INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

Projet 4A STI : Supervision et audit de la sécurité système dans un réseau

Diplôme d'Ingénieur, 4e année

Aymeric Berquin
&
Fayçal-Anoar Cherkaoui

Date de rendu de rapport : 10/02/2015

Remerciements

Avant d'entamer ce sujet nous saisissons la présence pour exprimer les remerciements les plus sincères et nos profonds respects à notre encadrant M. Jeremy Briffaut, pour son encadrement, son assistance et son soutien considérables qui ont pu rendre ce travail possible. Ainsi que Nous le remercions aussi pour toutes les connaissances qu'il nous a apportés.

Nous aimerions aussi gratifier M. Martial Szpieg et M. Pascal Berthome pour leur amabilité de nous avoir fourni les explications nécessaires, et les conseils pertinents qui nous ont accompagnés tout au long du projet.

Introduction

Dans le cadre de notre formation d'ingénieurs en Sécurité et Technologies Informatiques, un projet d'application sécurité nous est soumis. Dans notre cas il s'agit de concevoir une application client/-serveur permettant la supervision et l'audit de la sécurité dans un réseau. Il s'agit de nous mettre en situation de travail en binôme sur un projet donné et sur un moyen terme. On peut utiliser l'audit d'un réseau afin de prévenir et de réparer un problème lié aux ressources informatiques dans n'importe quelconque organisme. La prévention consiste à adresser régulièrement un état des lieux afin de connaitre les faiblesses qui pourraient se traduire dans le futur en sources de menaces et vulnérabilités exploitables par les pirates. Quant à la réparation, l'entreprise peut exploiter l'audit de son réseau afin d'ameliorer les performances de ce dernier.

Dans notre projet on s'interssera aux scans afin de mettre en place un système d'audit et de sécurité. Le rapport est divisé en 3 grandes parties, la première est un tutoriel d'installation de l'environnement de travail adopté, la deuxème grande partie vient décrire la mise en place d'un dépôt Git qui jouera un moyen de communication sûr est agréable à l'échange de données, la troisème partie consiste en la mise en place d'une base de données où seront stockées les informations regroupées, et enfin la dernière partie est la partie codage des sockets qui joueront le rôle du moyen des communications entre le serveur et les clients afin d'échanger les données.

Table des matières

1 Installation des machines virtuelles **1**
- 1.1 Installation du serveur Debian . . . 1
- 1.2 Installation du client ubuntu . . . 8

2 Git **14**
- 2.1 Gérer les dépots . . . 14
- 2.2 Etat du dépot . . . 14
- 2.3 Gestion des fichiers . . . 14
- 2.4 Gestion des commits . . . 14

3 Base De Données **16**
- 3.1 Client . . . 16
- 3.2 Script . . . 16
- 3.3 Result . . . 16

4 Client/Serveur **17**
- 4.1 Client . . . 17
- 4.2 Serveur . . . 17

5 Script **18**

6 Interface web **19**
- 6.1 Installation de l'environnement de travail . . . 19

7 Conclusion **20**

8 Bibliographie **21**

Table des figures

1 choix de l'installation 1
2 choix de la langue 2
3 choix de la localisation géographique 2
4 Nom de l'hôte : hostname 2
5 Nom du domaine de la machine 3
6 Définition du mot de passe du compte root 3
7 Confirmation du mot de passe 3
8 Création d'un compte utilisateur 3
9 Choix du login du compte utilistauer précédemment créé 4
10 Définition du mot de passe pour le compte utilisateur 4
11 confirmation du mot de passe pour le compte utilisateur 4
12 Partitionnement du disque 4
13 Partitionnement du disque 4
14 Partitionnement du disque 5
15 Partitionnement du disque 5
16 Partitionnement du disque 5
17 Configuration de l'outil de gestion du paquet 6
18 Configuration de l'outil de gestion du paquet 6
19 configuration de l'outil de gestion du paquet 7
20 Sélection des logiciels 7
21 Installation du programme de démarrage GRUB 7
22 Fin de l'installation 7
23 interface de la machine virtuelle VMware 8
24 Choix de la langue d'installation 9
25 Prérequis pour l'installation 9
26 Type d'installation 10
27 Choix du fuseau horaire. 10
28 Disposition du clavier. 11
29 Création du client1. 12
30 Redémarrage de la machine. 13
31 Page d'acceuil du serveur xampp 19

1 Installation des machines virtuelles

1.1 Installation du serveur Debian

Installation d'un debian classique sans interface graphique, qui jouera le rôle du maître.
On récupère l'iso depuis le site officiel : http ://www.debian.org/.
On configure les caractéristiques suivantes :

— 1GB en RAM
— 1 processeur
— 20GB en dique dur

On choisi une installation sans interface graphique.

