

But du TP

A l'issue de ce TP, l'étudiant sera capable de d'utiliser l'interface d'administration **WebMin** (SSH, Telnet).

1. 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 réseau (**Apache, Postfix, Sendmail, FTP, MySQL, PostgreSQL, Samba, SSH, BIND**, etc) depuis n'importe quel poste sur le réseau.



Dans ce TP Nous allons intéressés par le service **web apache**, service de résolution des noms **DNS (BIND), Samba** et les services de connexion à distance **telnet** et **ssh**. Webmin utilise le port 10000.

Travail demandé

Pour effectuer ce TP nous aurons besoin d'utiliser trois machines virtuelles. **Machine serveur** sous **Ubuntu 14.04.5-desktop-i386** (Adresse IP : 134.59.136.1), et deux machines cliente, **Client 1** sous Ubuntu 14.04 (Adresse IP : 134.59.136.2) et **Client 2** sous Windows 7 (Adresse IP : 134.59.136.3).

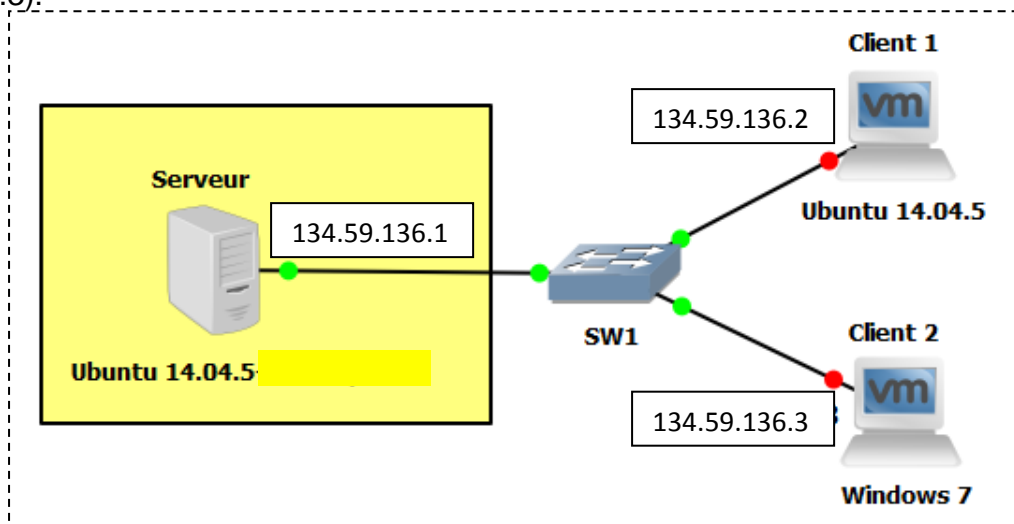


Figure 02: Topologie réseau de TP5.

Questions :

1. Tester les connexions entre les machines virtuelles (**Ping**):
2. Installer les paquets nécessaires à l'interface d'administration **WebMin** sur la machine serveur (Ubuntu 14.04.5). Puis tester la connexion à WebMin en utilisant l'URL **https://ubuntu:10000/** ou **https://localhost:10000/**.
3. Installer les paquets nécessaires de serveur **Bind** sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil **WebMin** selon les paramètres suivants : (Commande d'installation : ***apt-get install bind9***).

1- Zone principale	
Nom de domaine	<i>zarzara.edu</i>
Fichier d'enregistrement	<i>Automatique</i>
Serveur primaire	<i>umc. zarzara.edu</i>
Adresse électronique	<i>admin@ zarzara.edu</i>
Adresse	<i>134.59.136.1</i>
2- Enregistrements Adresse (serveur, client 1 et client2)	
Nom	<i>umc. zarzara.edu (serveur) client-1. zarzara.edu (Client 1) client-2. zarzara.edu (Client 2)</i>
Adresse	<i>134.59.136.1 (serveur) 134.59.136.2 (Client 1) 134.59.136.3 (Client 2)</i>
Mettre à jour l'adresse inverse	<i>oui</i>
3- De la même manière, créer la zone Inverse	

4. Installer les paquets nécessaires au serveur **OpenSSH** sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil **WebMin** (Commande d'installation : ***apt-get install ssh***).

Contrôle d'accès (autorisés)	<i>Serveur et client 1</i>
Contrôle d'accès (Refusé)	<i>Client 2</i>
Port (par défaut)	22
Connexion	<i>par mot de passe</i>

5. Installer les paquets nécessaires au serveur **web (apache)** sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil **WebMin** (Commande d'installation : ***apt-get install apache2***).
 - Créer et configurer un "**Serveur Virtuel**" ou hôte virtuel avec les paramètres suivants :

Document racine	<i>/var/www/</i>
port	<i>Par défaut</i>
Options du serveur virtuel	

e-mail du responsable	<i>Votre @ Electronique</i>
Nom d'hôte du serveur	<i>Automatique</i>
Page par défaut	<i>index.htm</i>

6. Installer les paquets nécessaires au serveur **MySQL server** sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil **WebMin**.

- Créer une nouvelle base de données avec les options suivantes :

Nom de la base	<i>fstech</i>
Set caracteres	<i>Utf8 (UTF-8-Unicode)</i>
Table 1	<i>Client 1</i>

Les champs

Field name	Data type	Type width	Key?	Auto-increment?	Allow nulls?	Unsigned?	Default value
num	int		<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	
Designation	char	50	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
adresse	char	15	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	
			<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	

7. Installer les paquets nécessaires au serveur **Samba server** sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil **WebMin**.

- Partager le dossier **/home** avec les machines clientes avec le nom de partage : **mon-dossier** et avec les permissions : **777**.

8. Installer les paquets nécessaires au serveur **Telnet** sur la machine serveur (Ubuntu 14.04.5), puis procéder à la configuration de ce dernier en utilisant l'outil **WebMin**.

- Quel est l'intérêt de **SSH** par rapport à **Telnet** ?
- Tester la console avec : Telnet 127.0.0.1.
- Etablir une connexion Telnet à la machine **Client 1**.

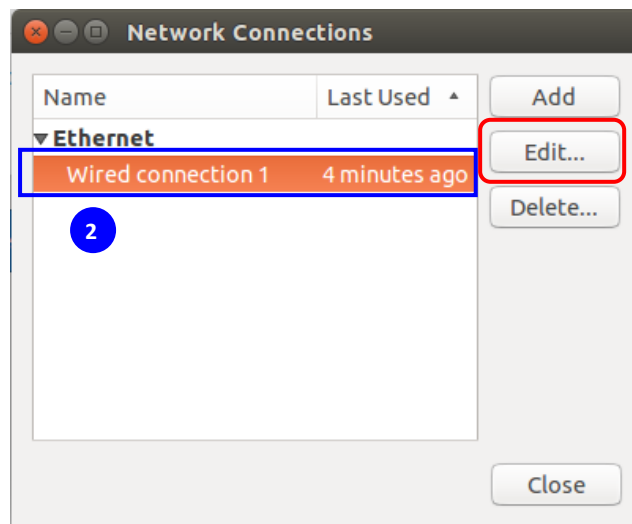
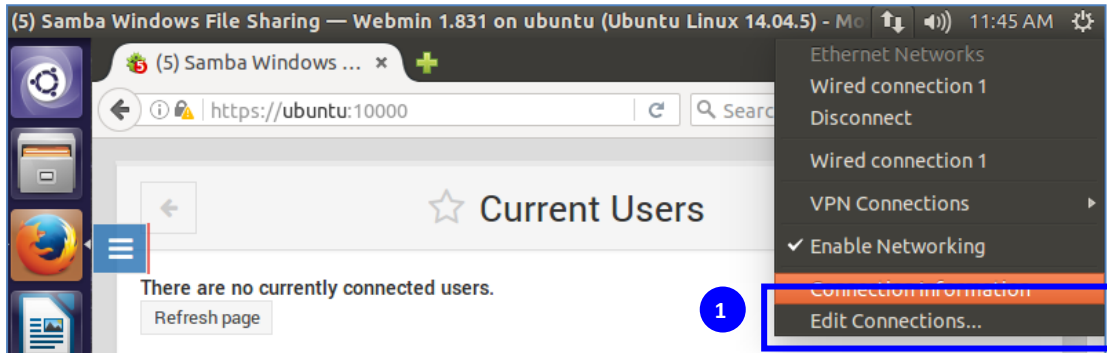
Solution

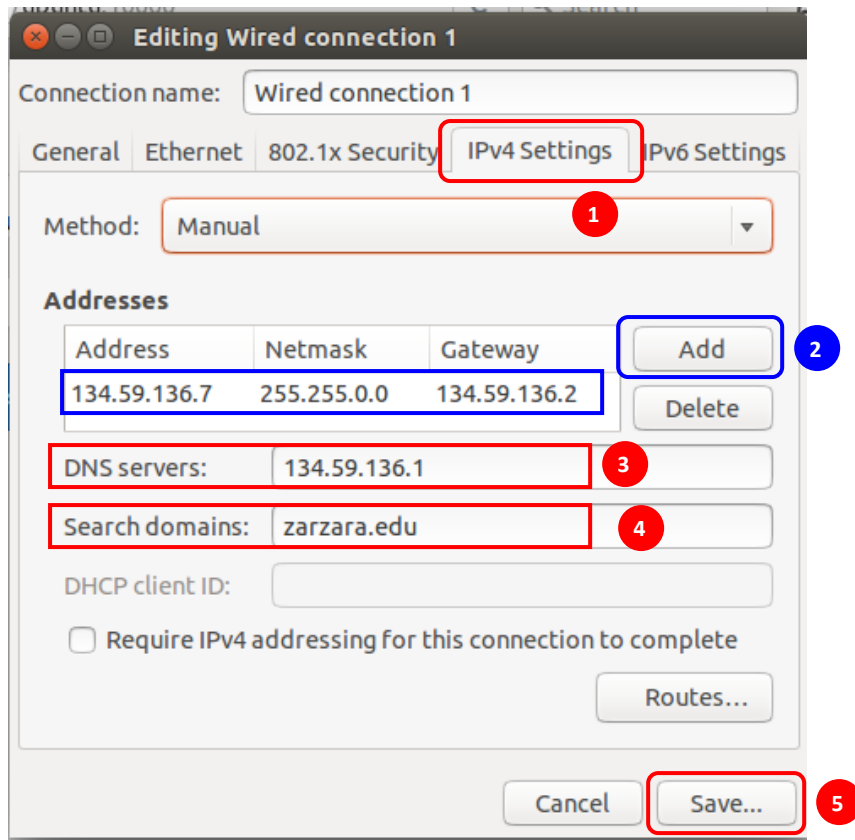
Etape 01 : Configuration du réseau

- Configuration de l'adresse IP statique et de la passerelle sous Ubuntu.

```
root@ubuntu:~# ifconfig eth0 134.59.136.7
root@ubuntu:~# route add default gw 134.59.136.2
root@ubuntu:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:3b:ba:60
          inet addr:134.59.136.7  Bcast:134.59.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20c:29ff:fe3b:ba60/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4045 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3813 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4724293 (4.7 MB)  TX bytes:293026 (293.0 KB)
          Interrupt:19 Base address:0x2000
```

