République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

Thème

Administration des services réseau par l'outil Webmin

Réalisé par :

- **Zerrouk Nadia**
- **Lahlouhi Maroua**

Encadré par :

- **Mr Benaissa M**

Présenté le 10 Juin 2014 devant la commission d'examination composée de MM.

- *Mr Benziane Y* *(Examinateur)*
- *Mr Belhoucine A* *(Examinateur)*

Année universitaire : 2013-2014

Remerciement

Nous tenons à remercier en premier lieu DIEU le tout puissant, pour ses faveurs et ses gratitudes, et de nous avoir donné le courage,

la santé et la patience pour terminer ce travail.

Nos vifs remerciements à Monsieur Benaissa mohammed Samir, notre encadreur, pour nous avoir fait l'honneur de nous encadrer, et nous guider par ses conseils avisés et son aide très précieuse.

Un grand remerciement à nos enseignants et enseignantes qui ont contribué à notre formation, depuis le cycle primaire au cursus universitaire.

Enfin, nos gratitudes s'adressent également à tous ceux qui nous ont aidés de près ou de loin à achever ce modeste travail.

Dédicaces

Je dédie ce travail à :

- *Mes parents ,mon père et spécialement à ma chère mère pour son dévouement immodéré à mon égard, son soutien indéfectible et ses conseils avisés qui me permettent de faire les bons choix dans tout ce que j'entreprends.*
- *Toute ma famille, spécialement à ma sœur « Nardjess », mon frère « Hichem », et mes cousines Meriem et Rima.*
- *Mes défunts grands-parents maternels « Bouyakoub Ghouti » et « El Hadja Zohra » ,que Dieu tout puissant les reçoive dans son vaste paradis.*
- *Mes amies, Chahrazad, Samia , Ahlem, Soumia, Baya, et Wafaa .*
- *My best friend Maroua et toute sa famille spécialement sa maman et sa sœur Mimi.*

Nadia

Dédicaces

Je dédie ce travail à :

- *Mon défunt grand-père paternel, « Si Amor » Moudjahid de la première heure, combattant de la liberté de l'Algérie, que Dieu tout puissant le reçoit dans son vaste paradis.*
- *Ma chère mère, qui s'est beaucoup sacrifiée, pour que je puisse réussir mes études, ses efforts titanesques et son dévouement à mon égard au détriment de mes frères et sœurs, m'ont permis d'atteindre ce niveau que j'ai tant voulu.*
- *Mon cher père, idole de ma vie, dont la culture, l'attention à mon endroit ainsi que son soutien indéfini et ses précieuses orientations, m'ont inspiré indéfiniment.*
- *Mes deux sœurs mériem et la petite Maria-Louiza, joie de ma vie.*
- *Mes quatre frères, particulièrement « Toutou ».*
- *Mon grand-père maternel « Hadj Derradji »*
- *Mes amies, Chahrazad, Samia.*
- *Mon amie binôme Nadia et toute sa famille.*

Maroua

Table des Matières

Table des matières

Chapitre II: Installation et Configuration des différents services réseau

Chapitre III: Administration des services réseau par l'outil Webmin

Liste des Figures

Liste des figures

Liste des Acronymes

Liste des acronymes

A

API: application program interface

ARPANET: Advanced Research Projects Agency Network

B

BIND9: Berkley Internet Naming Daemon

BOOTP:Bootstarps Protocol

D

DHCP:Dynamic Host Configuration Protocol

DNS: Domain Name system

H

http :HyperText Transfer Protocol

I

imap :Internet Message Access Protocol

IP :Internet protocol

ICMP :Internet Control Message Protocol

IANA : Internet Assigned Number Autority

O

OSI :Open Sytems Interconnection

P

pop3 : Post Office protocol

R

RARP :Reverse Address Resolution Protocol

S

SSH : Secure Shell

SMTP :Simple Mail Transfer protocol

SQL :Structured Query Language

T

Tftp : trivial File Transfer Protocol

Telnet : Terminal Ntework ou Telecommunication Network

Tcp : transmission Control Protocol

U

Udp : user DAtagram

Introduction Générale

Introduction générale

L’informatique joue un rôle clé dans la vie quotidienne de chacun d'entre nous.

Véritable système nerveux à partir duquel l’information circule, elle est pour beaucoup dans la réussite d’une organisation.

L'histoire des réseaux s'inscrit à la frontière de celles de l'informatique et des télécommunications. Au traitement de l’information, elle a permis d'ajouter le partage.

Les réseaux informatiques sont apparus au milieu 20^{ieme} siècle, dans le cadre de projets militaires (SAGE), universitaires (ARPANET) ou commerciaux (SABRE) .

Les réseaux locaux et nationaux se sont progressivement développés à partir des années 1970. L'interconnexion à l'échelle planétaire a ensuite donné lieu au réseau Internet que nous connaissons aujourd'hui.

Le réseau transmet de l'information d'un point à un autre d'un bureau, d'une entreprise, d'une école, d'un porte-avion. Le réseau n'a pas simplement une avancée technologique du type "Super disquette ", il bouleverse totalement la conception des systèmes informatiques traditionnels.

Pour s'assurer que les services rendus par les réseaux soient convenables, il est nécessaire de surveiller et d'agir quand une erreur se produit, nous parlons alors d'administration réseaux.

L'administration des réseaux informatiques évolue sans cesse et elle s'affirme aujourd'hui comme une activité clé de toute entreprise. En plus d'être constamment en fonction, ces outils d'échange de données et de partage d'information en temps réel doivent être en mesure d'offrir une confidentialité maximale et une sécurité à toute épreuve.

Pour ce faire, il faut obtenir les données de gestion des équipements des réseaux et, si nécessaire, contrôler ces équipements, d'où l'utilité de recourir aux outils de supervision des réseaux.

Pour l’administration d'un serveur Internet, nous avons choisi de nous appuyer sur l'utilisation de l'excellent outil graphique d'administration Webmin. Bien que peu connu du grand public, ce logiciel libre permet d'administrer entièrement un serveur Internet à partir de simples pages html : un simple navigateur suffit alors pour paramétrer la plupart des fonctionnalités. Il devient alors possible d'administrer son serveur depuis n'importe quel poste relié à Internet.

Bien que la perspective de passer son temps à éditer des fichiers de configuration en mode texte ne soit pas très alléchante et qu'il soit possible de se contenter d'utiliser l'interface graphique webmin, il est important d'avoir une bonne vision générale de l'administration en mode texte.

Ceci afin de pouvoir résoudre correctement les problèmes rencontrés mais aussi pour utiliser des fonctionnalités non paramétrables graphiquement, pour mieux comprendre les mécanismes en jeu ou encore pour être capable d'intervenir sur n'importe quel serveur Unix . Nous traitons dans ce mémoire l'administration des services réseaux à partir du puissant outil graphique Webmin.

Nous sommes intéressés par le service web apache, service de résolution des noms DNS et les services de connexion à distance telnet et ssh.

Voici donc les chapitres de notre mémoire :

1. Étude de l’architecture du modèle client serveur
2. Installation et configuration des services réseau : apache, bind(DNS), telnet et ssh
3. Installation de l’outil Webmin
4. Administration des services réseau par webmin
5. Test avec évaluation

Chapitre I
Généralités sur l'architecture Client/serveur

1.1 Introduction

Dans l'informatique moderne, de nombreuses applications fonctionnent selon un environnement client-serveur; cette dénomination signifie que des machines clientes (faisant partie du réseau) contactent un serveur - une machine généralement très puissante en termes de capacités d'entrées-sorties - qui leur fournit des services. Nous allons voir comment cette technologie permet d'exploiter au mieux les réseaux, et permet un haut niveau de coopération entre différentes machines sans que l'utilisateur se préoccupe des détails de compatibilité.

1.2 Définition

L'architecture client/serveur désigne un mode de communication entre plusieurs composants d'un réseau. Chaque entité est considérée comme un client ou un serveur. Chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique. [2]

Une application est bâtie selon une architecture client-serveur lorsqu'elle est composée de deux programmes, coopérant l'un avec l'autre à la réalisation d'un même traitement. La première partie, appelée module client, est installée sur le poste de travail alors que la seconde, appelée module serveur, est implantée sur l'ordinateur (ou même des ordinateurs éventuellement situés dans des lieux géographiques différents) chargé de rendre le service (micro, mini ou grand système). [3]

L'architecture client-serveur met en œuvre une conversation entre 2 programmes pour répondre aux objectifs précédemment cités.

On peut essayer de préciser ce que l'on entend par modèle client-serveur par ce qui suit :

1.2.1 Le serveur : est un programme qui tourne sur un ordinateur dans le seul but de répondre à des requêtes de logiciels tournant sur d'autres ordinateurs. Par extension, on parle de machine serveur généralement très puissante en termes de capacités d'entrée-sortie, qui fournit aux machines clientes des services (programmes fournissant des données telles que l'heure, des fichiers, et une connexion...).

1.2.2 Le client : est un programme tournant sur une machine cliente et qui permet, de soumettre des requêtes à un ou à plusieurs serveurs donc le client est un consommateur de services. [1]

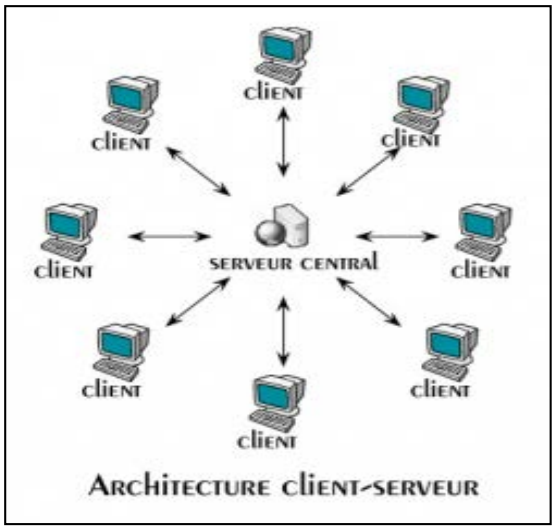

Figure 1.2 : Architecture Client/serveur

Pour le client, un serveur est une boite noire. Seuls les services rendus par le serveur sont connus du client par leurs noms, les paramètres à fournir et les paramètres qui lui seront rendus après exécution du service.

1.3 Générations de client-serveur

1.3.1. Client-serveur de première génération (Architectures centralisées -1970)

La première génération de client-serveur intègre des outils clients autour d'une base de données relationnelle. L'application est développée sur le client à l'aide d'un langage de $4^{ème}$ génération intégrant une interface graphique et de requêtes SQL au serveur. Autour des requêtes, les traitements gèrent l'affichage et les saisies.

Tout le code est exécuté sur le client, celui-ci envoie des requêtes SQL au serveur via un outil de connexion et récupère les résultats.

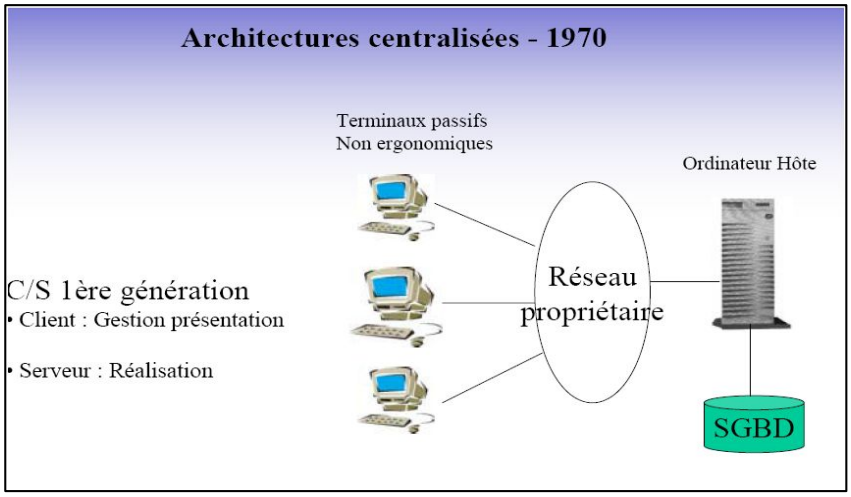

Figure 1.3.1. Client-serveur de première génération

1.3.2. Client-serveur de deuxième génération (Architecture décentralisée 1980)

Le client-serveur de deuxième génération est plus récent, il est caractérisé par l'évolution des outils dans trois directions :

> Possibilité de développer des traitements sous la forme de procédures stockées sur le serveur. Ces procédures sont soit appelées explicitement par les applications clientes, soit déclenchées par des événements survenant sur les données (triggers) ;

> Utilisation intensive de l'approche orientée objet aussi bien pour construire les interfaces que pour modéliser le système d'information ;

> Répartition des fonctions en trois niveaux : la présentation incombe au client, la gestion des données à un serveur de données, les traitements à un serveur d'applications.