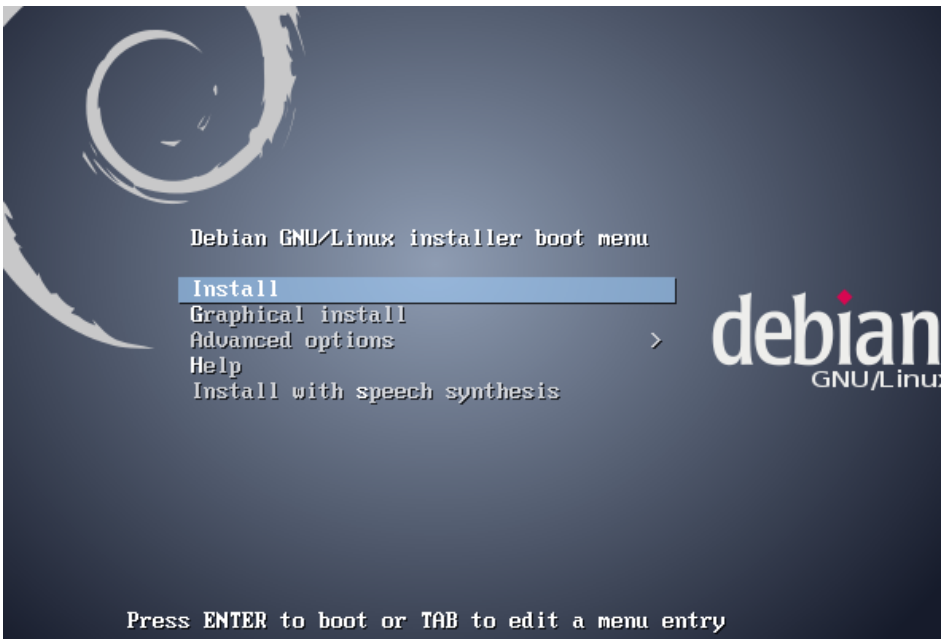

FIGURE 1 – choix de l'installation

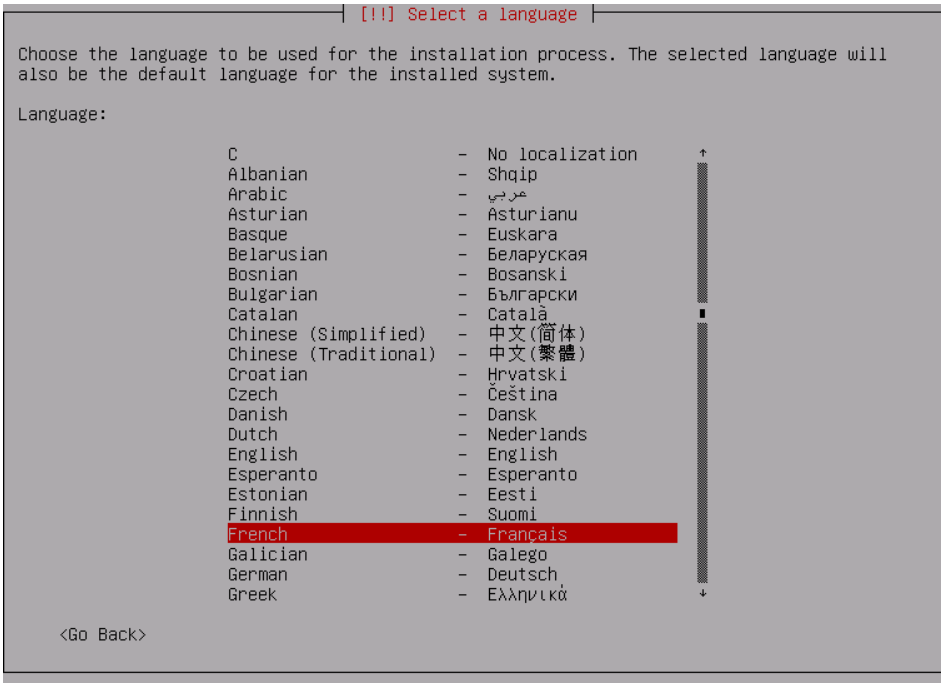

FIGURE 2 – choix de la langue

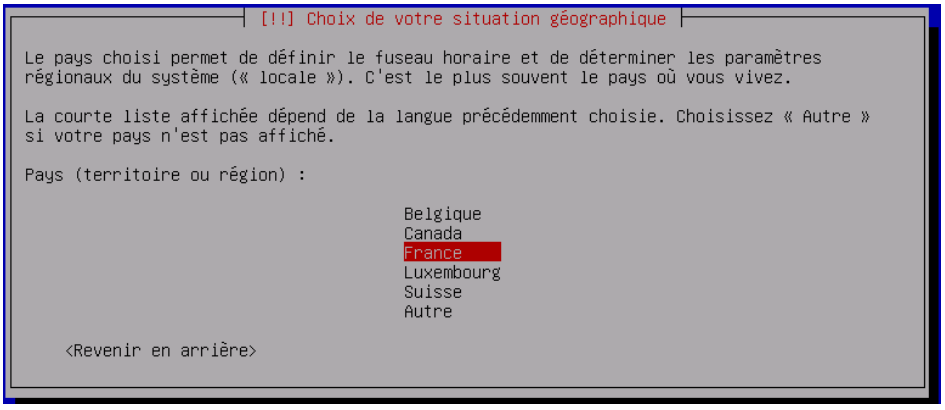

FIGURE 3 – choix de la localisation géographique

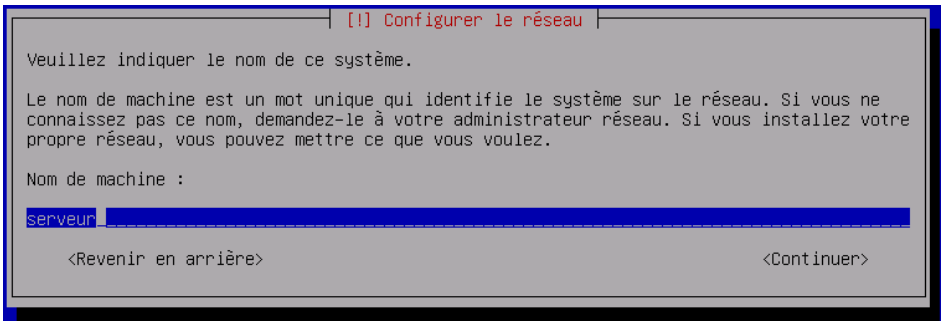

FIGURE 4 – Nom de l'hôte : hostname

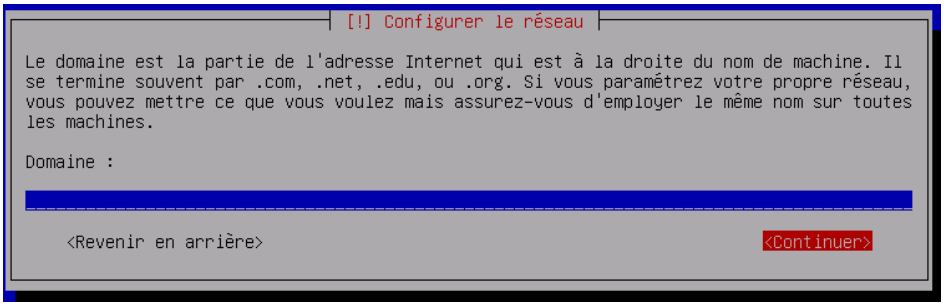

FIGURE 5 – Nom du domaine de la machine

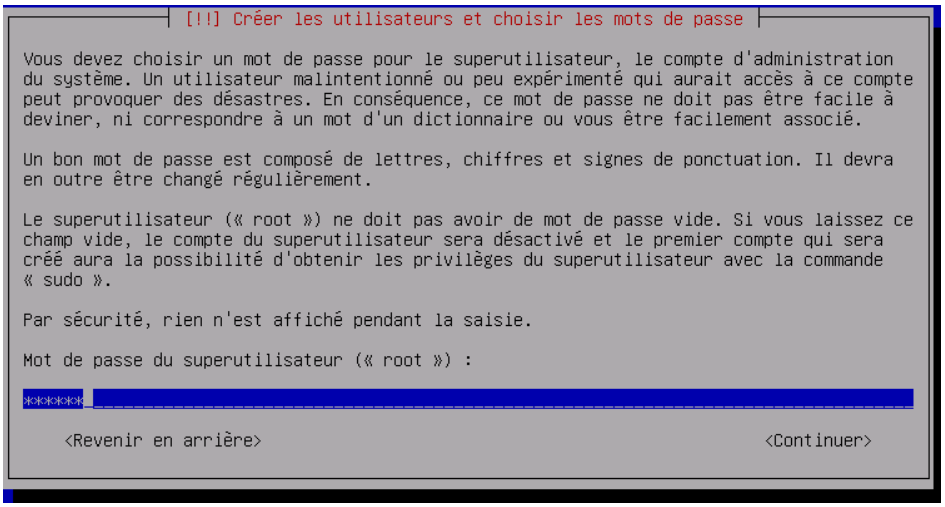

FIGURE 6 – Définition du mot de passe du compte root

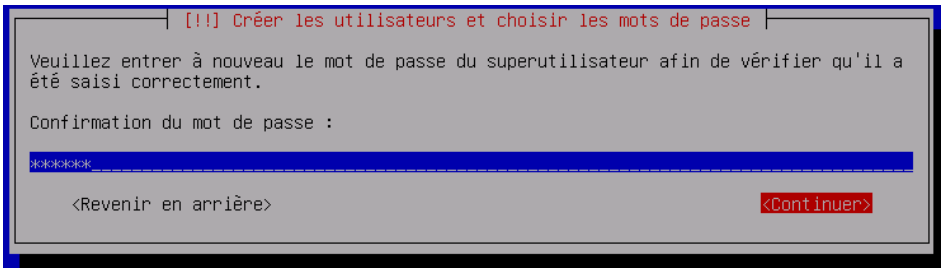

FIGURE 7 – Confirmation du mot de passe

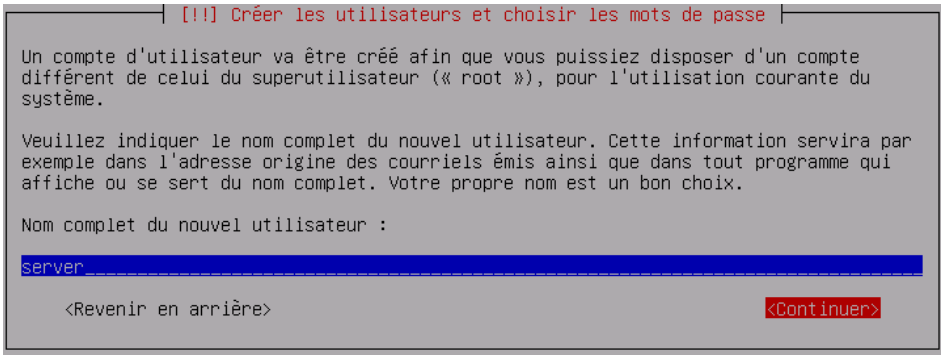

FIGURE 8 – Création d'un compte utilisateur

