

# **Installation d'un serveur WEB sous Linux**

Par LoiselJP  
Le 10/06/2013

# 1 Présentation de la situation

Vous travaillez dans une petite entreprise qui propose sur son site web un aperçu de ses produits ou de ses compétences.

Après quelques temps votre comptable, aidé par vos compétences dans le domaine, a déterminé qu'il serait plus « rentable » d'héberger votre propre serveur Web, car vous avez besoin d'une grosse bande passante, d'un gros volume de données... pour une sécurité moyenne (vous ne faites pas de vente en ligne, il n'y a pas de données confidentielles...).

Vous apprenez que plus vous avez de visite plus vous êtes connus, plus vous êtes connus plus le marché relatif à votre activité fonctionne mieux...

Après une étude supplémentaire, il apparaît que fournir un petit hébergement (gratuit ou non) à vos clients peut être très intéressant.

## 2 Objectifs

Face à cette situation qui pourrait être tout à fait possible (d'ailleurs réfléchissez-y), ce tutoriel va vous donner la marche à suivre, parmi d'autres, pour installer un serveur Web et proposer la possibilité de créer des dossiers qui pourront être confiés à des utilisateurs tiers.

- A cette occasion, il va alors être nécessaire d'installer le serveur, puis de le sécuriser.
- Ensuite le serveur Web sera installé sur celui-ci. Il ne sera pas question de création de site Web dans ce tutoriel (une page d'accueil sera juste créée sans aucune prétention).
- Afin de permettre à vos clients de déposer et modifier leur site web un serveur FTP non anonyme sera mis en place. Les clients seront alors enfermés dans un dossier virtuel et se verront attribué une taille limitée d'espace disque.
- Tout serveur Web digne de ce nom doit aujourd'hui proposer une (ou plusieurs) bases de données ainsi que des pages dynamiques en PHP.
  - o PHP permet d'envoyer des mails, mais vous ne désirez pas être embêté avec de plaintes liées au spam,
  - o De même que vos clients ne sont pas des programmeurs chevronnés vous ne désirez pas voir votre site Web utiliser des pages de redirection externes à votre serveur,
  - o Enfin, il sera nécessaire de limiter le nombre de requêtes (HTML et Mysql) afin d'éviter la saturation du serveur

Le sujet étant très vaste à lui seul (Côté sécurité, services et paramètres), la partie DNS ne sera abordée que pour le serveur Web principal, de même que le côté client ne sera qu'abordé à distance.

### 3 Sommaire

1 Présentation de la situation .....	2
2 Objectifs.....	2
3 Sommaire.....	3
4 Prérequis .....	5
5 Installation du serveur.....	5
Installer les outils d'administration.....	9
Installer et utiliser SSH.....	9
Installer Webmin.....	10
Utilisation de Webmin.....	12
Placer une adresse IP fixe .....	13
Améliorer l'affichage.....	14
6 Installer le serveur WEB .....	15
Déplacer le dossier de base.....	15
Déplacer les autres dossiers nécessaires .....	17
7 Installer PHP et Mysql.....	18
8 Configurer Mysql avec PHPMyadmin .....	21
Ajouter un nouvel utilisateur .....	22
Créer manuellement un nouvel utilisateur .....	24
9 Installer le serveur FTP .....	25
Installer le serveur.....	26
Configurer vsftpd.....	27
FTP anonyme .....	27
Chemin pour l'utilisateur anonyme .....	28
Autorise l'upload anonyme.....	28
Gérer les « Comptes utilisateur ».....	28
Message d'accueil.....	29
Valider et tester les modifications .....	29
10 Mettre en place des quotas.....	34
Mise en place des quotas .....	36
Activer les quotas.....	37
Fixer les quotas user .....	37
Définir des quotas par défaut .....	38
Afficher l'utilisation des quotas.....	39
11 Sécuriser le serveur.....	39
Installer et configurer l'antivirus .....	39
Configurer le pare-feu.....	41
DenyHosts.....	45
Bloquer les comptes clients .....	46
Mettre en place les ACLs.....	47
12 Configurer le serveur .....	48
Configurer le serveur PHP .....	49
Modifier les directives du fichier php.ini.....	49
Mettre une page d'information pour les « clients » .....	52
Configurer le serveur de DNS. ....	52
Installer Bind9.....	53
Vérifier le serveur .....	62
Ajouter des sous domaines .....	64
13 Gérer les comptes.....	67
Création d'un script de gestion des utilisateurs .....	67
Créer le formulaire d'enregistrement automatique du nouveau « client » .....	71
Créer un utilisateur avec sa base de données .....	72

Choisir un nouvel utilisateur et son mot de passe.....	72
Créer l'utilisateur .....	73
Supprimer un utilisateur.....	77
14 Petite page web .....	78
15 Conclusion .....	83
©Propriété.....	85
Licence.....	85

## 4 Prérequis

Dans ce document il est question de créer un serveur particulier qui devra entre-autre être en mesure de fournir du fichier et, dans le cas de nombreux « clients », devra être également en mesure de fournir de nombreuses pages dynamiques. Le serveur devra donc répondre à quelques règles de base (en matière de carte réseau, accès disque, mémoire...).

Quelques connaissances de base sont nécessaires ce qui permettra par la suite d'éviter de longues explications et se limiter aux différentes commandes et modifications de fichier de configuration.

Il n'est donc pas question ici de voir ou revoir toutes les bases de Linux. Pour appliquer cette opération, on suppose alors que sont acquis :

- L'installation de base d'un poste de travail linux,
- L'installation de logiciel:
  - Principe des paquets et leurs dépendances,
  - Les "dépôts" de paquets,
  - Principe d'installation manuelle,
- Les principales commandes du shell,
- Utilisation du programme man et l'aide des commandes (si vous ne savez pas appliquer ce qui est demandé),

On suppose pour ce type d'emploi que le lecteur (vous) a déjà quelques connaissances dans l'administration de Linux.

Pour le besoin de ce tutoriel, la machine utilisée est un processeur 4 cœurs avec un disque de 100Go (débit 100Mb/s), mémoire RAM 2Go et la carte réseau utilisée est une carte GigaBit... suivant le besoin et la destination le serveur pourra être dimensionné de manière différente (plus de RAM pour la gestion des caches clients, plus d'espace disque pour fournir plus d'espace au client...).

## 5 Installation du serveur

Chacun choisira la distribution de son choix, dans le cas présent toute l'installation est donnée pour une version à base de Debian (on recherche un serveur et non un poste de travail).

Cette version de Linux est plus destinée à un usage de serveur. D'autant plus qu'aucune interface graphique ne sera utilisée et que Debian est tout particulièrement prisée par les administrateurs pour ce type d'emploi.

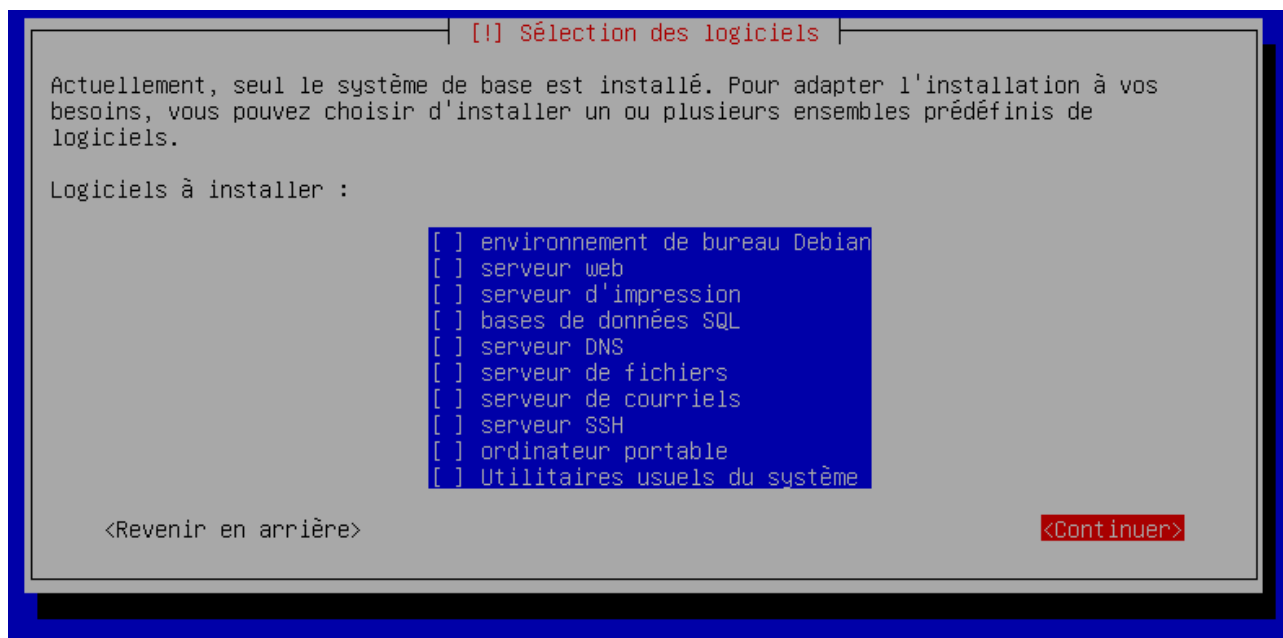
Ici on effectuera une installation suivant ce modèle:

# Installation à partir d'un CDrom netinstall Debian 7 (la dernière en date au jour de la création de ce tutoriel)

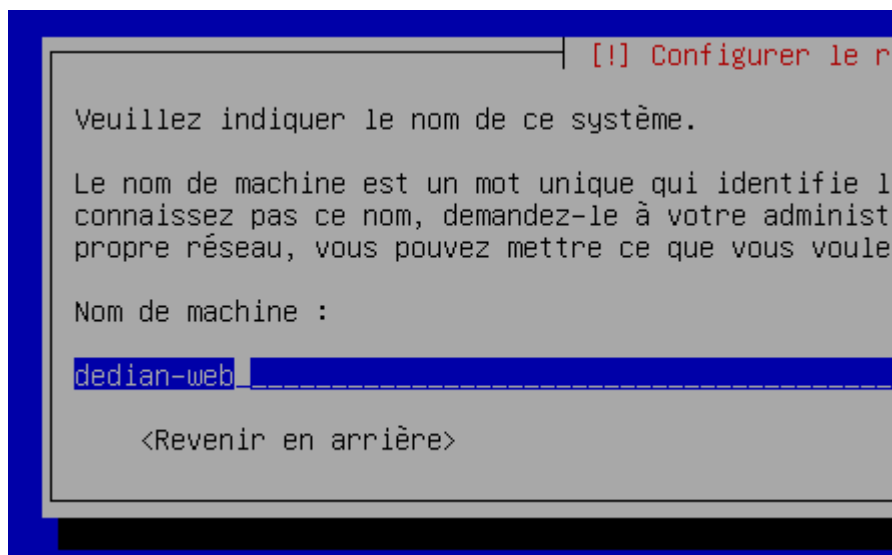
# Avec en sélection les paquets suivants: aucun.



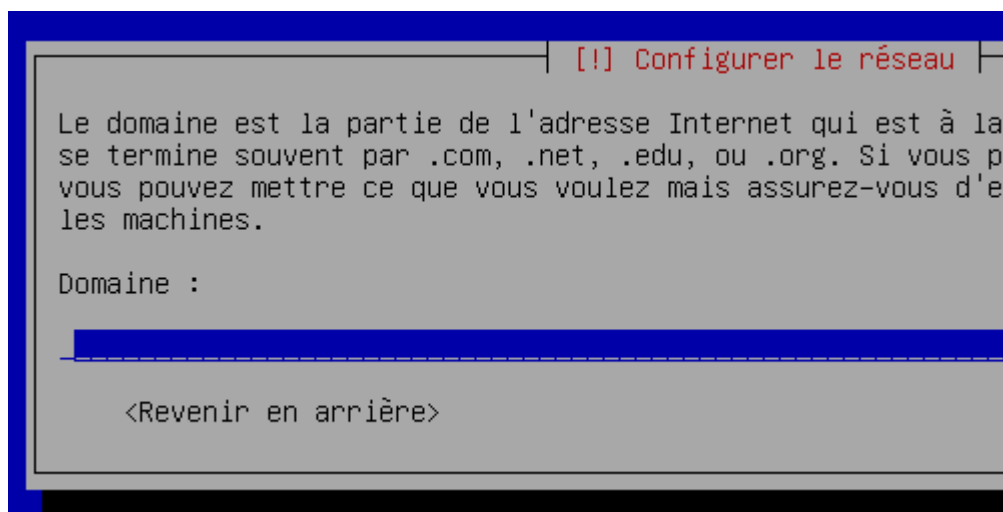
...



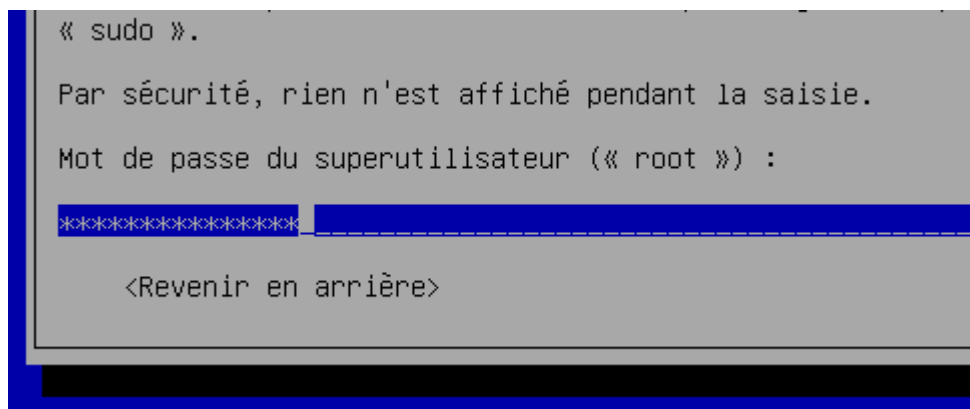
*Lorsque le choix de logiciel sera proposé, une installation minimale sera réalisée...*



A ce stade, le domaine n'a pas d'importance (il sera placé ultérieurement)



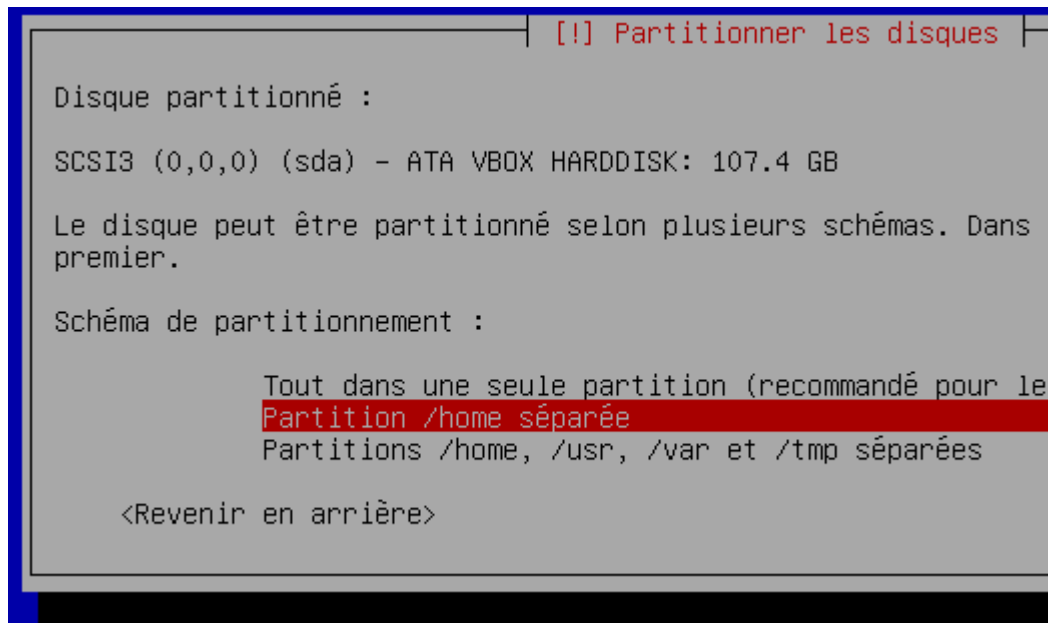
C'est évident mais il est préférable de le rappeler, le serveur va être accessible depuis l'Internet et donc de toutes ces personnes bien intentionnées qui ne demandent qu'à vous aider dans votre travail... il faut donc penser à mettre un mot de passe (Administrateur et utilisateur principal) assez fort pour pallier aux différents aléas de ce monde...



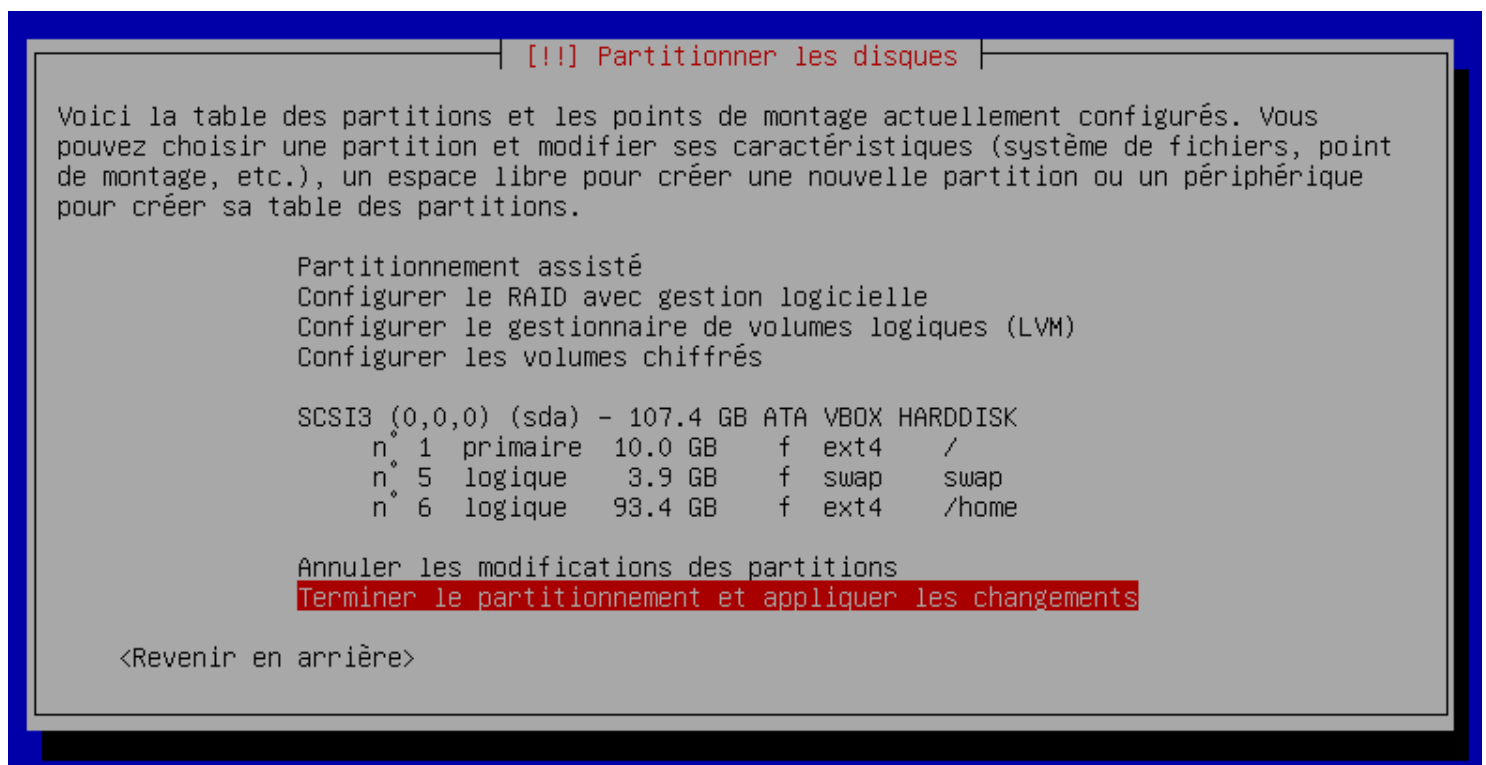
Un mot de passe entre 14 et 20 caractères est préférable (avec 3 type de caractères, Majuscules, chiffre, caractères spéciaux, caractère avec accent), qui n'est pas une expression que l'on trouve dans le dictionnaire, qui ne reprend pas le nom du serveur, propriétaire... Un vrai mot de passe quoi !

Par sécurité le minimum sera d'utiliser un serveur dont la partition /home sera séparée, les différents fichiers nécessaires au serveur Web seront déplacés dans cette partition. Au besoin il sera possible de créer une partition « Web », « Datas »...

Pour le système 10Go seront plus que suffisants d'autant que le serveur n'aura pas d'interface graphique et sera exclusivement dédié à cet emploi (C'est ce qui est proposé par défaut... y aurait-il une raison ?).



...



Pour les plus aisés il sera envisageable d'installer le serveur avec LVM (dans le cas actuel LVM ne change en rien l'installation ou la présentation du serveur).

***En tout état de cause, pour suivre ce tutoriel, il sera indispensable de placer les données utilisateur dans une partition séparée du système. En effet pour la mise en place de quotas cela sera indispensable !***



## Installer les outils d'administration

### Installer et utiliser SSH

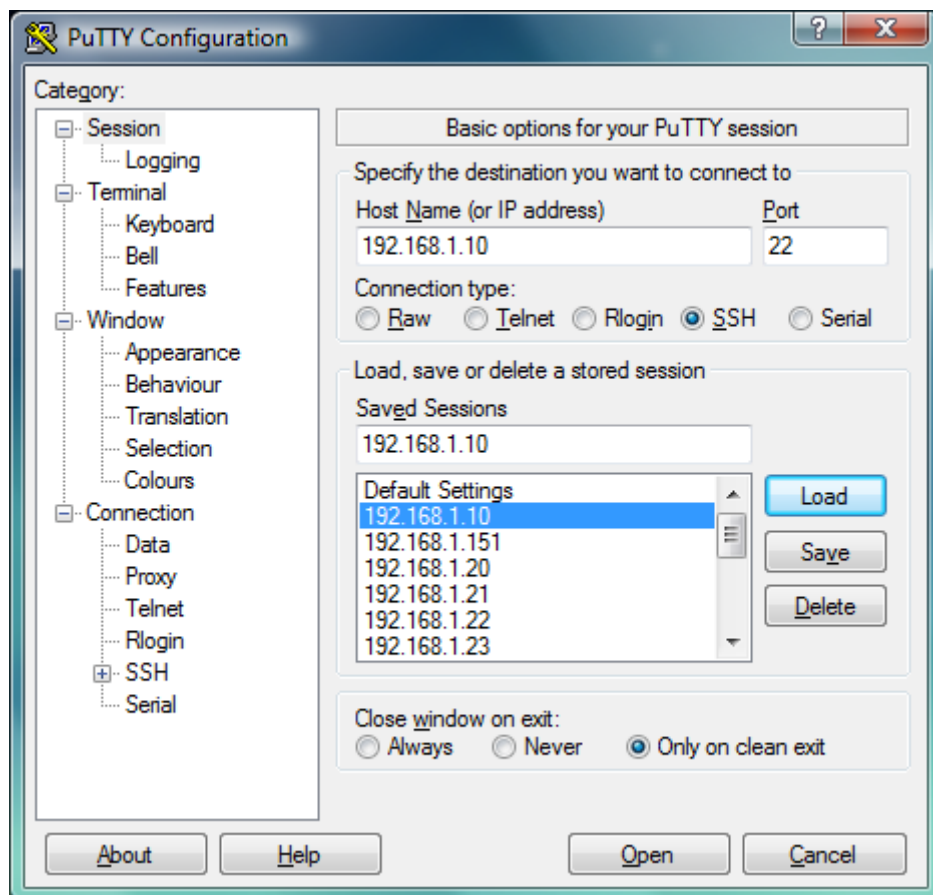
SSH est une solution de prise en main à distance sécurisée, toutes les commandes pourront être traitées par ce moyen.

Pour installer OpenSSH :

```
apt-get install openssh-server
```

```
web:~# apt-get install openssh-server_
```

Putty qui est un logiciel libre vous permet cela en toute simplicité:



Depuis un autre poste, par une fenêtre de commande, vous pourrez gérer votre serveur sans besoin de travailler directement dessus.

Dès lors que l'on utilise une connexion SSH il sera aisé d'utiliser le copier/coller en prenant directement les commandes de ce tutoriel.

## Installer Webmin

Webmin est l'un des outils d'administration Linux les plus populaires. Notre serveur ne dispose pas d'interface graphique, Webmin s'impose...

Installer les prérequis pour Webmin (« aptitude » et « zip » n'en font pas partie mais seront utilisés dans divers cas...

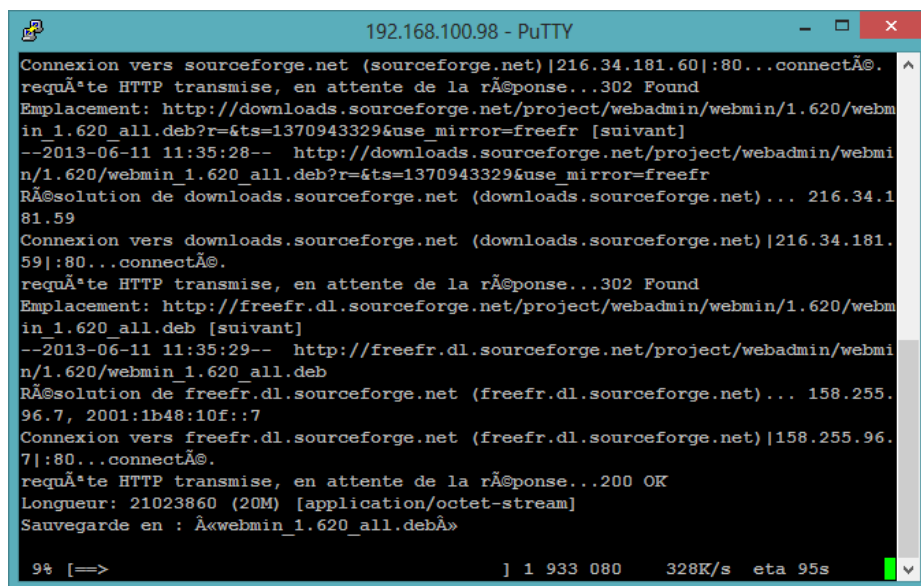
```
apt-get install aptitude zip libnet-ssleay-perl libauthen-pam-perl libio-pty-perl apt-show-versions
```

On pourra aussi installer python-minimal et openssl...