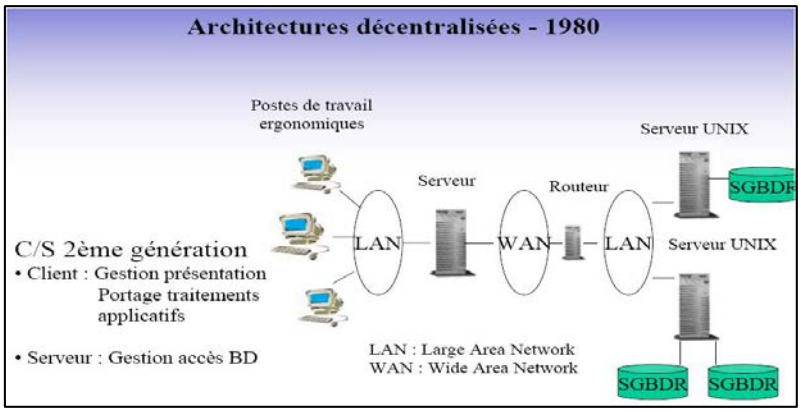

Figure 1.3.2Architecture Client-serveur de deuxième génération

1.3.3 Client Serveur Universel

Cette génération de client-serveur s'appuie sur le compte de client léger représenté par un navigateur web. Celui-ci est chargé de la présentation et possède des possibilités d'exécution locale de traitements. Les serveurs sont disséminés, souvent spécialisés (données ou applications) et s'appuient sur un réseau étendu (Internet) ou local (Intranet).

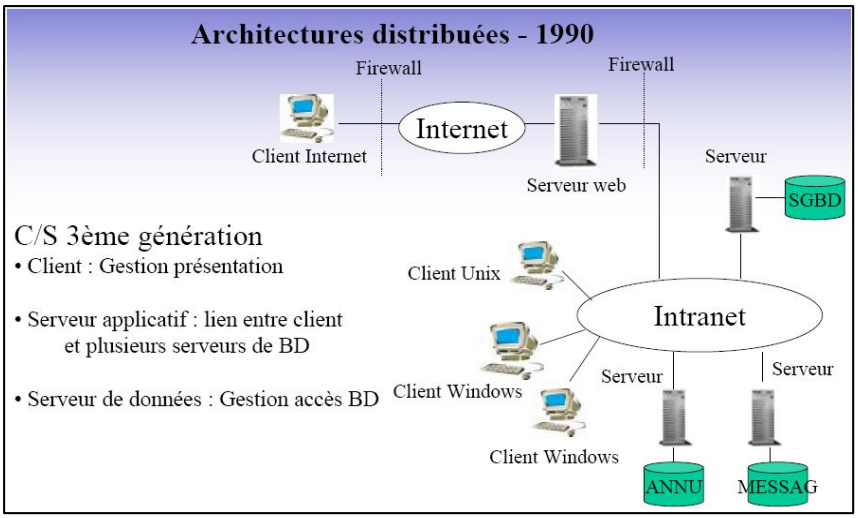

Figure 1.3.3 Client-serveur universel (Architectures Distribuées)

1.3.4 Tendances et évolutions

La tendance est très marquée actuellement vers l'adoption du client serveur universel. Notons aussi que l'évolution des besoins des entreprises pousse à l'intégration d'outils d'aide à la décision. Ces outils s'appuient sur l'exploitation de toutes les données de l'entreprise : base de données de production, fichiers...

Ces données sont extraites, agrégées et stockées dans des entrepôts de données (data Ware house) puis exploitées et restituées par des outils spécialisés (datamining). Plus récemment, les « services web » (web service) offrent de nouvelles perspectives dans le domaine du client-serveur universel.[4]

1.4 Notion de protocole et port

1.4.1 Notion de port

Quand un paquet contenant une requête arrive sur un serveur, comment l'OS sait à quel service il doit donner la requête ?

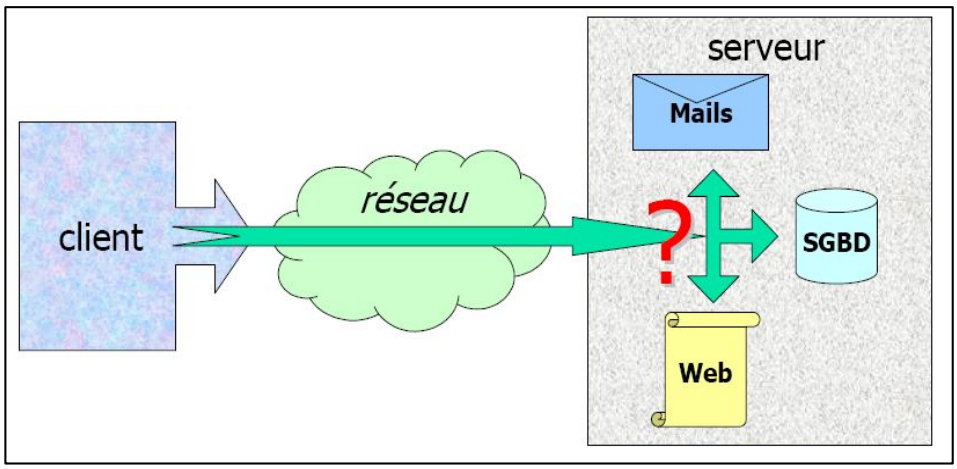

Figure 1.4.1 : Exemple d'un modèle client/serveur

En fait, chaque paquet réseau contient :

- L'adresse IP de la machine d'origine (le client dans le cas d'une requête),

- L'adresse IP de la machine de destination (le serveur dans notre cas),

et une information qui permet de savoir à quel « service » est destinée le paquet. On parle alors de « numéro de service » ou « numéro de port »

L'expéditeur possède aussi un numéro de port, alloué dynamiquement par le système, utilisé pour la réponse.[5]

Le numéro de port occupe 16 Bits (une valeur entière comprise entre 0 et 65 535).

Les numéros de ports entrent dans différentes catégories définies par l'IANA(Internet Assigned Numbers Authority) notamment :

Les ports bien connus (well known ports) : ce sont les ports entre 0 et 1023. Ils correspondent à des applications serveur très courantes.

Exemple : les services standards (dans **/etc/services** sous Unix, dans **%SystemRoot%\System32\drivers\etc\services** pour Windows).

Exemples :

telnet 23/tcp	HTTP 80/tcp
smtp 25/tcp	Pop3 110/udp
tftp 69/udp	imap 143/tcp

Les ports enregistrés (registered ports) : entre 1024 et 49151, utilisés par des applications clientes identifiées, ou des serveurs qui n'entrent pas dans la catégorie précédente.

Les ports dynamiques ou éphémères : au-delà de 49152, ce sont des ports qui ne peuvent pas être enregistrés, réservés aux connexions temporaires.[6]

Un site peut offrir plusieurs services. Chacun de ces services est fourni sur **un port de communication** identifié par un **numéro**. Ce numéro identifie le service quel que soit le site (ex. le service HTTP est offert sur le port numéro 80, FTP le numéro 21...).

Pour accéder donc à un service, il faut l'adresse du site et le numéro du port. Ce numéro peut être connu par le logiciel client ou être récupéré dans le fichier " **services** " en connaissant le nom du service. Sous UNIX ce fichier se trouve dans /etc/services.

En résumé

Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur.

Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port.

1.4.2 Notion de protocole

Un **protocole** est un ensemble de règles et procédures **standards** à respecter pour **émettre** et **recevoir** des données sur un réseau. Cette standardisation a pour but principal de permettre à deux programmes s'exécutant généralement sur différentes machines de communiquer et de se comprendre mutuellement et de manière harmonieuse.

Internet est un ensemble de protocoles regroupés sous le terme "TCP-IP" (Transmission Control Protocol/Internet Protocol).

Voici une liste non exhaustive des différents protocoles qui peuvent être utilisés :

HTTP : (Hyper Texte Transfert Protocol) : c'est celui que l'on utilise pour consulter les pages web.

FTP : (File Transfert Protocol) : C'est un protocole utilisé pour transférer des fichiers.

SMTP : (Simple Mail Transfert Protocol) : c'est le protocole utilisé pour envoyer des mails.

POP : C'est le protocole utilisé pour recevoir des mails .

Telnet : utilisé surtout pour commander des applications côté serveur en ligne.

IP (internet Protocol) : L'adresse IP vous attribue une adresse lors de votre connexion à un serveur .

On classe généralement les protocoles en deux catégories selon le niveau de contrôle des données que l'on désire :

Les protocoles orientés connexion: Il s'agit des protocoles opérant un contrôle de transmission des données pendant une communication établie entre deux machines.

Dans un tel schéma, la machine réceptrice envoie des accusés de réception lors de la communication, ainsi la machine émettrice est garante de la validité des données qu'elle envoie. Les données sont ainsi envoyées sous forme de flot. TCP est un protocole orienté connexion.

Les protocoles non orientés connexion: Il s'agit d'un mode de communication dans lequel la machine émettrice envoie des données sans prévenir la machine réceptrice, et la machine réceptrice reçoit les données sans envoyer d'avis de réception à la première. Les données sont ainsi envoyées sous forme de blocs (datagrammes). UDP est un protocole non orienté connexion.[5]

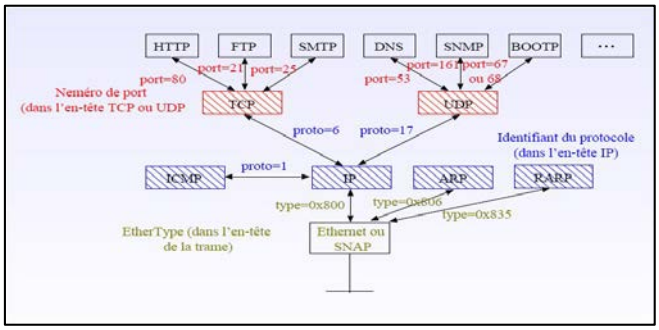

Figure 1.4.2 Notion de Protocole et Port

1.5 La notion de sockets

Il s'agit d'une interface de programmation (API) pour établir des communications inter-processus développée par l'université Berkley.[6]

« Tuyau » entre deux programmes

Quintuplé : (machine1, port1, protocole transmission, port2, machine2)

Exemple

Client sur machine 1 appelle serveur sur machine 2 / port 53.

La connexion s'établit, le canal de communication est ouvert

Il devient possible de communiquer suivant un protocole application (par exemple DNS)

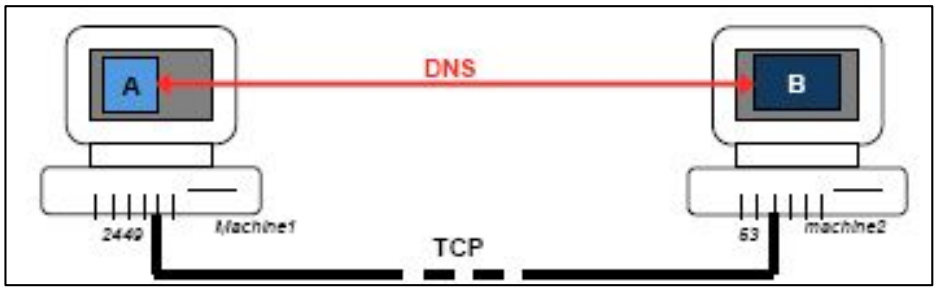

Figure 1.5 Utilisation des Sockets

La programmation des sockets se fait par le biais d'une **API** socket

Une API est une interface de Programmation d'Application : Mécanisme d'interface de programmation, permettant aux programmes d'échanger des données.

Les applications client/serveur ne voient les couches de communication qu'à travers l'API socket .[3].