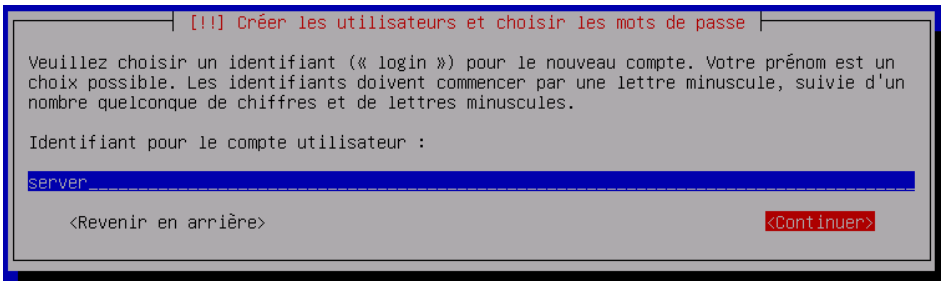

FIGURE 9 – Choix du login du compte utilistauer précédemment créé

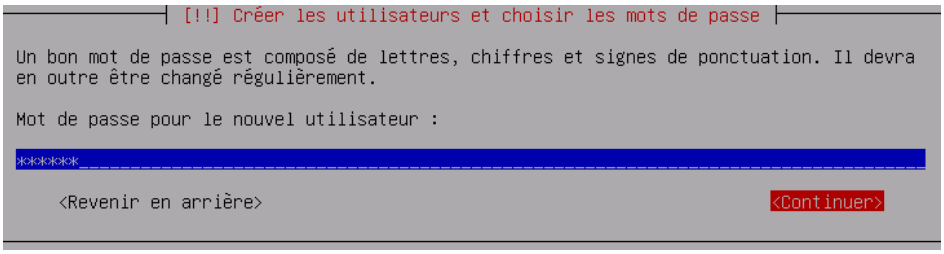

FIGURE 10 – Définition du mot de passe pour le compte utilisateur

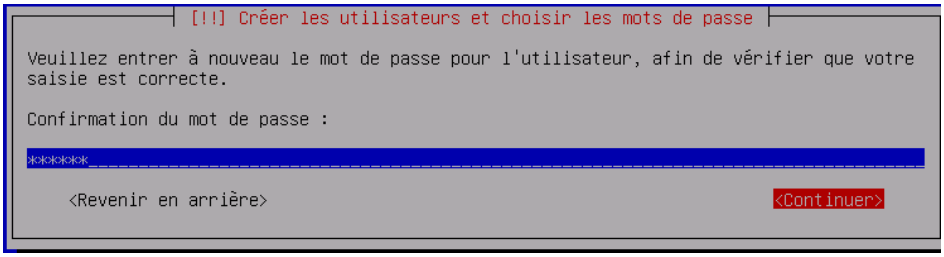

FIGURE 11 – confirmation du mot de passe pour le compte utilisateur

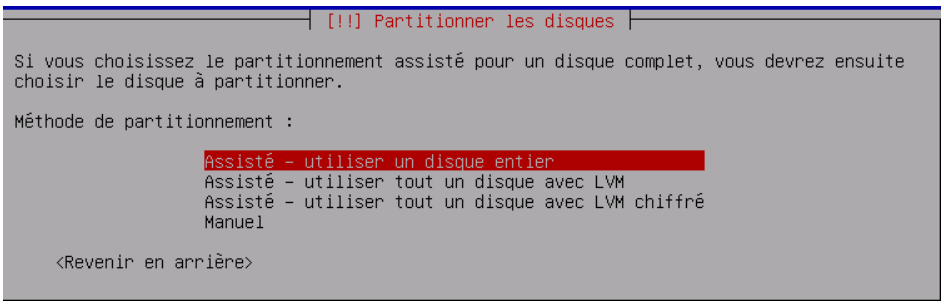

FIGURE 12 – Partitionnement du disque

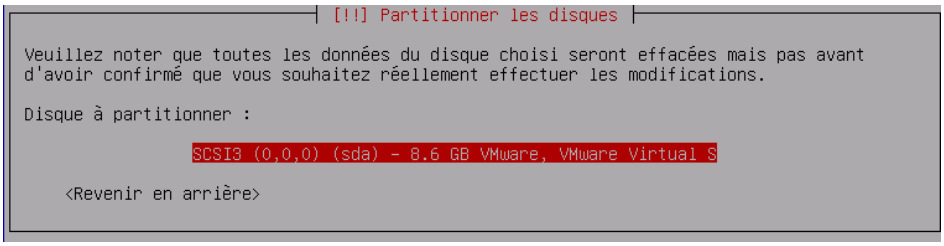

FIGURE 13 – Partitionnement du disque

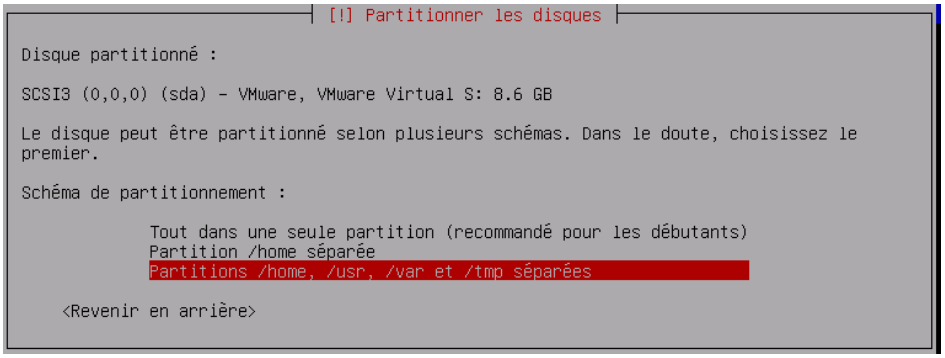

FIGURE 14 – Partitionnement du disque

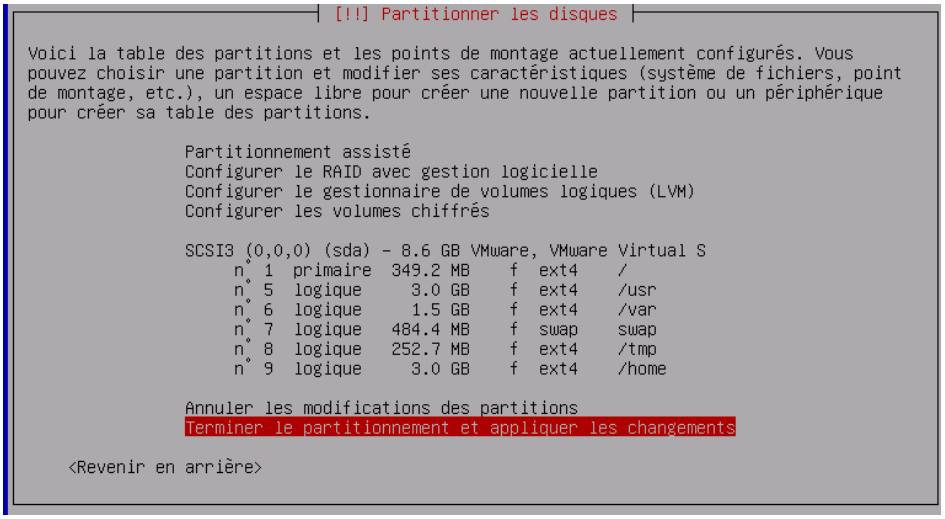

FIGURE 15 – Partitionnement du disque

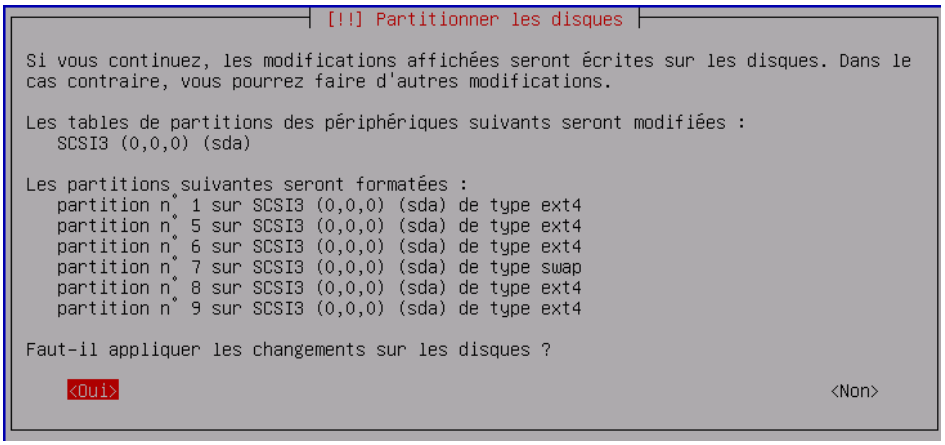