- Deuxième méthode (via l'interface graphique)



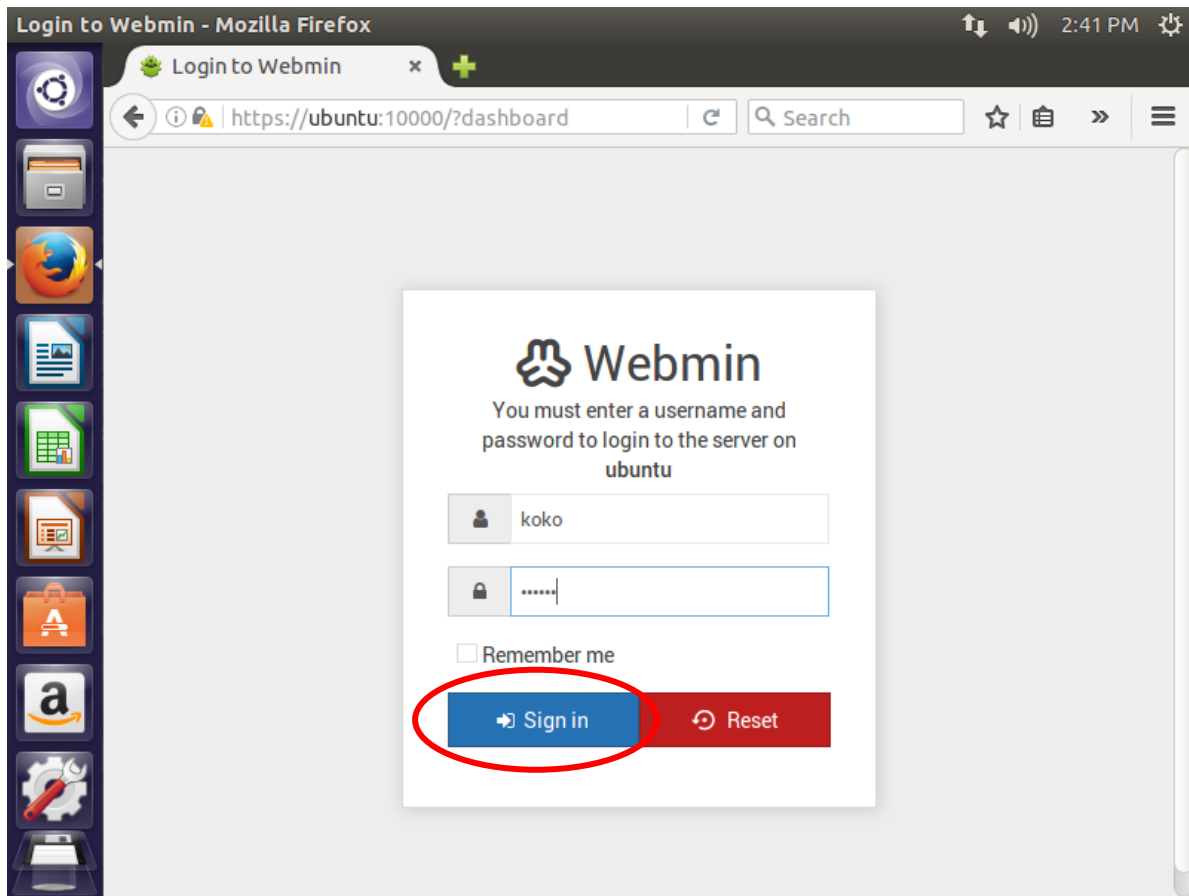


Etape 02 : Installation de l'outil WebMin

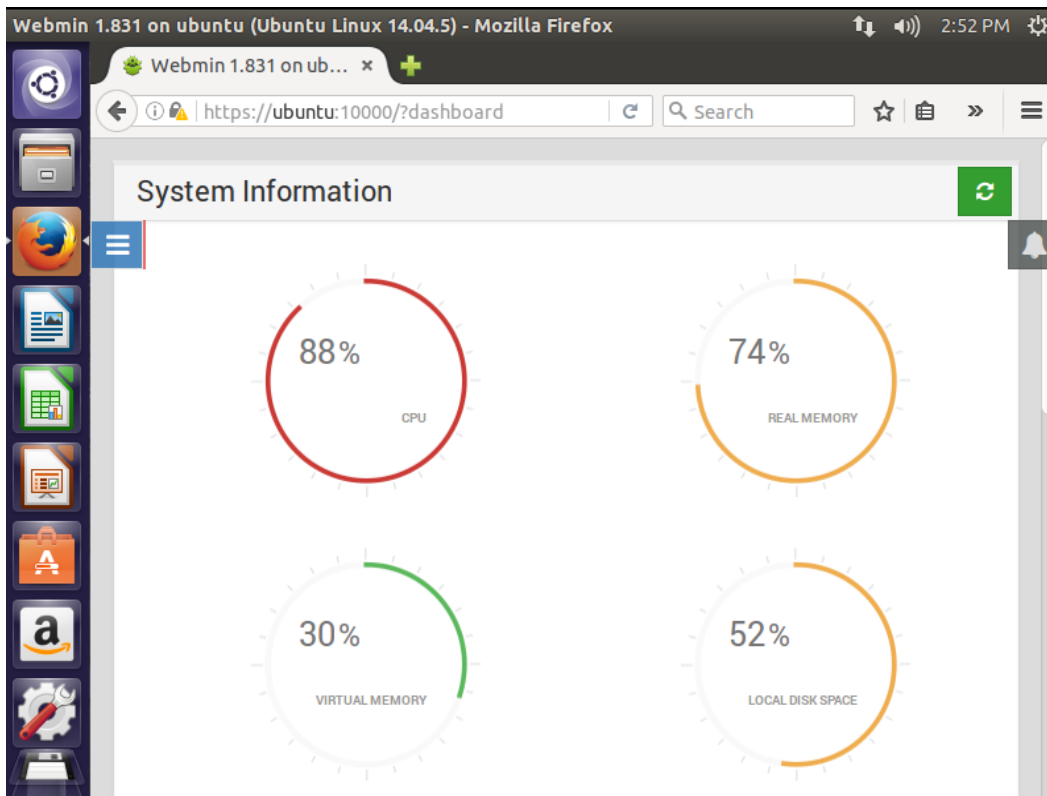
1. Installer les paquets nécessaires à **WebMin** en tapant les commandes suivantes dans le terminal:

```
sudo sh -c 'echo "deb http://download.webmin.com/download/repository sarge contrib" > /etc/apt/sources.list.d/webmin.list'  
wget -qO - http://www.webmin.com/jcameron-key.asc | sudo apt-key add -  
sudo apt-get update  
sudo apt-get install webmin
```

2. Pour se connecter à webmin ; taper l'URL suivante: **https://ubuntu:10000/** et se loguer avec le compte **root** crée "Ex : **koko**" et le mot de passe de la session Ex : « **123456** ».

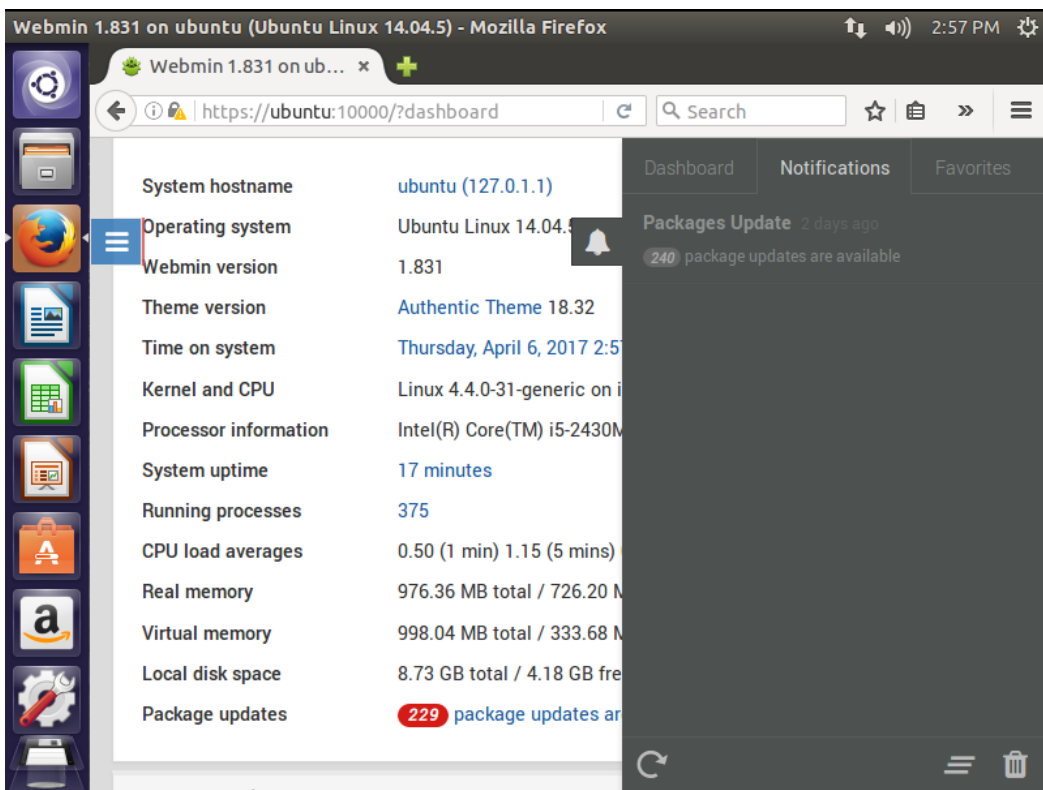
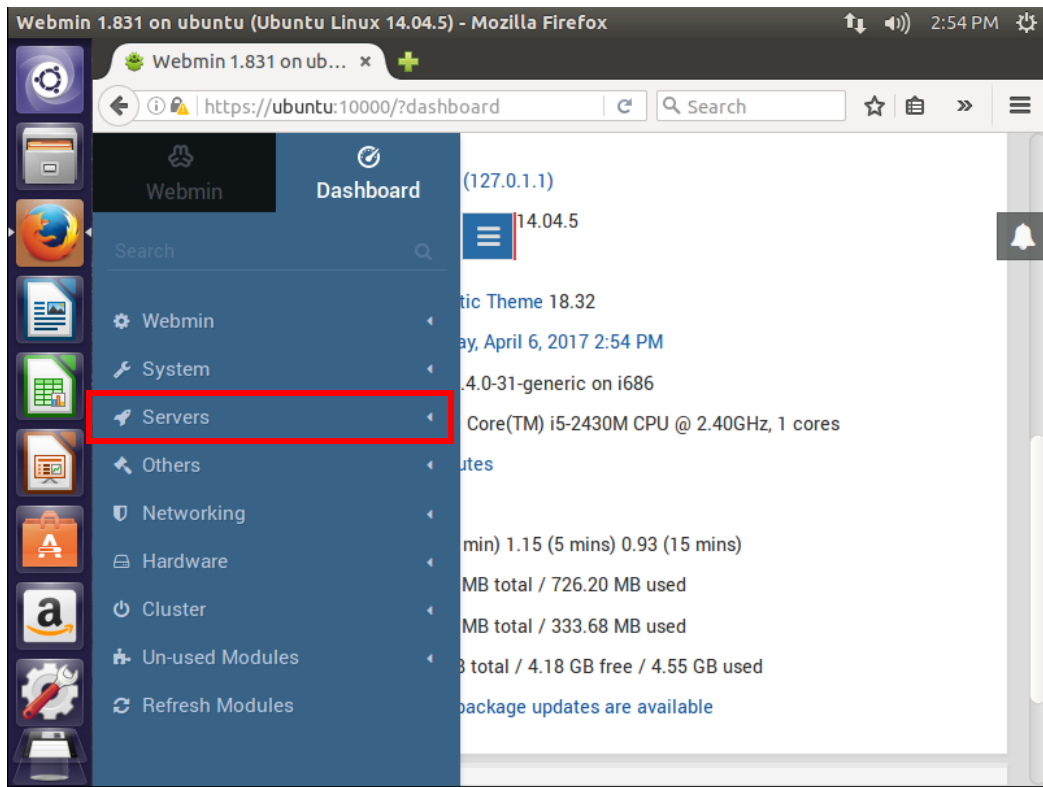