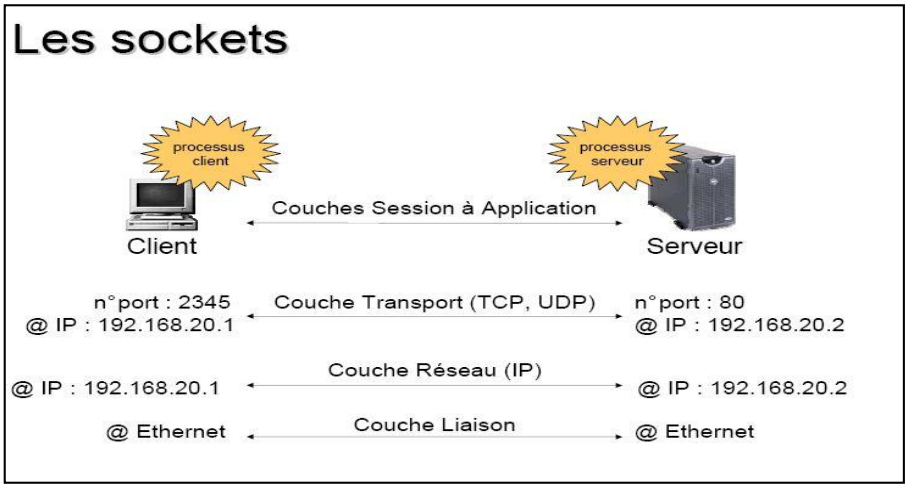

Figure 1.5 Les sockets

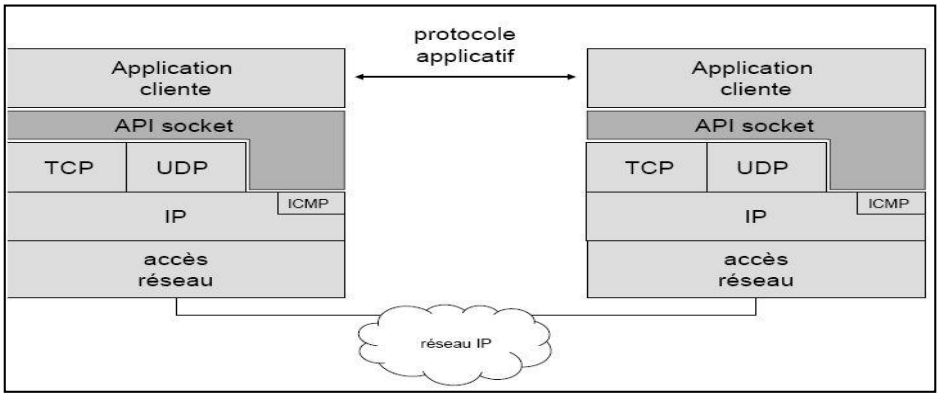

Figure 1.5 Modèle OSI et Sockets

1.6 Le Middleware

1.6.1 Définition

C'est un ensemble de services logiciels construits au-dessus d'un protocole de transport afin de permettre l'échange de requête/réponse entre le client et le serveur de manière transparente .[7]

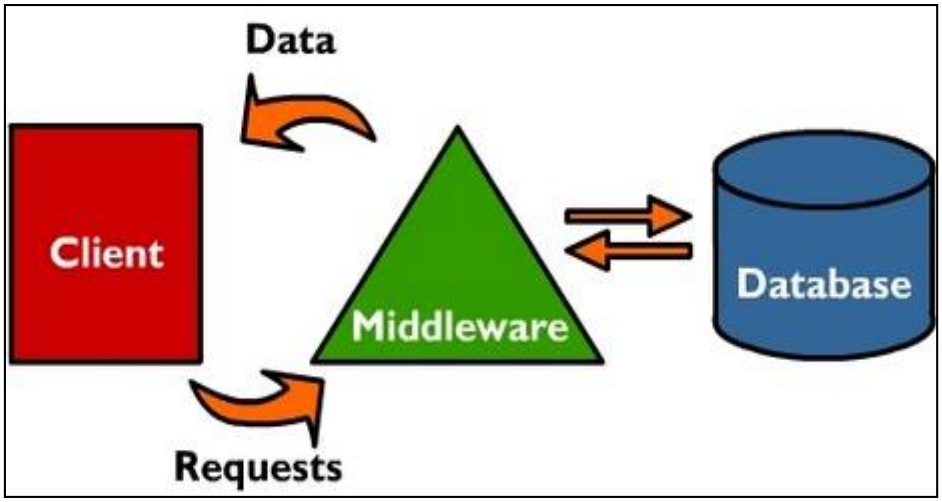

Figure 1.6.1 Middleware

1.6.2. Les services des middlewares

Un middleware est susceptible de rendre les services suivants :

Conversion :

Service utilisé pour la communication entre machine mettant en œuvre des formats de données différentes.

Adressage :

Permet d'identifier la machine serveur sur laquelle est localisé le service demandé afin d'en déduire le chemin d'accès dans la mesure du possible.

Sécurité :

Permet de garantir la confidentialité et la sécurité des données à l'aide de mécanismes d'authentification et de cryptage des informations.

Communication :

Permet la transmission des messages entre les deux systèmes sans altération. Ce service doit gérer la connexion au serveur, la préparation de l'exécution des requêtes, la récupération des résultats et la déconnexion de l'utilisation.

Le middleware masque la complexité des échanges inter-applications et permet ainsi d'élever le niveau des API utilisées par les programmes. Sans ce mécanisme, la programmation d'une application client/serveur serait complexe et difficilement évolutive.

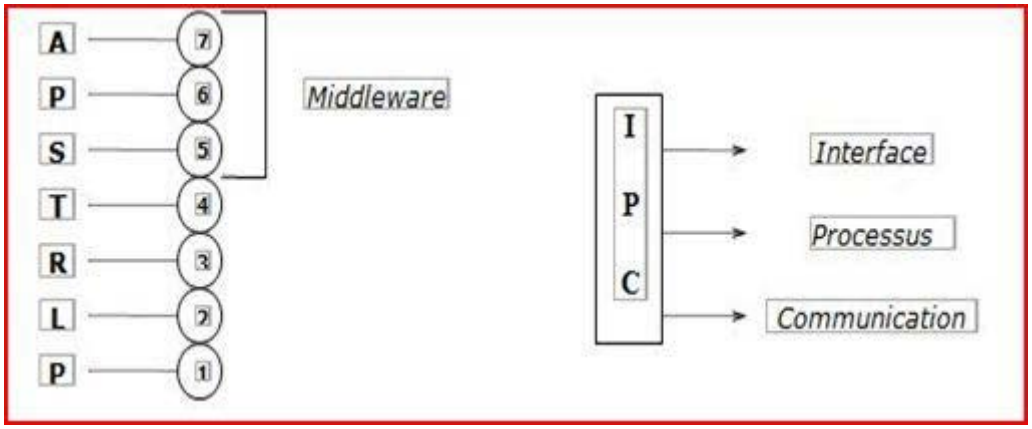

Figure 1.6.2 IPC

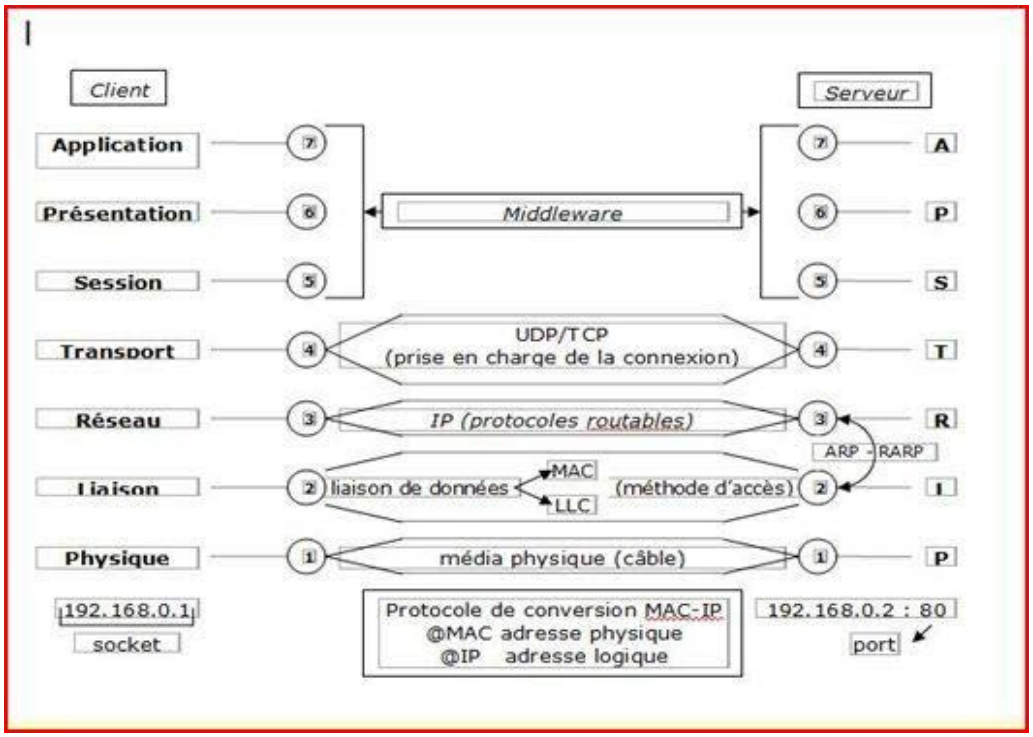

Figure 1.6.2 Modèle OSI et middleware

1.6.3 Remote Procedure Call (RPC)

Appel de procédure à distance : RPC

Technique permettant d'appeler une procédure distante comme une procédure locale en rendant transparent les messages échangés.

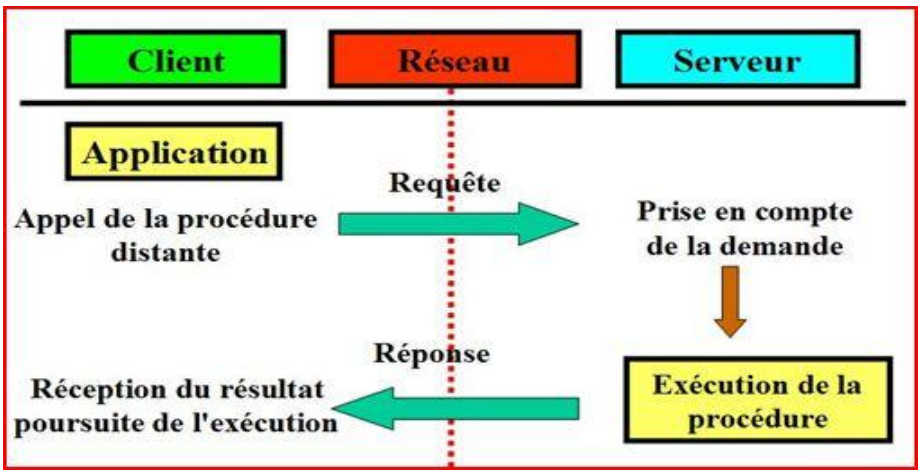

Figure 1.6.3 Appel de procédure à distance

Dans les RPC, la communication client/serveur : peut se faire par datagramme (par paquets), ou par connexion (flux de données dans un canal).

Elle peut être :

Synchrone : le serveur attend la requête du client ; et pendant que le serveur fait le traitement, le client attend.

Asynchrone : pendant qu'un des acteurs traite les informations, l'autre acteur, au lieu

d'attendre, continue de « vivre sa vie ». Il est interrompu (par une interruption système) quand l'autre acteur lui envoie de nouveau de l'information, afin qu'il aille traiter ce flot entrant. [5]

1.7 Conclusion

Le modèle client /serveur est la base de tous les services réseaux informatique.

Ce pourquoi nous nous sommes intéressés à l'étude de ce modèle. Le but de ce chapitre est de décrire ses différentes notions de base comme le middleware, les protocoles, les sockets et l'appel de procédure à distance. Afin de mieux configurer et administrer les différents services qui font l'objectif de notre projet. La configuration de la carte réseaux est une étape fondamentale pour faire cette configuration sur la distribution Linux Ubuntu. Dans le chapitre suivant, nous présentons les étapes de configuration et d'administrations des différents services réseaux.

Chapitre II
Installation et Configuration des différents services réseau

2.1 Introduction

Les services DNS (résolution de nom), SSH (Secure shell), Apache(serveur web) sont très importants pour l'administration de notre réseau. Dans ce chapitre nous présentons les étapes d'installation et de configuration de ces différents services réseau.

2.2 Service de résolution de nom (DNS)

2.2.1 Installation et configuration d'un service de noms DNS : BIND9

Le service DNS (Domain Name Service) est un service TCP/IP permettant la correspondance entre un nom de domaine qualifié (FQDN : Fully Qualified Domain Name) et une adresse IP, par exemple **www.licence-pfe.dz = 192.168.1.65**. Ainsi, grâce à DNS, il n'est pas nécessaire de se souvenir des adresses IP.

Un serveur qui héberge le service DNS est appelé "serveur de noms". Ubuntu est livré par défaut avec BIND9 (Berkley Internet Naming Daemon), le serveur DNS le plus utilisé sur Internet.

Etape1 : Installation serveur DNS BIND9

Pour installer le serveur BIND9, il suffit d'installer le paquet bind9 :

Installation le package bind9_9.6.1.dfsg.p1-3_i386.deb

Installation le package bind9-doc_9.6.1.dfsg.p1-3_i386.deb

Installation le package bind9utils_9.6.1.dfsg.p1-3_i386.deb

Le paquet dnsutils fournit des outils très pratiques pour tester et débugger le service DNS. La documentation BIND9 peut également être trouvée dans le paquet bind9-doc.

BIND9 peut être utilisé pour contenir les enregistrements DNS d'un nom de domaine enregistré. Un ensemble d'enregistrements DNS pour un nom de domaine est appelé une "zone".

Etape 2 : Configuration de la carte réseau eth0

Ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up

Etape3 : Configuration du serveur DNS BIND9

Les fichiers de configuration de BIND9 sont stockés sous :

/etc/bind/

La configuration principale de BIND9 est effectuée dans les fichiers suivants :

/etc/bind/named.conf

/etc/bind/named.conf.options

/etc/bind/named.conf.local

Éditer le fichier "**/etc/bind/named.conf.options**", positionner l'option **forwarders** en indiquant l'adresse ip de serveur DNS.