FIGURE 16 – Partitionnement du disque

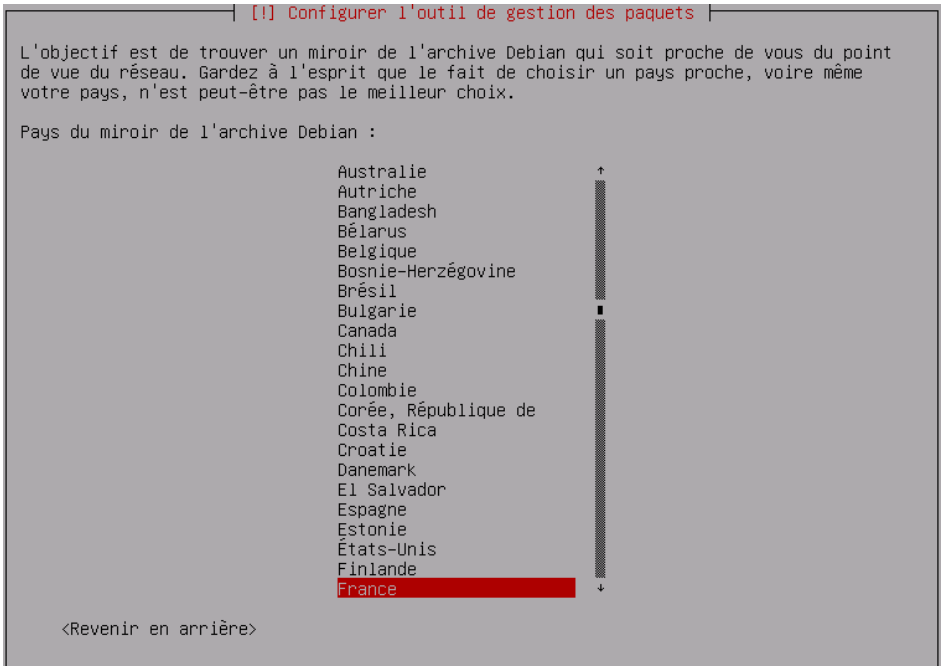

FIGURE 17 – Configuration de l'outil de gestion du paquet

FIGURE 18 – Configuration de l'outil de gestion du paquet

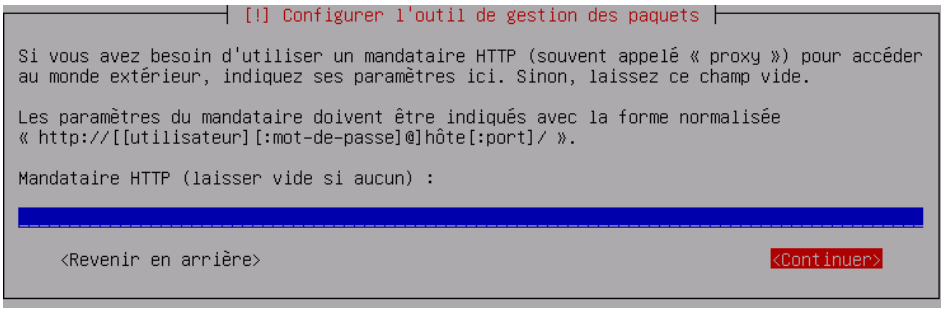

FIGURE 19 – configuration de l'outil de gestion du paquet

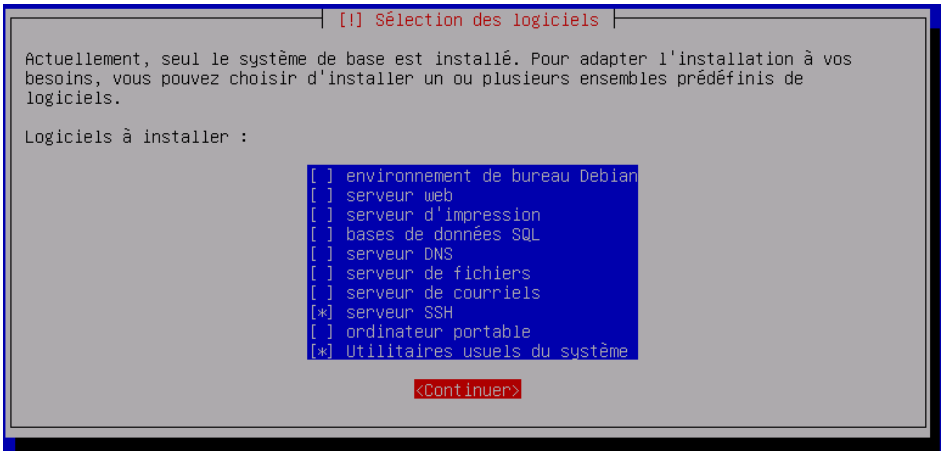

FIGURE 20 – Sélection des logiciels

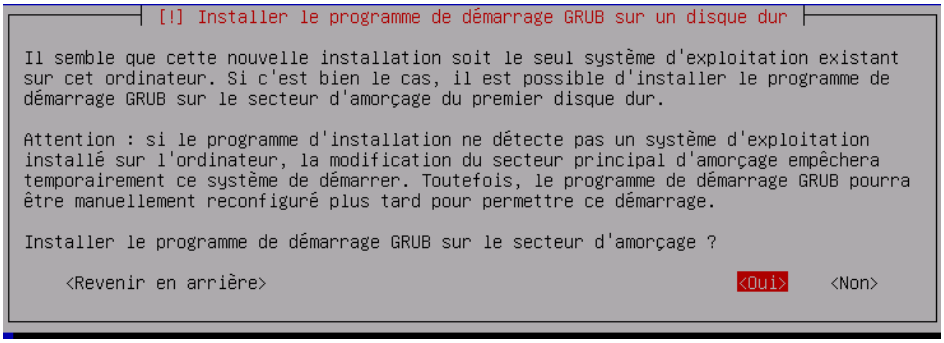

FIGURE 21 – Installation du programme de démarrage GRUB

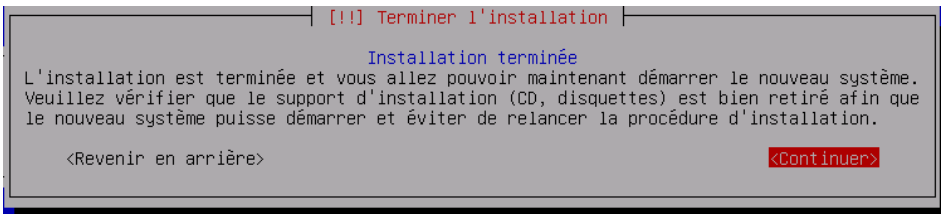

FIGURE 22 – Fin de l'installation

1.2 Installation du client ubuntu

On aura besoin d'installer deux machines virtuelles avec la dernière distribution Ubuntu stable. Elles auront pour nom client1 et client2.

