configuration de base)
- utiliser le directory `/srv` et stocker les sites dans des subdirectories avec dans ceux-ci :
 - `htdocs`
 - `webapps`
 - `mail`
 - ...

Nous utiliserons donc :

- `/srv/<DOMAINNAME>/` pour stocker les données relatives au domaine
- `/srv/<DOMAINNAME>/htdocs/` pour les pages html statiques

!! A compléter pour le setup des LVM !!

Il faut ensuite installer le package `httpd`. Le manuel en ligne conseille d'installer les packages pour la gestion ssl et pour le monitoring de domaine. Il suffit ensuite de démarrer le service `httpd` et de l'enable avec `systemctl`. La page d'accueil par défaut ressemble à ceci sur AlmaLinux :



Le menu de selection contient donc :

- Install web server
- Show httpd status
- Create web dir for user
- Remove web directory of user
- Display web directories

2.6 FTP server

Chaque utilisateur spécifié doit être en mesure d'utiliser le FTP pour accéder :

- à son dossier root
- à son dossier web

Le service choisi est vsftpd (Very Secure FTP Daemon). Il est le plus répandu au sein des distributions RedHat-like, peu gourmand, stable et sécurisé.

L'installation est similaire celle du serveur web. Par conséquent, il faudra en premier installer le service, le démarrer et puis ensuite le configurer.

Le fichier de configuration de vsftpd est `/etc/vsftpd/vsftpd.conf`.

Le menu présente diverses options :

- Install and enable ftp server
- Start ftp server
- Stop ftp server
- Enable ftp server
- Disable ftp server
- Show ftp server status
- Directory attribution for users :
 - enable srv for all users
 - enable home for all users
 - disable srv for all users
 - disable home for all users
 - enable srv for the specified user
 - enable home for the specified user

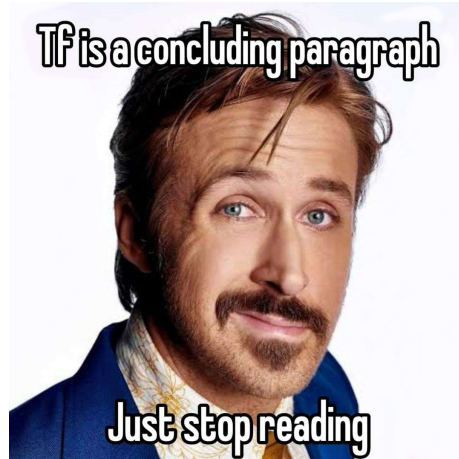
2.7 Backup

Le menu Backup doit comporter deux options :

- Backup
- Restore

2.8 Partitionnement

3 Conclusion



References

- [1] Author, A. (Year). Title of the article. *Journal Name*, Volume(Issue), Pages.