Modifier l'option forwarders du fichier "**/etc/bind/named.conf.options**" qui ressemble à ceci :

forwarders {

192.168.1.1; #adresse ip de serveur DNS

};

Si votre carte réseau est configurée pour utiliser DHCP, décommenter la ligne 20 du fichier "**/etc/dhcp3/dhclient.conf**" :

prepend domain-name-servers 127.0.0.1;

Redémarrer bind

sudo service bind9 restart # sudo /etc/init.d/bind9 restart

Tests

Si le package **dnsutil**s a été installé, il est possible de tester la nouvelle configuration en utilisant dig :

dig -x 127.0.0.1

ou

nslookup localhost # nslookup 127.0.0.1

Etape 4 : Création des fichiers de zone

Création du Fichier de la résolution de nom : db.licence-pfe.dz

Pour ajouter une zone, et faire de BIND9 un serveur maître : éditer le fichier named.conf.local :

[...]

zone "licence-pfe.dz" {

type master;

file "/etc/bind/db.licence";

};

[...]

Créer maintenant le fichier db.licence depuis un fichier existant :

Sudo etc/bind/db.local /etc/bind/db.licence

Editer le nouveau fichier pour la zone (/etc/bind/**db.licence**), et ajouter les informations suivantes :

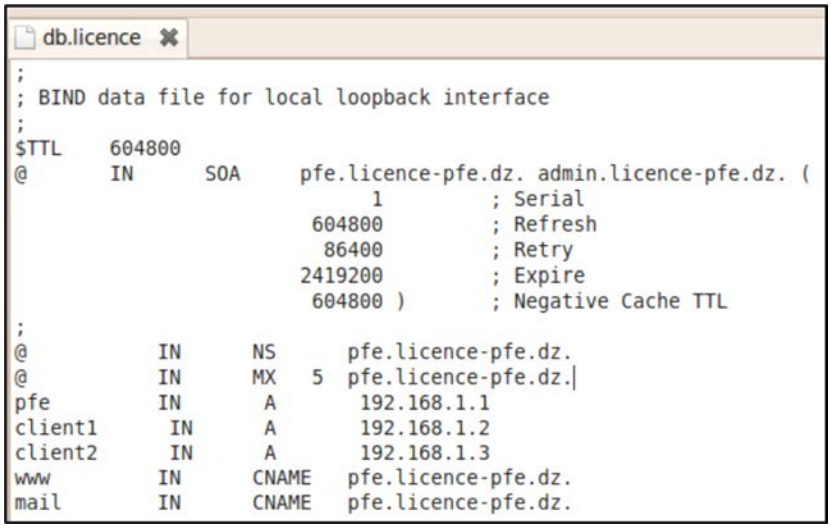

Figure 2.2.1 fichier db.licence

Le numéro de série doit être incrémenté à chaque changement dans le fichier de zone. En cas de multiples changements, une seule incrémentation suffit.

Une fois les changements dans le fichier de zone effectués, il faut redémarrer BIND9 pour qu'ils prennent effet :

sudo service bind9 restart

Création du Fichier de la résolution inverse : db.192

Maintenant que notre fichier de zone est configuré et que les adresses IP sont résolues,

une zone de recherche inversée est requise. Une zone de recherche inversée permet au DNS de convertir une adresse en nom.

Editer /etc/bind/named.conf.local et ajouter les lignes suivantes :

zone "1.168.192.in-addr.arpa" {

type master;

notify no;

file "/etc/bind/db.192";

};

Créer maintenant le fichier db.192 depuis un fichier existant :

sudo cp /etc/bind/db.127 /etc/bind/db.192

Editer le fichier /etc/bind/**db.192** et ajouter les informations suivantes :

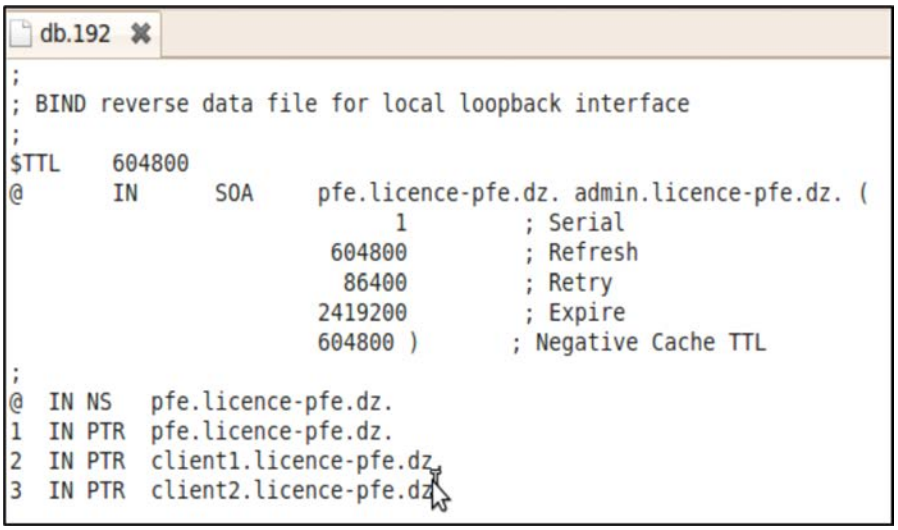

Figure2.2.1 Fichier db.192

Le numéro de série de la zone de recherche inversée nécessite d'être incrémenté à chaque changement. Pour chaque enregistrement A ajouté dans /etc/bind/db.licence, il faut créer un enregistrement PTR dans /etc/bind/db.192.

Après avoir créé le fichier de la zone de recherche inversée, redémarrez BIND9 :

sudo service bind9 restart

Tests

Il doit maintenant être possible de faire un ping sur licence-pfe.dz et la requête doit être résolue :

***ping licence*-pfe.dz**

L'utilitaire named-checkzone (inclus dans le package BIND9) peut également être utilisé :

Pour tester la recherche inversée, l'utilitaire dig peut être utilisé

dig 1.168.192.in-addr.arpa. AXFR

Etape 5 : Configuration du fichier resolv.conf de la machine serveur où le service bind9 est installé : resolv.conf

Le fichier resolv.conf se trouve dans le chemin /etc/resolv.conf

Ajouter les deux lignes suivantes dans le fichier resolv.conf :

Search licence-pfe.dz

nameserver 192.168.1.1 # ip de l'adresse de machine serveur de nom (DNS)

Redémarrer le service DNS (BIND9)

/etc/init.d/bind9 restart

Etape 6 : Configuration de la machine cliente sous linux

le fichier /etc/resolv.conf doit contenir les lignes suivantes:

search licence-pfe.dz

nameserver 192.168.1.1

Ce fichier doit contenir l'adresse du serveur de noms que l'on possède et de notre domaine. Nous pouvons spécifier plusieurs serveurs de noms grâce à la commande nameserver, on peut également positionner notre nom de machine à l'aide la commande hostname.

Pour que la commande de ping réussisse il faut indiquer le serveur DNS que l'on veut utiliser :

éditer le fichier resolv.conf de la machine client et ajouter les lignes suivantes :

nano /etc/resolv.conf :

search licence-pfe.dz # nom de domaine

nameserver 192.168.1.1 # @ip de serveur primaire dns

Redémarrer le service DNS (BIND9)

/etc/init.d/bind9 restart

Etape 7 : Utilisation de l'utilitaire d'interrogation d'un serveur de nom DNS : nslookup

Les utilitaires nslookup et dig permettent d'interroger un serveur de nom (serveur DNS) afin d'avoir des informations sur un domaine ou sur une machine. Par défaut, nslookup et dig utilisent le serveur de nom configuré sur votre machine. Vous pouvez toutefois interroger un autre serveur de nom.[8]

Test la résolution des noms

nslookup 192.168.1.1

nslookup pfe

nslookup pfe.licence-pfe.dz

nslookup 192.168.1.2

nslookup localhost

nslookup mail.licence-pfe.dz

nslookup www.licence-pfe.dz

2.3 Installation d'un service ssh : Secure Shell (connexion à distance sécurisée)

2.3.1 Le protocole SSH

Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisée. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible d'utiliser un sniffeur pour voir ce que fait l'utilisateur.[9]

2.3.2 Etapes installation et configuration d'un protocole ssh

Etape 1 : Installation du client SSH

Sur le poste client (qui va prendre l'accès à distance) **openssh-client** installé par défaut sous Ubuntu doit être présent. Si le client ssh n'existe pas, installer le paquet suivant :

openssh-server.deb

Etape 2 : Installation du serveur SSH

Installez le paquet **openssh-server** sur votre poste serveur.

Pour démarrer le service :

sudo /etc/init.d/ssh start ou sudo service ssh start

Pour arrêter le service :

sudo /etc/init.d/ssh stop ou sudo service ssh stop

Pour redémarrer le service :

sudo /etc/init.d/ssh restart ou sudo service ssh restart

2.4 Installation et configuration d'un serveur web (apache)

2.4.1 le serveur apache

Apache est un serveur http libre, c'est un des serveurs http les plus utilisés sur Internet avec plus de 60% des sites d'Internet.

En résumé, un serveur http est un serveur hébergeant un ou plusieurs sites Web, c'est à dire des pages html ou des programmes générant des pages html (programmes cgi) qui sont accessibles par des navigateurs internet. Le protocole, permettant l'échange de pages html est le protocole http, d'où le nom de serveur http. Ce protocole utilise généralement le port 80.

Pour lancer un serveur apache on écrit la commande suivante :

/etc/init.d/apache2 start.

Pour vérifier que le démon apache tourne :

ps –ef | grep apache

2.4.2 Configuration du serveur web Apache

a)Configuration de base

La configuration globale d'apache s'effectue par modification du fichier de configuration

/etc/apache2/apache2.conf.

Apache pouvant gérer plusieurs serveurs, on trouve des compléments pour la configuration de chaque serveur dans /etc/apache2/site-enabled.

Les fichiers de /etc/apache2/site-enabled sont inclus dans *apache2.conf* par un *Include* et suivent la même syntaxe.

Les fichiers dans /etc/apache2/site-enabled correspondent aux serveurs activés et sont en fait un lien symbolique vers un fichier dans /etc/apache2/site-available, qui contient la liste de tous les serveurs disponibles.

b)Les étapes d'installation d'un serveur web apache2

sudo apt-get install apache2

Pour démarrer le serveur Web :

sudo /etc/init.d/apache2 start

Pour stopper le serveur web :

sudo /etc/init.d/apache2 stop

Pour relancer le serveur web :

sudo /etc/init.d/apache2 restart

Le chemin des fichiers de configuration :

/etc/apache2 : contient le fichier de configuration apache2.conf

Le site web par défaut se trouve dans le chemin par défaut : /var/www/ (on trouve le site index.html) [8]

Authentification aux sites hébergés

Pour l'accès authentifié, nous utilisons les deux fichiers suivants : .htpasswd et .htaccess

Le fichier .htpasswd contient les utilisateurs qui ont le droit de consulter la page web

Le fichier .htpasswd se trouve dans le chemin /etc/apache2 (c'est un fichier caché)

Pour créer un nouvel utilisateur, on utilise la commande suivante :

htpasswd -c .htpasswd nom-utilisateur

Pour le fichier .htaccess, il sera placé dans le répertoire où se trouvent les pages web à protéger.

Il contient les informations suivantes :

Authname "securité web"

AuthUserFile /etc/apache2/.htpasswd

AuthType Basic

require valid-user |list-user

Le fichier /etc/apache2/site-enable/default contient les informations sur le chemin où se trouvent les sites web à consulter par les clients du navigateur Internet.

Créer un site web appelé index.html

Créer un répertoire web dans le chemin /home/web/

Mettre le site web index.html dans le répertoire web (/home/web)

Modifier le fichier : /etc/apache2/site-enable/default et ajouter les lignes enfoncées:(en gras)

Exemple 1

```
<VirtualHost *:80>
        ServerAdmin pfe@localhost
        DocumentRoot /var/www
        <Directory />
                Options FollowSymLinks
                AllowOverride None
        </Directory>
        <Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
```

Order allow,deny

allow from all

</Directory>

Alias /essai/ "/home/web/"

<Directory "/home/web/">

Options Indexes FollowSymLinks MultiViews

AllowOverride AuthConfig

Order allow,deny

allow from all

</Directory>

</VirtualHost >

Le site web à héberger se trouve dans le chemin /home/web avec un raccourci indiqué par /essai/

Pour accéder au site web on écrit, soit sous Linux ou Windows à partir d'un navigateur Internet sur une machine cliente :

http://www.licence-pfe.dz/essai/index.html

Ou

http://www.licence-pfe.dz/home/web/index.html

Ou

http://192.168.1.1/essai/index.html

2.5 Conclusion

Dans ce chapitre, nous avons configuré les services réseau textuellement, nous verrons par la suite (chapitre 3) comment configurer ces mêmes services graphiquement en utilisant l'outil Webmin .

Chapitre III
Administration des services réseau par l'outil Webmin

3.1 Introduction

Notre projet consiste à utiliser l'outil Webmin afin d'administrer de façon simple les différents services réseau.