Nous avons choisi d'utiliser un Xubuntu 14.04. A récuperer sur le site officiel :
http ://www.ubuntu.com/download/desktop
les paramètres à configurer :
Nom complet : user
Nom d'utilisateur : user
Mot de passe : resu
Hostname : client1

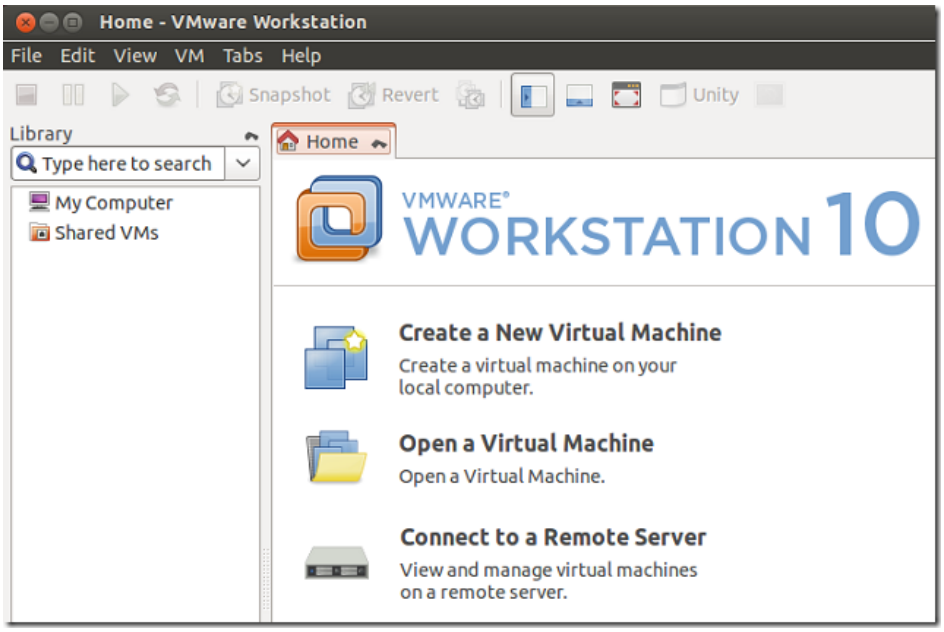

FIGURE 23 – interface de la machine virtuelle VMware

Elle aura pour configuration :
— 1GB en RAM
— 1 processeur
— 20GB en dique dur

On éditera la nouvelle machine virtuelle VMware dans laquelle en spécifiant le chemin de l'iso téléchargé.

On démarre la machine virtuelle.
Une nouvelle fenêtre s'affiche, on clique sur suivant.
On procédera à une installation par défaut.