Télécharger Webmin dans un dossier où il pourra être décompressé avant sa suppression :

```
cd /tmp
```

```
wget http://sourceforge.net/projects/webadmin/files/webmin/1.620/webmin_1.620_all.deb
```



Installation de webmin

```
dpkg -i webmin_1.620_all.deb
```

En cas de souci cette commande régler les différents soucis de dépendances (cela devrait être le cas si vous n'avez pas installé Python par exemple avant)...

```
apt-get install -f
```

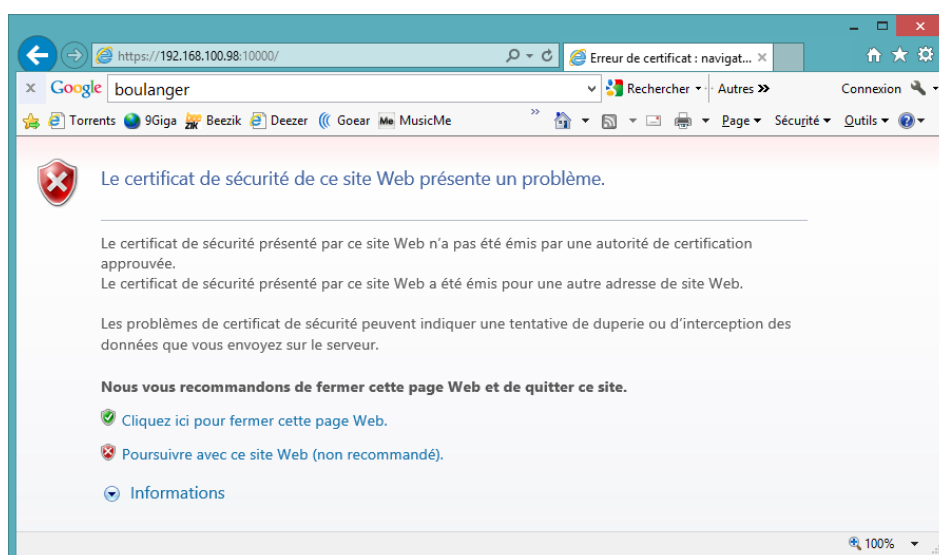
```
192.168.100.98 - PuTTY
root@dedian-web:/tmp# apt-get install -f
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Correction des dépendances... Fait
Les paquets supplémentaires suivants seront installés :
  file libmagic1 mime-support openssl python python-minimal python2.7
  python2.7-minimal
Paquets suggérés :
  ca-certificates python-doc python-tk python2.7-doc binfmt-support
Les NOUVEAUX paquets suivants seront installés :
  file libmagic1 mime-support openssl python python-minimal python2.7
  python2.7-minimal
0 mis à jour, 8 nouvellement installés, 0 à enlever et 0 non mis à jour.
1 partiellement installé ou enlevé.
Il est nécessaire de prendre 5 723 ko dans les archives.
Après cette opération, 19,3 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ?
```

Après installation le chemin d'accès sera indiqué par l'installation...

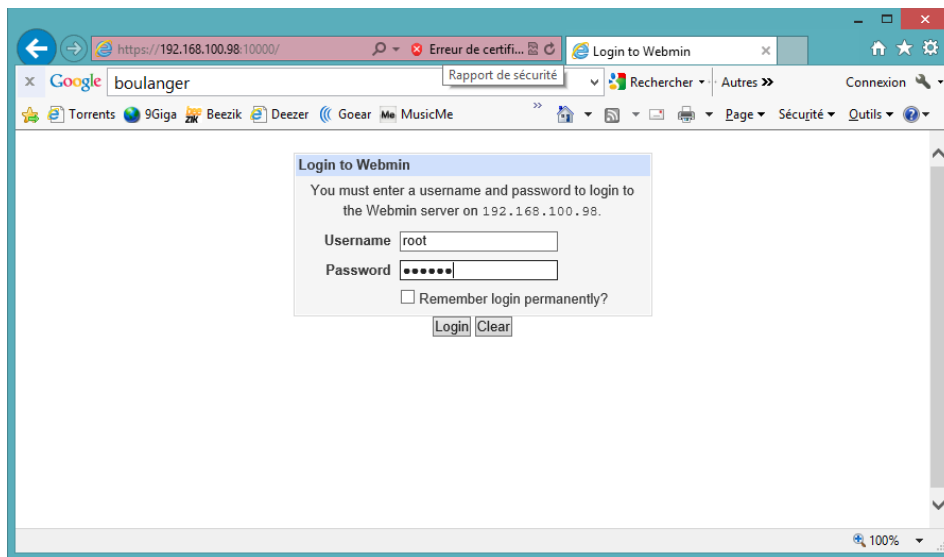
```
192.168.100.98 - PuTTY
Sélection du paquet file précédemment désélectionné.
Dépaquetage de file (à partir de ../archives/file_5.11-2_amd64.deb) ...
Traitement des actions différées (« triggers ») pour « man-db »...
Traitement des actions différées (« triggers ») pour « menu »...
Paramétrage de openssl (1.0.1e-2) ...
Paramétrage de python2.7-minimal (2.7.3-6) ...
Linking and byte-compiling packages for runtime python2.7...
Paramétrage de mime-support (3.52-1) ...
Paramétrage de python2.7 (2.7.3-6) ...
Paramétrage de python-minimal (2.7.3-4) ...
Paramétrage de python (2.7.3-4) ...
Paramétrage de webmin (1.620) ...
Webmin install complete. You can now login to https://dedian-web:10000/
as root with your root password, or as any user who can use sudo
to run commands as root.
Paramétrage de libmagic1:amd64 (5.11-2) ...
Paramétrage de file (5.11-2) ...
Traitement des actions différées (« triggers ») pour « menu »...
root@dedian-web:/tmp#
```

L'accès à Webmin se fait par:

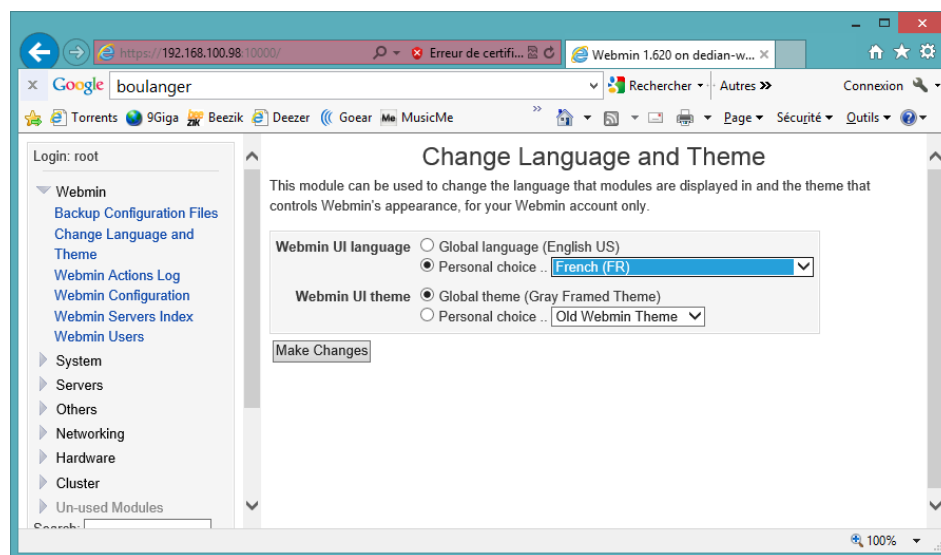
[https://ip\\_ou\\_nom\\_du\\_serveur:10000/](https://ip_ou_nom_du_serveur:10000/)



Après avoir confirmé le certificat, se connecter (en root avec votre mot de passe)

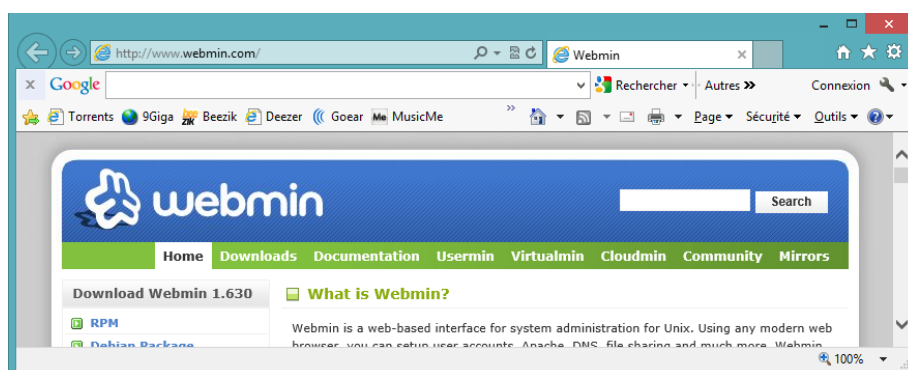


Passer dans « Webmin », « Change Language and Theme » passer en « french » (ça peut aider pour les non anglophones)



## Utilisation de Webmin

Le sujet de ce tutoriel n'est pas l'administration Linux, pour l'utilisation de « Webmin » et sa portée, on se reportera au site « <http://www.webmin.com/> » sur lequel toutes les informations d'installation, configuration emploi seront détaillées.



## Placer une adresse IP fixe

En fonction de l'organisation de la structure dans laquelle se trouve le serveur (derrière un routeur, avec redirection Nat ou pas, dans une DMZ...), placer une adresse IP fixe peut être intéressant. Dans le cadre de la sécurité, cela sera même indispensable pour une gestion et une administration plus aisée (mais c'est une question de choix).

Pour placer une adresse IP fixe :

```
vi /etc/network/interfaces
```

Placer l'ip fixe dans le fichier de configuration

```
# Interface reseau de bouclage

auto lo

iface lo inet loopback

address 127.0.0.1

netmask 255.0.0.0


# carte reseau en ip statique (ou auto)

allow-hotplug eth0

iface eth0 inet static

# <- a adapter suivant le besoin

address 192.168.1.10

broadcast 192.168.1.255

netmask 255.255.255.0

gateway 192.168.1.1

dns-nameservers 192.168.1.1

auto eth0
```

Puis redémarrer l'interface réseau

```
/etc/init.d/networking restart
```

Ou

```
service networking restart
```

Placer le hostname au besoin

```
vi /etc/hostname
```

## Améliorer l'affichage

Vous l'aurez peut-être rencontré certaines versions de Linux proposent les listes en couleur et quelques commandes comme « ll » qui remplace aisément « ls -l ».

Ces fonctions sont comprises dans le système mais non activées par défaut « ll » est un alias de « ls -l ». Ces fonctions pourront être activées à partir du fichier « .bashrc » du dossier de l'utilisateur (ici, c'est le compte root qui nous intéresse) :

```
nano /root/.bashrc
```

```
GNU nano 2.2.6      Fichier : .bashrc

# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want `ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "`dircolors`"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
```

Il suffira de décommenter les lignes voulues...

```
GNU nano 2.2.6      Fichier : .bashrc      Modifié : @

# ~/.bashrc: executed by bash(1) for non-login shells.

# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
umask 022

# You may uncomment the following lines if you want `ls' to be colorized:
export LS_OPTIONS='--color=auto'
eval "`dircolors`"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'
```

Les couleurs et commandes seront disponibles à la prochaine connexion...

```
dedian-web:/var# ll
total 44
drwxr-xr-x  2 root root  4096 dÃ@c.  14 13:56 backups
drwxr-xr-x  9 root root  4096 juin  11 12:04 cache
drwxr-xr-x 24 root root  4096 juin  11 11:46 lib
drwxrwsr-x  2 root staff 4096 dÃ@c.  14 13:56 local
lrwxrwxrwx  1 root root    9 juin  11 09:50 lock -> /run/lock
drwxr-xr-x  7 root root  4096 juin  11 19:26 log
drwxrwsr-x  2 root mail  4096 juin  11 09:50 mail
drwxr-xr-x  2 root root  4096 juin  11 09:50 opt
lrwxrwxrwx  1 root root    4 juin  11 09:50 run -> /run
drwxr-xr-x  4 root root  4096 juin  11 09:51 spool
drwxrwxrwt  2 root root  4096 juin  11 09:58 tmp
drwx----- 2 root bin   4096 juin  11 19:26 webmin
drwxr-xr-x  2 root root  4096 juin  11 12:04 www
dedian-web:/var#
```

## 6 Installer le serveur WEB

Le serveur est maintenant installé dans sa configuration de base.

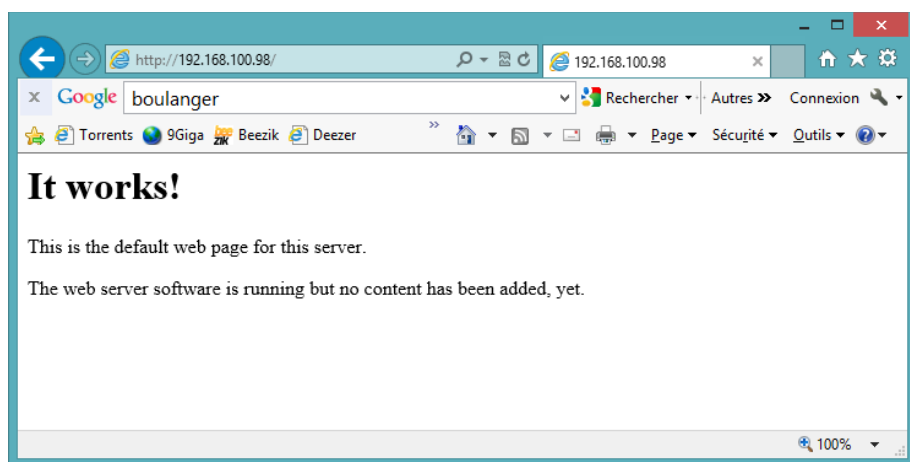
Le serveur Web peut maintenant être installé. A cette étape, le système va être laissé dans sa configuration d'origine. Il ne s'agit que d'installer le serveur Apache, de placer le dossier de base dans « /home » et de vérifier son fonctionnement.

Pour installer apache :

```
apt-get install apache2
```

On pourra vérifier son fonctionnement et son accès depuis n'importe quel navigateur web à cette adresse :

***http://ip\_du\_serveur***



Il l'annonce lui-même : « il fonctionne ! ».

## Déplacer le dossier de base

Le dossier de base de stockage est à cette adresse : ***/var/www***

```
cd /var/www
```

On y trouvera la page d'accueil par défaut

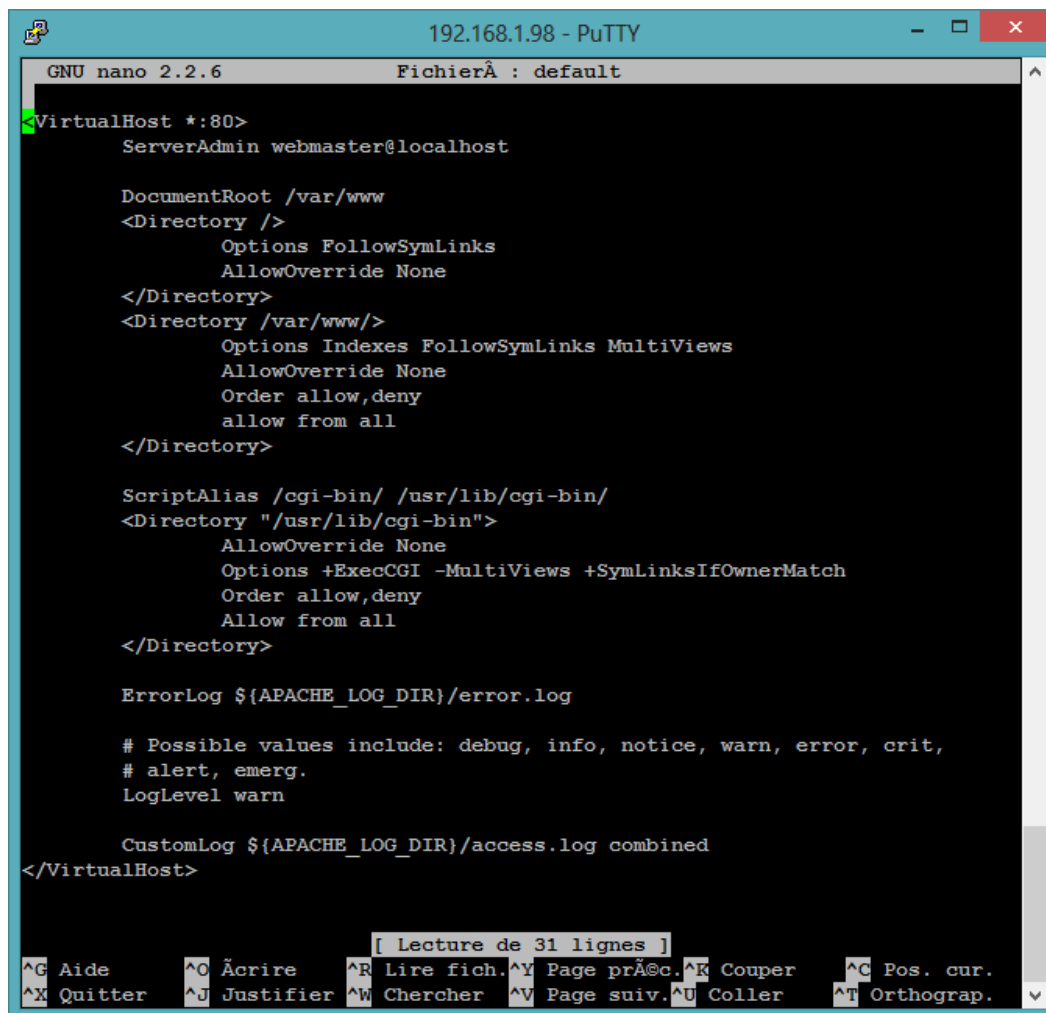
```
root@dedian-web:/# cd /var/www
root@dedian-web:/var/www# ls
index.html
root@dedian-web:/var/www#
```

Ce dossier peut être déplacé simplement en modifiant la configuration d'Apache .

Pour modifier le chemin par défaut d'apache se diriger dans « /etc/apache2/sites-available », éditer ensuite « default » :

```
cd /etc/apache2/sites-available
```

```
nano default
```



```
192.168.1.98 - PuTTY
GNU nano 2.2.6 Fichier : default
VirtualHost *:80>
  ServerAdmin webmaster@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>

  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/error.log

  # Possible values include: debug, info, notice, warn, error, crit,
  # alert, emerg.
  LogLevel warn

  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

[ Lecture de 31 lignes ]
^G Aide    ^C Quitter ^R Lire fich.^Y Page prÃsc.^K Couper    ^C Pos. cur.
^X Quitter ^J Justifier ^W Chercher  ^V Page suiv.^U Coller    ^T Orthograp.
```

La solution n'est pas compliquée mais déplace cependant le dossier par défaut et beaucoup s'y perdront (dont moi) car habitué au dossier « /var/www » on finirait par s'y perdre.

Bien que cela soit rarement le cas, certains logiciels pourraient utiliser (car trop habitué aussi) le chemin /var/www. Il va alors être préférable d'effectuer un lien symbolique.



Lors de l'installation du serveur nous avons placé le dossier home dans une partition séparée... le plus gros volume de données va être concentré dans le dossier /var/www. Le déplacer dans /home est le mieux...

```
mv /var/www /home
```

```
ln -s /home/www /var/www
```

```
dedian-web:/var# ll
total 40
drwxr-xr-x  2 root root  4096 dÃ©c.  14 13:56 backups
drwxr-xr-x  9 root root  4096 juin   11 12:04 cache
drwxr-xr-x 24 root root  4096 juin   11 11:46 lib
drwxrwsr-x  2 root staff 4096 dÃ©c.  14 13:56 local
lrwxrwxrwx  1 root root    9 juin   11 09:50 lock -> /run/lock
drwxr-xr-x  7 root root  4096 juin   11 19:26 log
drwxrwsr-x  2 root mail  4096 juin   11 09:50 mail
drwxr-xr-x  2 root root  4096 juin   11 09:50 opt
lrwxrwxrwx  1 root root    4 juin   11 09:50 run -> /run
drwxr-xr-x  4 root root  4096 juin   11 09:51 spool
drwxrwxrwt  2 root root  4096 juin   11 09:58 tmp
drwx----- 2 root bin   4096 juin   11 19:26 webmin
lrwxrwxrwx  1 root root    9 juin   11 19:51 www -> /home/www
dedian-web:/var#
```

On constate maintenant que /var/www pointe vers /home/www

## Déplacer les autres dossiers nécessaires

Il a été vu précédemment comment déplacer un dossier et pour quelle raison il sera déplacé dans home, toujours dans le but de ne pas saturer le système du serveur, d'autres dossiers vont être déplacés de la même manière, afin de limiter les dossiers en tous sens on créera auparavant un dossier destiné à accueillir ces fichiers système (logs, cache, temporaires..).

Comme précédemment, il sera créé des liens symboliques, dans le but de ne pas perturber l'ordre par défaut :

```
mkdir /home/fichiers_systeme
```

```
mv /tmp /home/fichiers_systeme/
```

```
ln -s /home/fichiers_systeme/tmp /tmp
```

```
mv /var/log /home/fichiers_systeme/
```

```
ln -s /home/fichiers_systeme/log /var/log
```

```
mv /var/cache /home/fichiers_systeme/
```

```
ln -s /home/fichiers_systeme/cache /var/cache
```

```
192.168.1.98 - PuTTY
dedian-web:/var# ll
total 32
drwxr-xr-x  2 root root  4096 dÃ©c.  14 13:56 backups
lrwxrwxrwx  1 root root    28 juin  11 20:12 cache -> /home/fichiers_systeme/cache
drwxr-xr-x 24 root root  4096 juin  11 11:46 lib
drwxrwsr-x  2 root staff 4096 dÃ©c.  14 13:56 local
lrwxrwxrwx  1 root root    9 juin  11 09:50 lock -> /run/lock
lrwxrwxrwx  1 root root   26 juin  11 20:10 log -> /home/fichiers_systeme/log
drwxrwsr-x  2 root mail 4096 juin  11 09:50 mail
drwxr-xr-x  2 root root 4096 juin  11 09:50 opt
lrwxrwxrwx  1 root root    4 juin  11 09:50 run -> /run
drwxr-xr-x  4 root root 4096 juin  11 09:51 spool
drwxrwxrwt  2 root root 4096 juin  11 09:58 tmp
drwx----- 2 root bin  4096 juin  11 19:26 webmin
lrwxrwxrwx  1 root root    9 juin  11 19:51 www -> /home/www
dedian-web:/var#
```

Plus loin, on déplacera également les fichiers mysql.

## 7 Installer PHP et Mysql

A cette étape PHP et Mysql vont être installés. PHPMyadmin est un très bon outil de configuration et gestion des bases de données Mysql, pour installer les paquets nécessaires, il suffira de taper cette commande :

```
apt-get install phpmyadmin
```

```
192.168.100.98 - PuTTY
Les paquets supplémentaires suivants seront installés :
 apache2-mpm-prefork dbconfig-common fontconfig-config libapache2-mod-php5
 libdbd-mysql-perl libdbi-perl libfontconfig1 libgd2-xpm libjpeg8 libltdl7
 libmcrypt4 libmysqlclient18 libnet-daemon-perl libonig2 libplrpc-perl
 libpng12-0 libqdbm14 libxml2 mysql-client mysql-client-5.5 mysql-common
 php5-cli php5-common php5-gd php5-mcrypt php5-mysql psmisc sgml-base
 ttf-dejavu-core xml-core
Paquets suggérés :
 php-pear libgd-tools libmcrypt-dev mcrypt libterm-readkey-perl mysql-server
 www-browser sgml-base-doc debhelper
Les paquets suivants seront ENLEVÉS :
 apache2-mpm-worker
Les NOUVEAUX paquets suivants seront installés :
 apache2-mpm-prefork dbconfig-common fontconfig-config libapache2-mod-php5
 libdbd-mysql-perl libdbi-perl libfontconfig1 libgd2-xpm libjpeg8 libltdl7
 libmcrypt4 libmysqlclient18 libnet-daemon-perl libonig2 libplrpc-perl
 libpng12-0 libqdbm14 libxml2 mysql-client mysql-client-5.5 mysql-common
 php5-cli php5-common php5-gd php5-mcrypt php5-mysql phpmyadmin psmisc
 sgml-base ttf-dejavu-core xml-core
0 mis à jour, 31 nouvellement installés, 1 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 19,7 Mo dans les archives.
Après cette opération, 87,7 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ?
```

Toutes les bibliothèques de dépendance du paquet seront installées (« php » car le module de gestion est en php, Mysql-server car c'est le module géré)... en une seule fois tout ce qui sera nécessaire sera installé...

Au cours de l'installation il va être demandé quel serveur Web reconfigurer, si vous avez installé Webmin avant (comme indiqué dans ce document), il sera proposé Apache2 et lighttpd. Apache2 car le serveur web est installé avec apache, et lighttpd car c'est le serveur sécurisé utilisé par Webmin...

```
192.168.100.98 - PuTTY
Outil de configuration des paquets

Configuration de phpmyadmin
à Veuillez choisir le serveur web à reconfigurer automatiquement pour
à exÃ©cuter phpMyAdmin.
à
à Serveur web à reconfigurer automatiquement :
à
à [*] apache2
à [*] lighttpd
à
à <Ok>
à
```

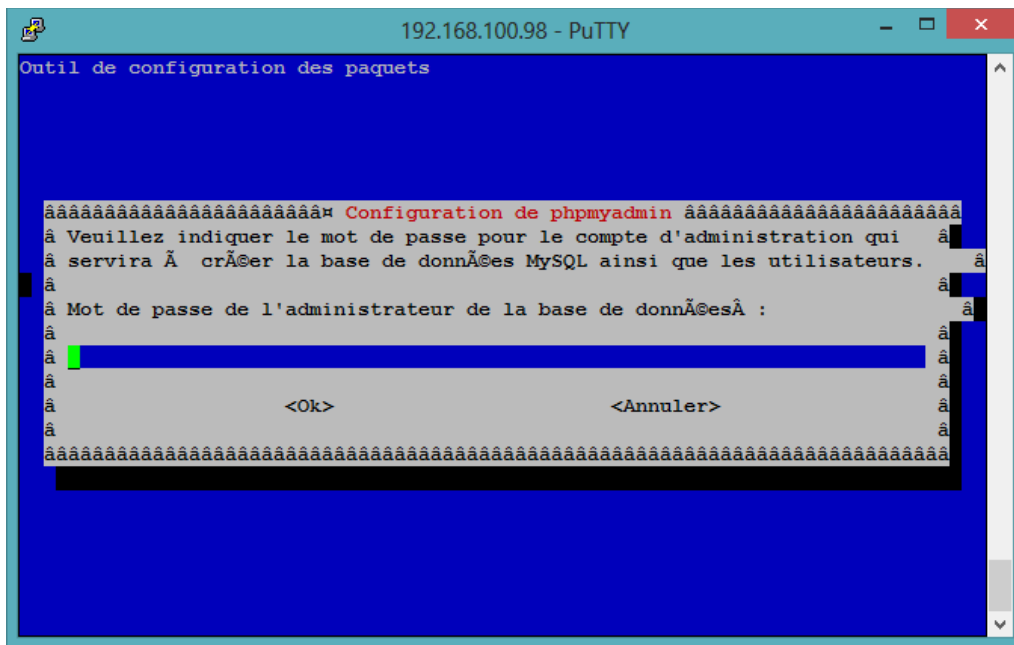
On va lui laisser modifier les deux serveurs (ou au moins apache2...).