Figure 3.1 Logo de Webmin

Nous avons utilisé l'outil VirtualBox pour construire notre réseau virtuel.

Ce réseau virtuel est composé par les machines suivantes :

1) Machine serveur : @ :192.168.1.1 sous Ubuntu 9.10

Deux Machines clientes :

1) Client 1: @ :192.168.1.2 sous Ubuntu 12.04

2) Client 2: @ :192.168.1.3 sous Windows7

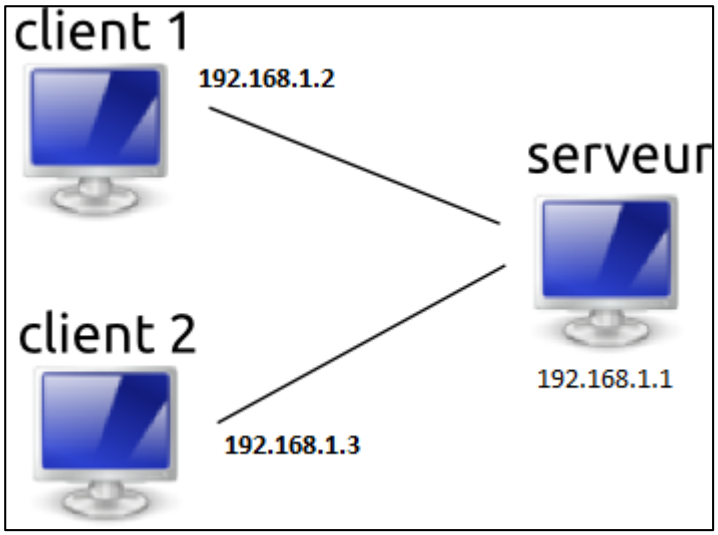

Figure 3 .1 Architecture de notre réseau

3.2 Présentation et Installation de l'outil Webmin

Webmin est un outil d'administration basé sur une interface Web pour les systèmes Unix/Linux. Il est très puissant, et également très simple à utiliser. A travers une connexion sécurisée, il peut être utilisé pour administrer de façon sûre la plupart des services depuis n'importe quel poste sur le réseau.

3.2.1 Installation

Voici les instructions pour installer rapidement l'interface d'administration WebMin sous Ubuntu :

1) Installer les paquets nécessaires à WebMin en tapant la commande suivante dans le terminal:

sudo apt-get install webmin

Pour se connecter à webmin ; taper l'URL suivante: https://localhost:10000/ et se loguer avec le compte "root" et le mot de passe de la session « root ».

La figure suivante est la page permettant l'accès à webmin après authentification

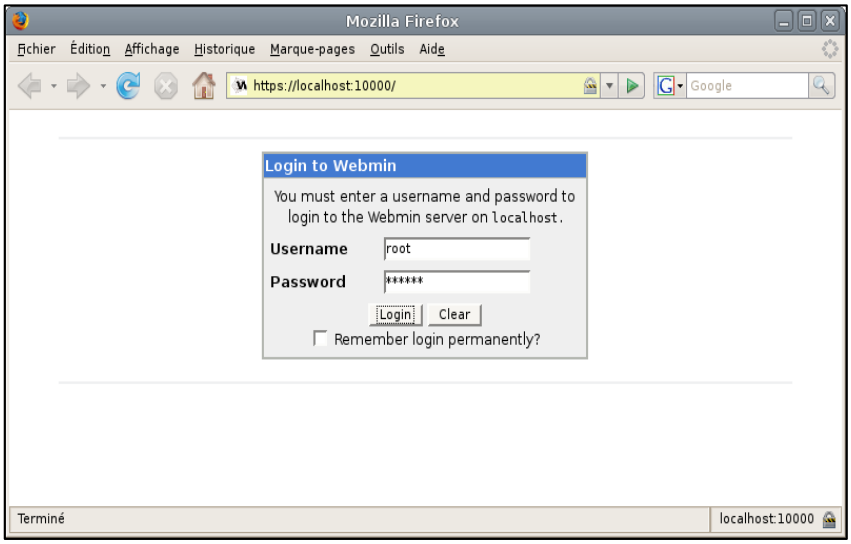

Figure 3.2.1 Page d'authentification

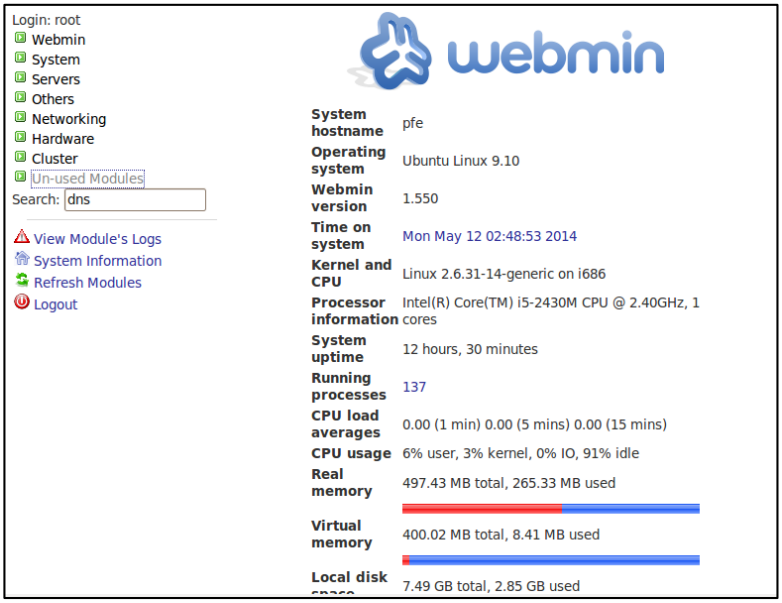

Figure 3.2.1 Page d'accueil de Webmin

3.3 Administration des services réseau par Webmin

3.3.1 Configuration du serveur DNS (BIND)

Après l'installation des paquets Bind sur la machine serveur (Ubuntu 9.10), il faut procéder à la configuration suivante :

Pour ce faire, nous allons utiliser Webmin

Etape1 : modification du fichier resolv.conf

Dans webmin, aller dans «Autres/Gestionnaires de Fichiers, rechercher le fichier resolv.conf dans le répertoire etc

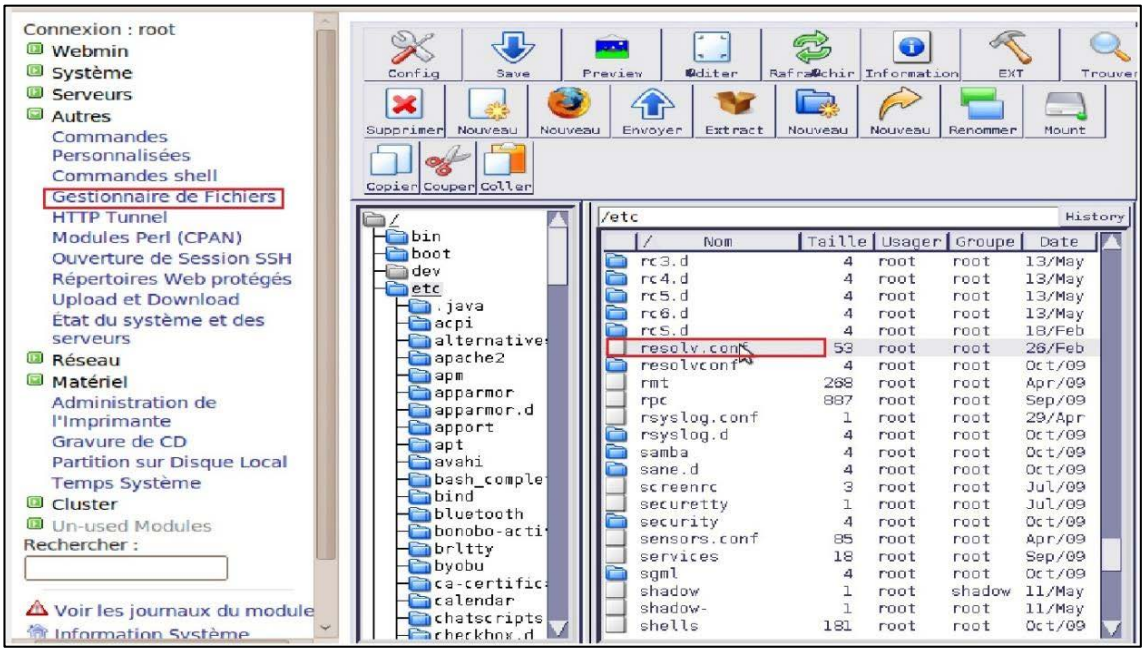

Figure 3.3.1 gestionnaire de fichiers

Editer le fichier resolv.conf . Le fichier doit contenir les lignes suivantes :

search licence-pfe.dz

nameserver 192.168.1.1

nom de domaine

@ip de serveur primaire dns

Redémarrer bind .

Etape 2 : création d'une zone primaire graphiquement

Aller dans Serveurs --- > Serveur de noms de domaines BIND

Figure 3.3.1 Onglet Serveurs

Créer une Zone Primaire en allant dans : Servers -> BIND DNS Server → Create master zone

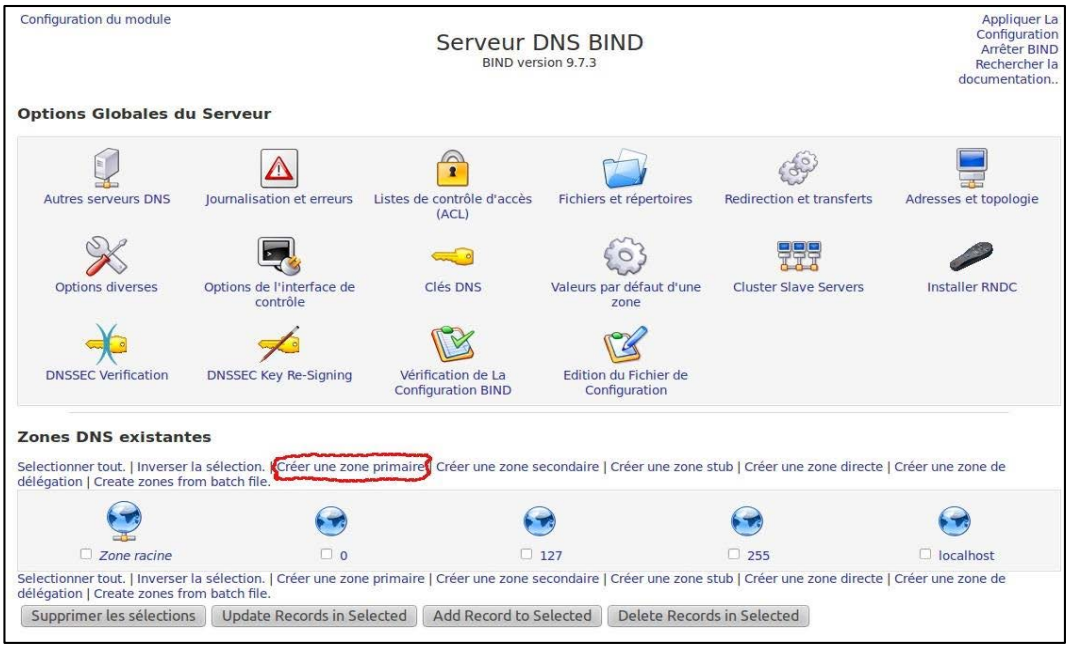

Figure 3.3.1 Fenêtre de configuration graphique Serveur DNS BIND

Il faut à présent renseigner les champs :

- Nom de domaine / Réseau : Renseignez ce champs avec votre nom de domaine (licence-pfe.dz dans notre cas).
- Serveur Primaire : Mettez ici le nom du serveur primaire (pfe.licence-pfe.dz dans notre cas).
- Adresse électronique : entrez-y l'adresse e-mail principale de votre site (admin@licence-pfe.dz dans notre cas).