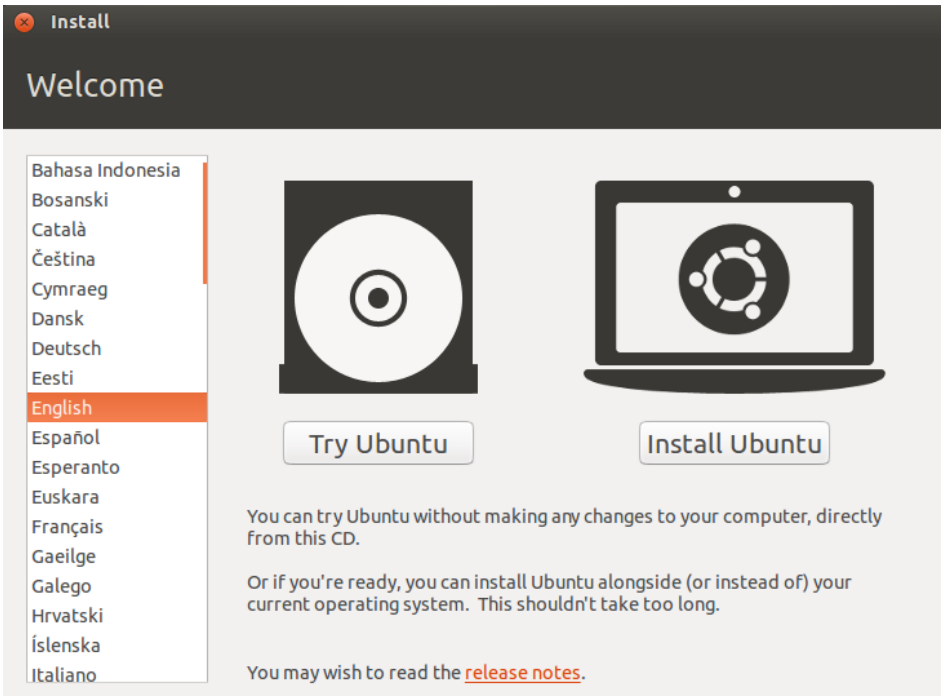

FIGURE 24 – Choix de la langue d'installation

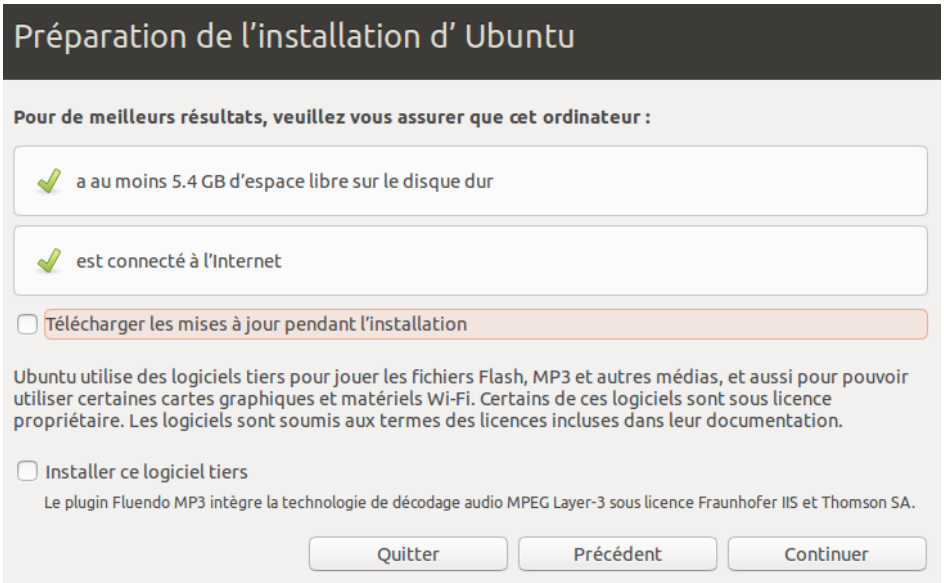

FIGURE 25 – Prérequis pour l'installation

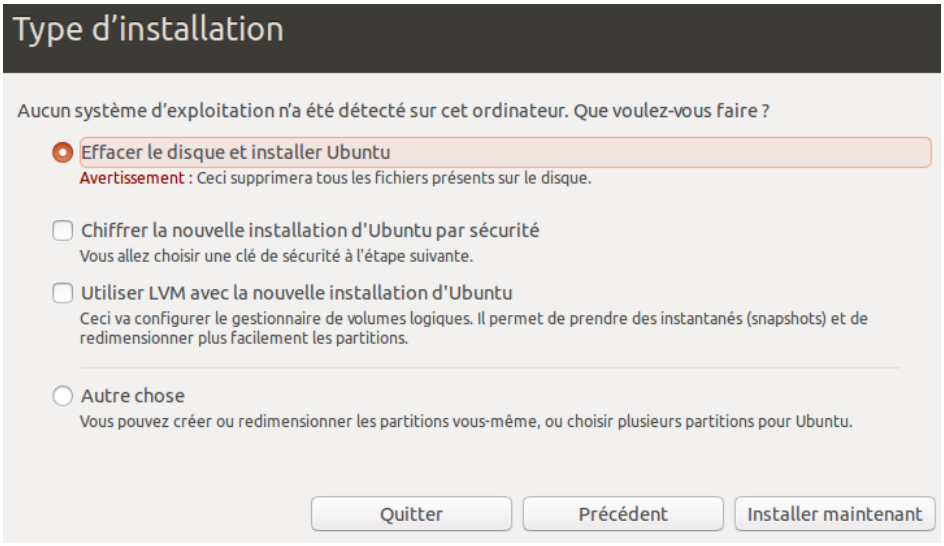

FIGURE 26 – Type d'installation

Une nouvelle fenêtre pour le choix du fuseau horaire. On choisie Paris.

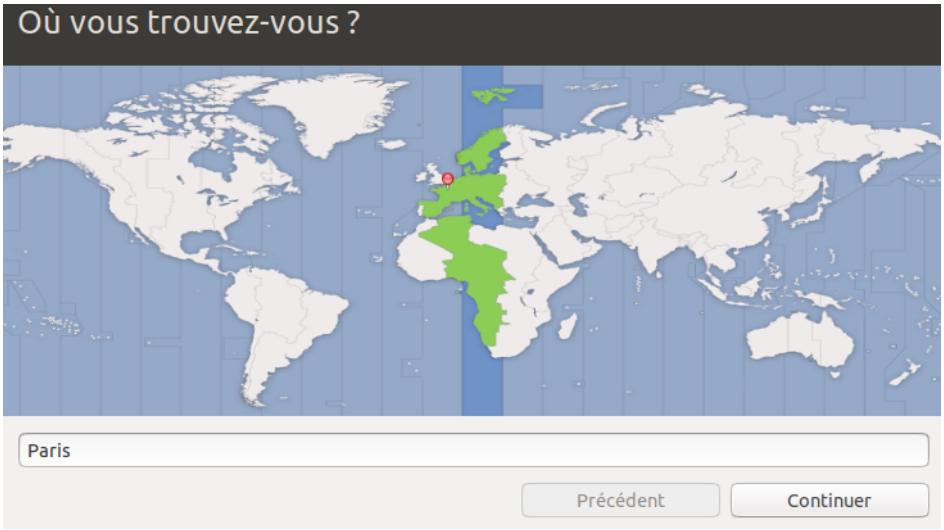

FIGURE 27 – Choix du fuseau horaire.

La nouvelle fenêtre qui s'affiche est pour le choix de la disposition du clavier.