Comme indiqué par la suite si vous êtes déjà un administrateur expérimenté, si la base de donnée est déjà installée,... vous pouvez choisir de configurer manuellement la base de données Mysql, donc à vous de voir...

```
192.168.100.98 - PuTTY
Outil de configuration des paquets

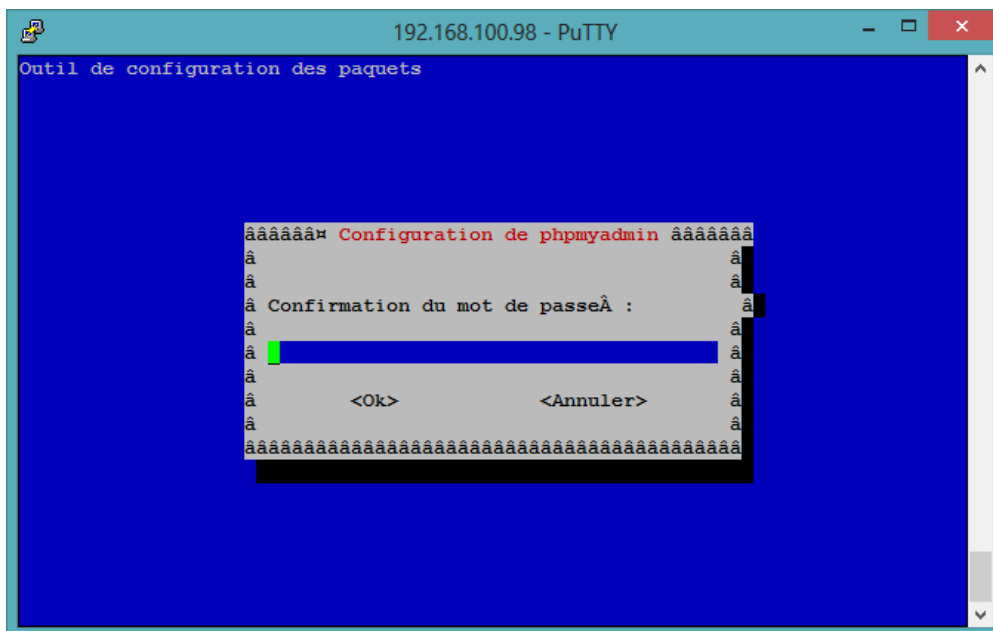
Configuration de phpmyadmin
à Le paquet phpmyadmin a besoin d'une base de données installée et
à configurée avant de pouvoir être utilisée. Si vous le souhaitez,
à dbconfig-common peut prendre cette opération en charge.
à
à Si vous êtes un administrateur de bases de données expérimenté et si
à vous savez que vous voulez procéder à cette configuration vous-même, ou
à si votre base de données est déjà installée et configurée, vous pouvez
à refuser cette option. Des précisions sur la procédure se trouvent dans
à /usr/share/doc/phpmyadmin.
à
à Autrement, vous devriez choisir cette option.
à
à Faut-il configurer la base de données de phpmyadmin avec
à dbconfig-common ?
à
à [*] <Oui>
à [*] <Non>
à
```

C'est le compte root qui s'occupera dans un premier temps de la base de données, donner alors le mot de passe qui va permettre d'y accéder :



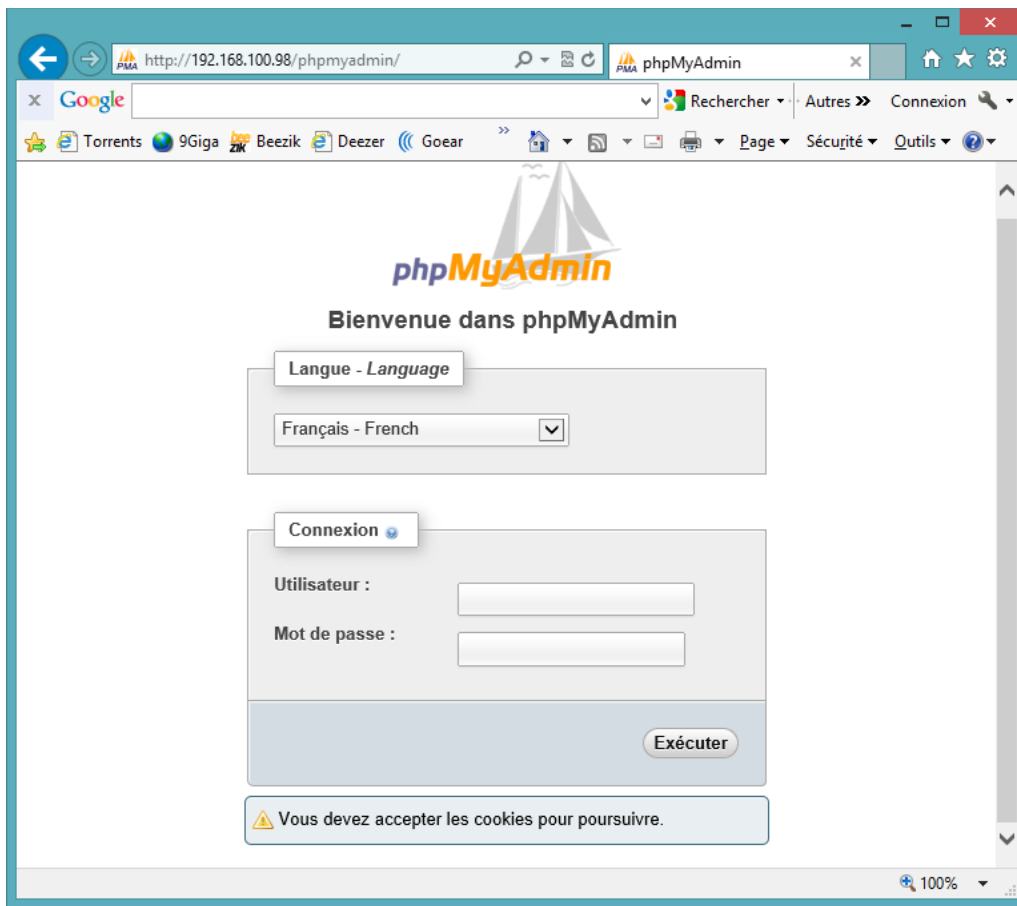
A l'étape suivante on va créer un mot de passe (à mettre !). Si aucun mot de passe n'est fourni, un mot de passe aléatoire est créé. Vous serez alors contraint de passer en ligne de commande pour le configurer. Avec un peu d'expérience l'opération est facile, dans ce tutoriel il s'agit d'installer un serveur Web et non de devenir administrateur de base de données Mysql (Mysql sera un support supplémentaire pour le site...).

Confirmer le mot de passe...



Dès lors que PHPmyadmin est installé (Avec Mysql...), l'interface de gestion est disponible à l'adresse suivante :

[http://nom\\_ou\\_ip\\_du\\_serveur/phpmyadmin](http://nom_ou_ip_du_serveur/phpmyadmin)



Dans certains cas (erreur de socket), l'utilisateur pourra être amené à revoir l'installation de phpmyadmin.

Dans ce cas il sera nécessaire de supprimer phpmyadmin :

```
apt-get remove phpmyadmin
```

Au cours de la suppression supprimer les bases, purger les fichiers

Réinstaller ensuite le serveur Mysql :

```
apt-get install mysql-server
```

Puis réinstaller phpmyadmin

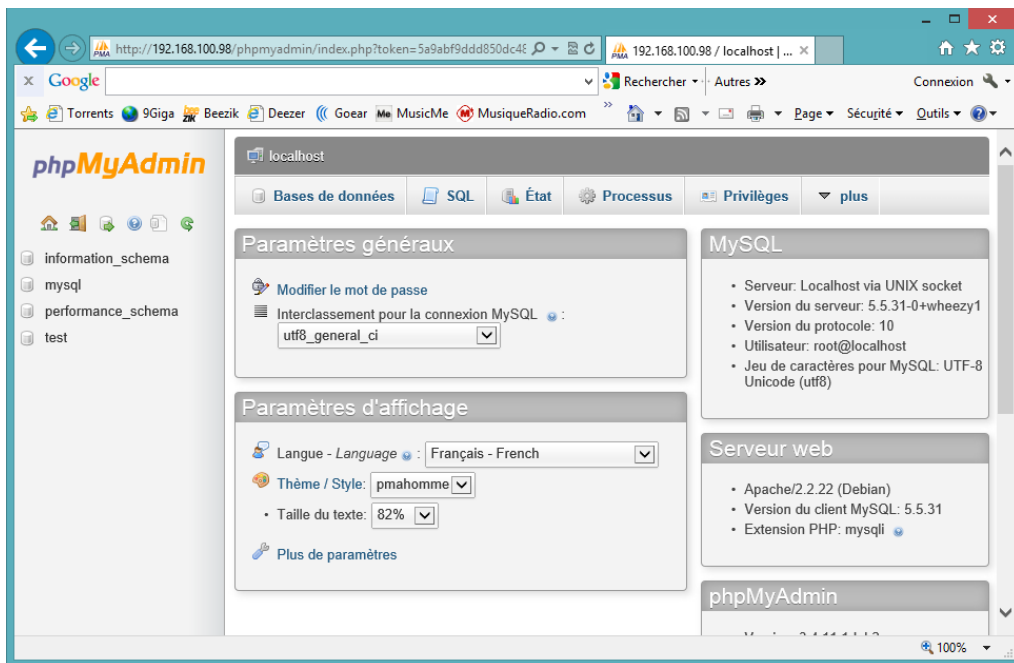
```
apt-get install phpmyadmin
```

## 8 Configurer Mysql avec PHPMyadmin

Après avoir installé PHPmyadmin, il est possible d'accéder à la gestion des bases par le chemin :

```
http://nom_ou_ip_du_serveur/phpmyadmin
```

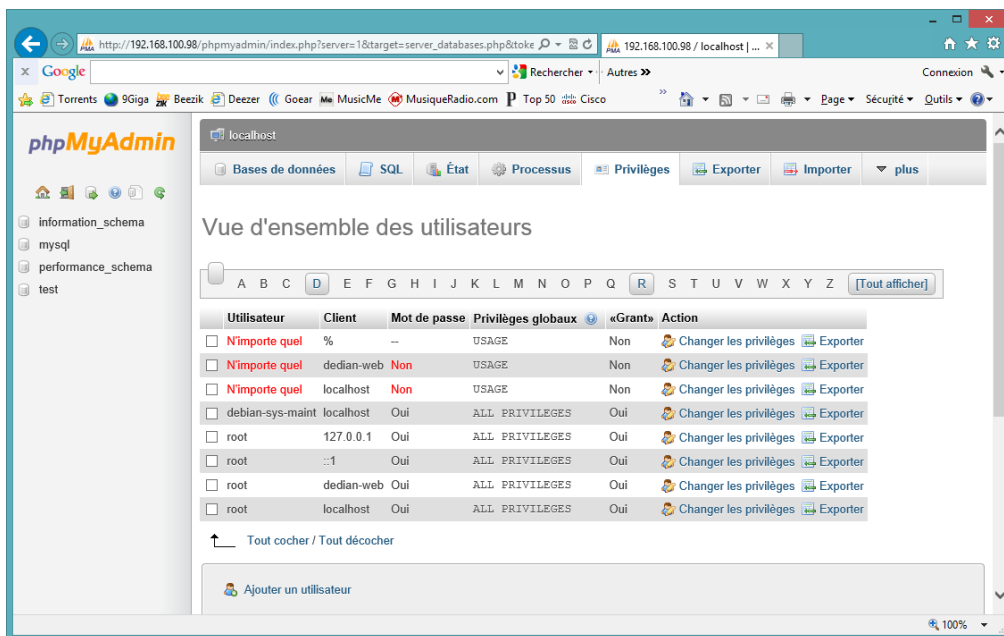
On y place le compte « root » avec le mot de passe de gestion (demandé au cours de l'installation)



Après connexion l'interface pourra ressembler à ceci (l'interface change régulièrement).

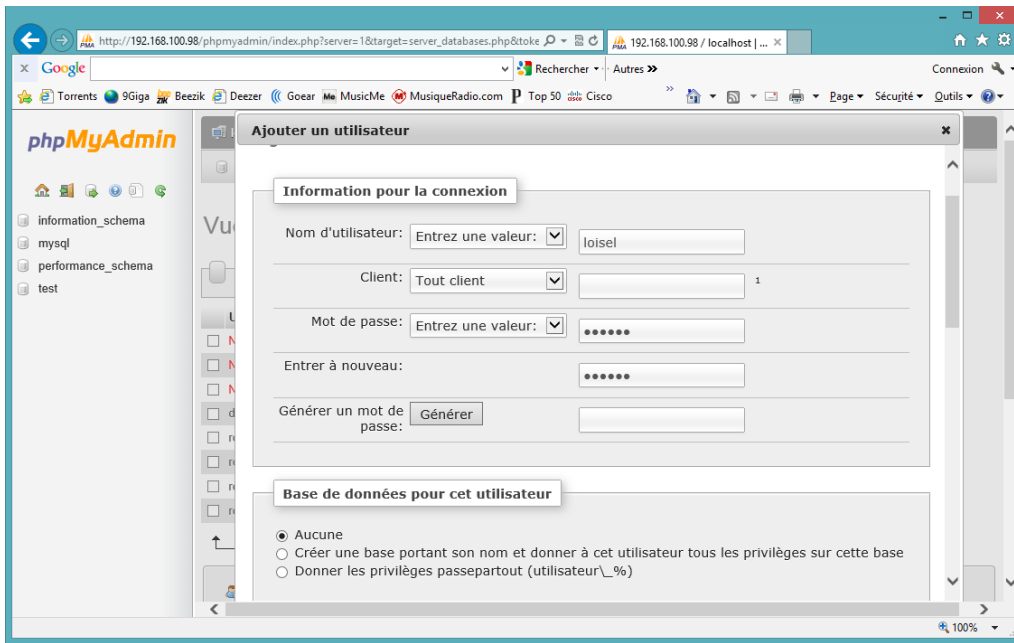
## Ajouter un nouvel utilisateur

La sécurité voudrait que le compte principal utilisant le serveur ne soit pas « root » (trop facile à trouver). Il est donc nécessaire de créer un nouvel utilisateur...

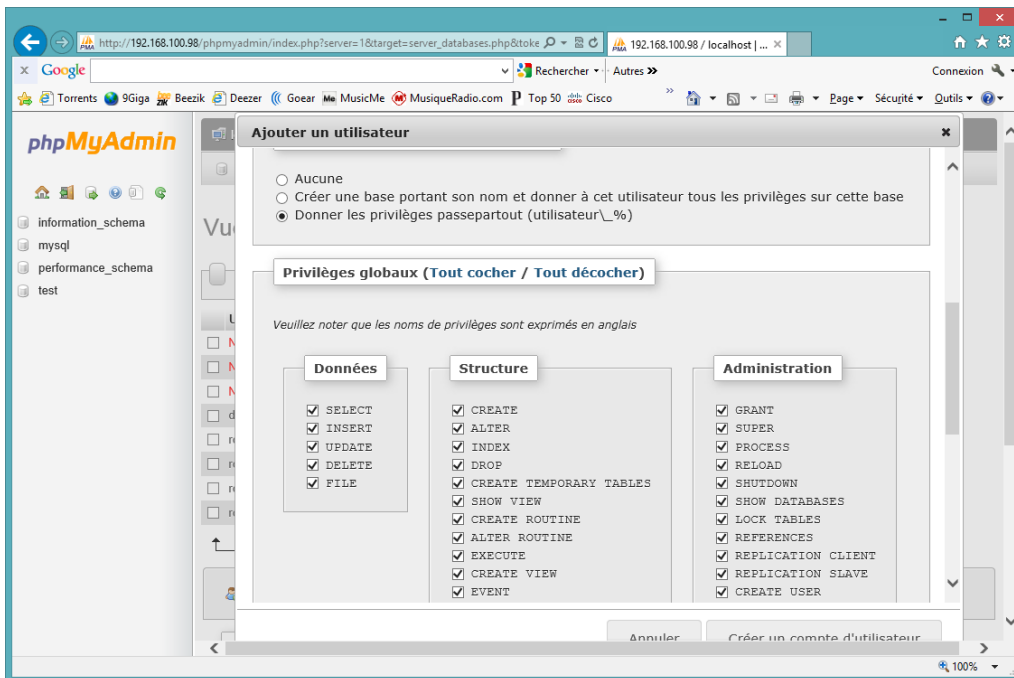


Se rendre alors dans « Privilèges » puis choisir « Ajouter un utilisateur »...

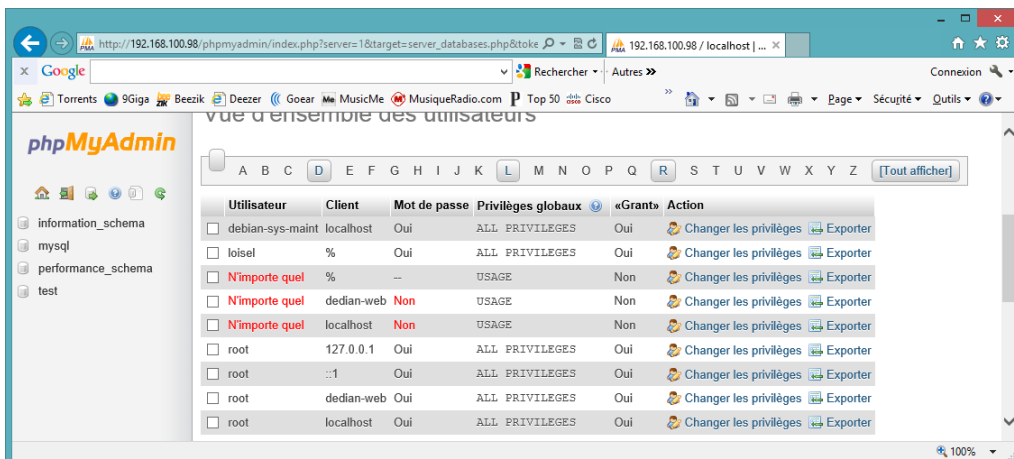
Lui donner un nom, un mot de passe (que l'on pourra générer automatiquement).



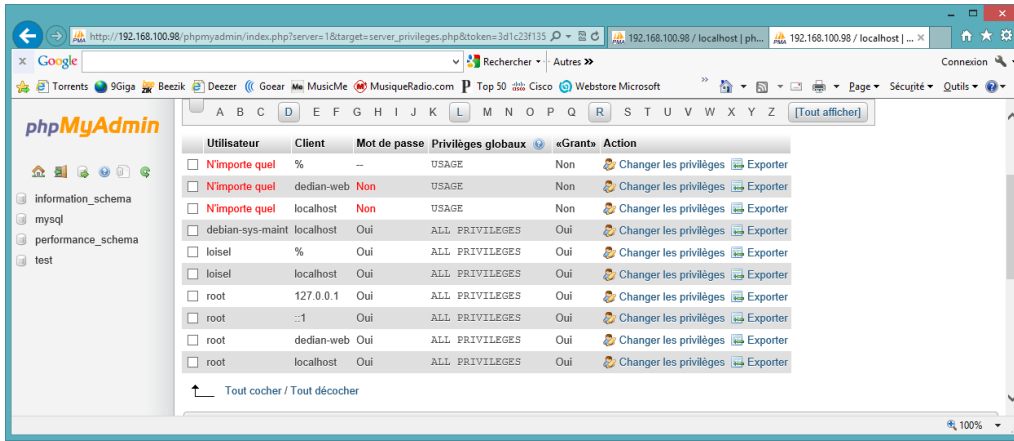
Lui donner des privilèges passepartout, sur tout !



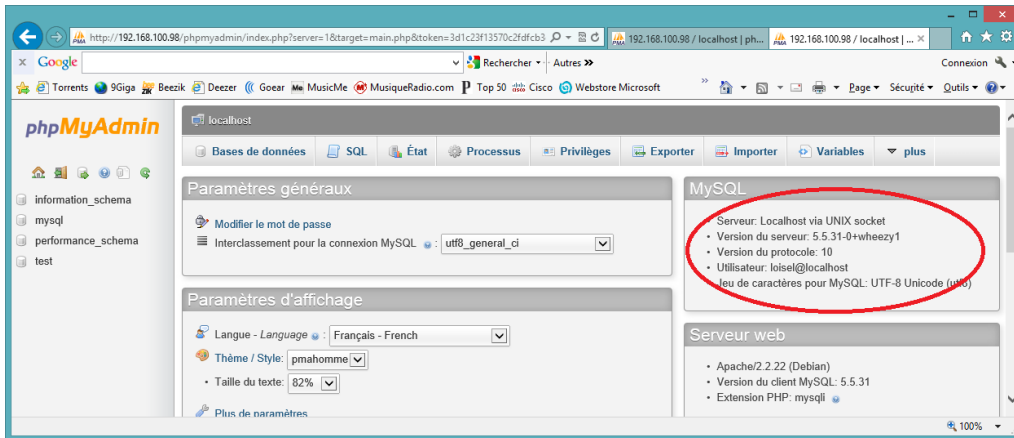
Créer le compte.



Recommencer l'ajout de l'utilisateur sur le serveur « localhost »



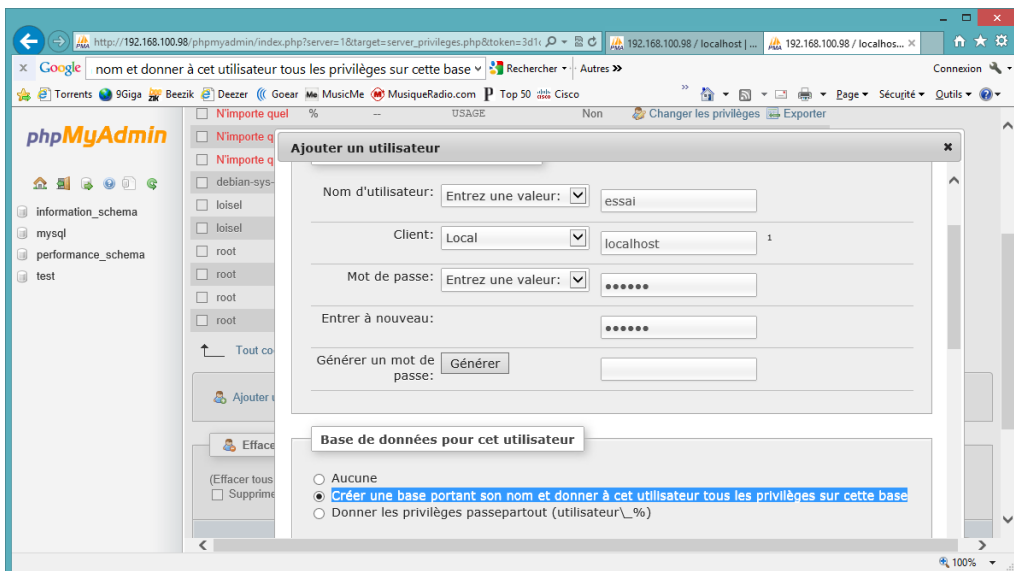
Se déconnecter puis tester le compte...



Avec tous les droits sur le serveur vous pouvez maintenant supprimer « root » l'administrateur qui doit être unique sera donc plus difficile à trouver...

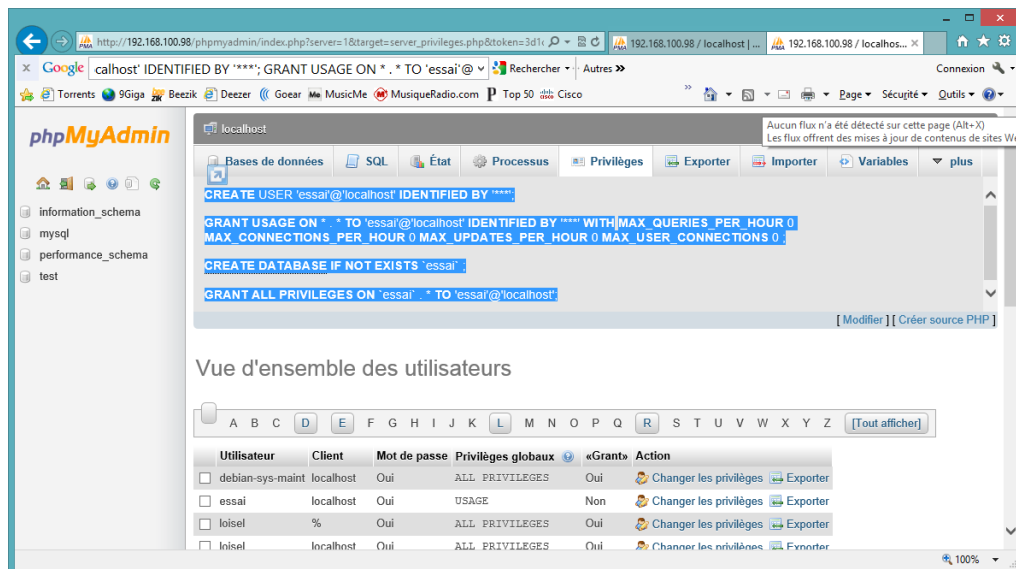
## Créer manuellement un nouvel utilisateur

Se rendre dans privilèges, puis « Ajouter un utilisateur », client « localhost », puis ajoutez lui un mot de passe, cochez «Créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base » :





Cliquer sur « créer le compte » :



Notez la commande « sql » permettant de générer le compte :

```
CREATE USER 'essai'@'localhost' IDENTIFIED BY '***';

GRANT USAGE ON * . * TO 'essai'@'localhost' IDENTIFIED BY '***' WITH MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;

CREATE DATABASE IF NOT EXISTS `essai` ;

GRANT ALL PRIVILEGES ON `essai` . * TO 'essai'@'localhost';
```

Vous pourrez utiliser cette commande dans PHP pour générer automatiquement vos nouveau utilisateurs... il suffira d'adapter dans le code « essai » et « \*\*\* » par le mot de passe voulu.

Les valeurs suivantes seront étudiées de près :

```
MAX_QUERIES_PER_HOUR
MAX_CONNECTIONS_PER_HOUR
MAX_UPDATES_PER_HOUR
MAX_USER_CONNECTIONS
```

Elles seront à placer au mieux afin de permettre aux utilisateurs de lire et exécuter les pages, mais aussi au mieux afin de ne pas saturer le serveur de requêtes trop nombreuses.

Il sera également nécessaire de définir le nombre maximum de connexion utilisateur.

Dans les pages Web il est courant d'ouvrir la connexion en début de page, et d'oublier de la fermer en fin d'ouverture de page. Un utilisateur qui lit 100 pages web se retrouvera vite avec 100 connexion ouvertes, chaque connexion prenant des ressources système, le serveur peut se trouver rapidement saturer.

## 9 Installer le serveur FTP

Plusieurs serveurs ftp sont disponibles sous linux, dans le cadre de ce tutoriel, c'est le serveur « vsftpd » qui sera choisi.

Vsftpd par son nom (Very Secure FTP Daemon) parle de lui-même il s'agit d'un démon de serveur ftp très sécurisé. De plus, il reste l'un des serveurs le plus facile à configurer, il s'agit dans ce tutoriel d'apprendre à créer un serveur Web sécurisé mais pour administrateur dont ce n'est pas le travail principal (même si de grandes enseignes ne vont pas plus loin que cela) il n'est donc pas question de chercher trop compliqué...

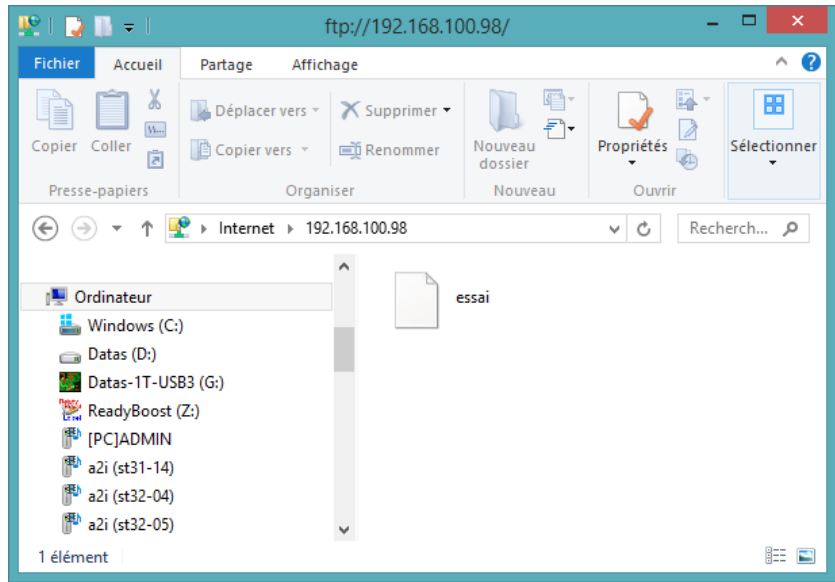
## Installer le serveur

Pour installer le serveur la commande est classique :

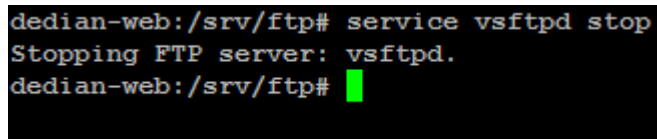
```
apt-get install vsftpd
```

Par défaut, le serveur est lancé et écoute le port 21 (c'est le port standard des serveurs ftp) et comme il va servir aux clients pour mettre leur données sur le serveur, il n'est pas question de le modifier.