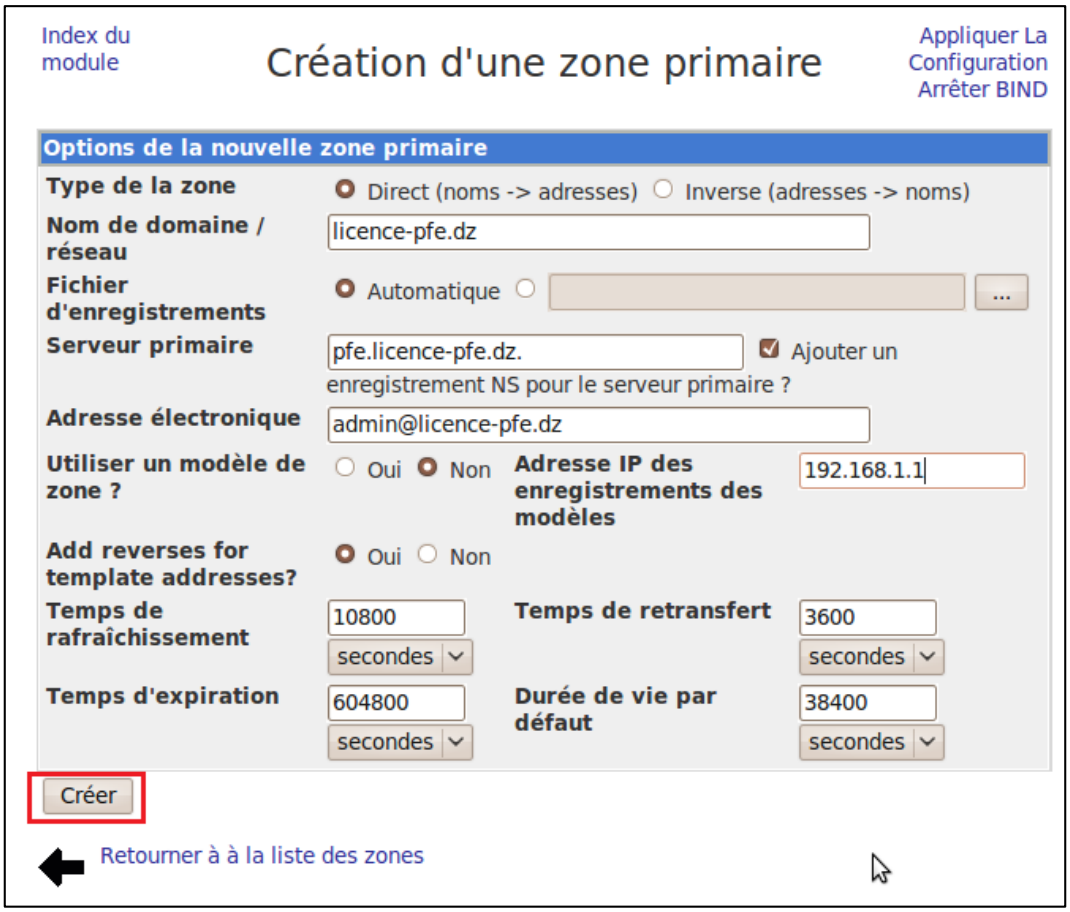

Figure 3.3.1 Fenêtre de création d'une zone primaire

Cliquez sur **"Créer"**, afin de créer votre serveur primaire.

Il faut dorénavant définir les noms de domaines, cliquez par conséquent sur "Adresse".

Ajouter l'enregistrement Adresse :

Pour ce faire, aller dans Serveurs de noms de domaines BIND, sélectionner la zone primaire(master zone) créé précédemment puis sélectionner «Adresse» dans la fenêtre qui apparaît.'(Pour plus de simplicité, enregistrer les adresses tout de suite après la création de la zone primaire car la fenêtre «Edition d'une zone primaire» apparait directement après la création).

Remplir les champs suivants :

- Nom : Indiquez- y votre Nom de Domaine. N'oubliez pas de le faire finir par un "." (dans notre cas : licence-pfe.dz. pour la machine serveur, client1.licence- pfe.dz et client2.licence-pfe.dz pour les machines clientes).

- Adresse : Indiquez les @ IP des machines concernées

(@IP serveur : 192.168.1.1; @IP Client1 :192.168.1.2 ;@IP Client2 :192.168.1.3 dans notre cas)

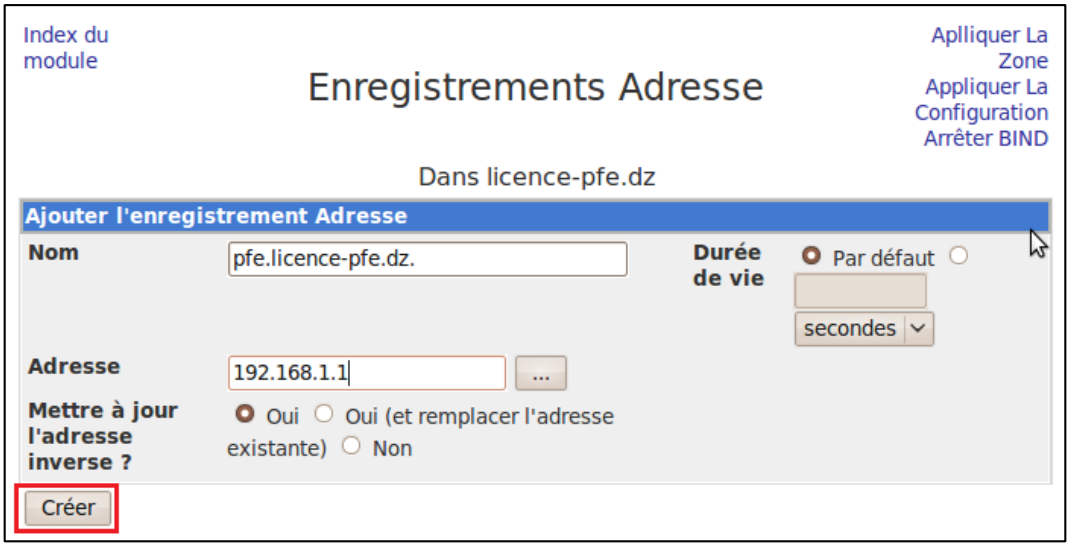

Figure 3.3.1 Fenêtre d'enregistrement des adresses

Une fois le tout complété, vous devriez avoir à peu près ceci :

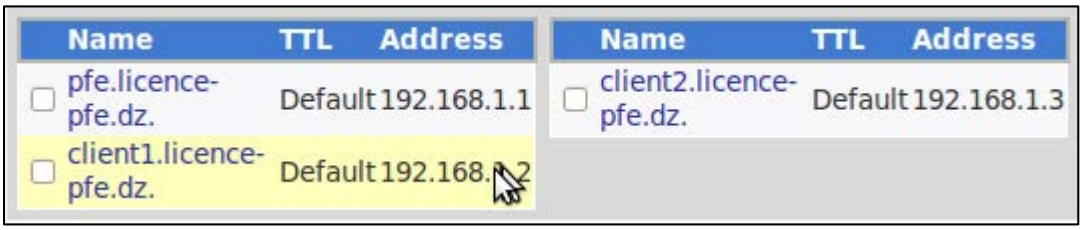

Figure 3.3.1 Liste des adresses créées

De la même manière, créer la zone Inverse : Servers -> BIND DNS Server → Create master zone (création d'un zone primaire):

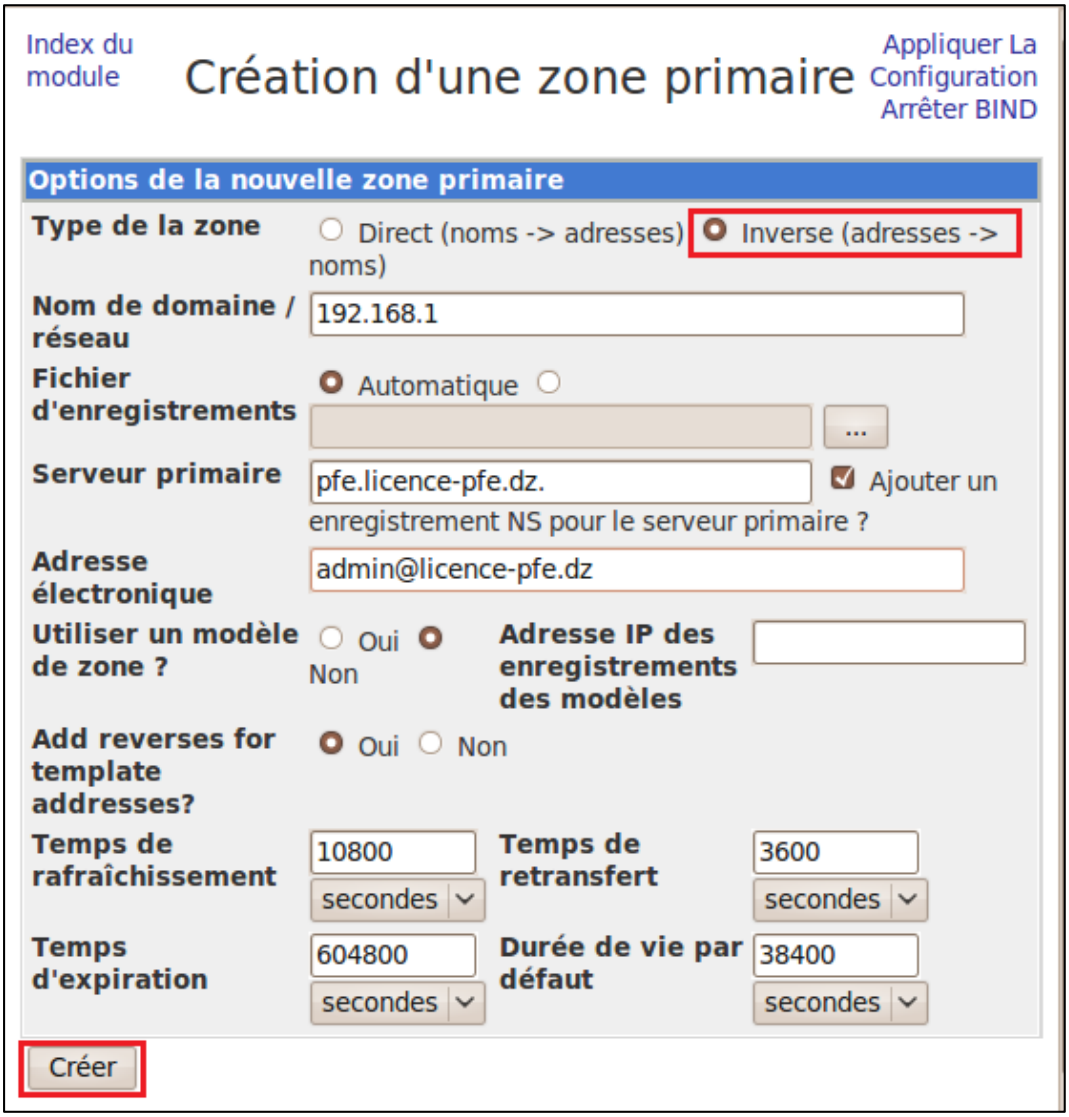

Figure 3.3.1 Fenêtre de création d'une zone primaire Inverse

Aller ensuite dans adresse inverse et créer les adresses inverses comme suit :

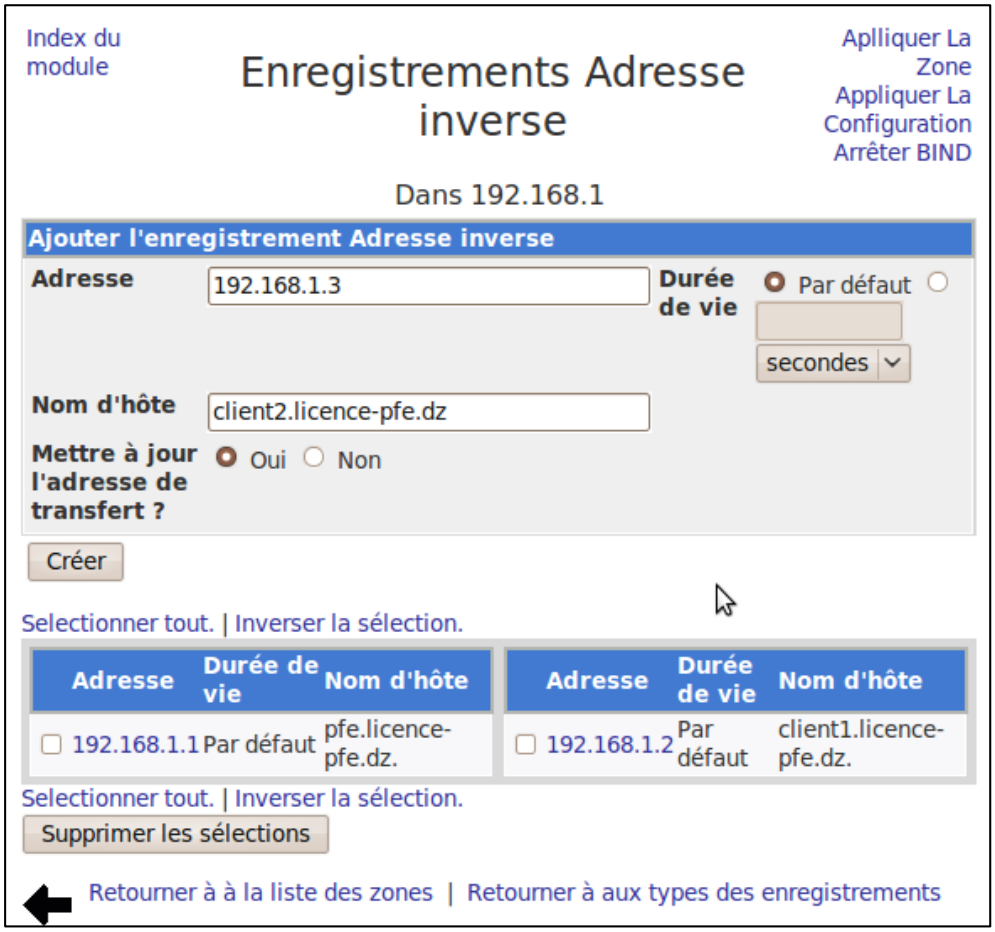

Figure 3.3.1 Enregistrement des adresses Inverse

3.3.2 Présentation et configuration du serveur SSH :

Après l'installation des paquets OpenSSH sur la machine serveur (Ubuntu 9.10), se connecter à webmin, et aller dans Serveurs->Serveur SSH. Vous aurez l'interface suivante :