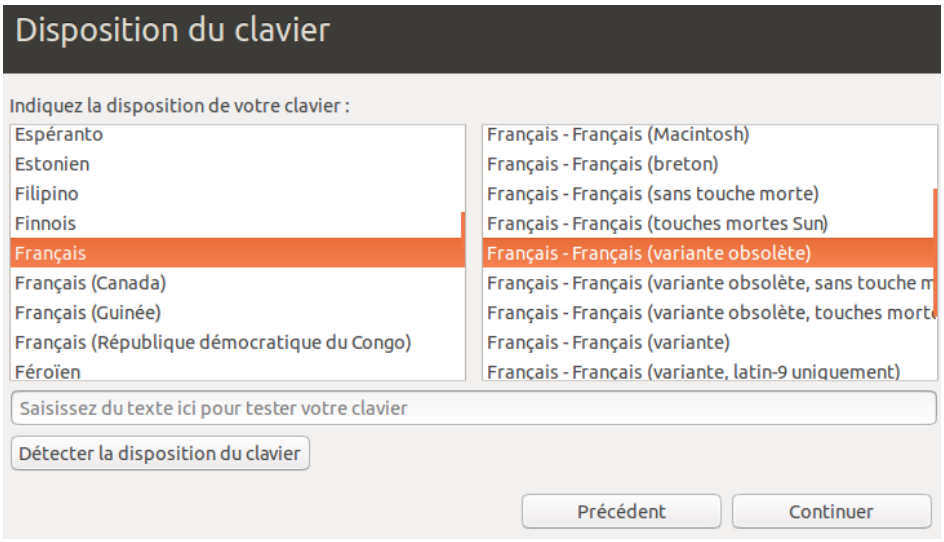

FIGURE 28 – Disposition du clavier.

On procède dans cette étape à la création du client1.

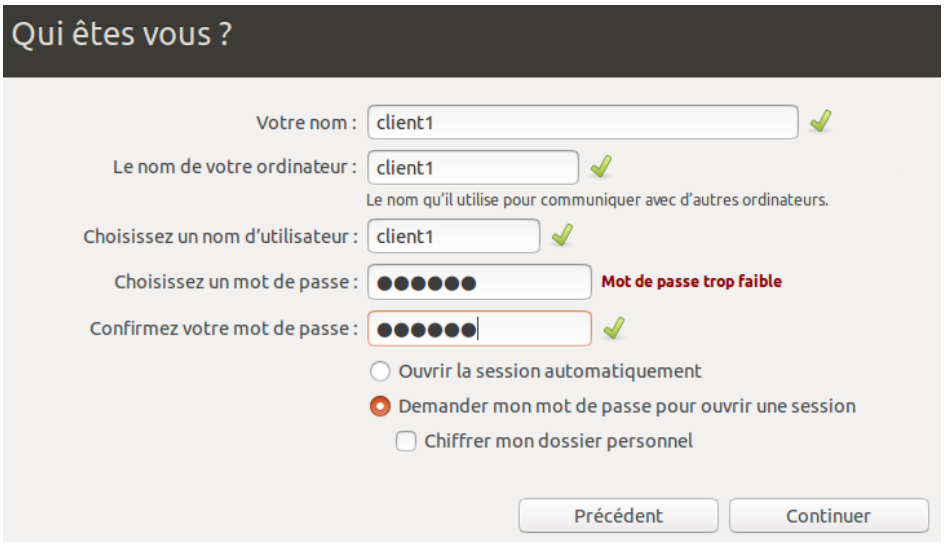

FIGURE 29 – Création du client1.

On redémarre la machine.

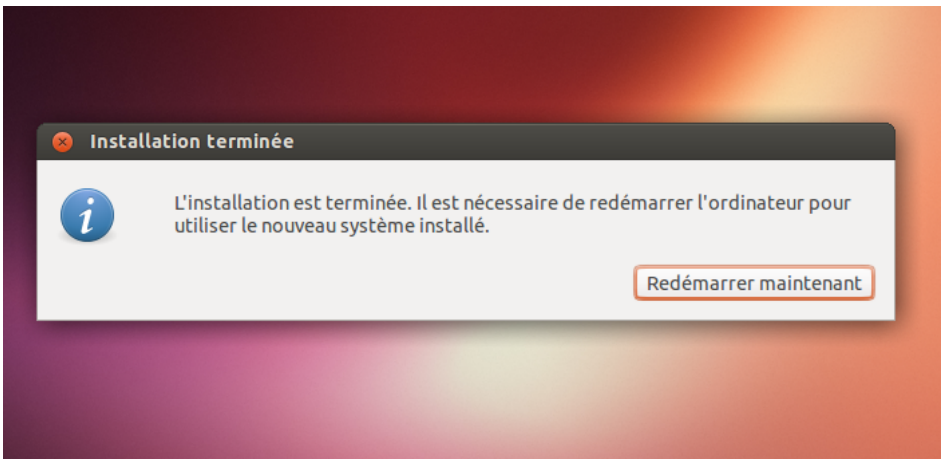

FIGURE 30 – Redémarrage de la machine.

On est arrivé à la fin de l'installation de la machine client1, pour le deuxième client il nous suffit de faire un clone du premier client. UN clic droit sur la machine client1 puis manage puis clone.

On démarre la machine clonnée puis dans /etc/hostname on modifie le nom en la nommant client2.

```
vim /etc/hostname
```

2 Git

Nous avons décidé d'utiliser un serveur git pour gérer les sources.

Git est un logiciel de gestion de versions décentralisé. C'est-à-dire que le développement ne se fait pas sur un serveur centralisé, mais chaque personne peut développer sur son propre dépôt. Git facilite ensuite la fusion (merge) des différents dépôts.

Pour pouvoir utiliser Git, il suffit d'installer le paquet git :

`apt-get install git`

2.1 Gérer les dépots

Pour pouvoir utiliser le Git il va falloir tout d'abord créer un dépot.

`mkdir proscan`

`cd proscan`

`git init`

2.2 Etat du dépot

On peut avoir l'état du dépot en utilisant les commandes :

`git diff`

`git diff <commit1> <commit2>`

Le git offre la possiblité de trouver les changements éffectués. SI vous avez des changements pas encore commités, la commande git diff affichera les modifications éffectuées depuis le dernier commit.

`git status` Permet de savoir tout ce qui n'a pas encore été validé.

`git log` Liste les commits éffectués dans le dépot. Et ainsi voir les modifications faites dans quelle date et par qui.

2.3 Gestion des fichiers

Pour ajouter au git un dossier ou un fichier on utilise la commande :

`git add <nom-du fichier-ou-du-dossier>`

Pour ajouter tout le contenu d'un fichier ou d'un dossier :

`git add *`

Pour supprimer le fichier de l'ordinateur, ainsi que du dépot git :

`git rm <nom-fichier>`

Pour dépolacer le fichier de l'ordinateur, ainsi que du dépot Git :

`git mv <nom-fichier> <nouvel-emplacement>`

2.4 Gestion des commits

Met à jour votre dépôt local (à faire avant de commencer à modifier des fichiers pour être sûr de travailler sur leurs dernières versions et avant tout commit pour éviter les éventuels conflits avec des modifications effectuées par d'autres utilisateurs entre temps).