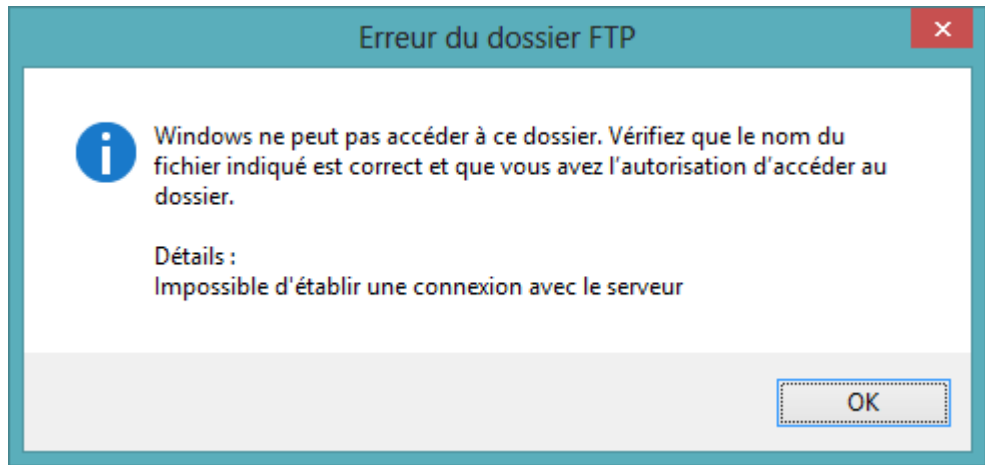
Par défaut « fstftpd » donne accès au dossier partagé « ftp » en mode anonyme... (pas bien) en attendant de l'avoir configuré il est peut-être sage de l'arrêter :



```
service vsftpd stop
```



C'est mieux...



## Configurer vsftpd

Dans ce paragraphe il va être question de configurer « vsftpd » afin de limiter les erreurs et le sécuriser suffisamment pour l'utilisation voulue.

Ouvrir le fichier de configuration:

```
nano /etc/vsftpd.conf
```

*Toutes les opérations suivantes se font dans ce fichier (à l'exception du changement de dossier par défaut)!*

Comme beaucoup de fichier configuration Linux, le fichier vsftpd.conf comporte beaucoup de commentaires qui permettront (après un peu de lecture) de configurer correctement le serveur.

Mais on remarquera que le fichier ne fait que 150 lignes (commentaires compris) ce qui le rend très abordable.

### FTP anonyme

Par défaut la connexion anonyme est autorisée, c'est le répertoire du serveur (« /srv/ftp ») qui est utilisé pour stocker les fichiers.

Pour le désactiver trouver le paramètre suivant et le placer à NO:

```
anonymous_enable=NO
```

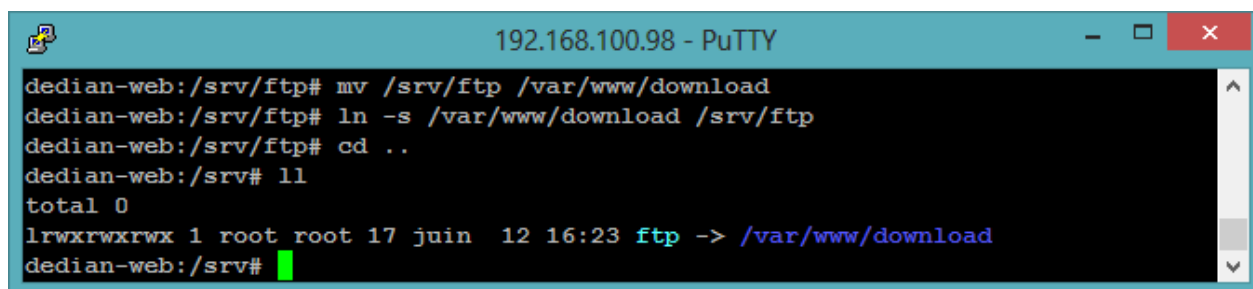
Dans le cadre de ce tutoriel, on désire proposer au «visiteur» un accès « ftp » sur le serveur. Mais ces mêmes fichiers pourront être disponibles au téléchargement (au travers de liens web sur les pages).

Le dossier /srv/ftp va donc être déplacés vers le site web (qui lui-même a été mis dans home) dans le dossier download :

```
mv /srv/ftp /var/www/download
```

Puis relier le dossier ftp par défaut à son nouvel emplacement :

```
ln -s /var/www/download /srv/ftp
```



```
192.168.100.98 - PuTTY
dedian-web:/srv/ftp# mv /srv/ftp /var/www/download
dedian-web:/srv/ftp# ln -s /var/www/download /srv/ftp
dedian-web:/srv/ftp# cd ..
dedian-web:/srv# ll
total 0
lrwxrwxrwx 1 root root 17 juin 12 16:23 ftp -> /var/www/download
dedian-web:/srv#
```

## Chemin pour l'utilisateur anonyme

Comme indiqué précédemment les comptes anonymes ont accès par défaut au dossier « /srv/ftp » il est possible de le changer en utilisant la directive « anon\_root ». Ce qui est effectivement une seconde méthode (peut être plus « propre ») mais pourra apporter parfois quelques confusion, alors qu'avec une méthode telle que celle décrite avec une redirection, l'administrateur aura plus d'aisance :

```
anon_root=/var/www/download
```

## Autorise l'upload anonyme

Chacun prend ses risques... il est possible de laisser l'accès en Ecriture pour les utilisateurs anonymes (on pourrait l'envisager dans un intranet).

Configurez les options « anon\_upload\_enable » pour autoriser ou non l'écriture et « anon\_mkdir\_write\_enable » pour la création de dossier.

```
anon_upload_enable = yes
```

```
anon_mkdir_write_enable= yes
```

Dans ce cas il sera utile de définir un compte du système comme propriétaire des fichiers envoyés :

```
chown_uploads=YES
```

```
chown_username=utilisateur
```

Ainsi, tout fichier envoyé sera considéré comme créé par l'utilisateur défini (ce qui permettra au besoin de créer les « acl » nécessaires).

## Gérer les « Comptes utilisateur »

Comme indiqué au début de ce document il est question dans ce tutoriel de créer un petit serveur Web et de proposer aux clients la possibilité de placer leur propre page web. Pour leur permettre de mettre à jour cette page web, ils pourront le faire au travers du serveur FTP, pour cela il va être nécessaire de leur donner la possibilité d'avoir leur propre dossier, et leur propre accès.

On utilisera les comptes système pour les clients.

Afin de limiter l'accès au site web du « client », il faut activer son authentification. Pour permettre l'identification des utilisateurs système, il est nécessaire de placer le paramètre « local\_enable » à oui :

```
local_enable=YES
```

Il sera nécessaire (pour qu'ils puissent mettre leur page web sur le site) leur autoriser l'écriture :

```
write_enable=YES
```

## Sécuriser le FTP (SFTP – SSL)

Les comptes utilisateurs peuvent avoir accès aux fichiers de tout le système. Dans le cadre de notre serveur FTP ce n'est pas souhaitable !

Il est alors nécessaire de les enfermer (confiner) dans leur dossier. C'est le principe de base de la virtualisation : l'utilisateur est enfermé dans un dossier (un machine, un programme) dont il ne peut pas s'écarter.

Pour cela activer le paramètre « chroot\_local\_user »

```
chroot_local_user=YES
```

La racine de leur FTP sera alors leur répertoire personnel.

**Malgré tout**, un compte peut être utilisé pour se connecter autrement qu'en « ftp » : ssh ou getty par exemples. Ce qui lui donnera la possibilité d'accéder au reste du système par le shell.

Le blocage des compte sera expliquer durant la sécurisation du serveur.

## Message d'accueil

Il est possible de modifier le message d'accueil de votre serveur FTP en plaçant dans ce paramètre le texte voulu « ftpd\_banner » :

```
ftpd_banner=Bienvenue sur le serveur LoiselJP debian-web.
```

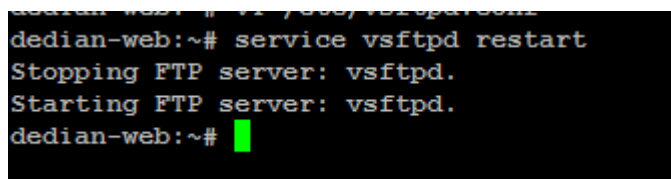
## Valider et tester les modifications

Cette fois le serveur est opérationnel, le chemins sont définis, le sécurité de login l'est également, il ne reste qu'à démarrer ou redémarrer le service :

```
service vsftpd start
```

ou

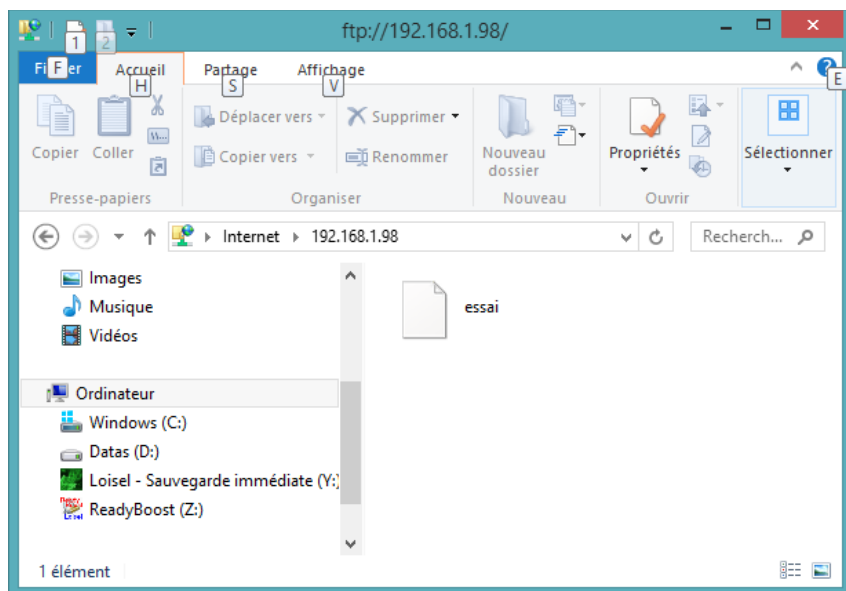
```
service vsftpd restart
```



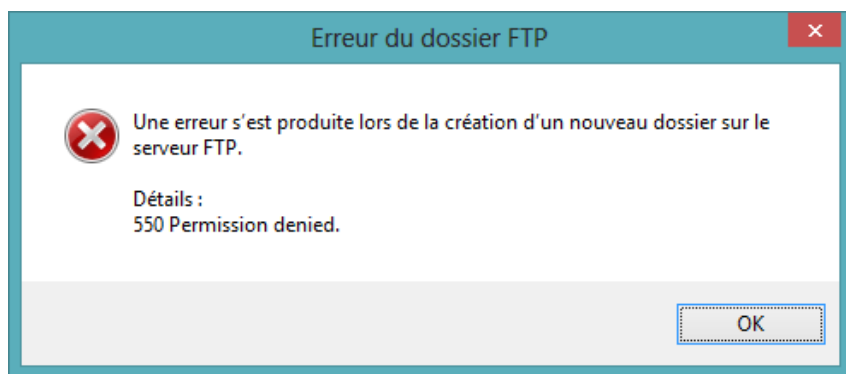
```
debian-web:~# service vsftpd restart
Stopping FTP server: vsftpd.
Starting FTP server: vsftpd.
debian-web:~#
```

## Faire un accès anonyme sur le serveur.

L'accès anonyme est simple car il suffira de taper l'adresse du serveur ftp depuis un exploreur de fichier en précisant le protocole ftp :



On pourra essayer de créer un nouveau fichier ou dossier...



Ce que le serveur refusera.

### ***Donner les droits en écriture pour les utilisateurs***

Dans les dernières versions de VSFTPD il est nécessaire de donner les droits en écriture sur le dossier de l'utilisateur (on parle de l'utilisateur enregistré qui aura le droit de créer son site web dans votre espace). Actuellement seul l'utilisateur «utilisateur» est accessible (ou l'utilisateur que vous avez créé à l'installation).

Pour donner les droits en lecture/écriture à « utilisateur », taper cette commande :

```
chmod a-w /home/utilisateur
```

(Le dossier de travail de l'utilisateur est dans /home/nom\_utilisateur)

Puis ajouter le paramètre suivant à /etc/vsftpd.conf :

```
nano /etc/vsftpd.conf
```

...

```
user_config_dir=/home/vsftpd_user_conf
```

Créer le dossier de configuration pour l'utilisateur (afin de lui donner les droits en écriture par exemple) :

```
mkdir /home/vsftpd_user_conf
```

Créer un dossier de configuration utilisateur par défaut :

```
nano /home/vsftpd_user_conf/default
```

Ajouter les paramètres de l'utilisateur par défaut :

```
# parametres des utilisateurs par default  
anon_world_readable_only=NO  
  
write_enable=YES  
  
anon_upload_enable=YES  
  
anon_mkdir_write_enable=YES  
  
anon_other_write_enable=YES
```

Ici on autorise un utilisateur anonyme a uploader un fichier sans pour autant voir le contenu du dossier...

Créer et copier le fichier pour l'utilisateur « utilisateur »

```
cp /home/vsftpd_user_conf/default /home/vsftpd_user_conf/utilisateur
```

Terminer avec un redémarrage du serveur :

```
service vsftpd restart
```

### ***Accéder au serveur depuis un compte enregistré.***

Pour l'heure actuelle, seul deux comptes sont disponibles : root et utilisateur qui ont été créés à l'installation.

Pour l'essai, il est possible de créer un fichier...

```
cd /home/utilisateur/
```

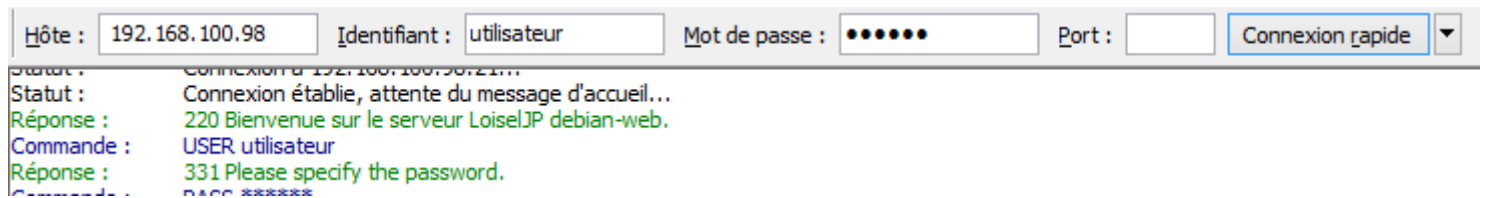
Après s'être déplacé dans le dossier de l'utilisateur, on crée un nouveau fichier vide (avec un nom caractéristique...) :

```
touch essai-utilisateur
```

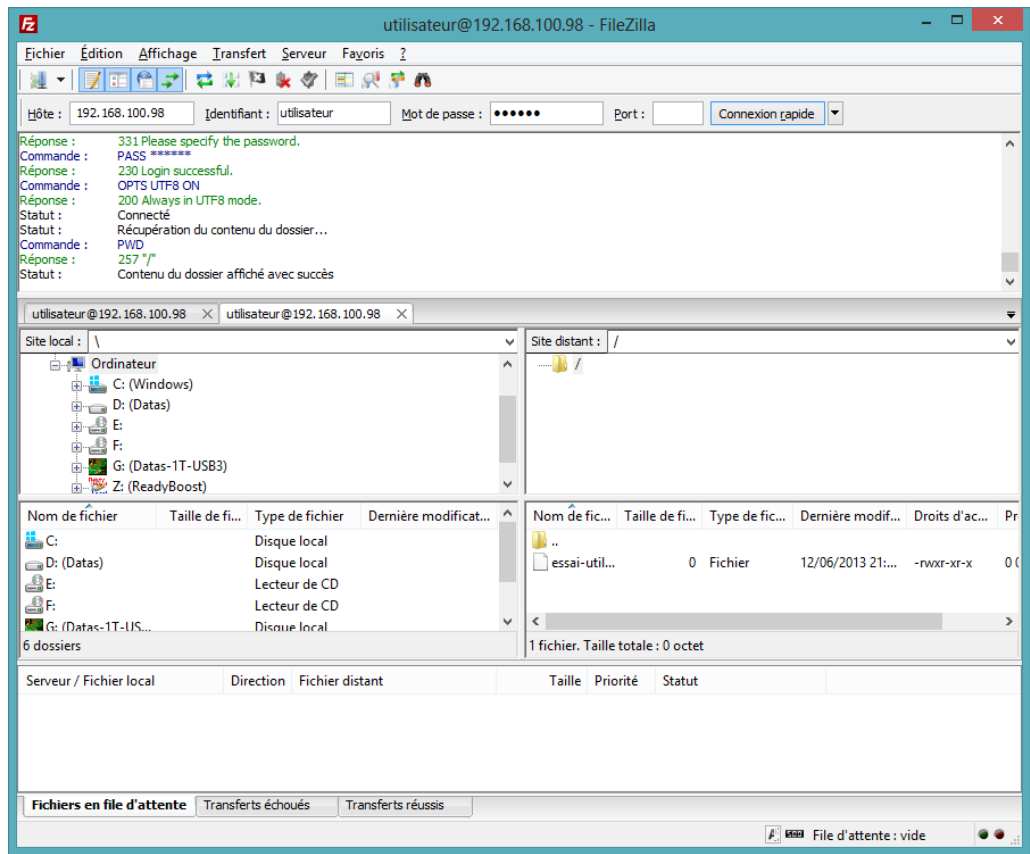
```
dedian-web:/# cd /home/utilisateur/  
dedian-web:/home/utilisateur# touch essai-utilisateur  
dedian-web:/home/utilisateur#
```

Depuis un client ftp se connecter au serveur. Pour cela on pourra utiliser de très bon freeware que l'on trouvera facilement sur l'Internet (Filezilla par exemple, que les autres m'en excuse).

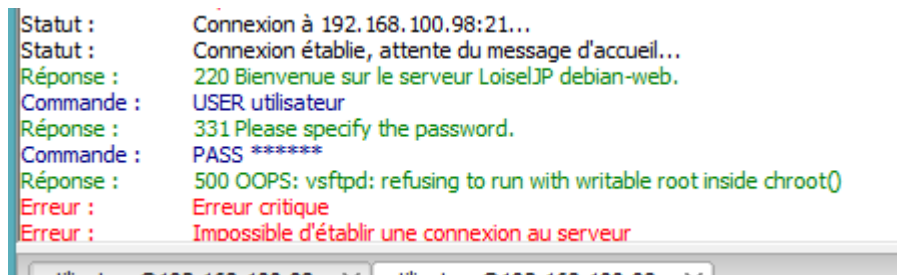
Dans les 2 impressions d'écran ci-dessous, on peut voir le login « utilisateur » l'adresse du serveur (son adresse IP), le mot de passe (mot de passe linux de l'utilisateur), ...



La seconde image permet de voir le chemin par défaut (/) ainsi que l'on est bien dans le dossier de l'utilisateur (on y voit le fichier créé précédemment).



Si vous rencontrez ce message :



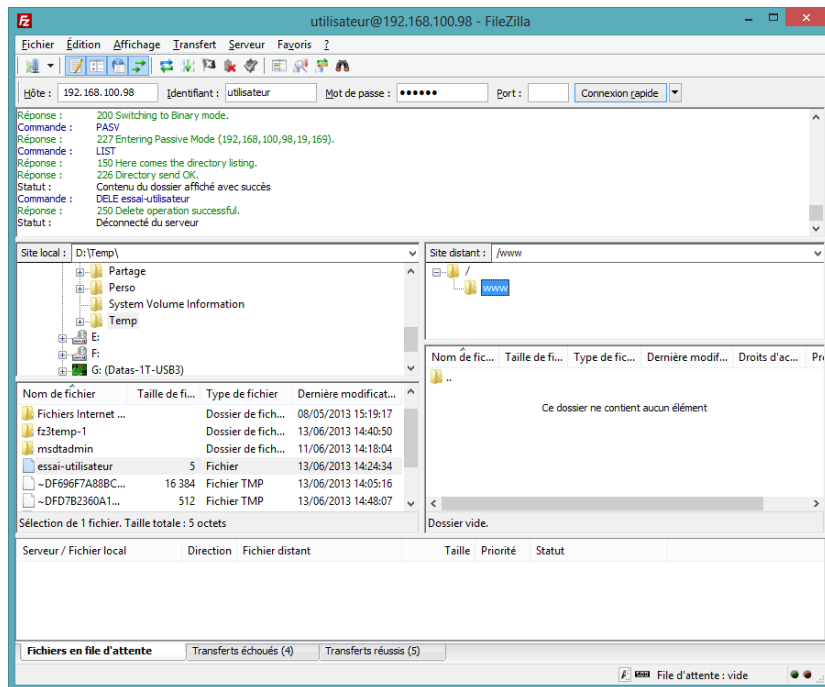
Penser à mettre les droits en écriture sur le dossier de votre utilisateur...

Par défaut l'utilisateur ne peut pas écrire directement dans la racine du dossier, nous allons donc lui créer un dossier « web » (« html », « www ») avec les droits en écriture :



```
mkdir /home/utilisateur/www
```

```
chmod 777 /home/utilisateur/www
```



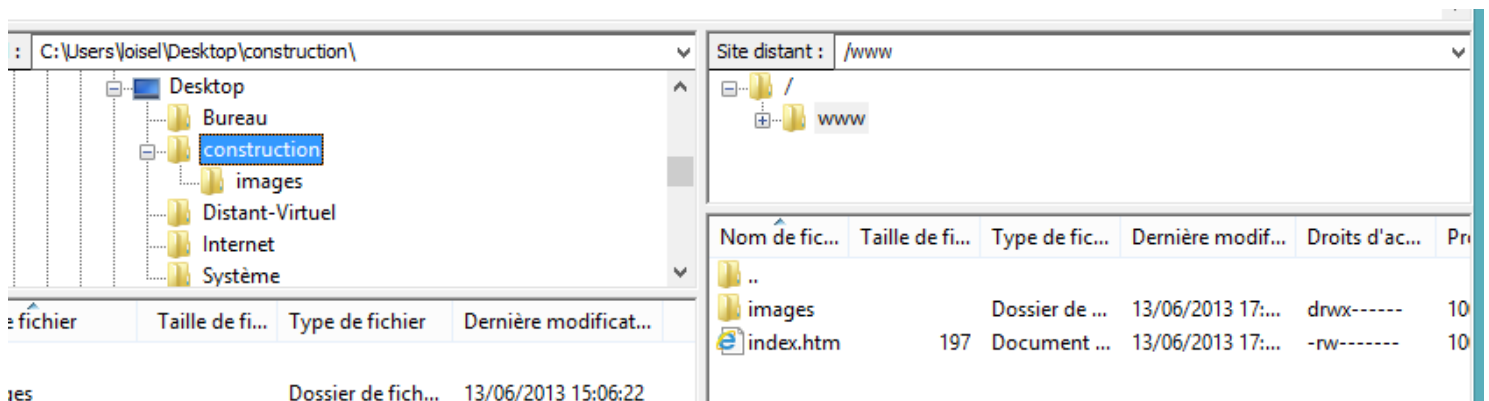
Cette fois « utilisateur » a un dossier dans lequel il peut écrire et modifier ce que bon lui semble.

### *Lier le dossier « www » avec le serveur http*

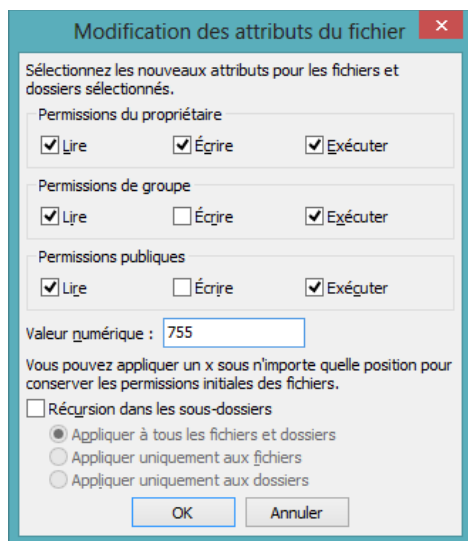
A cette dernière étape de dossier de l'utilisateur va cette fois être lié au serveur apache dans un dossier « utilisateur » :

```
ln -s /home/utilisateur/www /var/www/utilisateur
```

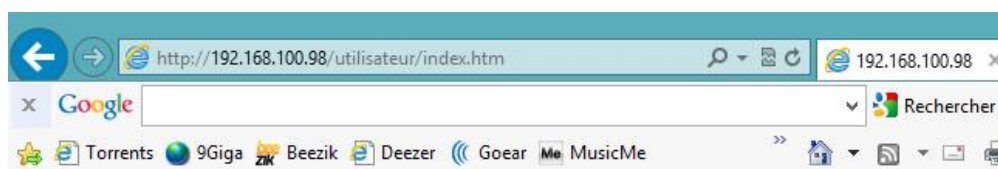
Transférer sur le dossier de l'utilisateur une petite page web du type « en construction »...



Un petit transfert...



Au besoin on y autorise les droits de lecture (+exécution)... on pourra y mettre les droits en écriture sur des fichiers de statistique par exemple.



**Bienvenue sur Le serveur Loiseljp**



La page s'affiche suivant l'adresse voulue et suivants les droits accordés.

*(La présentation et l'écriture de fichier PHP/HTML ne sont pas au sujet de ce tutoriel)*

A cette étape du tutoriel, le serveur fonctionne et est opérationnel.

## 10 Mettre en place des quotas

Ce paragraphe montre comment installer et configurer les quotas pour les utilisateurs. Ce qui est indispensable dans le cas de création de comptes utilisateur. Pour la taille accordée, c'est une question d'espace, d'emploi...

Il n'est pas ici question de traiter les quotas de bout en bout, on suppose que le lecteur (vous) aura quelques notions à ce sujet et notamment dans les définitions de quotas (stricte, souple, inode...).

***Pour rappel les quotas ne sont utilisables que sur une partition séparée du système !***

Les quotas sont gérés à partir d'un logiciel.  
Installer les programmes suivants:

```
apt-get install quota quotatool
```

```
dedian-web:~# apt-get install quota quotatool
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  dbus libdbus-1-3 libnl-3-200 libnl-genl-3-200 libsystemd-login0
Paquets suggérés :
  dbus-x11 libnet-ldap-perl rpcbind portmap default-mta mail-transport-agent
Les NOUVEAUX paquets suivants seront installés :
  dbus libdbus-1-3 libnl-3-200 libnl-genl-3-200 libsystemd-login0 quota quotatool
0 mis à jour, 7 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 1 317 ko dans les archives.
Après cette opération, 3 208 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ?
```

Le système de fichiers par défaut n'est pas positionné pour utiliser les options de quota. Il est donc nécessaire de modifier ces options.