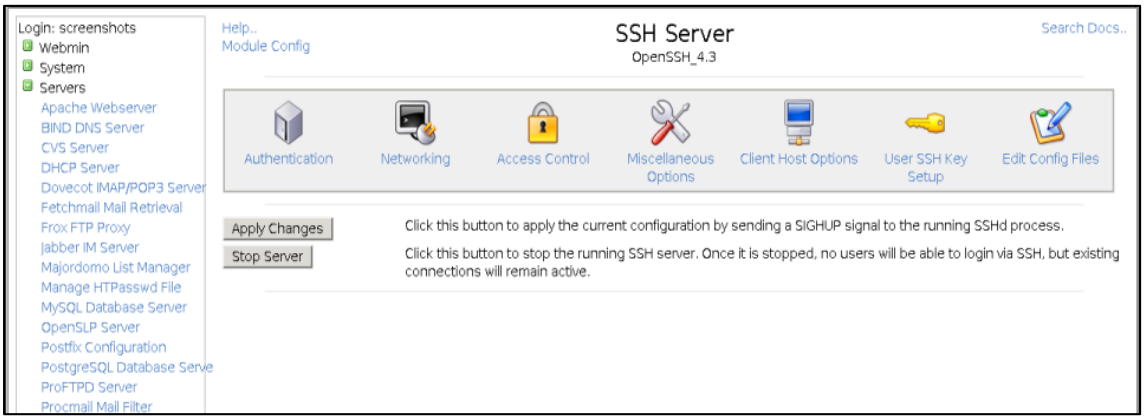

Figure 3.3.2 Interface graphique du module SSH sur Webmin

a)Restriction de l'accès au serveur SSH

Par défaut, tout utilisateur Unix sera autorisé à se connecter à distance au serveur SSH sur votre système, ou l'utiliser pour télécharger des fichiers.

Sur la page principale du module, cliquez sur l'icône ‘’contrôle d'accès’’ pour mettre en place une liste d'utilisateurs autorisés ou bien refusés.

b) Configuration Réseau

Le serveur SSH dispose de plusieurs options qui vous permettent de configurer l'adresse IP, il écoute sur le port qu'il utilise (port 22 par défaut) et les différents paramètres liés au protocole.

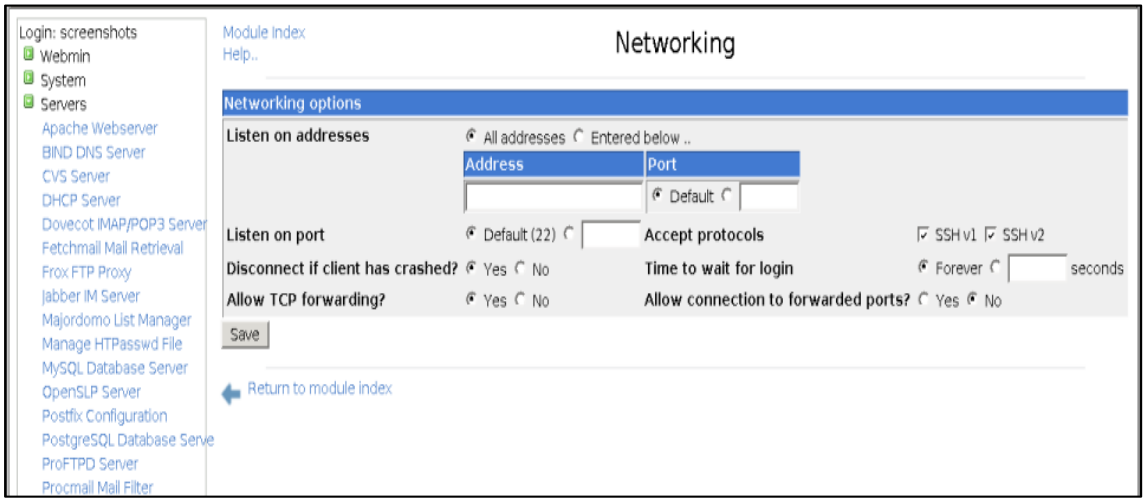

Figure 3.3.2 Options de la configuration réseau

c)Configuration du système d'authentification de service ssh

Toutes les implémentations de SSH ont des options liées à la façon dont les clients s'authentifient et les messages affichés à eux après la connexion. Plus précisément, vous pouvez autoriser ou refuser l'authentification par nom d'utilisateur et mot de passe ou par nom d'utilisateur et d'un certificat. Le service ssh utilise deux modes de connexion : par mot de passe ou par la génération de deux clés publique et privée.

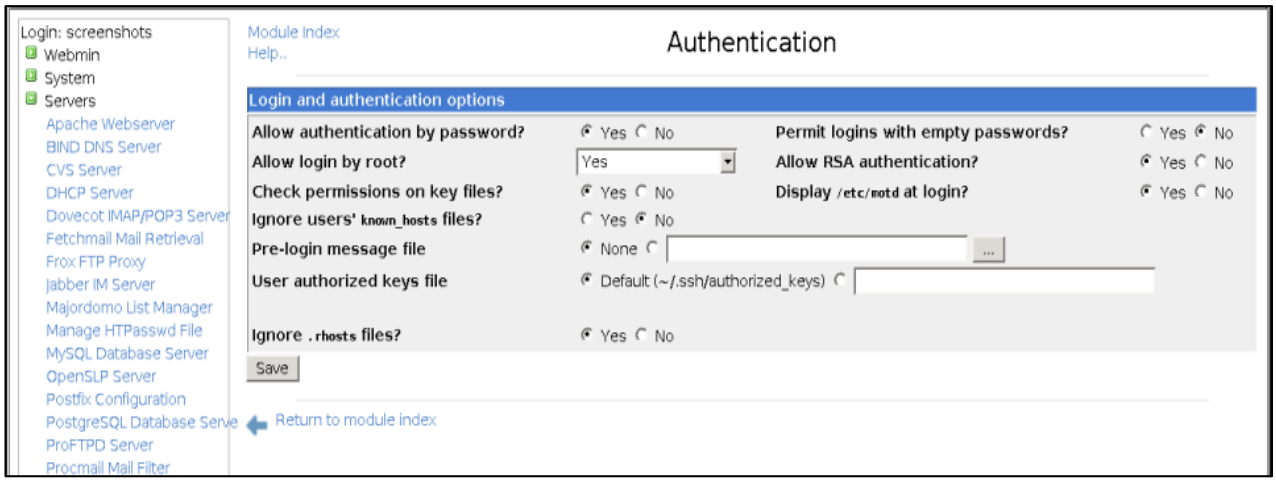

Figure 3.3.2 Options de connexion et d'authentification

3.3.3 Configuration du Serveur Apache sous Webmin

La configuration du Serveur Apache est accessible en allant dans :

Serveurs ->Serveur Web Apache

Vous arrivez sur la fenêtre principale de configuration, celle-ci se présente comme suit :

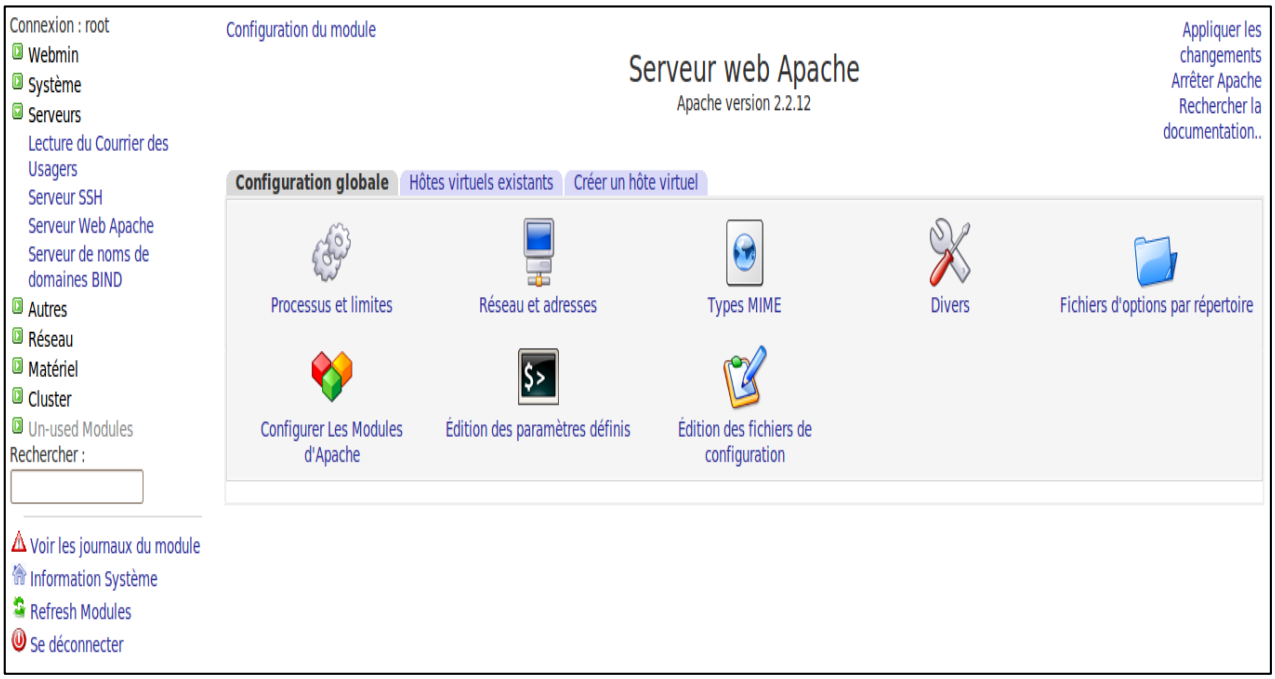

Figure 3.3.3 Fenêtre interface graphique du Serveur web Apache

a)Création et configuration d'un Serveur VirtuelHost :

La création et configuration d'un "Serveur Virtuel" va vous permettre de configurer Apache afin d'y faire fonctionner un site Web et de façon optimale.

Pour ce faire, aller dans Serveurs → Serveur Web Apache ->Créer un hôte virtuel

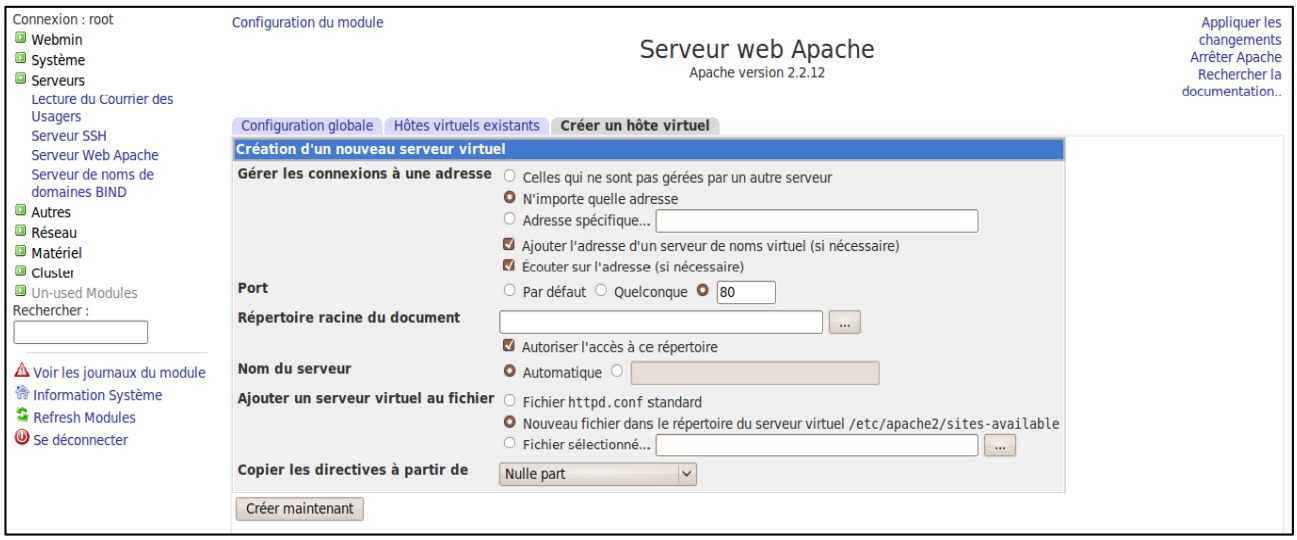

Figure 3.3.3 Fenêtre création d'un hôte virtuel

Il faut à présent renseigner les champs suivants :

- Adresse : y entrer l'adresse qu'aura communiqué l'hébergeur.
- Port : Utiliser celui que communique l'hébergeur, sinon, le laisser "par défaut".
- Répertoire Racine du Document : C'est le répertoire dans lequel sera enregistré le site, utiliser celui que fournit l'hébergeur
- Nom du serveur : entrer le nom que l'on veut donner au serveur et donc plus particulièrement au site, utiliser celui fourni par l'hébergeur, à moins que l'on possède déjà un nom de domaine, dans ce cas, l'entrer ici (licence-pfe.dz dans notre cas).