3. La figure suivante est la page permettant l'accès à webmin après authentification (*Page d'accueil de WebMin*).



The screenshot shows the Webmin 1.831 dashboard on Ubuntu Linux 14.04.5. The main section is titled "System Information" and contains a table of system details:

System hostname	ubuntu (127.0.1.1)
Operating system	Ubuntu Linux 14.04.5
Webmin version	1.831
Theme version	Authentic Theme 18.32
Time on system	Thursday, April 6, 2017 2:52 PM
Kernel and CPU	Linux 4.4.0-31-generic on i686
Processor information	Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 1 cores
System uptime	17 minutes
Running processes	375
CPU load averages	0.50 (1 min) 1.15 (5 mins) 0.93 (15 mins)
Real memory	976.36 MB total / 726.20 MB used
Virtual memory	998.04 MB total / 333.68 MB used
Local disk space	8.73 GB total / 4.18 GB free / 4.55 GB used
Package updates	229 package updates are available



Etape 03 : Installation des services réseau « Bind », « Apache », « SQL server », « Samba » et « Open ssh ».

1. Installation de service de noms DNS (**BIND9** ou *Berkley Internet Naming Daemon*) : Il faut passer en mode root (**sudo -i**), puis, tapez le mot de passe du compte **root**, puis lancez l'installation avec la commande (**apt-get install bind9**).

```
root@ubuntu: ~  
koko@ubuntu:~$ sudo -i 1  
[sudo] password for koko:  
root@ubuntu:~# apt-get install bind9 2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  bind9-host bind9utils dnstools libbind9-90 libdns100 libisc95 libisccc90  
  libisccfg90 liblwres90  
Suggested packages:  
  bind9-doc rblcheck  
The following NEW packages will be installed:  
  bind9 bind9utils  
The following packages will be upgraded:  
  bind9-host dnstools libbind9-90 libdns100 libisc95 libisccc90 libisccfg90  
  liblwres90  
8 upgraded, 2 newly installed, 0 to remove and 230 not upgraded.  
Need to get 1,436 kB of archives.  
After this operation, 1,588 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty-updates/main bind9-host i386 1  
:9.9.5.dfsg-3ubuntu0.13 [45.1 kB]  
3% [1 bind9-host 42.2 kB/45.1 kB 94%]
```

A la fin de l'installation, le service bind9 démarre automatiquement

```
#  
* Starting domain name service... bind9 [ OK ]  
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...  
Processing triggers for ureadahead (0.100.0-16) ...  
Processing triggers for ufw (0.34~rc-0ubuntu2) ...  
root@ubuntu:~#
```

2. Installation de **service web 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. un **serveur http** est un serveur hébergeant un ou plusieurs sites Web 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 l'installation d'un serveur apache on écrit la commande suivante :

```
koko@ubuntu:~$ sudo -i  
[sudo] password for koko:  
root@ubuntu:~# apt-get install apache2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done
```

3. Installation de **serveur SSH** : **Secure Shell** (SSH) est à la fois un programme informatique et un protocole de communication sécurisée. Il permet de se connecter à une machine distante avec une liaison sécurisée. Les données sont cryptées entre machines. Il permet d'exécuter des commandes sur un serveur distant.

Pour lancer l'installation d'un serveur ssh on écrit la commande suivante :

```
root@ubuntu:~# apt-get install ssh  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done
```

```
root@ubuntu:~# status ssh  
ssh start/running, process 1121
```

Remarque:

Pour se connecter depuis un poste Windows au **serveur SSH**, il faut utiliser un "**client SSH**". Exemples de clients SSH :

- **Putty** (client telnet / SSH) : <http://www.putty.org/>
- **WinSCP** (Transfert de fichiers - interface graphique) : <http://winscp.net/eng/docs/lang:fr>
- **JuiceSSH** sur Android : <https://play.google.com/store/apps/>

4. Installation de **serveur Samba** : Samba est une application qui permet d'utiliser sous Linux le protocole **SMB** (« *Session Message Block* ») également appelé le protocole *NetBIOS* ou *LanManager*. Le protocole SMB est utilisé par Microsoft Windows pour partager des disques et des imprimantes. En utilisant les outils Samba d'*Andrew Tridgel*, les systèmes Unix (Linux inclus) peuvent également partager des disques et des imprimantes avec des hôtes Windows.

Pour lancer l'installation d'un serveur samba on écrit la commande suivante :

```
root@ubuntu:~# apt-get install -y -f samba  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done
```

```
root@ubuntu:~# service samba status  
nmbd start/running, process 1763  
smbd start/running, process 720
```

5. Installation de **serveur SQL** : Le SQL server est un SGBD (*Système de Gestion de Base de Données*) qui possède toutes les caractéristiques pour pouvoir accompagner l'utilisateur dans la manipulation, le contrôle, le tri, la mise à jour, et bien d'autres actions encore, de bases de données grâce au langage SQL (*Structured Query Language*).

Pour lancer l'installation d'un serveur SQL on écrit la commande suivante :

```
root@ubuntu:~# apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
root@ubuntu:~# service mysql status
mysql start/running, process 1279
root@ubuntu:~#
```

6. Installation de **serveur Telnet** : (*Terminal Network*). Le protocole Telnet est un protocole standard d'Internet permettant de relier un client (système composé d'un affichage et d'un clavier) à un interpréteur de commande (côté serveur). Il permet d'ouvrir une connexion avec un serveur à distance comme si on le consultait sur place. Telnet est un protocole non sécurisé, les informations transitant n'étant pas cryptées contrairement à **SSH**.

Pour lancer l'installation d'un serveur Telnet on écrit la commande suivante :

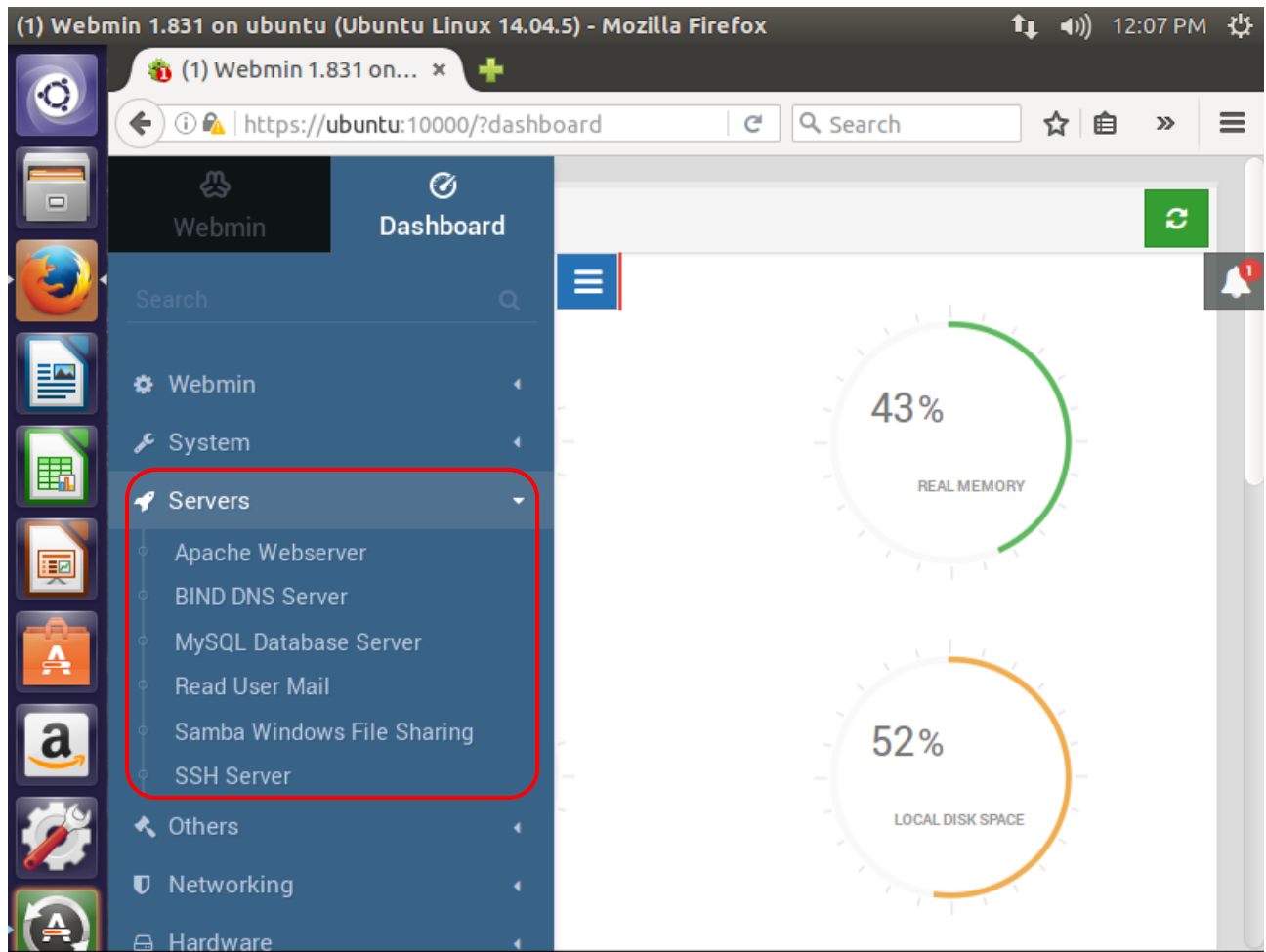
```
koko@ubuntu:~$ sudo -i
[sudo] password for koko:
root@ubuntu:~# apt-get install xinetd telnetd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  telnetd xinetd
0 upgraded, 2 newly installed, 0 to remove and 223 not upgraded.
Need to get 140 kB of archives.
After this operation, 438 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu/ trusty/main xinetd i386 1:2.3.15-3ubu
ntu1 [102 kB]
24% [1 xinetd 33.0 kB/102 kB 32%] 3,644 B/s 29s
```