Les options de gestion des quotas sont mises en place dès le montage de la partition. Le programme destiné à la gestion des partitions à son fichier de configuration sur /etc : /etc/fstab

```
cp /etc/fstab /etc/fstab_old
```

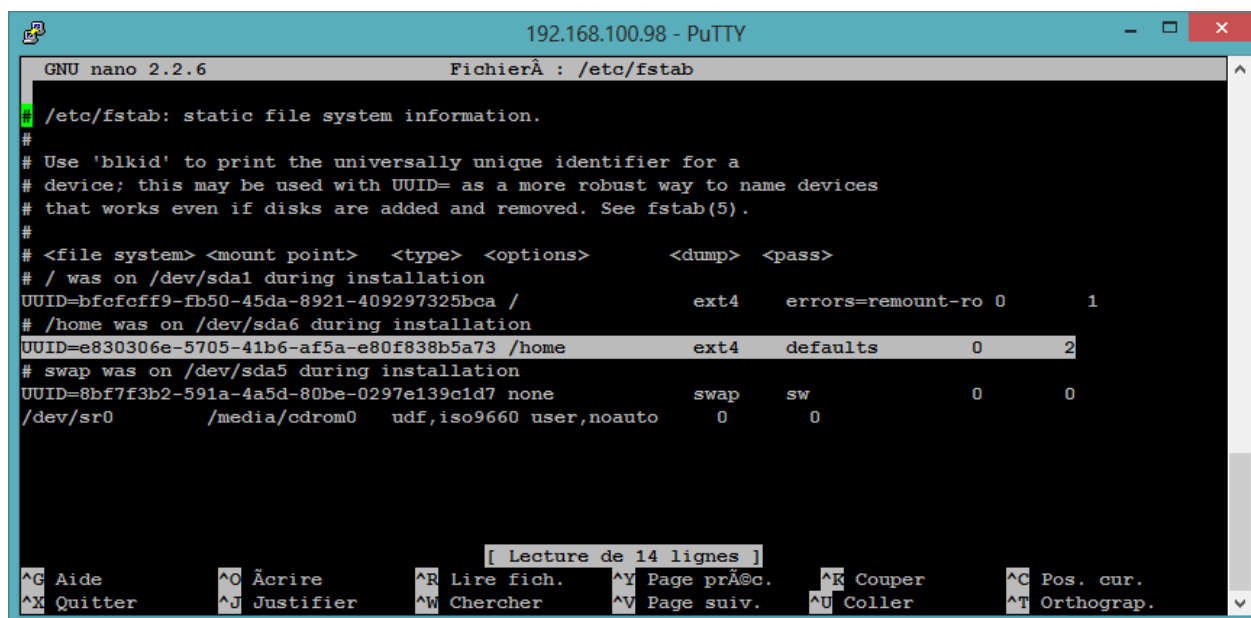
Ouvrir le fichier de configuration du système de fichier :

```
nano /etc/fstab
```

Puis modifier la ligne correspondant à la partition sur laquelle activer les quotas en lui ajoutant cette gestion dans les options de montage.

Trouver la ligne correspondant à la partition à modifier. Par exemple si «home » se trouve dans « /dev/sda3 »

```
/dev/sda6 /home auto defaults 0 2
```



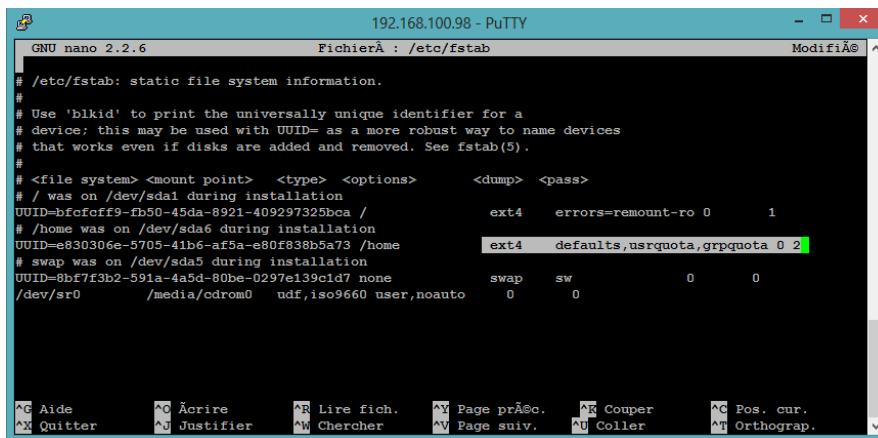
```
192.168.100.98 - PuTTY
GNU nano 2.2.6      Fichier : /etc/fstab
/etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>          <dump> <pass>
# / was on /dev/sda1 during installation
UUID=bfcfcff9-fb50-45da-8921-409297325bca /              ext4      errors=remount-ro 0      1
# /home was on /dev/sda6 during installation
UUID=e830306e-5705-41b6-af5a-e80f838b5a73 /home            ext4      defaults      0      2
# swap was on /dev/sda5 during installation
UUID=8bf7f3b2-591a-4a5d-80be-0297e139c1d7 none             swap      sw           0      0
/dev/sr0          /media/cdrom0    udf,iso9660 user,noauto     0      0

[ Lecture de 14 lignes ]
^G Aide      ^O Écrire    ^R Lire fich. ^Y Page préc. ^K Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher   ^V Page suiv. ^U Coller    ^T Orthograp.
```

Sur le serveur utilisé lors de ce tutoriel, celui-ci utilise les UUID pour identifier les partitions, ici la partition montée dans est surlignée.

La changer en :

```
/dev/sda6 /home auto defaults,usrquota,grpquota 0 2
```



```
192.168.100.98 - PuTTY
GNU nano 2.2.6 Fichier: /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=bfcfcff9-fb50-45da-8921-409297325bca / ext4 errors=remount-ro 0 1
# /home was on /dev/sda6 during installation
UUID=e830306e-5705-41b6-af5a-e80f838b5a73 /home ext4 defaults,usrquota,grpquota 0 2
# swap was on /dev/sda5 during installation
UUID=8bf7f3b2-591a-4a5d-80be-0297e139c1d7 none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0

^G Aide ^O Ãcrite ^R Lire fich. ^Y Page prÃsc. ^X Couper ^C Pos. cur.
^X Quitter ^U Justifier ^M Chercher ^V Page suiv. ^U Coller ^H Orthograp.
```

Le dossier /home étant monté... il sera nécessaire de démonter et remonter le volume.

Mise en place des quotas

Créer les fichiers nécessaires au fonctionnement des quotas.  
Les fichiers ne seront remplis que ultérieurement afin de créer des fichiers vides, tapez ces commandes :

```
touch /home/aquota.user
touch /home/aquota.group
```

Ces fichiers doivent être en écriture et lecture pour root uniquement.

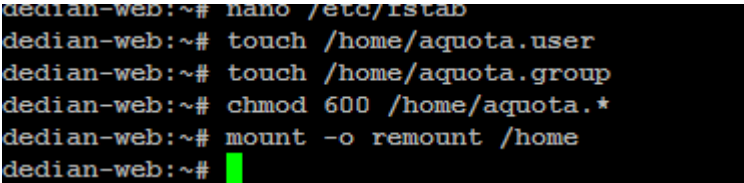
```
chmod 600 /home/aquota.*
```

Remonter le système de fichiers concerné pour prendre en compte l’utilisation de quotas pour ce système de fichiers.

```
mount -o remount /home
```

On pourra utiliser cette commande qui monte automatiquement les partitions inscrites dans fstab

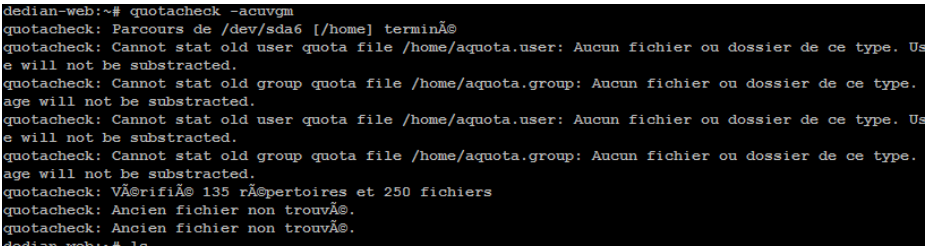
```
mount -a
```



```
dedian-web:~# nano /etc/fstab
dedian-web:~# touch /home/aquota.user
dedian-web:~# touch /home/aquota.group
dedian-web:~# chmod 600 /home/aquota.*
dedian-web:~# mount -o remount /home
dedian-web:~#
```

La commande quotacheck permet de scanner le système de fichier.

```
quotacheck -auvgm
```



```
dedian-web:~# quotacheck -auvgm
quotacheck: Parcours de /dev/sda6 [/home] terminÃ©
quotacheck: Cannot stat old user quota file /home/aquota.user: Aucun fichier ou dossier de ce type. Us
e will not be substracted.
quotacheck: Cannot stat old group quota file /home/aquota.group: Aucun fichier ou dossier de ce type.
age will not be substracted.
quotacheck: Cannot stat old user quota file /home/aquota.user: Aucun fichier ou dossier de ce type. Us
e will not be substracted.
quotacheck: Cannot stat old group quota file /home/aquota.group: Aucun fichier ou dossier de ce type.
age will not be substracted.
quotacheck: VÃ©rifiÃ© 135 rÃ©pertoires et 250 fichiers
quotacheck: Ancien fichier non trouvÃ©.
quotacheck: Ancien fichier non trouvÃ©.
dedian-web:~#
```

Si les anciens fichiers de quota « **aquota.user** » et « **aquota.group** » ne sont pas trouvé au besoin ils seront générés avec la commande.

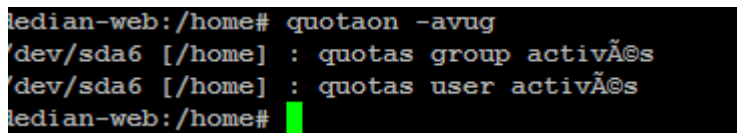
**En cas d'erreur c'est justement ce que l'on pourra faire : supprimer les fichiers aquota.\* puis relancer la commande :**

```
rm /home/aquota.*  
  
quotacheck -acuvgm
```

## Activer les quotas

Pour activer les quotas, exécuter cette commande

```
quotaon -avug
```



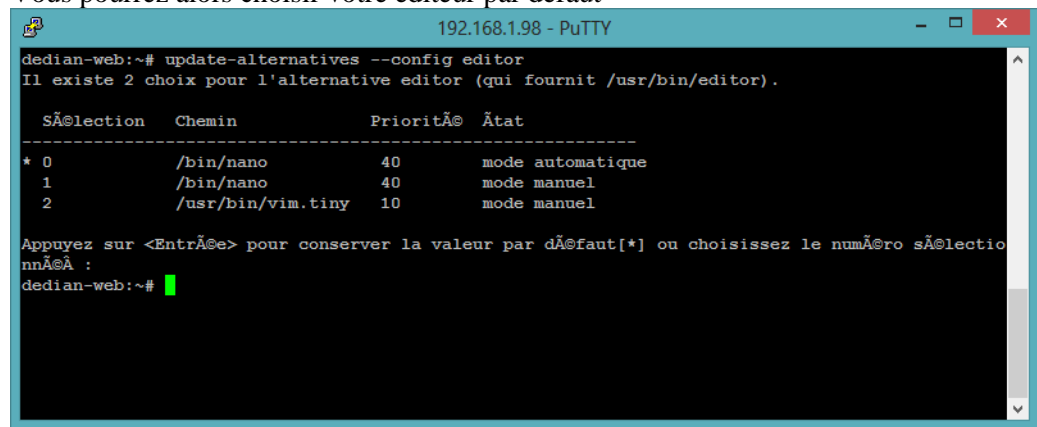
## Fixer les quotas user

Mettre en place les quotas sans les activer ou les fixer est un peu inutile.  
Pour les fixer, une commande fait le travail : « **edquota** ».

A son utilisation elle ouvre le fichier demandé avec l'éditeur « **nano** ».  
Certains administrateurs utilisent « **vi** » ou habituellement « **vim** » (il ne faut pas changer les habitudes).  
Il est possible alors de modifier la configuration en utilisant cette commande :

```
update-alternatives --config editor
```

Vous pourrez alors choisir votre éditeur par défaut



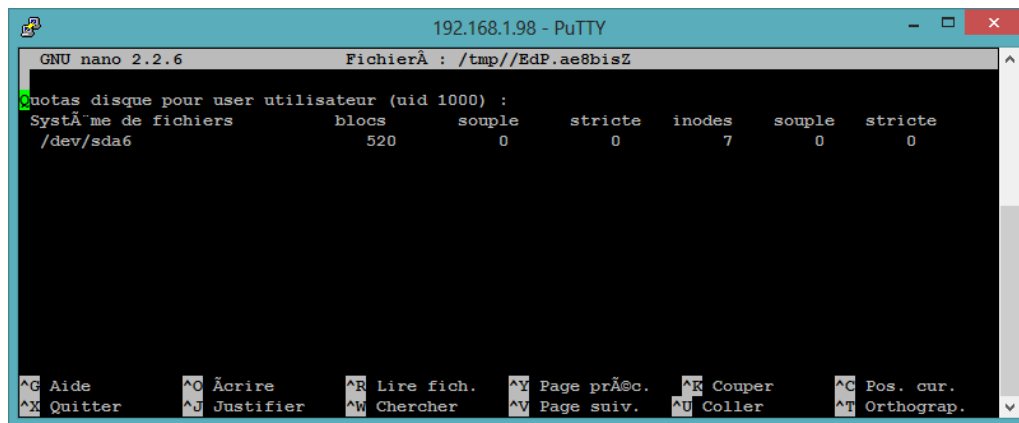
Il va s'agir d'éditer les quotas pour les utilisateurs... on pourra en obtenir la liste avec cette commande :

```
cut -f1 -d: /etc/passwd
```

Après avoir trouvé l'utilisateur voulu (ici « **utilisateur** ») activer ses quotas :

```
edquota -u utilisateur
```

Quotas disque pour utilisateur « utilisateur » :



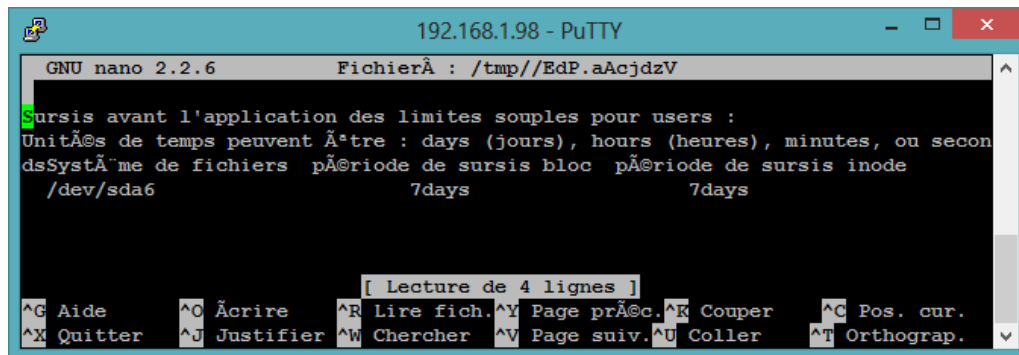
```
GNU nano 2.2.6 Fichier : /tmp/EdP.ae8bisZ
Quotas disque pour user utilisateur (uid 1000) :
Syst me de fichiers      blocs      souple      stricte      inodes      souple      stricte
/dev/sda6                520        0          0            7          0          0
```

Pour m moire : 1 bloc=1ko

La commande suivante permettra de fixer les limites d'utilisation des quotas.

edquota -t

Par d faut elles sont fix es   7 jours. D s lors que la p riode de sursit est d pass e, la souplesse est alors convertie en strict. Ainsi un utilisateur qui d passera sa limit  souple sera d finitivement bloqu  apr s 7 jours (par d faut)...



```
GNU nano 2.2.6 Fichier : /tmp/EdP.aAcjdzV
Sursis avant l'application des limites souples pour users :
Unit s de temps peuvent  tre : days (jours), hours (heures), minutes, ou seconds
Syst me de fichiers      p riode de sursis bloc      p riode de sursis inode
/dev/sda6                7days                7days
```

Ce sursit est destin  aux uns et aux autres   corriger les  ventuels d bordements.

## D finir des quotas par d faut

IL n'y a pas de m thode particuli re pour placer des quotas par d faut. On pourra retenir comme solution de d finir les quotas d'un utilisateur, apr s d finition de ces quotas, ils pourront  tre r pliqu s avec la commande suivante :

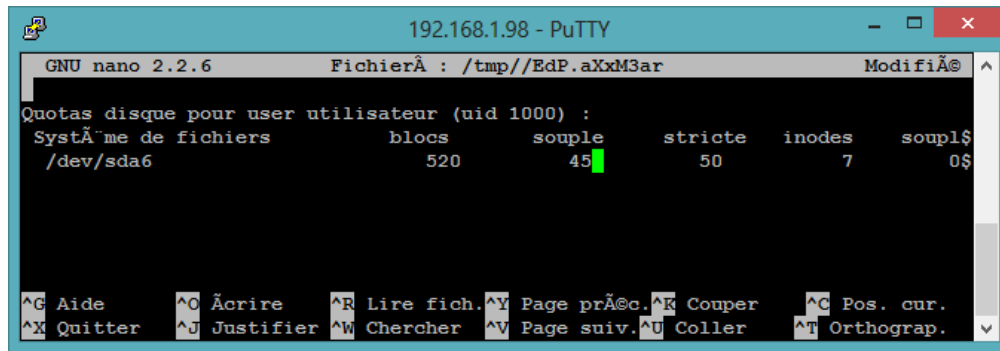
edquota -p utilisateur1 utilisateur2

On pourra alors envisager de cr er un utilisateur « User15 », un autre « User50 » avec une attribution de 15 Mo pour User15 et 50 Mo pour User50...

D s lors on pourra affecter les quotas de l'un ou de l'autre au besoin.

Dans le cadre de ce tutoriel, on utilisera comme profile, celui de « utilisateur ». On va alors fixer les quotas de cet utilisateur

à 50Mo (ce qui est un espace moyen généralement proposé).



```
GNU nano 2.2.6      Fichier : /tmp//EdP.aXxM3ar      Modifié : ^
Quotas disque pour user utilisateur (uid 1000) :
Système de fichiers      blocs      souple      stricte      inodes      souple$
/dev/sda6                520        45         50          7          0$

^G Aide      ^O Écrire    ^R Lire fich.^Y Page précédente.^K Couper      ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher   ^V Page suiv.^U Coller     ^T Orthograp.
```

## Afficher l'utilisation des quotas

Pour l'administrateur, il est assez simple en ligne de commande d'avoir un rapport d'utilisation avec la commande « repquota » :

```
repquota -avu
```

Cette commande affichera les quotas de manière lisible en Ko Mo Go.

```
repquota -avus
```

## 11 Sécuriser le serveur

Le serveur présenté dans ce tutoriel est destiné à une utilisation sur le Web, il serait suicidaire de ne pas y mettre un minimum de sécurité, le niveau de sécurité présenté dans ce chapitre est bien entendu suffisant pour un niveau plus élevé, il n'y aura pas d'autre solution que de basculer vers des solutions externes ou propriétaires.

### Installer et configurer l'antivirus

On le dit, Linux ne contient pas de virus ce qui est faux, on en trouve, certes moins que sur d'autres systèmes et cela pour plusieurs raisons : les logiciels sont libres donc transparents (on peut facilement les comparer, obtenir les sources), les enjeux financiers sont inexistant, c'est également un jeu pour certains, la sécurisation du système est nettement plus forte que d'autres OS...

Mais les fichiers joints, les fichiers à télécharger, le code n'est pas pour autant protégé. Il est donc indispensable de mettre un antivirus sur le serveur.

Clamav est une solution libre qui présente de plus de bons résultats et qui est mis à jour très régulièrement. Facile à mettre en œuvre, particulièrement adapté à la mise en œuvre sur serveur, la solution est donc intéressante.

#### Installer et configurer Clamav

```
apt-get install clamav clamav-freshclam
```

```

median-web:/var/www# apt-get install freshclam
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
E: Impossible de trouver le paquet freshclam
median-web:/var/www# nano /etc/crontab
median-web:/var/www#
median-web:/var/www# apt-get install clamav clamav-freshclam
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  clamav-base libclamav6 libtommath0
Paquets suggérés :
  clamav-docs libclamunrar6
Les NOUVEAUX paquets suivants seront installés :
  clamav clamav-base clamav-freshclam libclamav6 libtommath0
0 mis à jour, 5 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 5 351 ko dans les archives.
Après cette opération, 12,6 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] n ? o

```

```
apt-get install clamav-unofficial-sigs
```

Mettre à jour la base de données virale :

```
dedian-web:/# freshclam
ClamAV update process started at Thu Jun 13 16:26:01 2013
Downloading main.cvd [ 3%]
```

Il peut être intéressant de mettre en place une vérification régulière du système

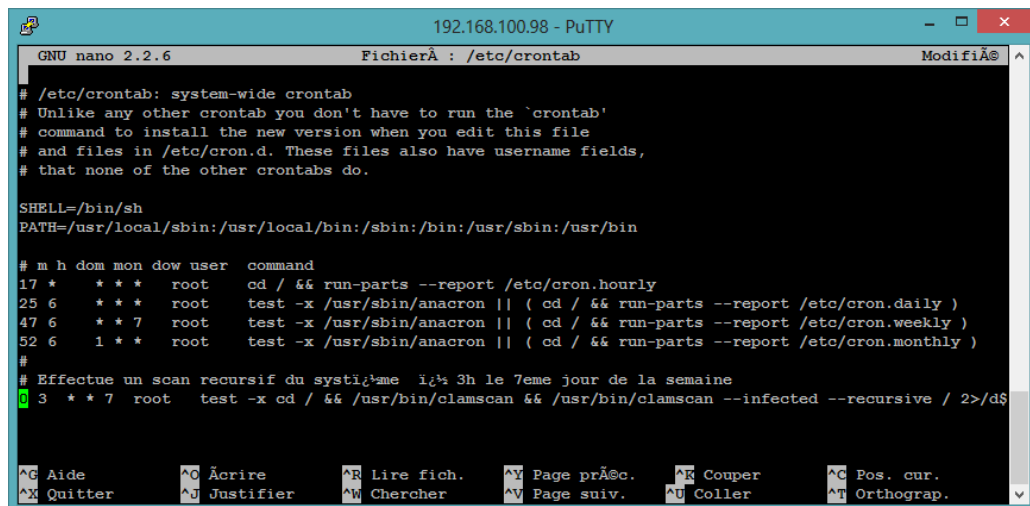
```
nano /etc/crontab
```

Administration serveur Installation d'un serveur WEB sous Linux page 40/85  
LoiselJP ©2013



```
# Effectue un scan recursif du système à 3h le 7eme jour de la semaine
```

```
0 3 * * 7 root test -x cd / && /usr/bin/clamscan && /usr/bin/clamscan --infected --recursive / 2>/dev/null" \ >
"/etc/cron.d/clamscan-weekly"
```



```
GNU nano 2.2.6 Fichier : /etc/crontab Modifié : 10/10/2013 14:10:10
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anaacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anaacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anaacron || ( cd / && run-parts --report /etc/cron.monthly )
#
# Effectue un scan recursif du système à 3h le 7eme jour de la semaine
0 3 * * 7 root test -x cd / && /usr/bin/clamscan && /usr/bin/clamscan --infected --recursive / 2>/dev/null" \ >
"/etc/cron.d/clamscan-weekly"
```

## Commandes de base

Scanner le répertoire courant

```
clamscan
```

Scanner tout le système

```
clamscan -r /
```

Affichage des fichiers infectés

Scanner un répertoire et ses sous-répertoires en n'affichant que les fichiers infectés :

```
clamscan -r -i /dossier_a_scanner
```

Se débarrasser ensuite du virus :

```
clamscan --remove /chemin_du_fichier_infecté
```

## Configurer le pare-feu

Pour les utilisateurs en grande difficulté devant l'utilisation et la configuration d'un pare-feu, il sera préférable de le configurer depuis une interface graphique... pour n'en citer qu'un, Firestarter s'acquittera très bien et facilement de cette tâche....

Malheureusement, le serveur n'a pas d'interface graphique...

Le pare-feu va donc être configuré manuellement dans le cadre de ce tutoriel.

### Présentation de Iptables

A l'origine, le pare-feu est installé mais est ouvert à toutes les communications, laissant passer toutes les communications, dans les deux sens, il ne remplit pas ses objectifs.

Dans ce chapitre, un script va être utilisé pour sauver la configuration en cours après avoir été configuré, le même script restaurera la configuration à chaque démarrage. En cas de mauvaise manipulation, dans le cas où l'utilisateur désirera remettre le pare feu à son état d'origine, le script utilisera un paramètre ayant ce rôle (clean).

### ***Créer un script de mise en place***

Dans un premier temps, un 'script' va être créé. Ce script sera destiné à recevoir les commandes vers le pare-feu.

Le script sera configuré pour être démarré avec le serveur, il sera mis dans « /etc/init.d ».

Créer le fichier 'parefeu' :

```
touch /etc/init.d/parefeu
```

Rendre le fichier exécutable avec commande :

```
chmod +x /etc/init.d/parefeu
```

Ouvrir et éditer le fichier « parefeu » :

```
nano /etc/init.d/parefeu
```

Dans le script suivant, il va être donné la possibilité d'utiliser 3 paramètres : « clean », « start », « stop » et « restart ».

- Le script utilisera le fichier « /etc/config\_parefeu » pour configurer le pare feu. Suivant la commande utilisée, il mettra en place (start) le contenu du fichier ou sauvera (stop) le fichier correspondant à la stratégie en place au moment de l'arrêt.