`git pull`

Créer un commit contenant fichier1 et fichier2. Ces fichiers auront dû être au préal able ajoutés au dépôt avec la commande git add. Il s'agit de la validation d'une transaction.

Pour envoyer un commit dans la branche principale du dépot (master) :

```
git push origin master
```

3 Base De Données

3.1 Client

Table Client La table client est constituée de :

- id : identifiant auto-indexé, clé primaire
- ip : adresse ip du client
- hmac : hmac du client (unique)
- hostname : Nom d'hôte du client
- pid : pid du processus chargé de communiquer avec ce client

3.2 Script

Table Script La table script est constituée de :

- id : identifiant auto-indexé, clé primaire
- nom : Nom du script
- description : Description rapide du script
- code : code du script

3.3 Result

Table Result La table result qui enregistre les résultats des scripts est constituée de :

- id : identifiant auto-indexé, clé primaire
- idclient : Identifiant du client
- idscrip : Identifiant du script
- result : résultat du script

4 Client/Serveur

4.1 Client

Fonctionnement Le client ouvre la connexion vers le serveur dont l'adresse lui est passée en paramètre. Ensuite il attend de recevoir le script à exécuter.

4.2 Serveur

Fonctionnement Le serveur démarre, écoute sur une socket. Lorsqu'une connexion entrante arrive sur la socket, le serveur fork. Le fils accepte la connexion tandis que le père lance le menu afin de recevoir la commande à envoyer. Une fois la commande enregistrer le serveur l'écrit dans un tube nommé puis envoi un signal au fils qui communique avec le client choisi. Ainsi le fils sait qu'il doit lire le tube puis envoyer le script à exécuter au client. Ceci fait il attend le résultat qu'il enregistre dans la base de données.

Le menu 4 choix sont disponible :

- 1. Afficher la liste des scripts
- 2. Afficher la liste des clients connectés
- 3. Choix un script à exécuter
- 4. Ne rien faire et repasser à l'attente d'une connexion entrante.

5 Script

Permissions Bien que la majorité de nos scripts puissent s'exécuter avec les permissions d'un utilisateur, certain d'entre eux nécessitent les droits d'administrateur.

N°	**Résultats**
01	Hostname, Interfaces réseaux, nom de la distribution, version de la distribution, version du noyau, table de routage.
02	Espace des partitions montées.
03	Affiche les connections internet actives.
04	Processus actif.
05	Variables d'environnement.
06	Informations CPU, Interruptions, Mémoire utilisée, Fichier$s$ Swap$s$, version du noyau,systèmes de fichiers montés, périphériques CPU, périphériques usb.
07	Affiche les processus en cours dans une arborescence qui commence à la racine.
08	Récupération de tous les fichiers d'extension ".log".
09	Table de routage.
10	interfaces réseaux.
11	User loggé, heure du dernier démarrage, affiche les processus morts, runlevel courant.
13	Liste des utilisateurs.
14	Affiche l'état de la mémoire de la partition courante.
17	Vérification de l'intégrité de /bin, /usr/bin, /sbin, /usr/sbin.

6 Interface web

L'interface web vient afficher les données stockées au niveau de la base de données. Elle est divisée en trois parties chacune de ces trois permet l'affichage des données stockées dans l'une des tables.

6.1 Installation de l'environnement de travail

Pour pouvoir consulter l'interface web, il nous faudra avoir une interface graphique sur le poste.

L'implémentation de l'interface web sur un environnement linux requiert un serveur xampp. Par défaut le serveur xampp inclut :

— MySQL
— Php 5
— phpMyAdmin
— Apache 2
— ...

L'installation de xampp est très facile. Tout d'abord il faudra récupérer l'archive sur le site d'Apache Friends : https ://www.apachefriends.org/fr/download.html.

Pour changer les droits sur le fichier d'installation. Ouvrir après un terminal et en mode sudo tapez la commande suivante :

```
sudo chmod 755 xampp-linux-*-installer.run
```

Puis tapez la commande suivante :

```
sudo ./xampp-linux-*-installer.run
```

Ainsi vous aurez fini l'installation du serveur xampp.

Pour démarrer le serveur, tapez la ligne de commande suivante :

```
sudo /opt/lampp/lampp start
```

On verra défiler un texte qui nous signalera le lancement du serveur.

L'accès au serveur local se fait en tapant dans le champ du l'url : http ://localhost.

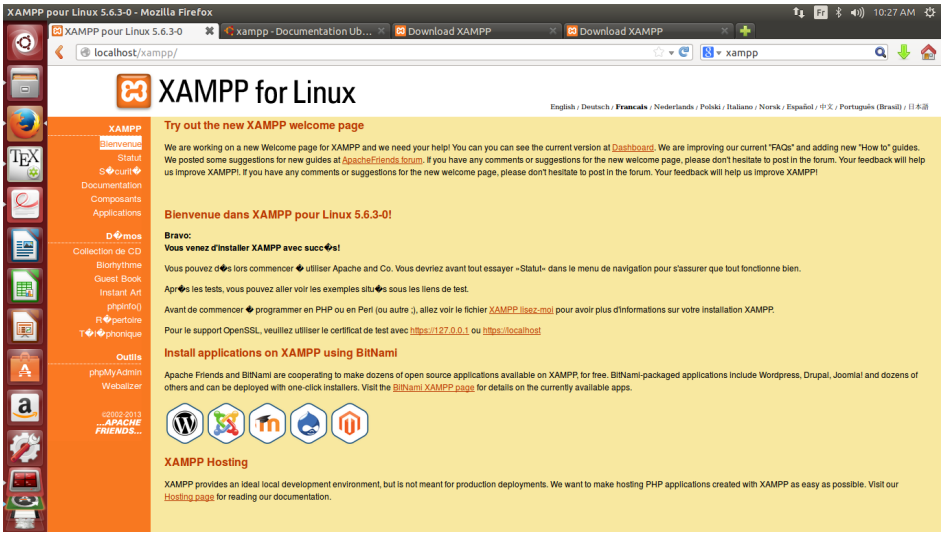

FIGURE 31 – Page d'acceuil du serveur xampp

On mettra notre dossier contenant la page web au niveau de /opt/htdocs.

```
cd /opt/htdocs
```

Pour pouvoir y accéder depuis votre navigateur, il suffira de taper : localhost/nomDuDossier

7 Conclusion

Ce projet nous a permis d'expérimenter le travail à moyen terme (5 mois). Il a aussi été l'occasion de découvrir des technologies que nous n'avions jusque là jamais utilisée (exemple : latex pour l'un et php pour l'autre). Cela nous a également permis d'améliorer nos connaissances en langage C.

Ce projet de programmation C a été intéressant tant du coté humain avec la gestion d'un travail en équipe, que du coté technique. En effet, nous avons été confronté à de multiples problèmes techniques et avons du faire des recherches afin d'essayer de les résoudre.

8 Bibliographie

Le "man" linux