Remarque:

Pour quitter le mode root, tapez la commande **exit** :

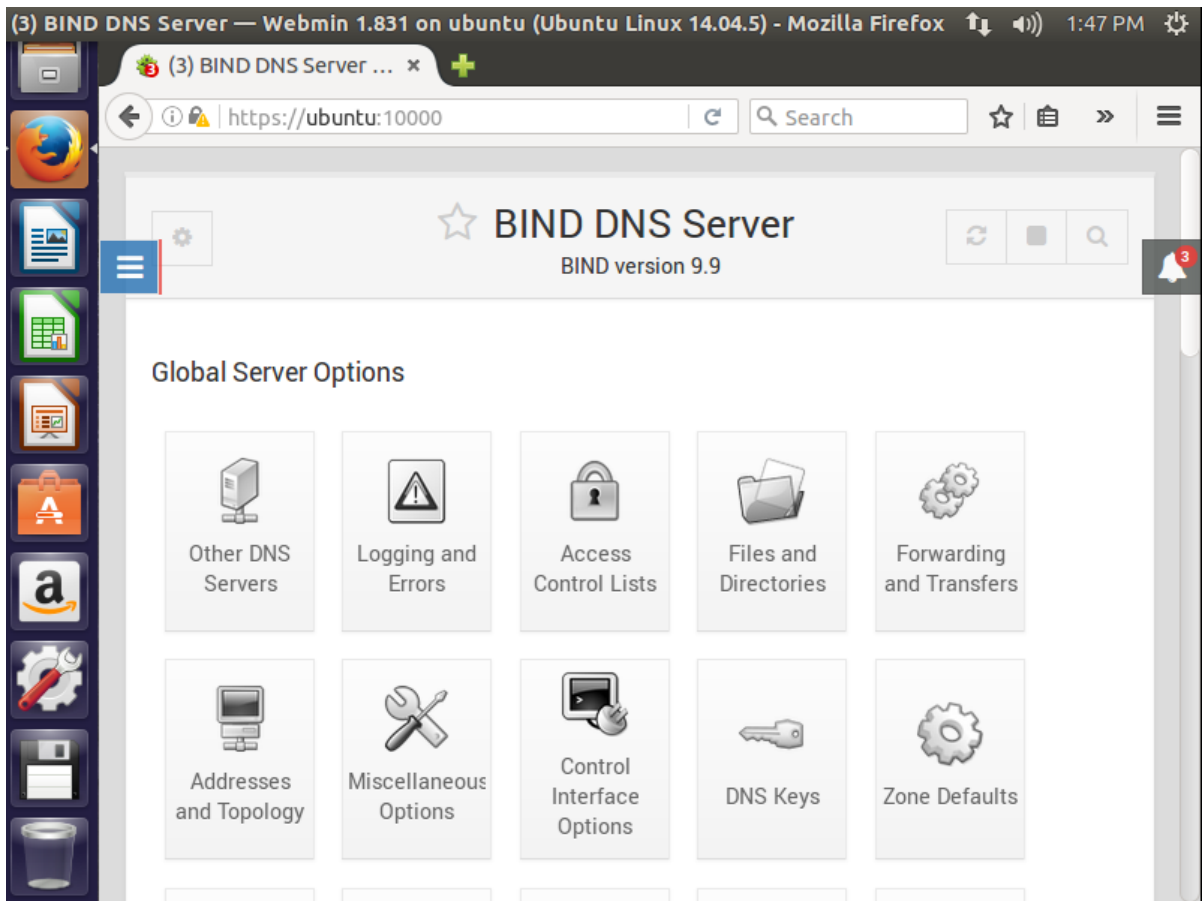
```
root@ubuntu:~# exit
logout
koko@ubuntu:~$
```

A la fin de l'installation, les serveurs s'affichent automatiquement dans la liste des serveurs dans l'onglet servers de webmin. Nous pouvons donc procéder à la configuration de ces derniers via l'interface de Webmin et à partir de n'importe quelle machine du réseau.



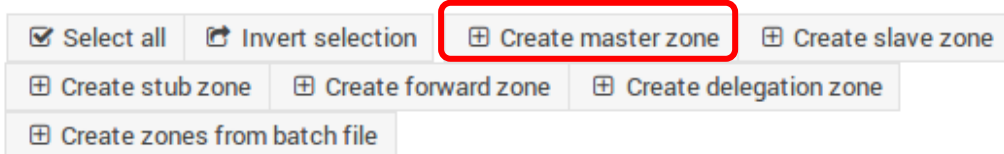
Etape 04 : Configuration des serveurs via Webmin

1. Configuration du serveur Bind via l'interface Webmin.



- **Création d'une nouvelle zone**

Existing DNS Zones



☆ Create Master Zone

New master zone options

Zone type **1** Forward (Names to Addresses) Reverse (Addresses to)

Domain name / Network **2**

Records file **3** Automatic

Master server **4** 5 Add NS record

Email address **6**

Use zone template? Yes No IP address for template records

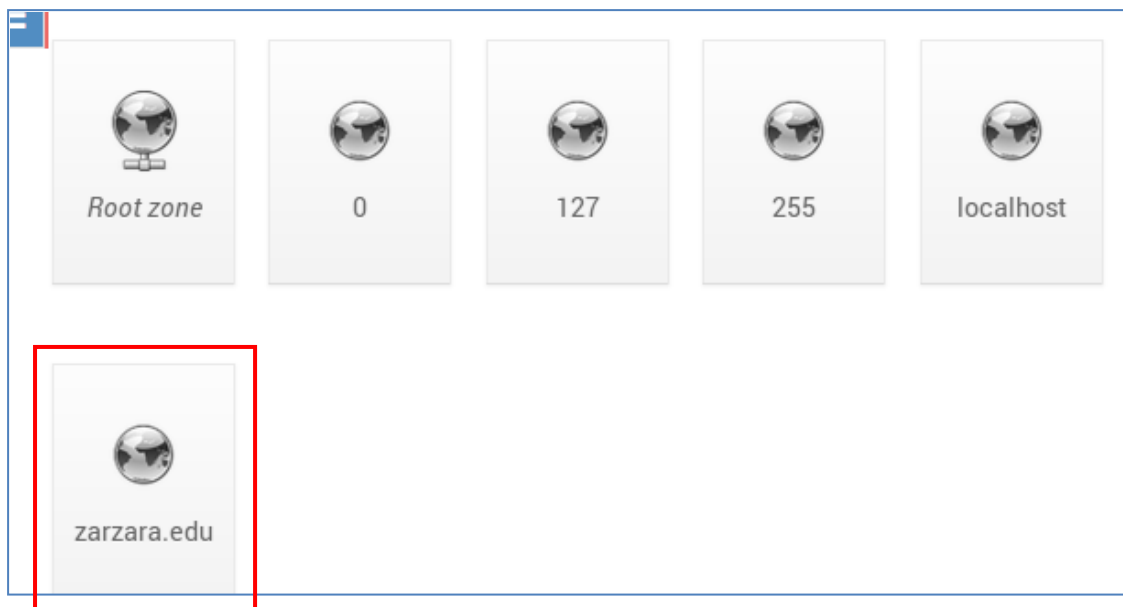
Add reverses for template addresses? Yes No

Refresh time seconds

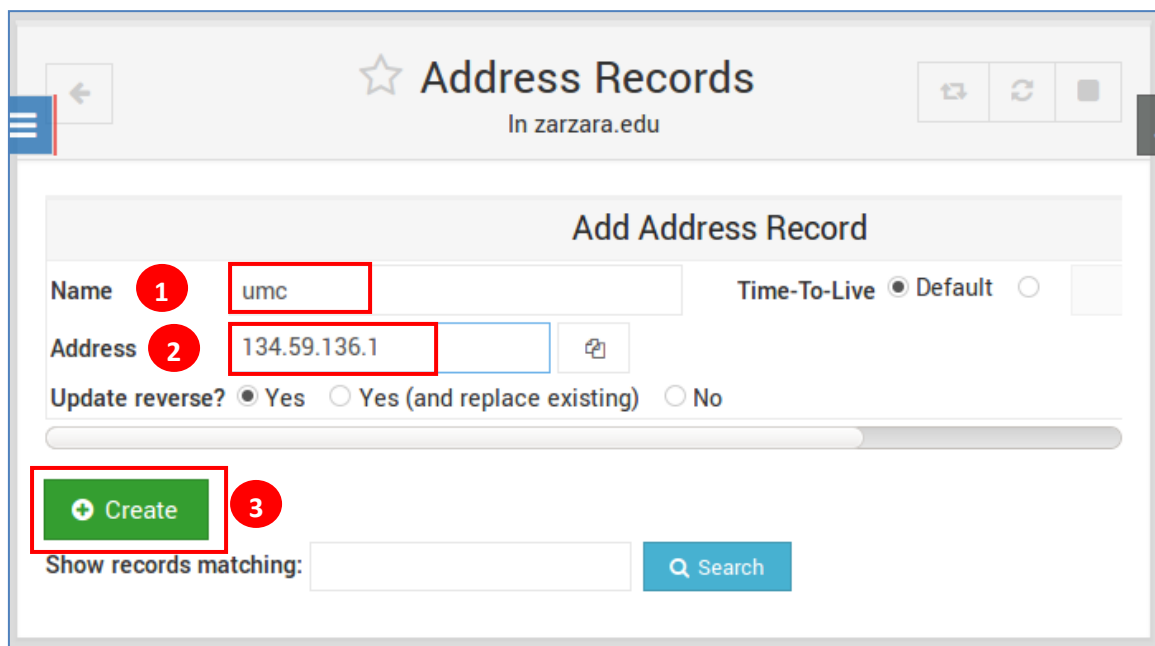
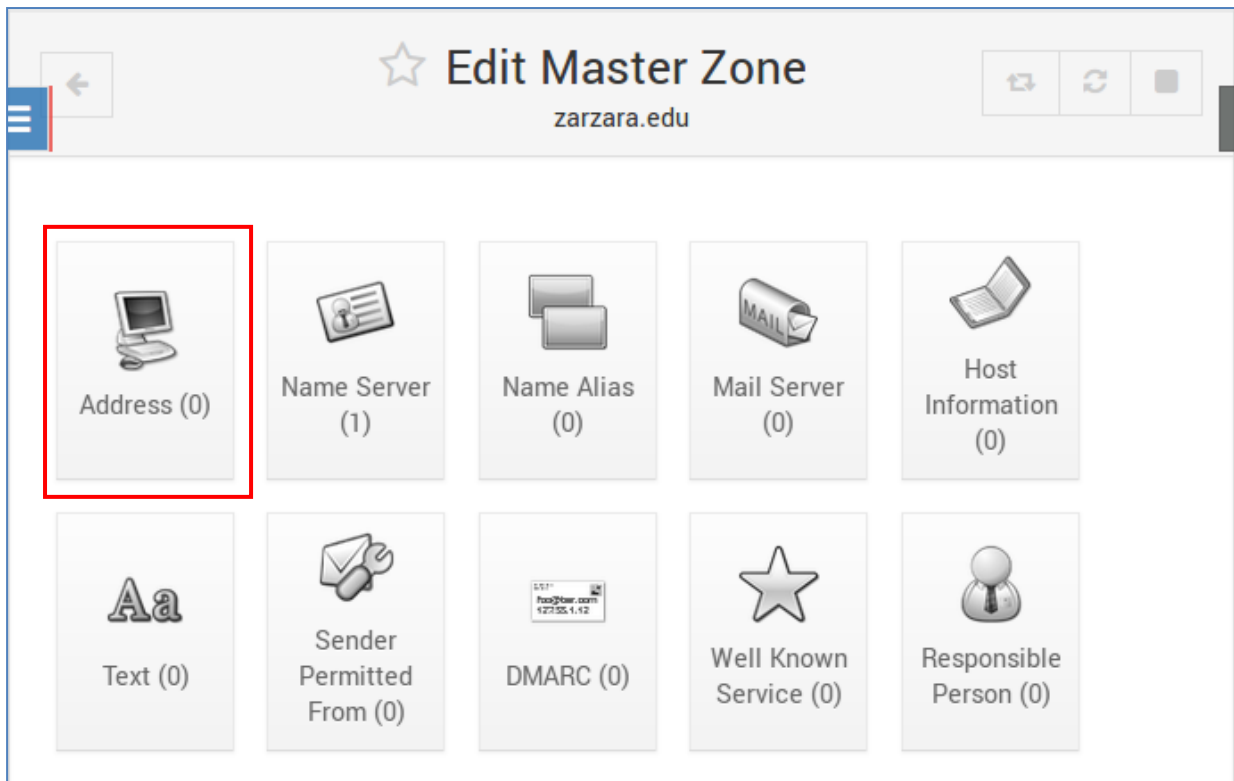
Expiry time seconds