Pour obtenir le script effectuant la commande précédemment vue, ajouter au fichier le code suivant :

```
#!/bin/sh

# Script de démarrage et d'arrêt d'iptables

case "$1" in

'start')

/sbin/iptables-restore < /etc/config_parefeu

RETVAL=$?

;;

'stop')

/sbin/iptables-save > /etc/config_parefeu

RETVAL=$?

;;

'clean')

# nettoie le parefeu
```

```

/sbin/iptables -t filter -F
/sbin/iptables -t nat -F
/sbin/iptables -t mangle -F
/sbin/iptables -t raw -F
/sbin/iptables -t filter -P INPUT ACCEPT
/sbin/iptables -t filter -P OUTPUT ACCEPT
/sbin/iptables -t filter -P FORWARD ACCEPT
/sbin/iptables -t nat -P PREROUTING ACCEPT
/sbin/iptables -t nat -P POSTROUTING ACCEPT
/sbin/iptables -t nat -P OUTPUT ACCEPT
/sbin/iptables -t mangle -P PREROUTING ACCEPT
/sbin/iptables -t mangle -P OUTPUT ACCEPT
/sbin/iptables -t mangle -P POSTROUTING ACCEPT
/sbin/iptables -t mangle -P FORWARD ACCEPT
/sbin/iptables -t mangle -P INPUT ACCEPT
/sbin/iptables -t raw -P OUTPUT ACCEPT
/sbin/iptables -t raw -P PREROUTING ACCEPT
RETVAL=$?
;;
'restart')
$0 stop && $0 start
RETVAL=$?
;;
*)
echo "Usage: $0 { start | stop | restart | clean}"
RETVAL=1
;;
esac
exit $RETVAL

```

Pour mettre en place le pare feu on utilisera la commande suivante :

```
update-rc.d parefeu start 1 S . stop 99 0 6 .
```

## ***Initialiser le pare-feu***

Les commandes suivantes initialiseront le pare feu (elles vont mettre en place les stratégies):

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

# Pour le serveur FTP

```
Iptables -A INPUT -p tcp -m tcp --dport 21 -j ACCEPT
```

# Pour le serveur SSH

```
iptables -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
```

# Pour le serveur Web

```
iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

# pour rendre CUPS accessible (suivant le besoin)

```
iptables -A INPUT -p tcp -m tcp --dport 631 -j ACCEPT
```

Pour les autres ports on utilisera le terme « udp » ou « tcp » en fonction du type de paquets, et le numéro de port correspondant (la ligne suivante n'est pas une commande mais une explication de la commande) :

***iptables -A INPUT -p tcp\_ou\_udp -m tcp\_ou\_udp --dport num\_port -j ACCEPT***

Sauver le pare feu actuel :

```
/etc/init.d/parefeu stop
```

Puis le démarrer :

```
/etc/init.d/parefeu start
```

Il est préférable de travailler directement sur le serveur plutôt qu'en SSH à cette étape ! En cas de blocage du pare feu, il n'y aura pas d'autre recours que de travailler sur le serveur et éventuellement ajouter ou supprimer les règles nécessaires.

## ***Enlever une règle***

Pour connaître le numéro de ligne de la règle à supprimer, la commande suivante affichera les différentes règles avec leur numéro de ligne :

```
iptables -L --line-numbers
```

Il suffira ensuite de taper la ligne suivante (que l'on adaptera en fonction du besoin) avec XX qui correspondra au numéro de ligne à supprimer :

***iptables -D INPUT\_ou\_FORWARD XX***

Enfin cette commande remplacera le pare feu à son état d'origine :

```
/etc/init.d/parefeu clean
```

## DenyHosts

Lorsqu'un serveur en ligne sur internet, de nombreuses tentatives de connexion sur le port SSH sont effectuées.

La solution est, bien entendu de le supprimer les ports ouverts dans le pare feu, mais dès lors, il ne sera plus possible de l'atteindre par SSH (la méthode est bien pratique...).

Alors comment sécuriser contre les attaques en SSH

Denyhosts est un paquet qui va permettre de filtrer les clients.

Denyhosts analyse à chaque connexion le fichier de log « /var/log/auth.log » après plusieurs tentatives refusées, dont le nombre est défini dans la configuration, va « blacklister » l'IP en cause dans le fichier /etc/hosts.deny.

L'administrateur pourra même être averti par courriel lorsqu'une IP sera bannie.

Installer le paquet denyhosts :

```
apt-get install denyhosts
```

Dès son installation, le programme « denyHosts » protégera des intrusions.

Il sera encore possible de personnaliser la configuration en éditant le fichier « /etc/denyhosts.conf ».

```
nano /etc/denyhosts.conf
```

Quelques variables intéressantes :

```
BLOCK_SERVICE = ALL ou sshd # pour bloquer l'accès à tous les services ou seulement à ssh
DENY_THRESHOLD_INVALID = 5 #Nombre d'échec avec un login invalide avant blocage
DENY_THRESHOLD_VALID =10 # Nombre d'échec avec un login valide avant blocage
DENY_THRESHOLD_ROOT = 1 # Nombre d'échec avec le login root avant blocage
ADMIN_EMAIL = root@localhost # Adresse mail qui sera informé à chaque blocage d'une IP
SMTP_HOST = localhost # Relais SMTP pour l'envoi des mails de notification
SMTP_PORT = 25 # Port utilisé pour le SMTP
SMTP_FROM = DenyHosts <nobody@localhost> # définir le nom de l'expéditeur de la notification
SMTP_SUBJECT = Rapport DenyHosts (protection intrusion SSH) # définir le sujet du mail de notification
```

Dans le cas où l'on se trompe trop souvent de mot de passe il est possible de ne plus être en mesure de se connecter.

Le fichier « /etc/hosts.allow » pourra contenir un ensemble d'adresses IP toujours autorisées à se connecter.

```
nano /etc/hosts.allow
```

Ajouter des adresses IP individuelles ou groupées sous cette forme :

```
ALL: 192.168.1.
```

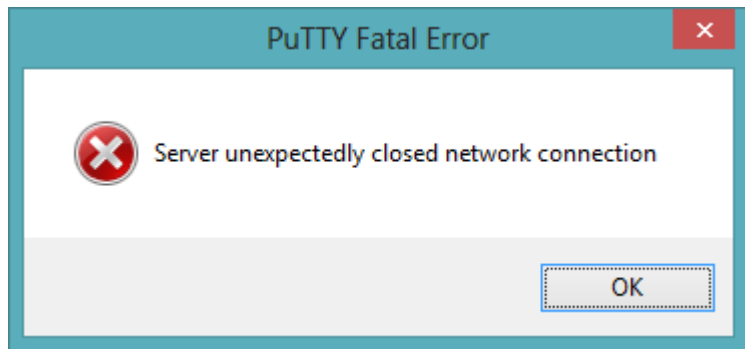
```
ALL: 192.168.2.
```

```
ALL: 192.168.100.
```

```
ALL: .loiseljp.fr
```

(« loiseljp.fr » est un exemple de domaine, vous pouvez mettre autre chose, voir aucun domaine !)

Après quelques essais infructueux... le compte est bloqué (pour la circonstance « hosts.allow » ne contenait pas l'IP du poste):



On pourra également obtenir la liste des IP bloquées :

```
nano /etc/hosts.deny
```

```
# ALL: PARANOID
sshd: 192.168.1.86
```

Il faudra supprimer l'adresse pour être autorisé de nouveau à se connecter... (Ou ajouter le poste dans le fichier « hosts.allow »).

## Bloquer les comptes clients

Lors de la configuration du serveur « vsftpd » il a été vu que le client utilise un compte utilisateur. Un utilisateur peut utiliser les fonctionnalités du serveur (c'est le principe).

En rencontrera deux type d'utilisateurs le client qui utilise son accès FTP sur le serveur pour y placer ses pages web (ou autres fichiers). Le second utilisateur est celui qui se logue en ftp pour télécharger des fichiers.

Par défaut dans vsftpd, le compte anonyme passe sur « utilisateur », ce dernier n'a pas à travailler sur le serveur.

La solution la meilleure est donc de lui interdire l'emploi du shell.

Pour cela on assigne le shell de l'utilisateur vers false. False est un simple binaire qui renvoi un signal d'erreur (il est déjà intégré au système):

```
usermod -s /bin/false utilisateur
```

Il faut ensuite ajouter false à la liste des shells :

```
/bin/false >> /etc/shells
```

A la connexion de « utilisateur », ou tout utilisateur qui aura été bloqué, le shell ne pourra être lancé, la fenêtre de connexion reviendra sur login !

## Mettre en place les ACLs

Les Listes de Contrôle d'Accès sont assez mal connues. Ce tutoriel n'a pas la prétention de présenter ou d'expliquer le fonctionnement des ACL. L'utilisateur désirant aller plus loin dans ce domaine, trouvera de très bons articles sur le sujet sur Internet.

Avant de pouvoir utiliser ces acl, il faudra installer le programme de gestion

```
apt-get install acl
```

Le terme ACL désigne deux choses en sécurité informatique :

- Un système permettant de faire une gestion plus fine des droits d'accès aux fichiers que ne le permet la méthode employée par les systèmes UNIX,
- En réseau, une liste des adresses et ports autorisés ou interdits par un pare-feu.

Dans le cas qui nous intéresse, ce sont les droits sur les fichiers qui nous intéressent (les listes de ports sur le parefeu ont été traitées précédemment).

Une ACL est une liste d'Access Control Entry (ACE) ou entrée qui permet de donner ou supprimer des droits d'accès à une personne ou un groupe.

Sous Unix, on ne trouve que trois types de droits (RWX) pour trois types d'utilisateurs (le propriétaire, le groupe, tous les autres utilisateurs).

Les deux commandes permettant de visualiser les droits ACL et de les modifier sont, sous UNIX :

**getfacl** : pour visualiser

**setfacl** : pour modifier

Utilisation des commandes :

```
getfacl fichier_ou_dossier
setfacl -m u:nom_utilisateur:droits_rwx fichier_ou_dossier
setfacl -m g:nom_groupe:droits_rwx fichier_ou_dossier
setfacl -x u:nom_utilisateur # pour retirer les droits
setfacl -m o:: - fichier_ou_dossier # pour remettre a normal les autres permissions
setfacl -m d:u:nom_utilisateur:droits_rwx dossier # pour placer les droits par défaut
```

Ce qui nous intéresse ici c'est cette dernière commande qui permettra de placer les droits par défaut sur un fichier ou dossier

```
setfacl -m d:u:utilisateur:rwx /home/utilisateur
```

On notera l'absence d'espaces entre les paramètres sauf pour le nom de dossier ou de fichier

### Activer les ACL

L'activation des ACLs passe par une option de montage. Tout comme il l'a été fait lors des quotas, il va être nécessaire de les activer dans « fstab ».

```
dedian-web:/home# setfacl -m d:u:utilisateur:rwx /home/utilisateur
dedian-web:/home# ll
total 48
-rw----- 1 root      root      7168 juin 14 14:41 aquota.group
-rw----- 1 root      root      7168 juin 14 14:41 aquota.user
drwxr-xr-x 5 root      root      4096 juin 11 20:07 fichiers_systeme
drwx----- 2 root      root     16384 juin 11 09:50 lost+found
dr-xrwxr-x+ 3 utilisateur utilisateur 4096 juin 13 14:50 utilisateur
drwxr-xr-x 2 root      root      4096 juin 13 14:38 vsftpd_user_conf
drwxr-xr-x 3 root      root      4096 juin 13 15:01 www
dedian-web:/home# _
```

On remarque maintenant le « + » à côté des attributs signifiant que le dossier contient des acl.

```
dedian-web:/home# getfacl /home/utilisateur
getfacl : suppression du premier « / » des noms de chemins absolus
# file: home/utilisateur
# owner: utilisateur
# group: utilisateur
user::r-x
user:utilisateur:rwx
group::r-x
mask::rwx
other::r-x
default:user::r-x
default:user:utilisateur:rwx
default:group::r-x
default:mask::rwx
default:other::r-x
```

## 12 Configurer le serveur

Si les différentes consignes d'installation ont été suivies dans les chapitres précédents, le serveur devrait être prêt à subir toutes les maltraitances des utilisateurs.

Cependant, quelques paramètres seront encore à définir pour vivre en toute tranquillité. Suivant la stratégie de gestion du serveur, il sera peut être nécessaire de limiter l'emploi de certains services. Ces services seront configurés dans le serveur PHP.

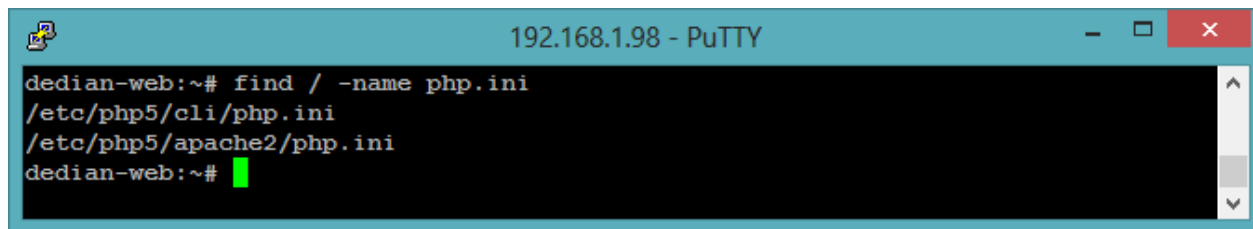


## Configurer le serveur PHP

Afin d'éviter toute confusion entre les différents fichiers `php.ini` (c'est le fichier de configuration), nous allons éliminer un fichier pour le lier à l'autre. Ainsi un seul et même fichier pourra être employé :

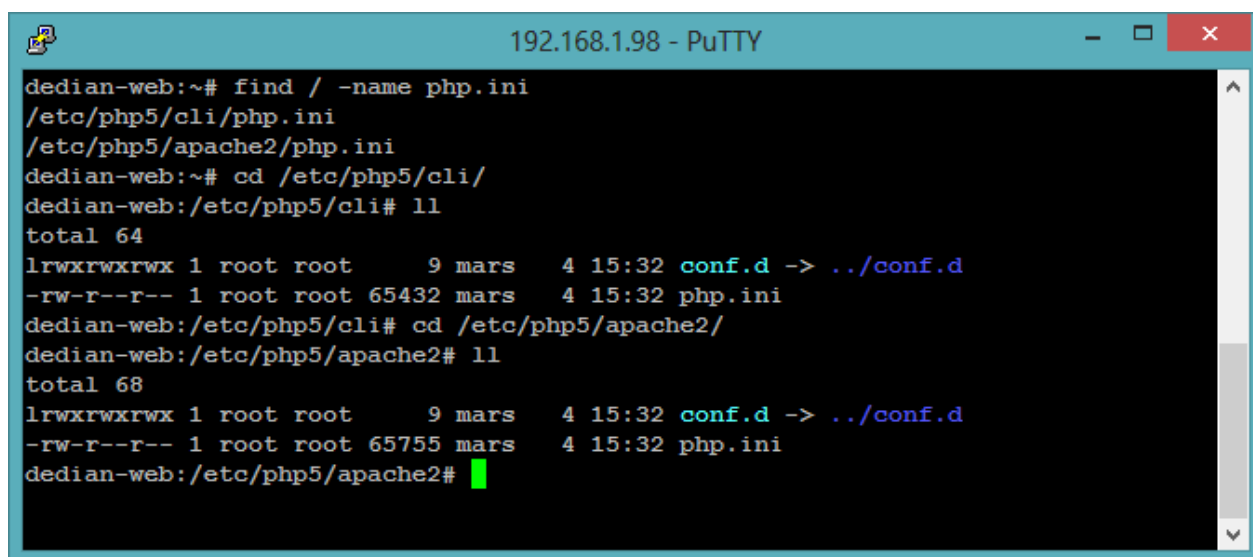
On pourra rechercher auparavant les fichiers présents sur le serveur :

```
find / -name php.ini
```



```
192.168.1.98 - PuTTY
dedian-web:~# find / -name php.ini
/etc/php5/cli/php.ini
/etc/php5/apache2/php.ini
dedian-web:~#
```

Après vérification des fichiers (s'ils ont le même rôle, s'ils ne sont pas déjà liés...),



```
192.168.1.98 - PuTTY
dedian-web:~# find / -name php.ini
/etc/php5/cli/php.ini
/etc/php5/apache2/php.ini
dedian-web:~# cd /etc/php5/cli/
dedian-web:/etc/php5/cli# ll
total 64
lrwxrwxrwx 1 root root    9 mars  4 15:32 conf.d -> ../conf.d
-rw-r--r-- 1 root root 65432 mars  4 15:32 php.ini
dedian-web:/etc/php5/cli# cd /etc/php5/apache2/
dedian-web:/etc/php5/apache2# ll
total 68
lrwxrwxrwx 1 root root    9 mars  4 15:32 conf.d -> ../conf.d
-rw-r--r-- 1 root root 65755 mars  4 15:32 php.ini
dedian-web:/etc/php5/apache2#
```

Supprimer le ou les fichiers supplémentaires :

```
rm /etc/php5/cli/php.ini
```

Puis lier les fichiers :

```
ln -s /etc/php5/apache2/php.ini /etc/php5/cli/
```

## Modifier les directives du fichier `php.ini`

Le fichier de configuration `php.ini` est un fichier très complet et très largement commenté. Bien que peu compliqué, il sera facile de le modifier, mais devant le nombre de directives disponibles, il sera facile de s'y perdre.

Ce chapitre ne peut pas toutes les détailler car trop nombreuses, mais aussi car ce n'est pas dans l'objectif de ce tutoriel.

Seules les plus intéressantes (pour la configuration de base d'un serveur) seront décrites.

Afin de pouvoir changer la configuration du serveur, passer le fichier php.ini en édition :

```
nano /etc/php5/apache2/php.ini
```

Pour désactiver une fonction (mail par exemple) trouver la directive `disable_functions` :

```
; It receives a comma-delimited list of function names. This directive
; *NOT* affected by whether Safe Mode is turned On or Off.
; http://php.net/disable-functions
disable_functions = pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_w
; This directive allows you to disable certain classes for security
; It receives a comma-delimited list of class names. This directive
```

Puis ajouter la directive à supprimer :

```
disable_functions = mail,
```

Désactiver le rapport d'erreurs, en production il est conseillé de mettre ces valeurs :

```
= E_ALL & ~E_DEPRECATED & ~E_STRICT
```

```
display_errors = Off
```

```
log_errors = On
```

```
error_log = "error.log"
```

La plus intéressante des directives est « `safe_mode` », elle est recommandée pour les FAI et les services d'hébergement partagé (pour ce serveur !), dans la mesure où elle limite les actions qu'un utilisateur peut faire:

```
safe_mode = Off
```

Il est possible de limiter les opérations sur les fichiers avec la directive « `open_basedir` ».

Cette directive va agir de la même manière qu'un chroot en définissant le répertoire de base de toute opération sur le serveur, par du code déterminé, un utilisateur ne pourra pas utiliser un fichier se trouvant dans un dossier différent que celui défini par la directive:

```
open_basedir = /home/www/
```

Toujours dans le cadre de la sécurité, on désactivera la possibilité de placer des scripts dans les formulaires (ils sont malins ces internautes) :

```
register_globals = Off
```

La directive « `max_execution_time` » définit le nombre de secondes maximal pendant lesquelles le serveur attendra qu'un script ait fini de s'exécuter.

Ce choix est important afin de limiter la charge du serveur (les clients ne se gêneront pas de faire travailler le serveur). Dès lors que temps d'exécution du script dépassera le temps défini, le script sera « tué ». Attention cependant à ne pas tuer des scripts qui pourraient être importants seuls quelques essais permettront de terminer le temps adéquat (20 à 50 secondes est une bonne moyenne), cela dépendra du nombre de clients, taille du serveur, puissance...

```
max_execution_time = 20
```

Le téléchargement de fichiers à l'aide de la variable « file\_uploads » pourra sauver votre serveur de téléchargements liés aux formulaires.

Si à tout hasard vous laissez ce paramètre sur « on », il sera encore possible de définir la taille maximale d'un téléchargement à l'aide de la variable upload\_max\_filesize.

On choisira généralement une taille de fichier très petite (quelques centaines de ko maximum), à moins d'en avoir besoin:

```
file_uploads = On
```

```
upload_max_filesize = 500k
```

On pourra aussi définir le nombre de fichiers maximum pouvant être uploadés, à traiter suivant la stratégie :

```
max_file_uploads = 10
```

Une autre directive à placer sur les formulaires est de limiter la taille des données envoyées, ce choix se fera en fonction de la stratégie et de l'emploi du serveur, mais on notera que l'internaute a souvent une connexion de quelques centaines de ko maximum en upload, il est très facile de saturer les connexions... bon nombre d'hébergeurs limitent les données à quelques Koctets,... à calculer suivant les besoins :

```
post_max_size = 1M
```

La directive « max\_input\_time » va permettre de limiter les temps d'upload, cette directive sera intéressante pour limiter les chargements très lents... elle s'exprime en secondes :

```
max_input_time = 30
```

Pour améliorer les performances du serveur PHP. Il est possible de définir des limites d'utilisation de la mémoire.

La directive « memory\_limit » spécifie la mémoire maximale qui peut être utilisée par un script:

```
memory_limit = 8M
```

Pour plus d'information sur les directives cette page sera d'un grand secours : <http://php.net/manual/fr/ini.core.php>

Php.ini est lu à chaque script, il n'est donc pas nécessaire de le redémarrer en fin de modification. Mais, pour ceux qui le désireront (ce qui ne peut pas faire de mal non plus), il est possible d'utiliser cette commande :

```
/etc/init.d/apache2 restart
```

**On notera également** que les directives par défaut sont souvent différentes de la réalité, afin de pouvoir héberger plus d'utilisateurs sur un même serveur, ces ressources sont souvent placées à leur valeur minimale

## Mettre une page d'information pour les « clients ».

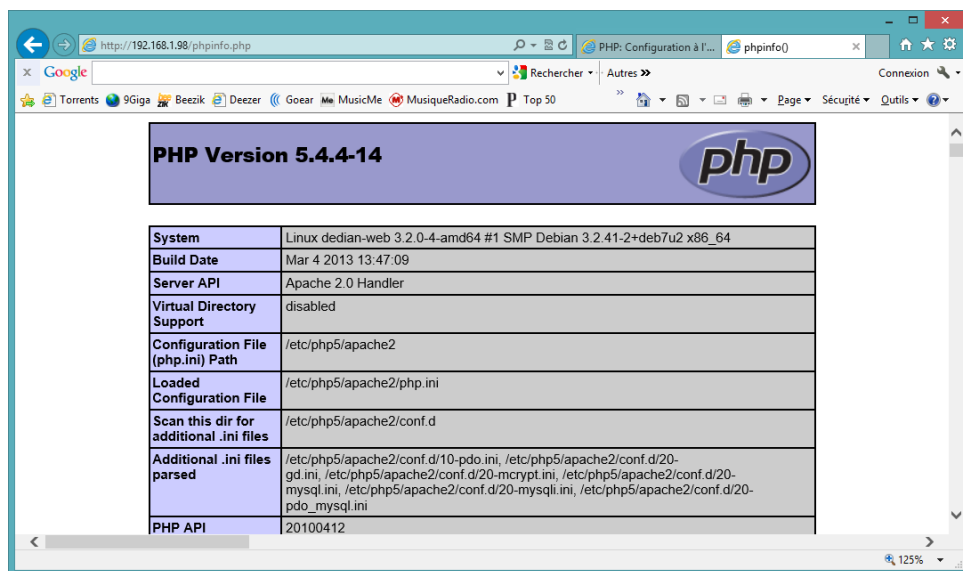
Il est possible de mettre en place une page web qui permettra de donner à vos clients une information sur la configuration php proposée :

Ouvrir une nouvelle page :

```
nano /var/www/phpinfo.php
```

Puis y ajouter le texte suivant :

```
<?php phpinfo(); ?>
```

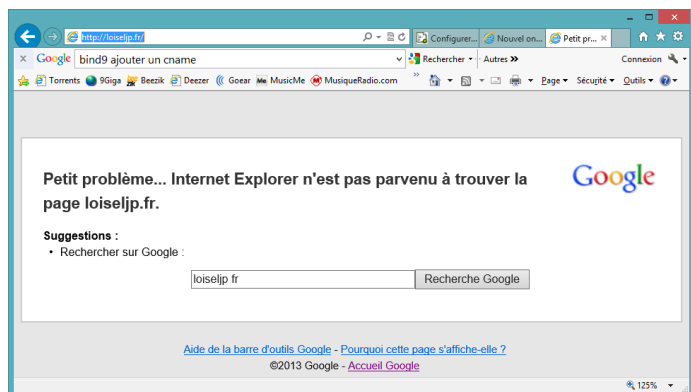


Pour plus d'information au sujet de « php » cette page sera d'un grand service : <http://www.php.net>

## Configurer le serveur de DNS.

A cette étape de la configuration du serveur, il s'agit d'enregistrer et configurer le serveur de DNS. Dans ce paragraphe la méthode complète (y compris le changement de nom du serveur) va être donnée.

Avant de configurer les DNS même sur un serveur de test, il sera préférable de vérifier que le nom de domaine choisi n'existe pas sur l'Internet, cela risquerait de fausser les résultats, voire de créer des conflits...



## Installer Bind9

BIND (Berkely Internet Name Domain) est un serveur DNS open source utilisé pour la résolution de noms Internet.

Dans le cadre d'un serveur web, il n'est pas totalement indispensable d'installer un serveur de DNS.

Dans ce cas deux cas se présentent : il n'y a pas de serveur DNS et le client se connecte en « http://domaine\_principal/client » ou il y a un serveur de dns et le client se connecte en « http://client.domaine\_principal/ » ce qui est plus sérieux !

Dans le cas de ce tutoriel, c'est le serveur Bind qui sera choisi, notamment pour sa robustesse. Après installation, il sera nécessaire de le configurer.

Installer le serveur de DNS :

```
apt-get install bind9
```

Il sera alors nécessaire de télécharger 1 à 2Mo environ (40Mo environ pour certaines installations sur Debian).



```
192.168.1.98 - PuTTY
dedian-web:~# apt-get install bind9
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  bind9utils
Paquets suggérés :
  bind9-doc resolvconf ufw
Les NOUVEAUX paquets suivants seront installés :
  bind9 bind9utils
0 mis à jour, 2 nouvellement installés, 0 à enlever et 2 non mis à jour.
Il est nécessaire de prendre 493 ko dans les archives.
Après cette opération, 1 291 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer [O/n] ?
```

📖 Dans ce chapitre, le nom de la machine (au besoin) et les accès DNS vont être modifiés, il convient de retenir que le nom du serveur devra respecter les conventions de nommage. Ainsi :

- Il ne doit pas contenir d'espace,
- Il ne doit contenir que des lettres, chiffres ou « - »
- Il ne pourra pas contenir de caractères spéciaux / \ "...

En cas de non-respect de ces règles, le réseau pourrait bien ne plus fonctionner, dans tel cas, il pourrait être nécessaire de vérifier ces impératifs.

### *Modifier le fichier /etc/hosts*

/etc/hosts est un fichier de gestion d'alias hôte et Ip.

Le fichier host va donc être chargé de fournir la translation de nom en adresses Ip (et inversement).

Pour la configuration du serveur de messagerie, il sera nécessaire d'enregistrer le serveur dans les DNS..

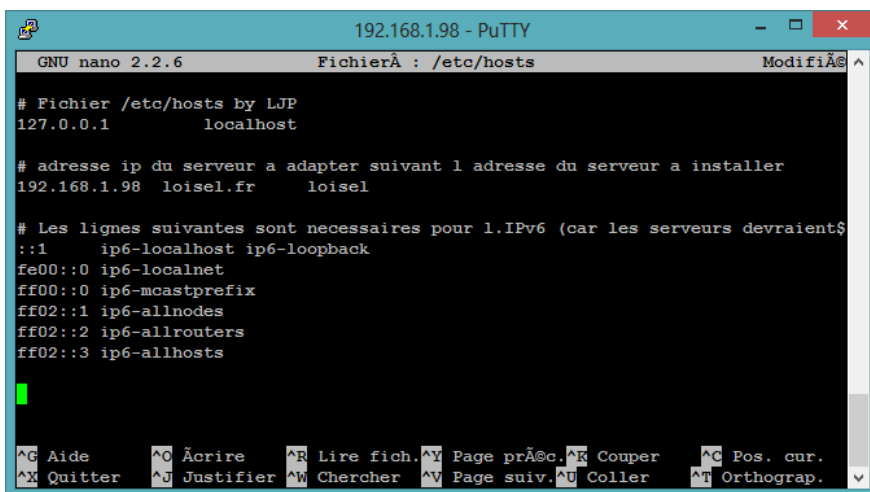
Modifier le fichier host en l'adaptant à votre besoin (nom, adresse IP...) :

```
# Fichier /etc/hosts by LJP
127.0.0.1    localhost

# adresse ip du serveur a adapter suivant l adresse du serveur a installer
192.168.1.98 loiseljp.fr  loiseljp

# Les lignes suivantes sont necessaires pour l'IPv6 (car les serveurs devraient l utiliser !)
::1    ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

**Remarque** : cette fois on donne un nom de domaine pour le serveur...



```
192.168.1.98 - PuTTY
GNU nano 2.2.6      Fichier : /etc/hosts      ModifiÃ©
# Fichier /etc/hosts by LJP
127.0.0.1    localhost

# adresse ip du serveur a adapter suivant l adresse du serveur a installer
192.168.1.98 loiseljp.fr  loiseljp

# Les lignes suivantes sont necessaires pour l'IPv6 (car les serveurs devraient$
::1    ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts

^G Aide      ^O Ã©crire   ^R Lire fich.^Y Page prÃ©c.^R Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher   ^V Page suiv.^U Coller     ^T Orthograp.
```

## Modifier le nom de la machine

On pourra en profiter pour modifier le nom de la machine...

```
nano /etc/hostname
```

Lui mettre le nom de la machine...

```
loiseljp
```

(vous pouvez choisir un autre nom)

## *Modifier le fichier resolv.conf*

Le fichier « /etc/resolv.conf » permet d'indiquer le ou les domaines de recherche et les différents serveurs DNS à utiliser.