Cliquer sur **"Créer"**, après avoir saisi toutes les informations nécessaires, pour ajouter le "Serveur Virtuel".

On doit maintenant voir le nouveau **"Serveur Virtuel"** apparaître dans la liste des "Serveurs Virtuel" comme dans la figure suivante :

Figure 3.3.3 Fenêtre des hôtes virtuels existants

A présent, il faut paramétrer le serveur créé

Pour ce faire, cliquer sur l'icône correspondant à ce serveur.

On arrive ensuite sur la page des "Options du serveur virtuel".

On se retrouve alors devant une page qui permet de personnaliser certaines fonctionnalités du serveur

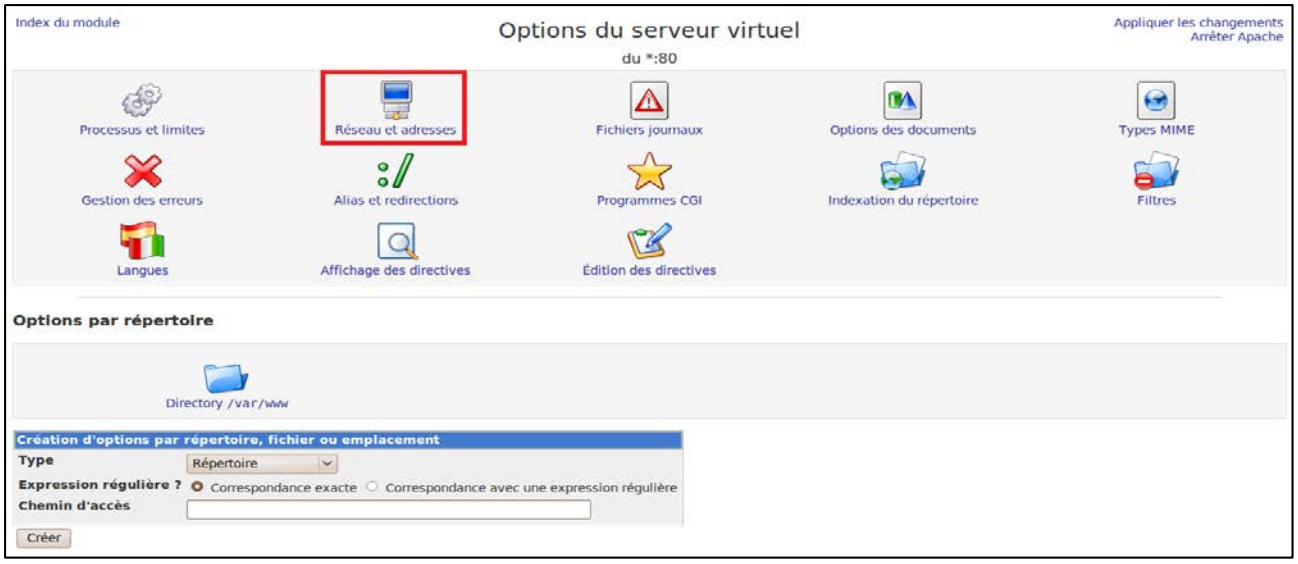

Figure 3.3.3 Fenêtre des options du serveur virtuel

On aura besoin d'accéder à l'interface de configuration du réseau : cliquer sur l'icône "Adresses et Réseaux" pour y accéder

b) Adresses et Réseaux :

- Adresse électronique de l'administrateur du serveur : permet de spécifier l'adresse e-mail du responsable de ce site.
- Nom d'hôte du serveur: Normalement, il devrait afficher automatiquement le nom de domaine que configuré lors de la création du "Serveur Virtuel". Si ce n'est pas le cas, l'indiquer.
- Noms de substitution des serveurs virtuels: on peut à l'aide de ce service créer des "Alias" au site. Si on possède d'autres noms de domaines, on peut les y entrer pour rediriger les visiteurs vers notre Nom de Domaine.

Cliquer sur "Sauvegarder" pour enregistrer les modifications.

On se retrouve ensuite sur l'écran de configuration du "Serveur Virtuel".Cliquer maintenant sur l'icône "Indexation du Répertoire" pour accéder à son sous-menu

Figure 3.3.3 Fenêtre Réseaux et adresses

On se retrouve ensuite sur l'écran de configuration du "Serveur Virtuel". Cliquer maintenant sur l'icône "Indexation du Répertoire" pour accéder à son sous-menu

c)Indexation du Répertoire :

- Options d'index de répertoire : Les sous-options doivent y être indiquées "par défaut". De même, les cases suivantes doivent être cochées, si ce n'est pas le cas, les cocher: "Allow user sorting of columms", "Output HTML header tags", Show file size", "Show file descriptions", et "Show last modified".
- Fichiers d'index du répertoire: Mettre ici les pages à charger par défaut au démarrage du site. Dans l'exemple, si un visiteur tape l'url "test.com", alors le serveur chargera dans l'ordre : "index.htm", si elle n'est pas présente sur le serveur, ce sera "index.html", etc. …

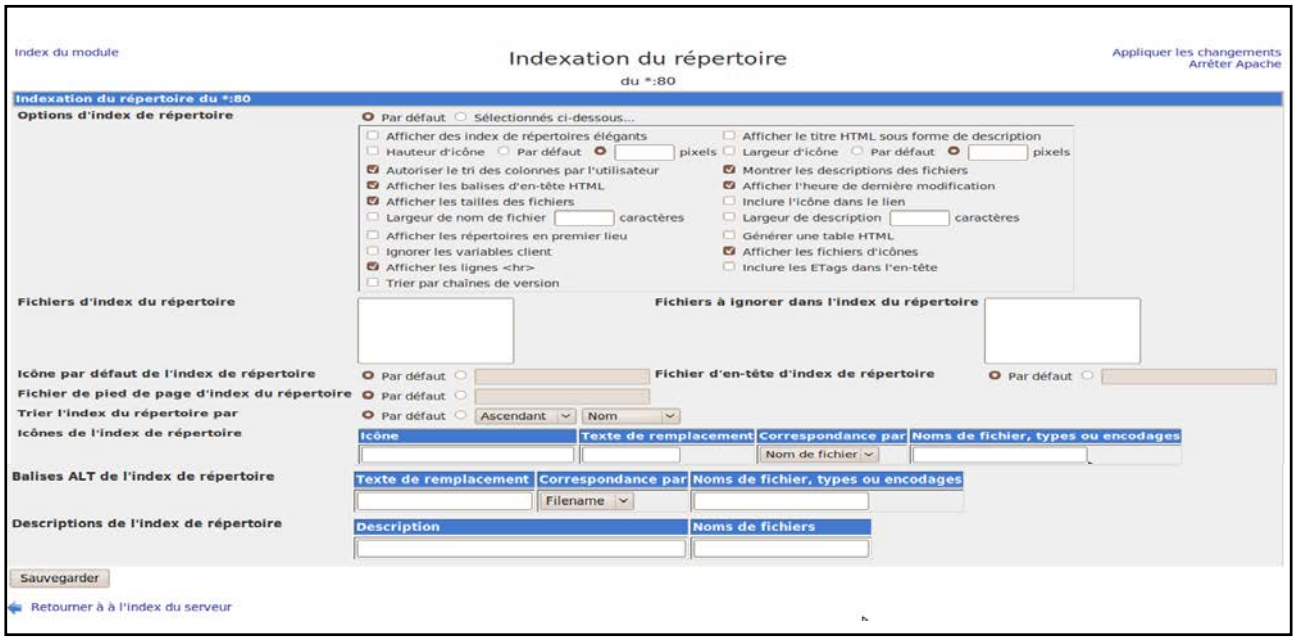

Figure 3.3.3 Fenêtre d'indexation du répertoire

Appliquer les modifications

On doit se retrouver à nouveau sur la page de configuration du **"Serveur Virtuel"**. Là, tout en haut à droite de l'écran, se trouve un lien **"Démarrer Apache"**.

Cliquer dessus pour l'activer, le **"Serveur Web"** sera activé en même temps Il ne reste plus qu'à placer les pages dans le répertoire.

3.4 Conclusion

Dans ce chapitre, nous avons montré l'importance de l'outil webmin dans la gestion et l'administration des différents services réseau. Nous somme intéressés par une configuration graphique du serveur de résolution de nom DNS, le serveur web et le serveur de connexion à distance ssh.

L'outil webmin est un outil simplifiant considérablement l'administration des différents services réseaux.

Conclusion Générale

Conclusion Générale

Le domaine des réseaux informatiques est un domaine très vaste, évoluant très vite. L'administration réseau rassemble les services Internet (*Telnet, ftp, mail,...*) pour tout le réseau qu'elle protège. Ceci nécessite de créer des comptes dédiés à ces services, ces comptes sont accessibles par un certain nombre de personnes du réseau

La gestion et l'administration de ces différents services nécessitent une compétence très importante pour un administrateur réseau.

L'administrateur trouve plusieurs difficultés pour gérer et administrer ces différents services d'une façon textuelle. Nous nous sommes intéressés dans notre mémoire à la solution basée sur la configuration graphique de ces différents services réseaux.

Notre choix s'est donc orienté vers l'outil Webmin. L'avantage de cette solution ? Outre sa simplicité, elle est peu gourmande en ressources

Après comparaison des deux types de configuration (textuelle et graphique), il nous semble évident que la configuration de tous les services cités précédemment est plus simple si elle se fait graphiquement.

L'outil webmin est un outil fondamental et indispensable dans la simplification de la tâche d'administration des différents services réseaux car c'est une interface de configuration relativement intuitive donc accessible aux experts comme aux débutants en administration réseau.

Webographie
&
Bibliographie

Webographie

[1] Couture Nadine Les architectures Client/Serveur.

http://profils.estia.fr/n.couture/ mars 2004

[2] Spécialité Système d'Information et Numérique, cours :Architecture client-serveur.

http://robert.cireddu.free.fr/

[3] Gilleron Rémi Tommasi Marc Architecture client-serveur .

http://www.grappa.univ-lille3.fr/polys/frime/

[4] OMARI Kitambala La réalisation d'une application de contrôle total des processus d'un ordinateur distant (Licence 2010)

http://www.memoireonline.com/

[5] BENAISSA Mohamed, Chapitre N°2 : Architecture client-serveur, cours, 4 avril 2011

[7] GLÜCK Olivier , Architecture et communications Client/Serveur

http://perso.univ-lyon1.fr/olivier.gluck/supports_enseig.html/

[8] BENAISSA Mohamed Tps Master –réseau et système distribué 2011

[9] Secure Shell

http://fr.wikipedia.org/

Bibliographie

[6] Cateloin Stéphane, Gallais Antoine, Stella Marc-Zwecker et Montavont Julien . Mini-manuel Réseaux Informatiques. Mini Manuel, Dunod 2012 - 192 pages

Résumé :

L'administration des réseaux informatiques évolue sans cesse et elle s'affirme aujourd'hui comme une activité clé de toute entreprise. En plus d'être constamment en fonction, ces outils d'échange de données et de partage d'information en temps réel doivent être en mesure d'offrir une confidentialité maximale et une sécurité à toute épreuve.

Nous sommes intéressés dans notre mémoire par une configuration graphique de quelques services réseaux. Notre objectif principal donc est d'utiliser un outil performant appelé webmin pour simplifier cette tâche d'administration des services réseaux.

Abstract :

Network administration is a wide field that keeps evolving, it is the engine that roles the world Because networking is used To exchange, communicate, share information..etc, besides assuring a high confidentiality and security as well under certain protocols,our final project is demonstrating the use and efficiently of one of the most powerful network administration tool called "webmin" to simplify the administration of the different services offered by networks.

ملخص

- تسيير وادارة خدمات الشبكات مجال واسع ومتشعب وهو في تطور دائم بحيث أصبح المحرك الفعال الذي يسير العالم بحكم أن الشبكات تستعمل للاتصال وتبادل المعلومات... الخ. بجانب ذلك يجب الحفاظ على السرية وضمان حماية البيانات. في مشروعنا هذا نتطرق الى وسيلة فعالة وسهلة الاستعمال لتنظيم الخدمات تسمى ويب من ونوضح مختلف خصائصها.