7

Lisez tous le reste par défaut, puis cliquez sur l'icône de la nouvelle zone.



Nous pouvons à présent ajouter des adresses de machines.



La liste des enregistrements créer dans la zone zarzara.edu

Name	Type	TTL	Values
<input type="checkbox"/> zarzara.edu	NS	Default	ubuntu.
<input type="checkbox"/> umc.zarzara.edu	A	Default	134.59.136.1
<input type="checkbox"/> client-1.zarzara.edu	A	Default	134.59.136.2
<input type="checkbox"/> client-2.zarzara.edu	A	Default	134.59.136.3

Select all Invert selection

Nous pouvons tester le fonctionnement de serveur **Bind** avec la commande « **nslookup** ».

Remarque : il faut démarrer le serveur avant de lancer le test.

```
root@ubuntu:~# service bind9 start
* Starting domain name service... bind9
root@ubuntu:~#
```

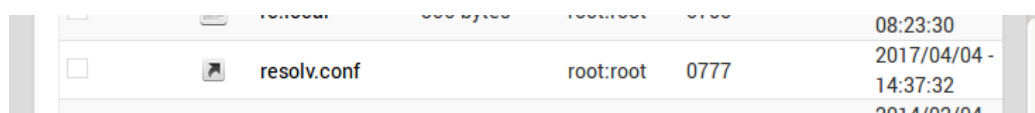
```
root@ubuntu:~# /usr/sbin/named
root@ubuntu:~# /usr/sbin/named
root@ubuntu:~# nslookup
> server localhost
Default server: localhost
Address: 127.0.0.1#53
>
```

```
> umc.zarzara.edu
Server:          localhost
Address:         127.0.0.1#53

Name:   umc.zarzara.edu
Address: 134.59.136.1
```

Pour changer l'adresse IP du serveur DNS il faut aller vers le fichier **/etc/resolv.conf**

- Dans webmin, aller dans «**Others/File Manager**,
- Rechercher le fichier **resolv.conf** dans le répertoire **etc**.



- Avant de modifier le fichier il faut changer les droits sur ce dernier.

```
root@ubuntu:~# chmod 664 /run/resolvconf/resolv.conf
```

```
resolv.conf x
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 134.59.136.7
search zarzara.edu
```

```
root@ubuntu:~# nslookup
> ubuntu
Server:         134.59.136.7
Address:       134.59.136.7#53

Name:   ubuntu.zarzara.edu
Address: 134.59.136.7
>
```

Test de fonctionnement de
serveur DNS

```
> ubuntu.zarzara.edu
Server:         134.59.136.7
Address:       134.59.136.7#53

Name:   ubuntu.zarzara.edu
Address: 134.59.136.7
>
```

```
> client.zarzara.edu
Server:         134.59.136.7
Address:       134.59.136.7#53

Name:   client.zarzara.edu
Address: 134.59.136.4
>
```

```
root@ubuntu:~# ping ubuntu.zarzara.edu
PING ubuntu.zarzara.edu (134.59.136.7) 56(84) bytes of data.
64 bytes from 134.59.136.7: icmp_seq=1 ttl=64 time=0.025 ms
64 bytes from 134.59.136.7: icmp_seq=2 ttl=64 time=0.074 ms
64 bytes from 134.59.136.7: icmp_seq=3 ttl=64 time=0.101 ms
64 bytes from 134.59.136.7: icmp_seq=4 ttl=64 time=0.112 ms
```

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

☆ Create Master Zone

New master zone options

Zone type Forward (Names to Addresses) Reverse (Addresses to)

Domain name / Network

Records file Automatic

Master server Add NS record

Email address

Use zone template? Yes No IP address for template records

Add reverses for template addresses? Yes No

Refresh time seconds Transfer retry time

Expiry time seconds Negative cache time

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

☆ Reverse Address Records
In 134.59

Add Reverse Address Record

Address Time-To-Live Default

Hostname

Update forward? Yes No

Show records matching:

	Address	TTL	Hostname
<input type="checkbox"/>	134.59.136.7	Default	ubuntu.
<input type="checkbox"/>	134.59.136.4	Default	Client.

Select all Invert selection

The screenshot shows a network configuration interface with several IP address ranges represented by globe icons and text labels below them:

- Root zone
- 0
- 127
- 134.59** (highlighted with a red box)
- 255
- localhost
- zarzara.edu

```
C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.
C:\Users\keltoum>ping ubuntu.zarzara.edu

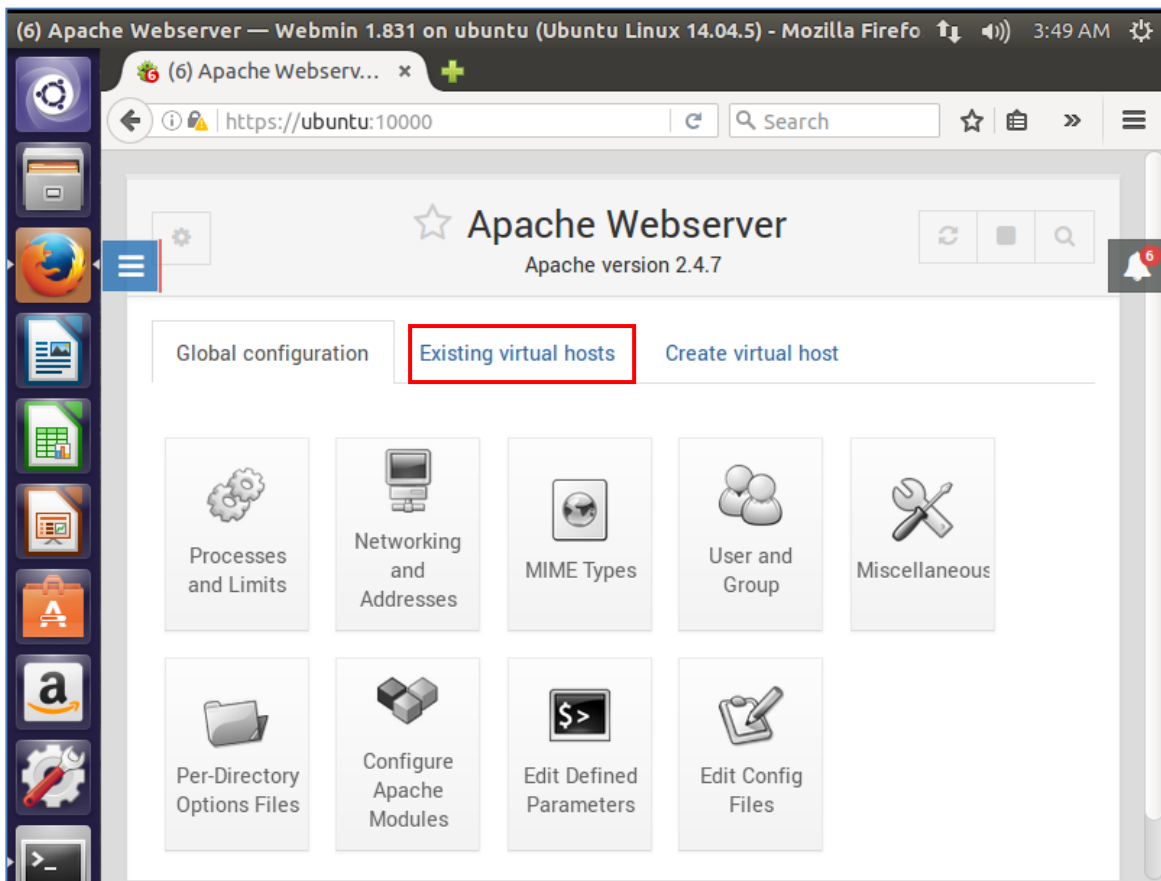
Envoi d'une requête 'ping' sur ubuntu.zarzara.edu [134.59.136.7] avec 32 octets
de données :
Réponse de 134.59.136.7 : octets=32 temps=1 ms TTL=64
Réponse de 134.59.136.7 : octets=32 temps=1 ms TTL=64
Réponse de 134.59.136.7 : octets=32 temps<1ms TTL=64
Réponse de 134.59.136.7 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 134.59.136.7:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms

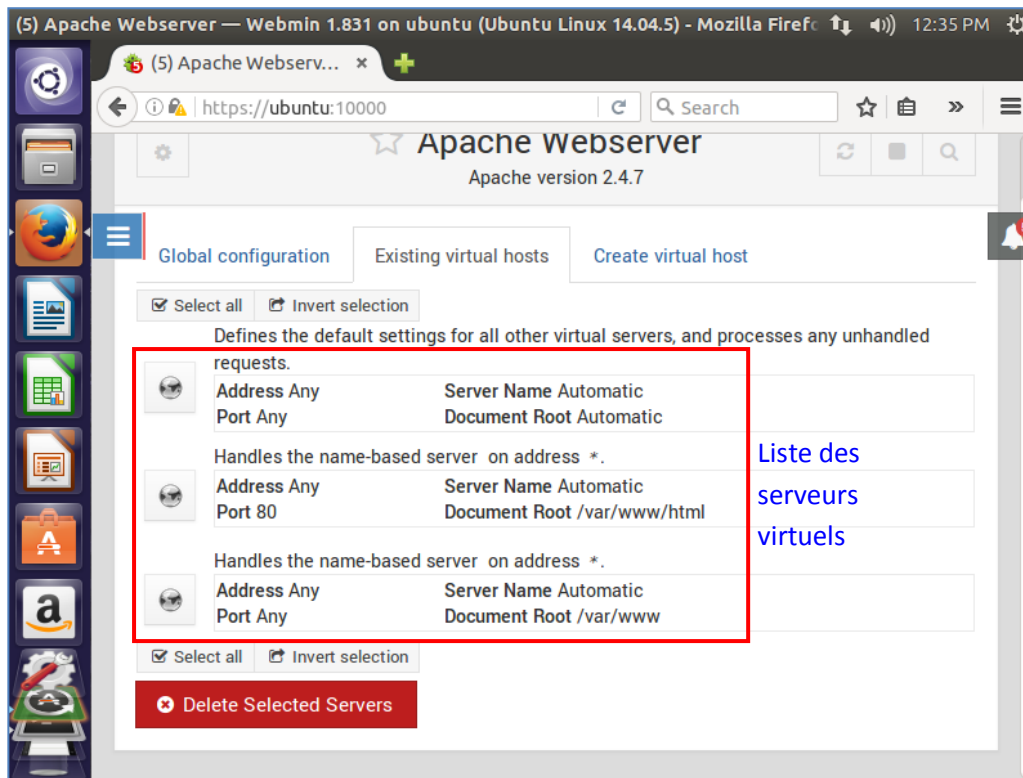
C:\Users\keltoum>_
```