Ce fichier va permettre d'utiliser plusieurs adresses de serveur, ainsi, si l'un est défaillant, il pourra passer au suivant...

Editer le fichier résolution

```
nano /etc/resolv.conf
```

Modifier et ajouter les lignes nécessaires dans le fichier :

```
# La recherche commence sur le serveur local

search loiseljp.fr

nameserver 192.168.1.98

nameserver 127.0.0.1


# Si le reseau a un autre serveur de DNS

nameserver 192.168.1.1


# Les dns sont a adapter en fonction du FAI et des disponibilites

nameserver 212.27.40.240

nameserver 212.27.40.241

nameserver 8.8.8.8
```



### Configuration des fichiers /etc/bind

Si les modifications précédentes non pas demandé de précautions particulières, il pourrait être intéressant de faire une sauvegarde avant modification des fichiers de configuration de Bind :

Sauvegarder les fichiers :

```
cd /etc/bind  
mkdir save  
cp -rp *.* save/
```

### Configurer le fichier /etc/bind/named.conf

Cette fois il va être question de modifier le serveur de DNS il pourra donc être préférable de l'arrêter avant (plus par principe que pour une autre raison).

Comme dans de nombreux cas, lorsque la configuration d'un système devient trop lourde, celle-ci est séparée en différents fichiers qui pourront être traités individuellement.

Par arrêter le serveur taper cette commande :

```
/etc/init.d/bind9 stop
```

Ou par cette commande :

```
service bind9 stop
```



Le fichier `/etc/bind/named.conf` est donc le fichier principal de Bind, à l'exécution, ce fichier vient intégrer d'autres fichiers tels que le fichier d'option (« `/etc/bind/named.conf.options` ») ou le fichier de zone local (« `/etc/bind/named.conf.local` ») qui seront modifiés après dans ce chapitre.

Configurer le fichier `/etc/bind/named.conf`...

Ouvrir le fichier :

```
nano /etc/bind/named.conf
```

Modifier le fichier de manière à obtenir ceci :

```
// << Fichier /etc/bind/named.conf by LJP>>

// #####

include "/etc/bind/named.conf.options";


zone "." {
    type hint;
    file "/etc/bind/db.root";
};

zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
```

```

};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

include "/etc/bind/named.conf.local";

// Ajout des liens vers le fichier de configuration du domaine
// A adapter suivant la configuration que vous lui avez installée
// #####

zone "loiseljp.fr" {
    type master;
    file "/etc/bind/zones/loiseljp.hosts";
};

// Ajout des liens vers le fichier de configuration du reverse pour le LAN
// A adapter suivant votre réseau local
// #####

zone "0.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/rev.192.168.0.in-addr.arpa";
};

```

**Attention** : vous devrez adapter les fichiers à vos besoins, un fichier « /etc/bind/zones/loiseljp.hosts » va être créé, vous devez le nommer suivant votre choix et les modifications apportées au fichier « named.conf »

 On remarquera que le fichier utilise « // » au lieu de « # » habituel pour insérer un commentaire.

## Modifier /etc/bind/named.conf.options

Le fichier « options » qui vient s'intégrer au fichier /etc/bind/named.conf, est destiné à venir intégrer à la configuration les options de base.

Ouvrir le fichier pour la modification (un peu plus gros que les autres on va le vider avant !):

```
echo > /etc/bind/named.conf.options
```

```
nano /etc/bind/named.conf.options
```

Modifier (et adapter au besoin) le fichier:

```
options {
    directory "/var/cache/bind";

    query-source address * port 53;

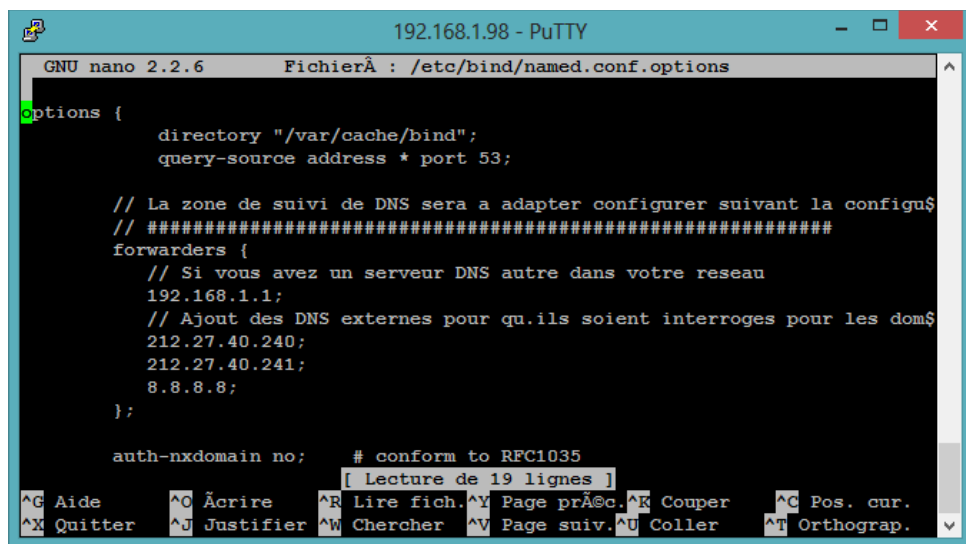
    // La zone de suivi de DNS sera a adapter configurer suivant la configuration
    // #####

    forwarders {
        // Si vous avez un serveur DNS autre dans votre reseau
        192.168.1.1;

        // Ajout des DNS externes pour qu'ils soient interrogés pour les domaines non internes
        212.27.40.240;
        212.27.40.241;
        8.8.8.8;
    };

    auth-nxdomain no; # conform to RFC1035

    listen-on-v6 { any; };
};
```



```
GNU nano 2.2.6 FichierÃ : /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";
    query-source address * port 53;

    // La zone de suivi de DNS sera a adapter configurer suivant la configu$
    // #####
    forwarders {
        // Si vous avez un serveur DNS autre dans votre reseau
        192.168.1.1;
        // Ajout des DNS externes pour qu'ils soient interrogés pour les dom$
        212.27.40.240;
        212.27.40.241;
        8.8.8.8;
    };

    auth-nxdomain no;          # conform to RFC1035
                                [ Lecture de 19 lignes ]
^G Aide      ^O Ãcrire    ^R Lire fich.^Y Page prÃc.^K Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher   ^V Page suiv.^U Coller     ^T Orthograp.
```

## Création du fichier de zone

Pour le serveur Web, un fichier de zone va devoir être créé afin de gérer le serveur lui-même.

Afin de rester cohérent dans l'organisation du serveur, ce fichier de « zones » sera placé dans un nouveau dossier (dossier qui recevra également d'autres fichiers):

```
mkdir /etc/bind/zones

# Adapter suivant le besoin (voir fichier /etc/bind/named.conf...)

nano /etc/bind/zones/loiseljp.hosts
```

**Attention** : le nom du fichier correspond à celui placé dans le fichier « named.conf » !

Modifier et adapter le fichier suivant les besoins :

```
$ttl 86400

@ IN SOA loiseljp.fr. loiseljp. (
    2008061801
    21600
    3600
    604800
    86400 )

;ENREGISTREMENT "A" DNS <-> IP CLASSIQUES
```

```
@      IN    NS      loiseljp.fr.
      IN    A      192.168.1.98
sql     IN    A      192.168.1.98
mysql   IN    A      192.168.1.98
```

### Création du fichier reverse DNS

Le rôle d'un serveur de DNS n'est pas uniquement de traduire une IP en nom de domaine mais de faire l'inverse également, un fichier va donc être créé afin d'y apporter toutes les caractéristiques du serveur. Éditer le nouveau fichier:

```
nano /etc/bind/zones/rev.192.168.0.in-addr.arpa
```

Modifier le fichier pour obtenir ceci (adapter le nom du serveur au besoin) :

```
@ IN SOA loiseljp. loiseljp.fr. (
    2006081401;
    28800;
    604800;
    604800;
    86400);

    IN    NS      loiseljp.fr.
12      IN    PTR  loiseljp.fr.
```

**Note** : les « . » et espaces sont à respecter dans ce fichier...

 Le nom du fichier et noms de serveur seront adaptés en fonction du contenu du fichier « named.conf ».

### Redémarrer le service Bind

Si le serveur était arrêté taper cette commande :

```
/etc/init.d/bind9 start
```

ou

```
service bind9 start
```

Sinon, pour redémarrer le service (s'il est en cours) :

```
/etc/init.d/bind9 restart
```

ou

```
service bind9 restart
```

Si le serveur ne redémarre pas, il sera nécessaire de revoir les fichiers modifiés (« ; » en fin de ligne, espaces mal placés, noms de fichiers, nom du serveur..) :

## Vérifier le serveur

📖 Pour une utilisation directe le serveur pourra répondre à son nom de domaine ou ses alias en quelques minutes (rarement plus de 5 minutes), dans le cadre de la diffusion des DNS sur les serveurs Internet il faut compter 24 à 48h...

Maintenant que le serveur DNS est démarré et configuré, il peut être intéressant de tester son bon fonctionnement. La commande « dig » se chargera de cette fonction.

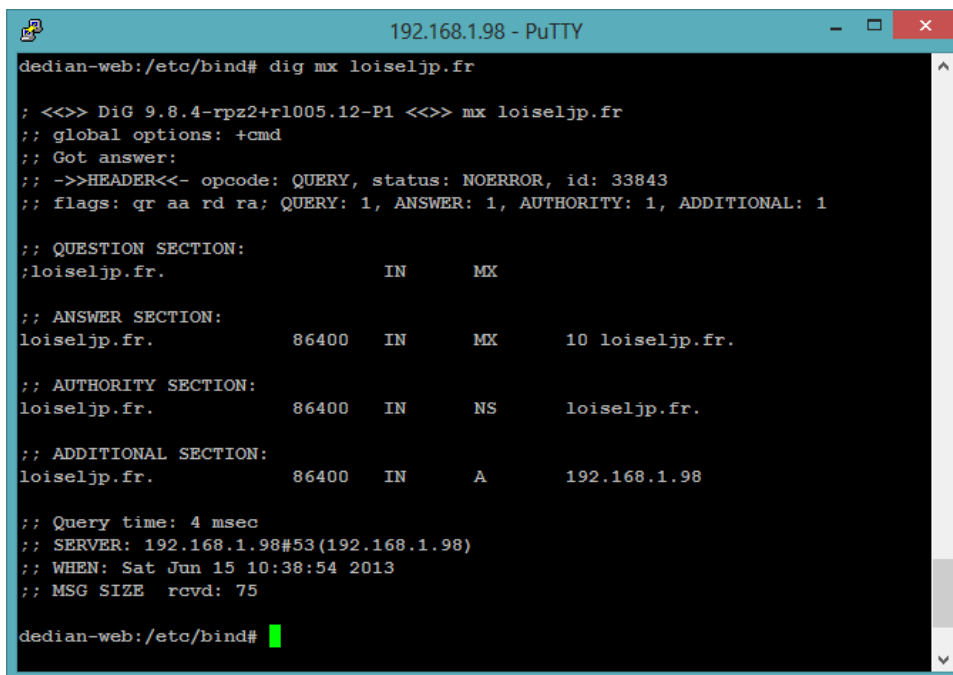
Cette commande se trouve dans le paquetage « dnsutils ». Pour l'installer (si ce n'est pas fait) taper la commande :

```
apt-get install dnsutils
```

On peut ensuite exécuter la commande (à adapter suivant le nom réseau) et suivants les besoins:

```
dig mx loiseljp.fr
```

Correctement configuré la commande « dig » devrait retourner sensiblement ceci :



```
192.168.1.98 - PuTTY
dedian-web:/etc/bind# dig mx loiseljp.fr

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> mx loiseljp.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33843
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;loiseljp.fr.                IN      MX

;; ANSWER SECTION:
loiseljp.fr.                86400   IN      MX      10 loiseljp.fr.

;; AUTHORITY SECTION:
loiseljp.fr.                86400   IN      NS      loiseljp.fr.

;; ADDITIONAL SECTION:
loiseljp.fr.                86400   IN      A      192.168.1.98

;; Query time: 4 msec
;; SERVER: 192.168.1.98#53(192.168.1.98)
;; WHEN: Sat Jun 15 10:38:54 2013
;; MSG SIZE rcvd: 75

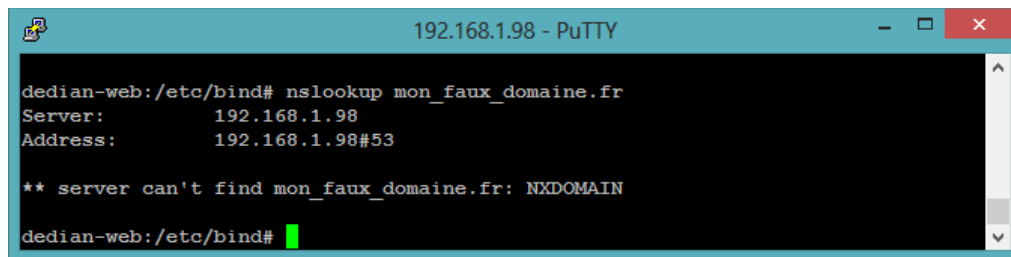
dedian-web:/etc/bind#
```

La commande suivante retourne suivant la demande l'adresse Ip ou le nom de domaine :

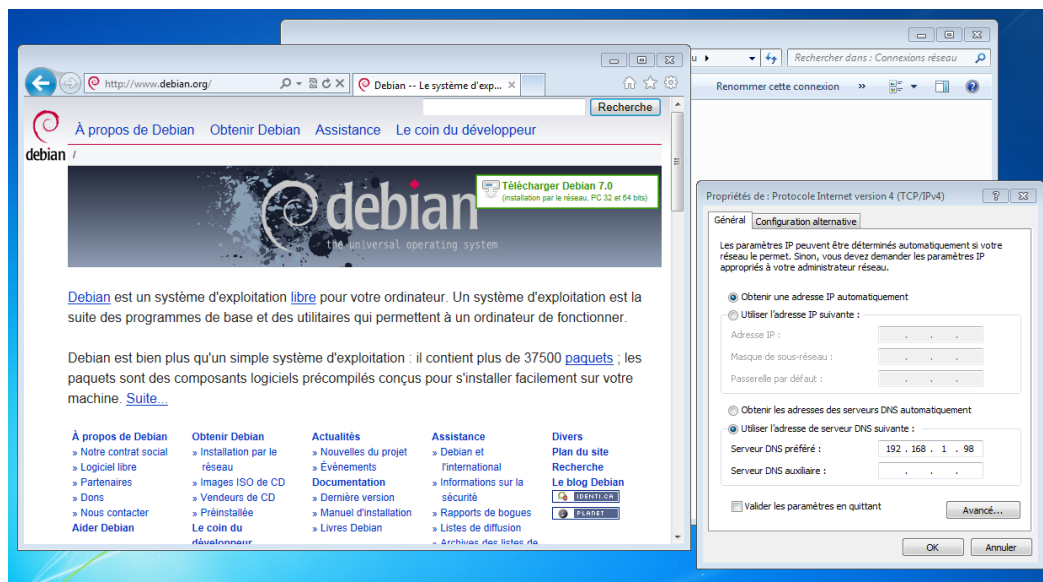
```
nslookup loiseljp.fr
```



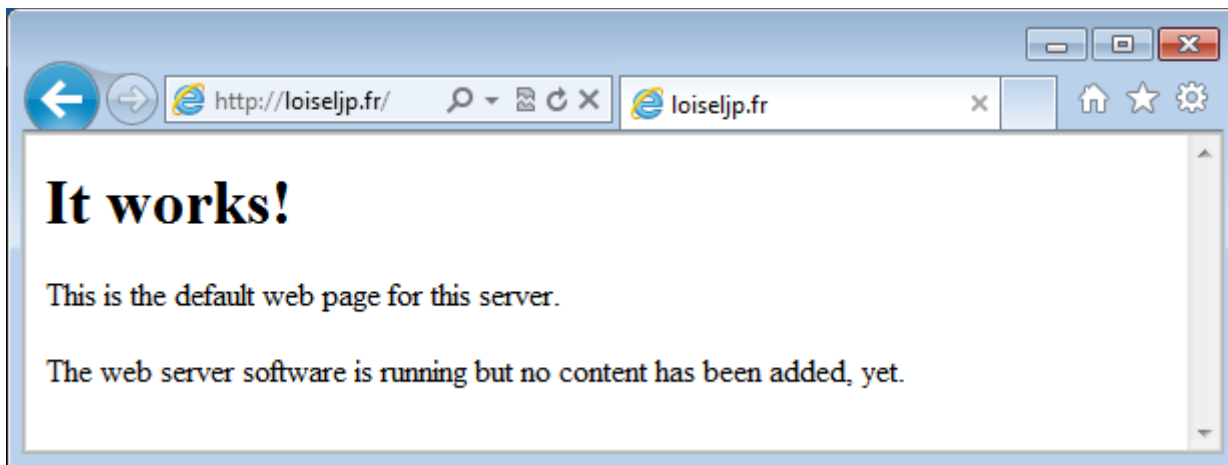
En cas de mauvaise configuration, on obtiendra une erreur :



La solution de tester depuis un client réseau (configuré pour utiliser le serveur « dns » installé) est aussi efficace :

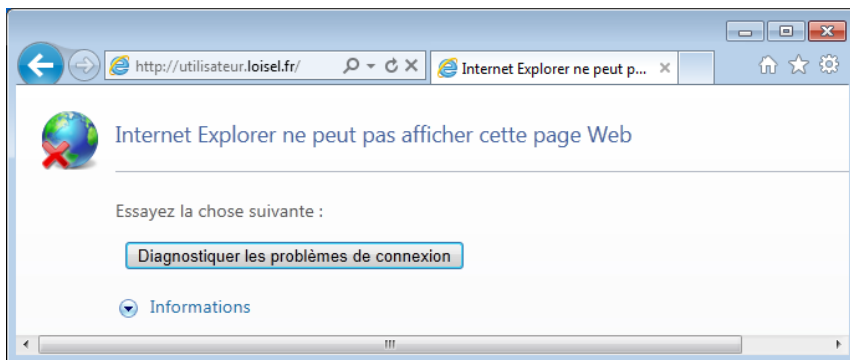


Après avoir testé une adresse DNS Internet, on pourra tester notre serveur :



## Ajouter des sous domaines

Maintenant que le serveur web répond à son nom dns, il serait intéressant qu'il dirige les sous domaines au bon endroit (par exemple <http://utilisateur.loisel.fr> vers le dossier où est stocké le site web de notre client).



(On s'en doutait, la redirection ne fonctionne pas encore actuellement)

L'ajout d'un sous domaine s'effectue en Deux temps : enregistrer le nouveau nom DNS, puis configurer Apache pour qu'il renvoie la page correspondante au domaine.

Pour ajouter un nouveau nom de domaine (sous-domaine), il suffira d'éditer le fichier de zone puis d'y ajouter une nouvelle redirection :

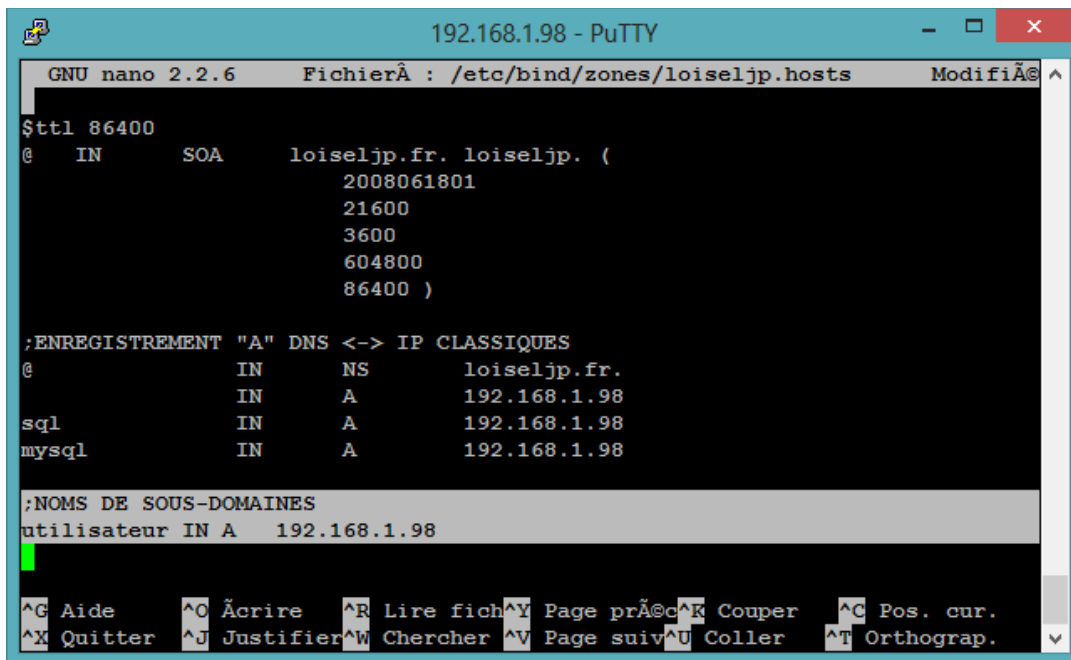
```
nano /etc/bind/zones/loiseljp.hosts
```

Ajouter les lignes suivantes (la première étant un commentaire pour indiquer les lignes qui suivront)

```
;NOMS DE SOUS-DOMAINES
```

```
utilisateur IN A 192.168.1.98
```





```
GNU nano 2.2.6 Fichier : /etc/bind/zones/loiseljp.hosts Modifié : 2013-06-10 14:00:00
$ttl 86400
@ IN SOA loiseljp.fr. loiseljp. (
    2008061801
    21600
    3600
    604800
    86400 )

;ENREGISTREMENT "A" DNS <-> IP CLASSIQUES
@ IN NS loiseljp.fr.
@ IN A 192.168.1.98
sql IN A 192.168.1.98
mysql IN A 192.168.1.98

;NOMS DE SOUS-DOMAINES
utilisateur IN A 192.168.1.98
```

Dans ce tutoriel il ne sera pas fait mention d'ajouter qu'un alias (« .utilisateur.loiseljp.fr CNAME loiseljp.fr »). Bien que la commande fonctionne de manière identique, elle demande plus de requête auprès du serveur (convertir l'alias puis demander l'adresse). Cette redirection sur l'adresse IP est donc plus efficace.

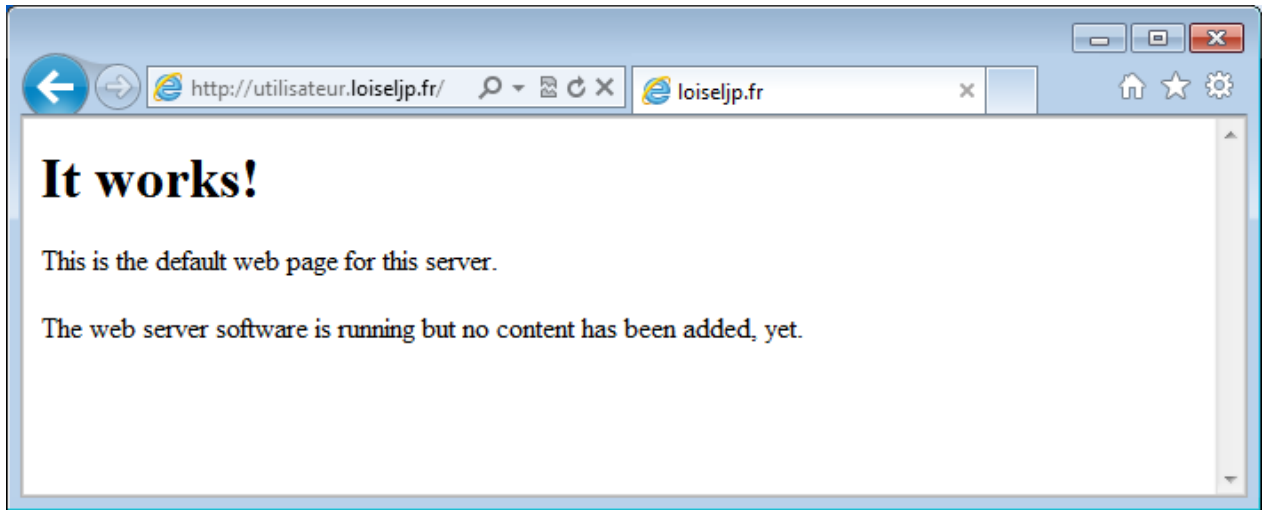
Après modification, redémarrer le service (s'il est en cours) :

/etc/init.d/bind9 restart

ou

service bind9 restart

On peut de nouveau faire les essais :



La dernière étape consiste à demander au serveur Apache de diriger les pages correspondantes au compte « utilisateur » (pour l'instant c'est celui qui nous intéresse).

Editer le fichier de gestion des sites par défaut :

```
nano /etc/apache2/sites-available/default
```

Indiquer à apache la nouvelle redirection :

```
<VirtualHost *:80>

DocumentRoot /home/www/utilisateur

ServerName utilisateur.loiseljp.fr

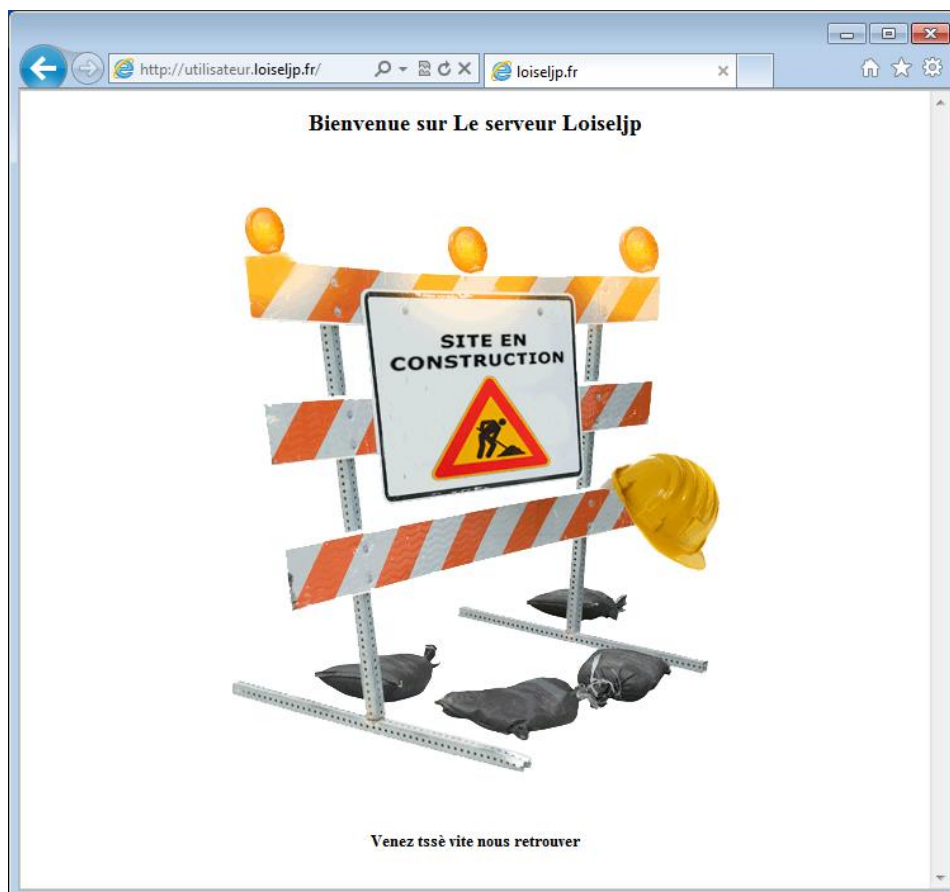
</VirtualHost>
```

A adapter suivant les besoins.