2. Configuration du serveur web Apache sous webmin

Ouvrir la fenêtre principale de configuration à partir de webmin, celle-ci se présente comme suit :



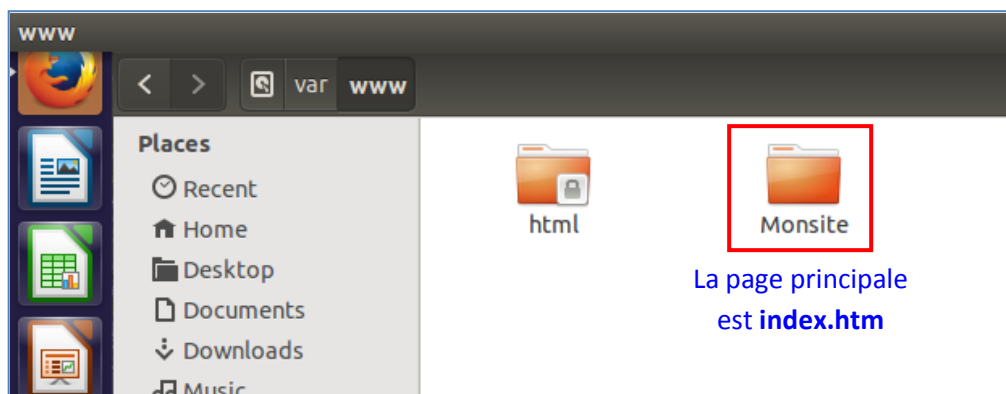
Pour voir la liste des serveurs virtuels, clic sur l'onglet « **Existing virtual hosts** ». c'est à partir de cette fenêtre que vous pouvez créer votre serveur virtuel hébergeant votre site.



• Créations d'un serveur virtuel

- La création et la configuration d'un "Serveur Virtuel" va vous permettre de configurer Apache afin d'y faire fonctionner un site Web et de façon optimale.
- Mais avous vous devez copier le dossier de votre site dans le répertoire (var/www). Pour ce faire, il faut modifier les permissions sur ce dossier de la manière suivante.

```
root@ubuntu:~# chmod 777 /var/www/  
root@ubuntu:~#
```



Global configuration Existing virtual hosts **Create virtual host** 1

Handle connections to address Those not handled by another virtual host Specific address .. 134.59.134.7 2

Listen on address (if needed)

Port 3 Default Any 80

Document Root 4 /var/www/Monsite

Allow access to this directory

Server Name Automatic

Add virtual server to file Standard *httpd.conf* file New file under virtual servers directory

Copy directives from Nowhere

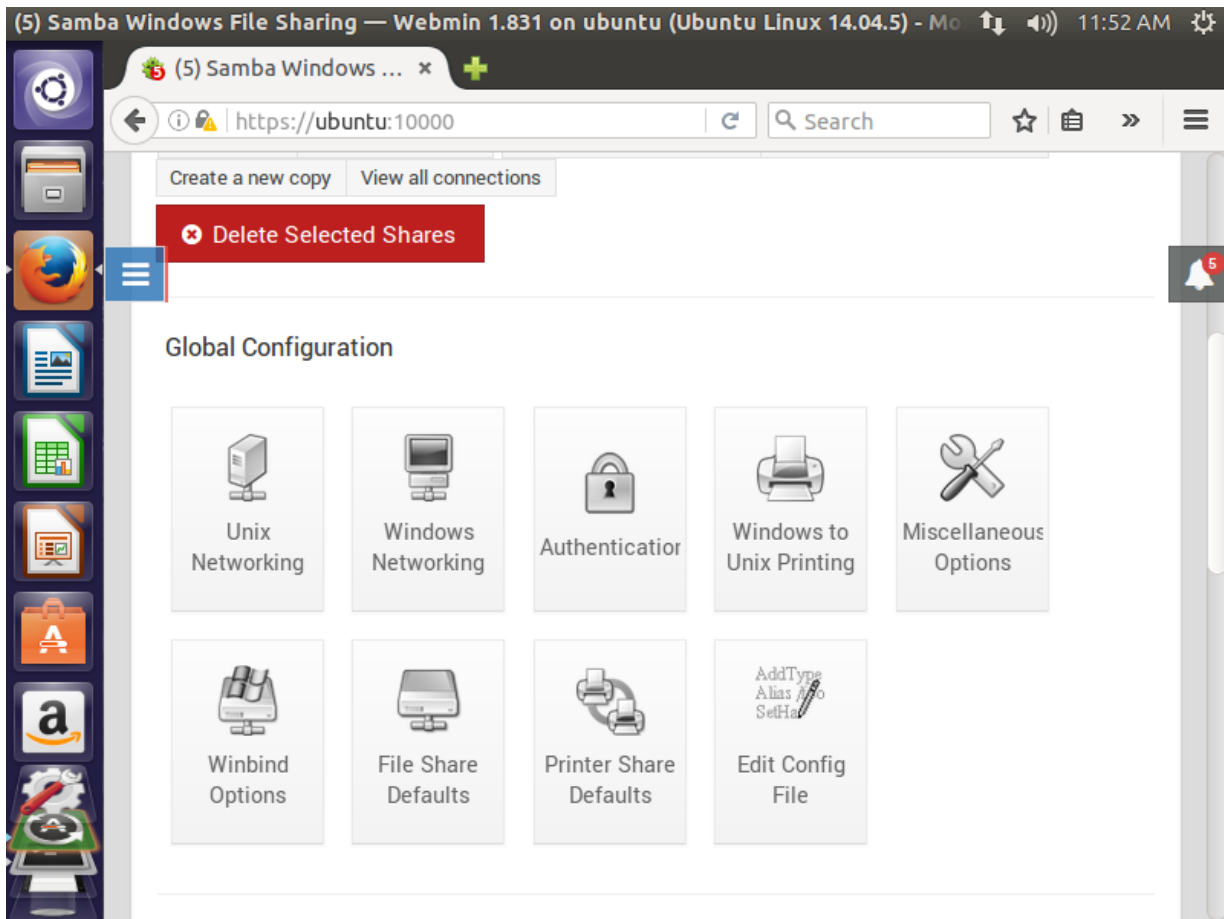
Create Now 5

Handles the name-based server <i>ubuntu</i> on address *.	
Address Any	Server Name <i>ubuntu</i>
Port 80	Document Root <i>/var/www/Monsite</i>

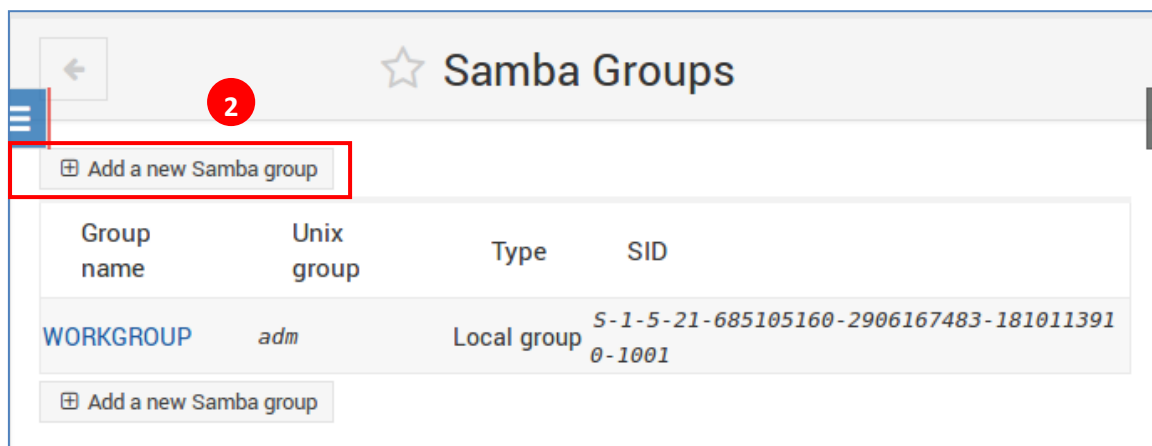
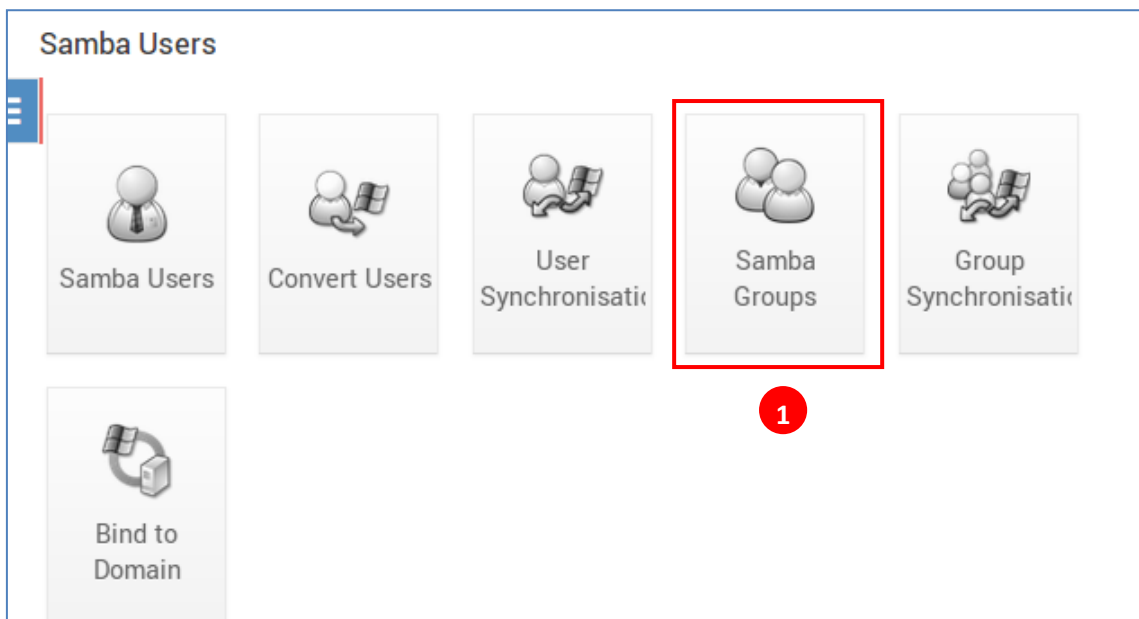
Tester le fonctionnement de votre site via le navigateur Web

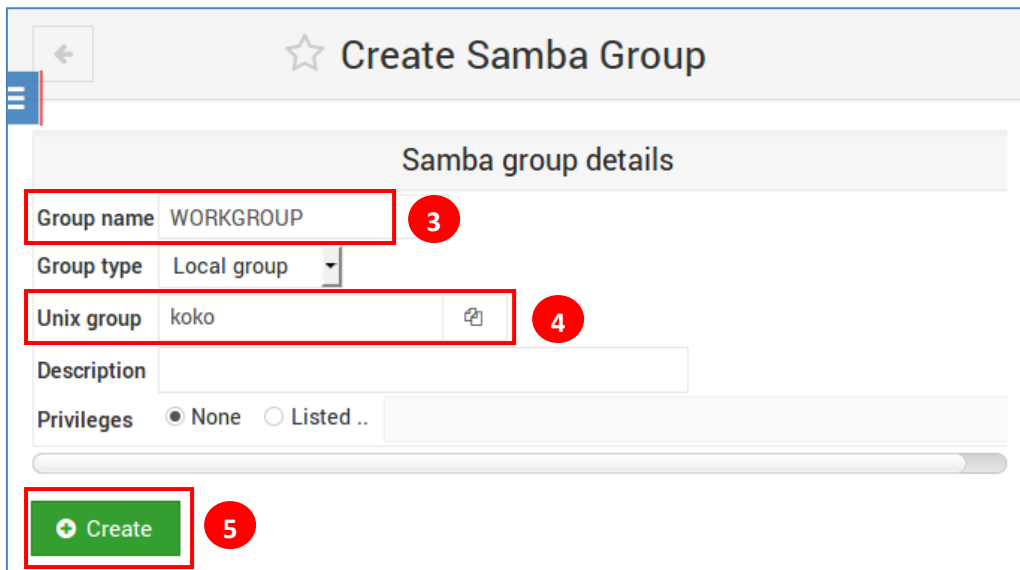


Etape 04 : Configuration du serveur Samba



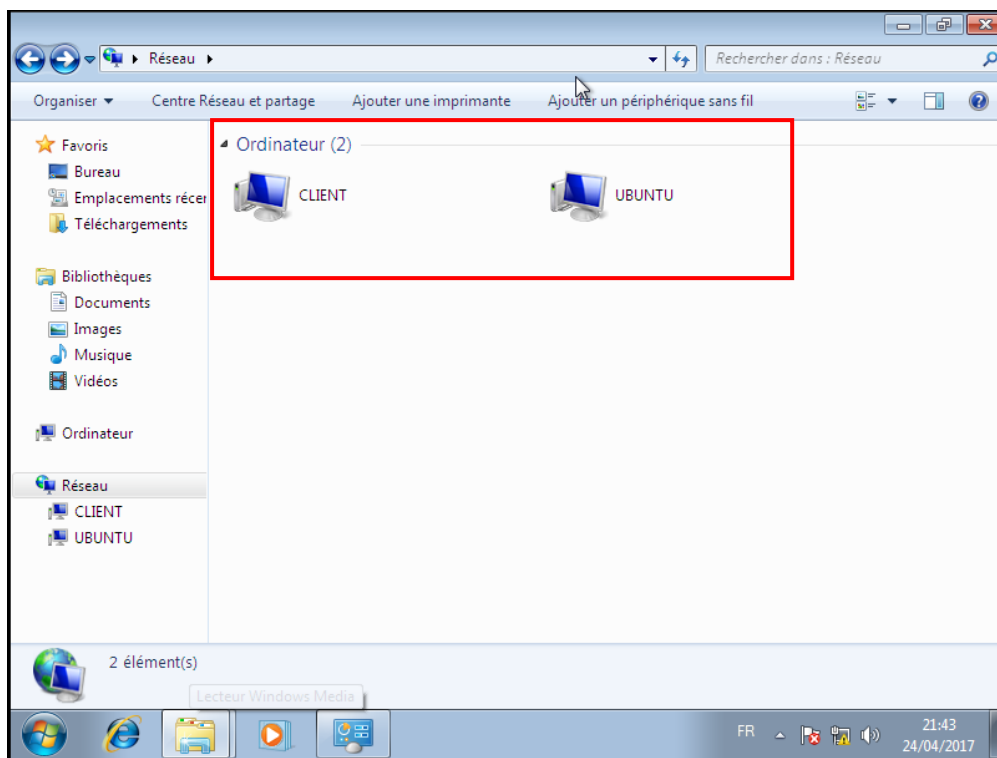
Ajouter un nouveau groupe « **WORKGROUP** »

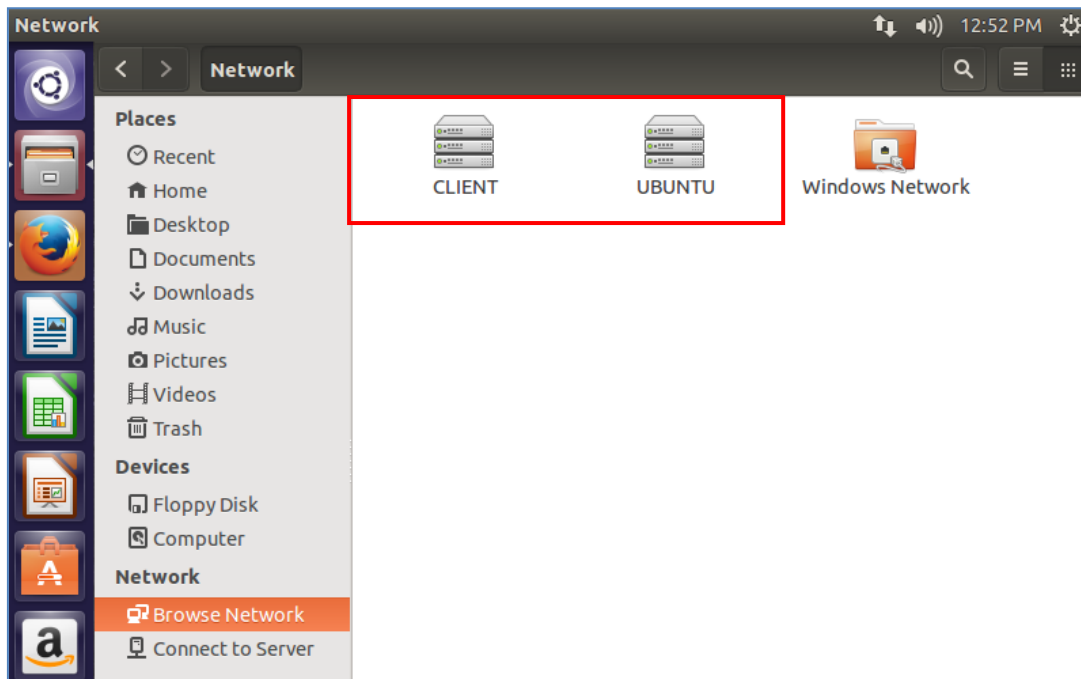




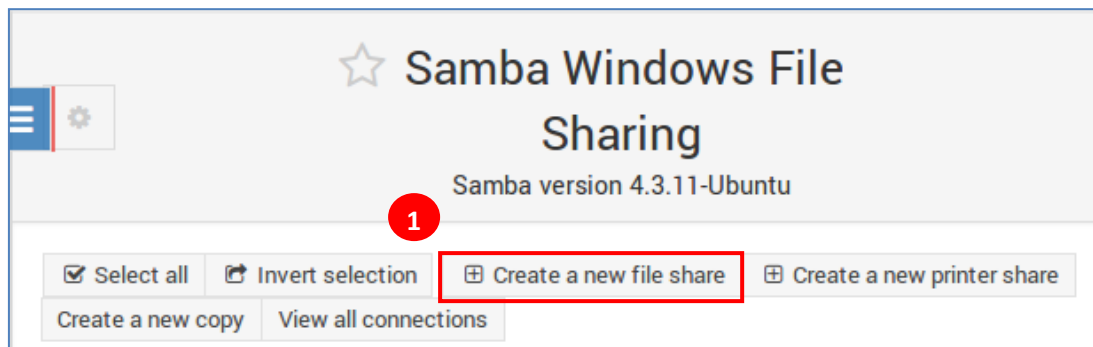
Le nouveau groupe à été créé

Group name	Unix group	Type	SID
WORKGROUP	adm	Local group	S-1-5-21-685105160-2906167483-1810113910-1001





Partager un fichier avec des machines Windows



☆ Create File Share

Share Information

Share name PARTAGE-UB Home Directories Share

Directory to share

Automatically create directory? Yes No

Create with owner

Create with permissions

Create with group

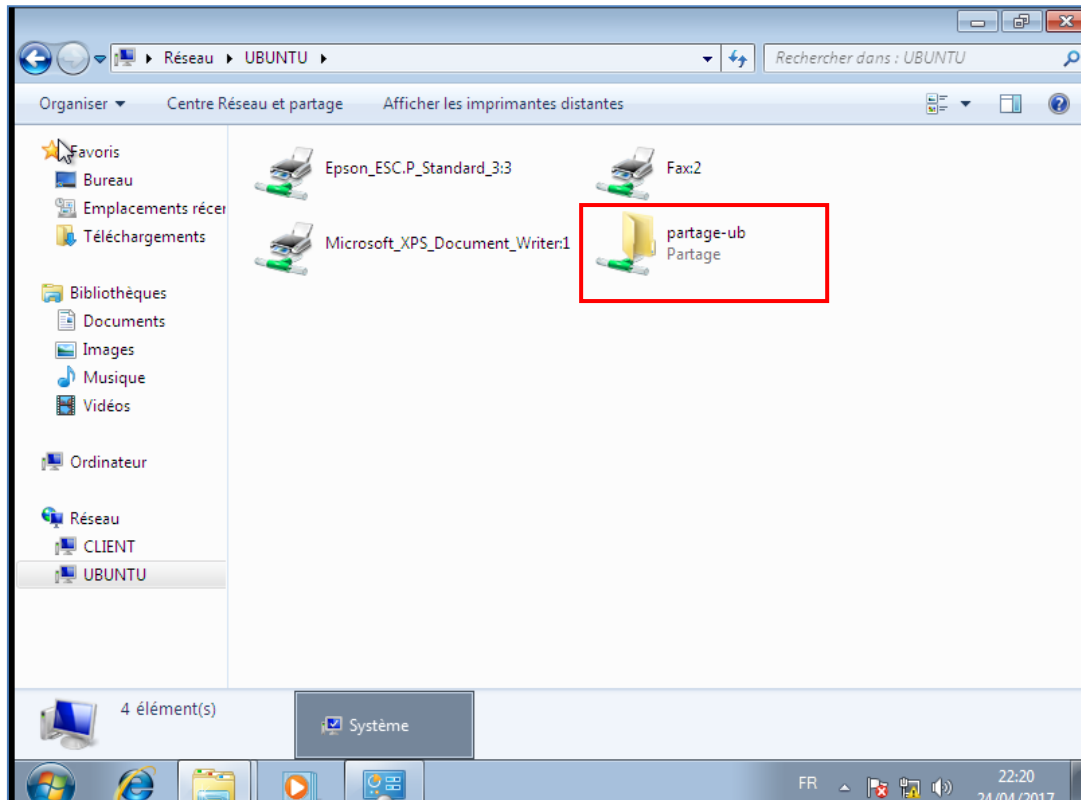
Available? Yes No

Browseable? Yes No

Share Comment

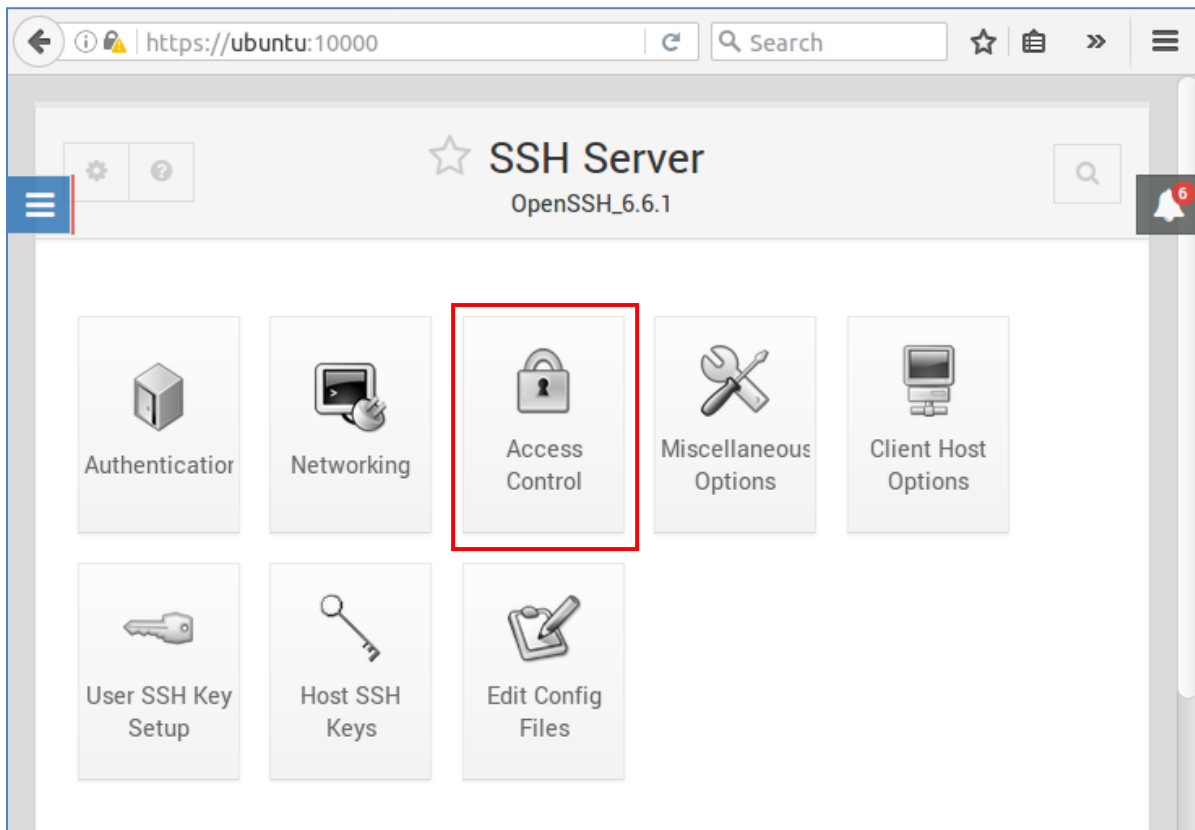
	Share Name	Path	Security
<input type="checkbox"/>	printers	All Printers	Printable to all known users
<input type="checkbox"/>	print\$	/var/lib/samba/printers	Read only to all known users
<input type="checkbox"/>	PARTAGE-UB	/home	Read only to all known users

Select all Invert selection Create a new file share Create a new printer share



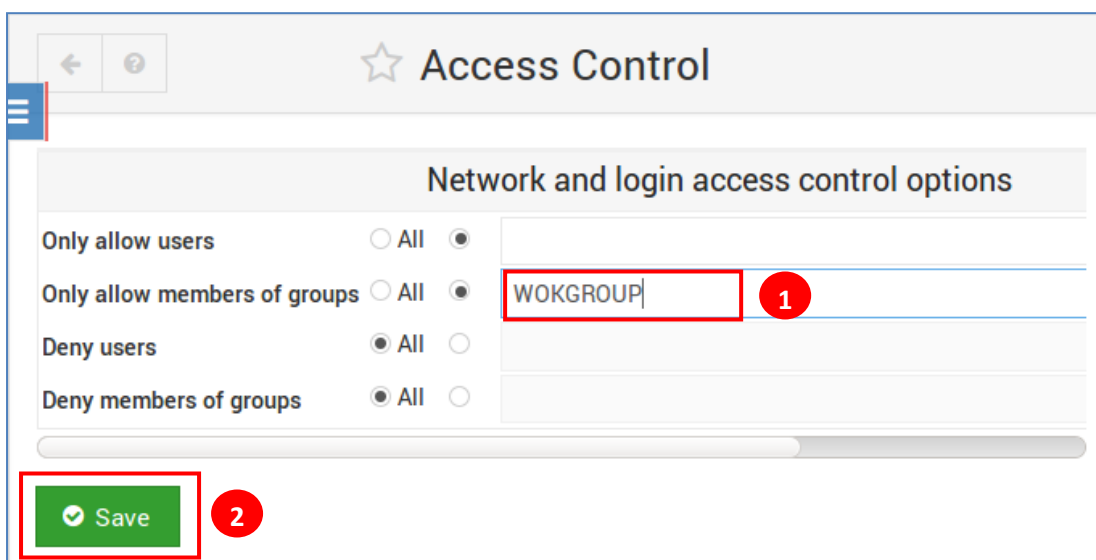
3. Configuration du serveur SSH sou Webmin

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



- **Restriction de l'accès au serveur SSH**

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.



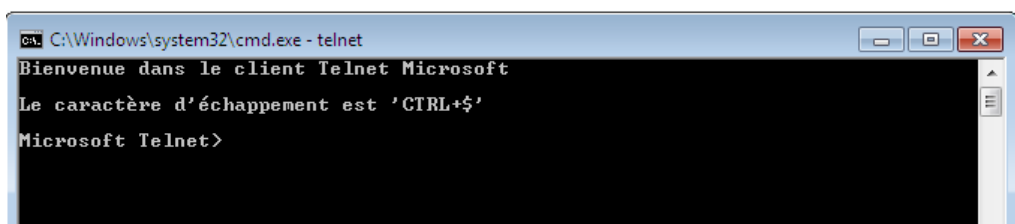
• Commandes Telnet

Les commandes **telnet** permettent de communiquer avec un ordinateur distant qui utilise le protocole Telnet.

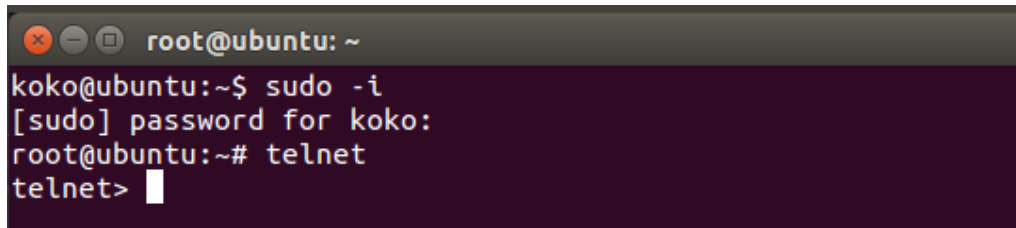
- Sous Windows, Telnet n'est pas installé par défaut (pour des raisons de sécurité). Pour l'activer, suivez la procédure ci-dessous :
 - Cliquez sur **Démarrer** et ensuite sur **Panneau de configuration**.
 - Dans la rubrique **Programmes**, cliquez sur **Désinstaller un programme**
 - Cliquez sur le menu de gauche "**Activer ou désactiver des fonctionnalités Windows**". Après quelques secondes, la fenêtre ci-dessous s'affiche.
 - Cochez la case à gauche de Client **Telnet** et pour finir cliquez sur **OK**



- Puis, allez dans **Démarrer** > **Exécuter**, et tapez simplement la commande : **telnet**.



- Pour les utilisateurs de **Linux** ou **Mac OS**, c'est quasiment pareil. Il suffit d'ouvrir un Terminal et d'entrer la commande : **telnet**.



```
root@ubuntu: ~  
koko@ubuntu:~$ sudo -i  
[sudo] password for koko:  
root@ubuntu:~# telnet  
telnet> █
```

Les commandes standard de Telnet sont les suivantes :

Commande	Description
?	Affiche l'aide
close	Termine la session Telnet
display	Affiche à l'écran les paramètres de la connexion (type de terminal, port)
environ	Permet de définir les variables d'environnement du système d'exploitation
logout	Permet de se déconnecter
mode	Bascule entre les modes de transfert ASCII (transfert d'un fichier en mode texte) et BINARY (transfert d'un fichier en binaire)
open	Permet de lancer une autre connexion à partir de la connexion en cours
quit	Quitte l'application Telnet
set	Modifie les paramètres de la connexion
unset	Charge les paramètres de connexion par défaut