Redémarrer Apache :

```
service apache2 restart
```

Essayer de se connecter sur le domaine de l'utilisateur :



**ATTENTION :** sur l'internet on trouve beaucoup d'informations à ce sujet, on trouve de tout et son contraire pour dire de résoudre ce sujet simple en apparence.

## 13 Gérer les comptes

Dans les chapitres précédents il a été expliqué comment créer un compte, comment configurer le compte FTP, comment lui placer les quotas...

A cette étape il va alors être question de combiner l'ensemble des commandes. En plus de la création de l'utilisateur :

- 1) Nous avons vu dans la configuration du serveur vsftp que l'utilisateur à son compte dans « /home/%user% » par défaut.
- 2) Dans le dossier « utilisateur » il faut lui créer un dossier dans lequel il pourra écrire : « /home/%user% /web».
- 3) Il va alors être nécessaire de lier ce compte (le dossier créé précédemment « www ») avec le dossier diffusé sur le web.  
Nous allons donc lier « /home/%user%/www » avec « /home/www/%user% ».  
L'utilisateur aura désormais son site web en « http://ip\_du\_serveur/%user% » (transformer « http://ip\_du\_serveur/%user% » en « http://nomdedomaine\_user » ne sera plus qu'une question de redirection DNS).
- 4) Les ACL devront être mises en place.
- 5) Chaque utilisateur aura une base de donnée de créée à son nom avec un mot de passe (identique au compte ftp).
- 6) L'utilisateur aura son nom de sous domaine, il sera alors nécessaire d'ajouter le domaine au fichier de zone, ajouter le domaine au fichier de configuration des sites Apache et redémarrer les services après configuration.
- 7) Une page web de type « En construction » devra être placée dans le dossier par défaut.

*Ce tutoriel a pour but de présenter la création d'un serveur Web, mais n'a pas la même prétention sur la création de scripts (php ou sh). Ici il est proposé une solution intermédiaire pour la création du compte utilisateur.*

*Pour aller plus loin dans ce domaine, le lecteur pourra se reporter vers des sites spécialisés sur PHP, MySQL...*

### Création d'un script de gestion des utilisateurs

A cette étape il va s'agit de créer un utilisateur depuis un script en passant simplement quelques paramètres. Le script sera directement intégré dans « /usr/bin » ce qui permettra un emploi direct du fichier. Le script sera utilisé en administrateur !

Créer le fichier « user\_web »:

```
touch /usr/bin/user_web
```

Rendre le fichier exécutable avec commande :

```
chmod +x /usr/bin/user_web
```

Pour placer le mot de passe automatiquement, il doit être crypté. Pour cela on utilisera le module mkpasswd, qu'il faudra installer :

```
apt-get install libstring-mkpasswd-perl
```

Ouvrir et éditer le fichier «user\_web» :

```
nano /usr/bin/user_web
```

Pour obtenir le script effectuant la commande précédemment vue, ajouter au fichier le code suivant :

```
#!/bin/sh

# Script de creation d un utilisateur web

# #####

# Retourne l'usage du script au besoin
usage()
{
    echo ""
    echo "#### Creation utilisateur Web par LoiselJP ####"
    echo "#####"
    echo "Utilisation du script: "
    echo "$0 nom_utilisateur mot_de_passe"
    echo "Utilisateur en caracteres alphanumeriques"
    echo "#####"
    echo ""
}

existe()
{
    echo ""
    echo "#### Creation utilisateur Web par LoiselJP ####"
    echo "#####"
    echo "Le dossier utilisateur existe"
    echo "#####"
    echo ""
}
```

```

}

format()
{
    echo ""

    echo "#### Creation utilisateur Web par LoiselJP ####"

    echo "#####"

    echo "Le format du compte n est pas bon !"

    echo "#####"

    echo ""
}

# Teste l entree passee en parametre => si pas d entree retourne l usage
if [ -z $2 ]; then
    usage
    exit 0
fi

# Teste le dossier (donc l utilisateur)
user=$(echo "$1" | tr A-Z a-z)

if [ -d "/home/$user" ];then
    existe
    exit 0
fi

# supprime la plupart des caracteres speciaux a ne pas mettre dans le nom
if [ ! `expr index "$user" [àâêéèëïîôöûù`<>%$`!`*`+`,`-`,`/`,`?`,`@`,`_`,`\`,`.=`,`~`,`\`,`{`,`}\`,`[`,`]`,`\`,`\`,`)` = "0" ]; then
    format
    exit 0
fi

```

```

# On code $2 pour l'envoyer a useradd

var=$(perl -e 'print crypt("azerty", "salt"), "\n"')

# Cree l'utilisateur $user avec comme mot de passe $2 cree son dossier personnel

useradd "$user" -m -p "$var"

# Cree le dossier web avec un lien vers le dossier www

mkdir /home/$user/www

ln -s /home/$user/www /var/www/$user

# Copier la page -En construction- => on utilise la page de -utilisateur- qui servira d'exemple pour les clients

cp -r /var/www/utilisateur/* /var/www/$user/

# On place les droits au besoin sur les fichiers

chown -R $user:$user /home/$user

chmod 755 -R /home/$user

chown -R $user:$user /var/www/$user

chmod 755 -R /var/www/$user

# Mettre les acl sur le dossier

setfacl -m d:u:$user:rwX /home/$user

# on fixe les quotas de $user avec ceux par default de utilisateur

edquota -p utilisateur $user

# mettre les droits d'écriture correctement sur les dossiers (sinon vsftpd n'est pas content)

chmod 555 /home/$user

chmod 755 -R /home/$user/www

# On ajoute la zone dans le fichier de DNS /etc/bind/zones/loiseljp.hosts (à adapter au serveur)

echo "$user IN A 192.168.1.98" >> /etc/bind/zones/loiseljp.hosts

```

```
# On ajoute la zone dans le fichier apache /etc/apache2/sites-available/default (a adapter au serveur)
```

```
echo "" >> /etc/apache2/sites-available/default
```

```
echo "<VirtualHost *:80>" >> /etc/apache2/sites-available/default
```

```
echo "DocumentRoot /home/www/$user" >> /etc/apache2/sites-available/default
```

```
echo "ServerName $user.loiseljp.fr" >> /etc/apache2/sites-available/default
```

```
echo "</VirtualHost>" >> /etc/apache2/sites-available/default
```

```
# Redemarrer les services
```

```
service bind9 restart
```

```
service apache2 restart
```

Le script présenté ici suffira à créer un utilisateur, après avoir testé son existence.

La création de la base de données associée à l'utilisateur, ainsi que la création automatisée de l'utilisateur pourra se faire depuis un formulaire de demande sur le site principal en php.

## Créer le formulaire d'enregistrement automatique du nouveau « client »

Ce tutoriel ne développera pas la partie web du site et notamment la partie création d'un compte depuis une page PHP. Ce type de création se rapprochant fortement de la programmation Web, développer un tel formulaire à ce stade de la création du serveur demanderait de nombreuses pages supplémentaires à ce document.

Seules quelques explications seront données ici.

*Auparavant, « \$variable\_nom\_utilisateur » aura été débarrassée des caractères non conformes à la création de l'utilisateur ( \_ - \* et autres caractères spéciaux)*

Le formulaire d'inscription pour un nouvel utilisateur se décomposera en 3 pages :

- La page d'inscription,
- La page de création du compte,
- La page de résultats.

### *La page d'inscription*

La page d'inscription pourra comporter au minimum 2 champs : le nom de l'utilisateur et son mot de passe.

Mais celle-ci pourra (et c'est généralement le cas) demander d'autres renseignements tels que adresse mail, nom du propriétaire... L'adresse mail pourra servir à la récupération du mot de passe.

### ***La page de création***

Cette page sera composée de plusieurs éléments :

- Récupération des données passées au formulaire
- Test de la disponibilité de l'utilisateur (soit par un accès sur une table MSQL, ou depuis un test d'existence de la base de données ou dossier utilisateur,
- Création de la base de données qui sera composée de 4 requêtes (création de l'utilisateur, création des droits généraux pour l'utilisateur, mise en place de la base de données, mise en place des droits utilisateur sur cette base).

L'exemple ci-dessous donne le contenu des requêtes (traité lors de la mise en place de MySQL).

```
CREATE USER '$variable_nom_utilisateur'@'localhost' IDENTIFIED BY '$variable_password_utilisateur';

GRANT USAGE ON *.* TO '$variable_nom_utilisateur'@'localhost' IDENTIFIED BY '$variable_password_utilisateur' WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;

CREATE DATABASE IF NOT EXISTS ` $variable_nom_utilisateur ` ;

GRANT ALL PRIVILEGES ON ` $variable_nom_utilisateur `.* TO '$variable_nom_utilisateur'@'localhost';
```

- Création du compte utilisateur sur le serveur pour l'utilisation du ftp et de l'espace Web qui lui est accordé.

Dans le formulaire PHP de création de l'utilisateur, la commande « exec() » se chargera de lancer l'exécution de la création de cet utilisateur en exécutant le script précédemment créé.

```
<?php exec(' user_web $variable_nom_utilisateur $variable_password_utilisateur' 2>&1); ?>
```

### ***La page de résultat***

Cette page indiquera à l'utilisateur si la création de son compte s'est bien déroulée.

## **Créer un utilisateur avec sa base de données**

Comme il vient de l'être montré au paragraphe précédent la création des pages Web pour l'inscription des clients est une autre aventure qui sera plus dans les compétences d'un programmeur Web plutôt que celui d'un administrateur.

Cependant, il va être nécessaire à l'administrateur du serveur de créer quelques utilisateurs pour y faire des essais. Ce que ce paragraphe va proposer.

## **Choisir un nouvel utilisateur et son mot de passe**



Le rôle du formulaire d'inscription est de mettre en place un masque d'inscription qui se chargera de mettre en forme le compte du client.

Pour éviter toute confusion, tout doublon, prendre un nom d'utilisateur en **minuscules**,

Le nom d'utilisateur de **ne devra pas comporter l'espace**,

Il ne devra pas non plus comporter de caractères spéciaux tels que « / \ ! ? »... par principe **il ne faut autoriser que des lettres ou des chiffres (mais doit commencer par une lettre)**

**Refuser les points** ce qui pourrait porter à confusion,

Limiter le nom de domaine et sous-domaine à **128 caractères maximum**, les chemins de sous dossiers aidant, il pourraient survenir des erreurs de nom de fichiers trop long...

Les accents ne faisant pas très bon ménage, **les accents seront retirés des lettres disponibles**.

Des lors, choisissez le nom de sous-domaine à créer :

- monsiteweb

- mesplusbellesidees

- monpetitmonde

Même s'il s'agit d'un essai, le mot de passe n'est pas à négliger, choisissez alors un mot de passe pour le compte de l'utilisateur à créer.

**Refuser les mots de passe vide !**

## Créer l'utilisateur

Après avoir fait le choix de l'utilisateur et de son mot de passe, la création d'un nouvel utilisateur se fera en 2 temps :

- Création de l'utilisateur Linux
- Création de l'utilisateur Mysql

Pour l'exemple un nouvel utilisateur va être créé sous le login : **demohebergement** et aura pour mot de passe **mon-super-password-2013**

Ouvrir une console puis créer l'utilisateur en utilisant le script crée au paragraphe précédent :

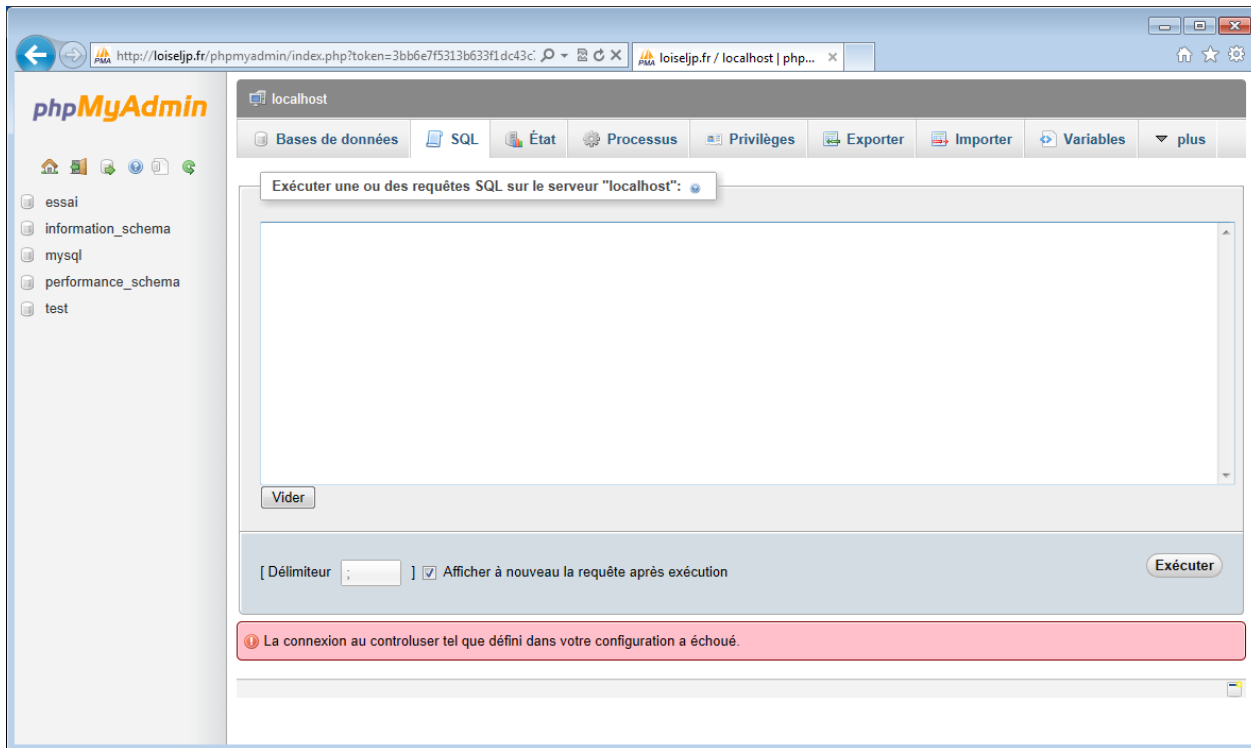
```
user_web demohebergement mon-super-password-2013
```

Si le script a été écrit comme il se doit, aucune erreur ne devrait être retournée (éventuellement, si le serveur n'a pas été redémarré depuis l'installation des DNS une erreur de jonction à l'ancien nom peut survenir)



```
loiseljp:~# user_web demohebergement mon-super-password-2013
[ ok ] Stopping domain name service...: bind9.
[ ok ] Starting domain name service...: bind9.
[ ok ] Restarting web server: apache2 ... waiting .
loiseljp:~#
```

Se connecter sur phpmyadmin puis ouvrir la console « sql »



Entrer les requêtes de création :

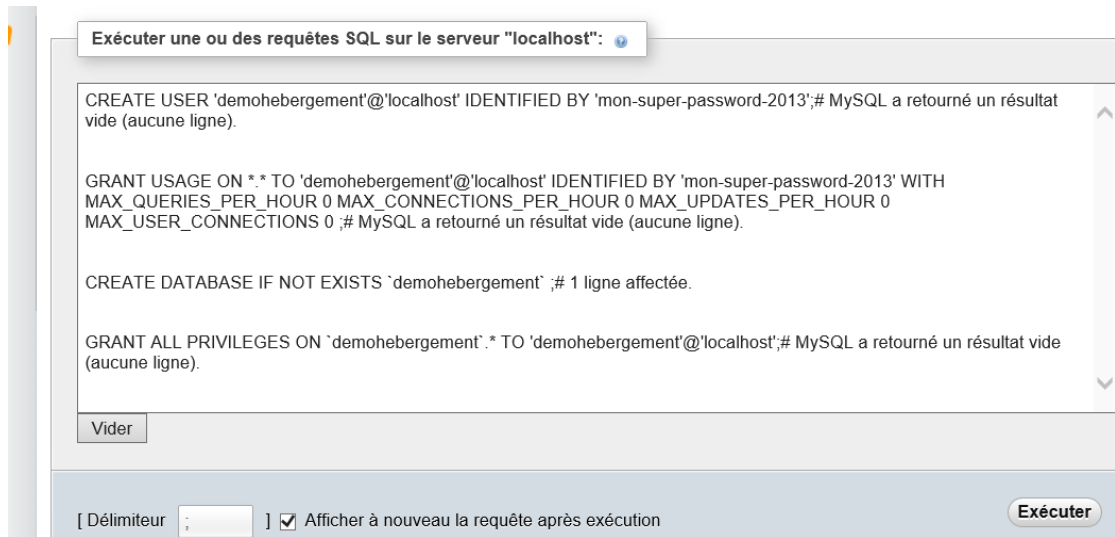
```
CREATE USER 'demohebergement'@'localhost' IDENTIFIED BY 'mon-super-password-2013';

GRANT USAGE ON *.* TO 'demohebergement'@'localhost' IDENTIFIED BY 'mon-super-password-2013'
WITH          MAX_QUERIES_PER_HOUR          0          MAX_CONNECTIONS_PER_HOUR          0
MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;

CREATE DATABASE IF NOT EXISTS `demohebergement` ;

GRANT ALL PRIVILEGES ON `demohebergement`.* TO 'demohebergement'@'localhost';
```

Cliquer ensuite sur « exécuter »...

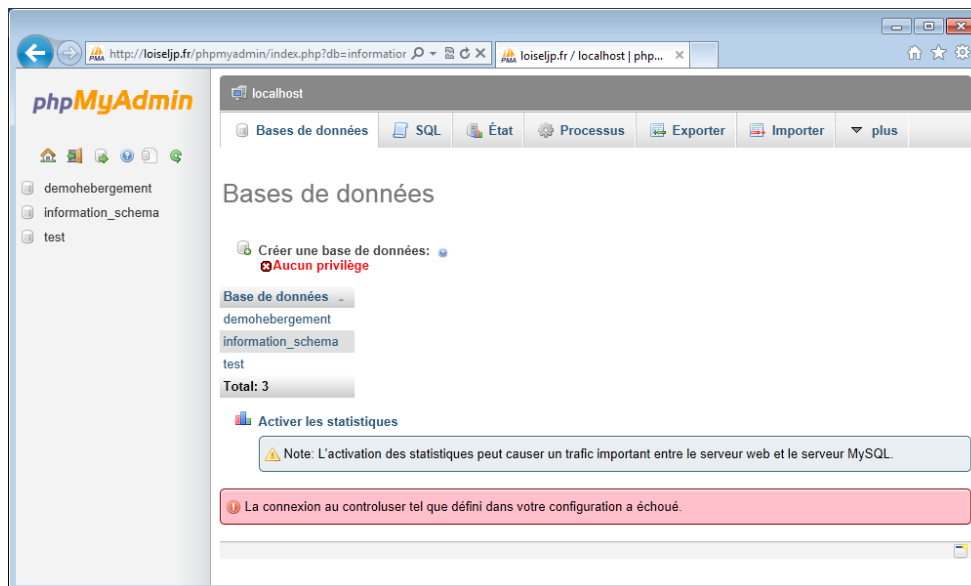


Aucune erreur ne devrait être retournée.

Le compte du client est opérationnel...

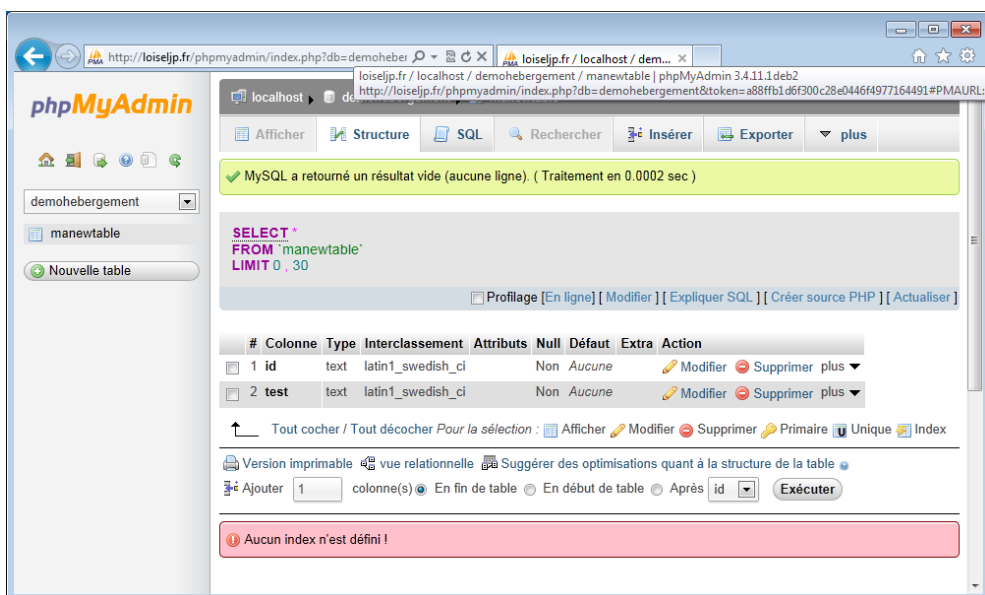
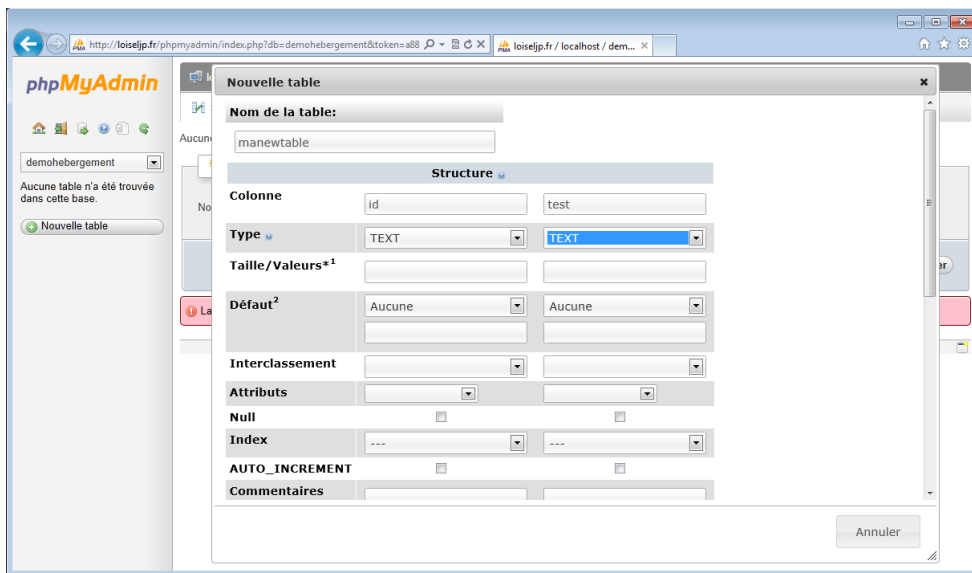
### Vérifier le compte

Depuis le serveur se déconnecter de PhpMyadmin, puis se reconnecter depuis le compte du client...



De manière logique il ne peut pas créer de bases de données supplémentaires,...

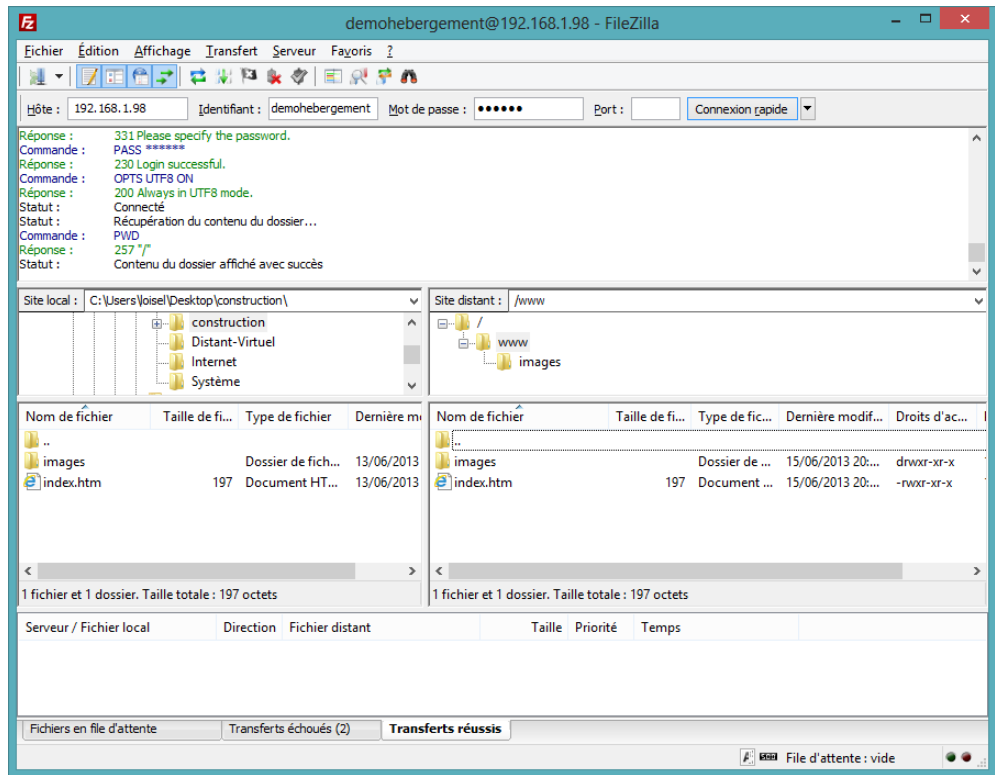
Mais rien ne l'empêche de créer une nouvelle table...



Se rendre ensuite sur le serveur et vérifier la disponibilité depuis le nom DNS, on devrait y voir apparaître la page « en construction » copiée depuis le compte utilisateur (référence)



*Restera le ftp à contrôler...*



Dès lors que le l'utilisateur a été créé automatiquement, si celui-ci ne présente pas de problème, les autres n'en poseront pas plus!

## Supprimer un utilisateur

La suppression de l'utilisateur se fera dans le sens inverse :

On supprime l'utilisateur unix...

```
userdel demohebergement
```

Sur la très grande majorité des serveurs, la suppression s'arrête là... tant que la place occupée par le site n'est pas nécessaire... (c'est une question de stratégie et de visibilité sur le web)

Mais on pourra complètement le supprimer en utilisant ces commandes :

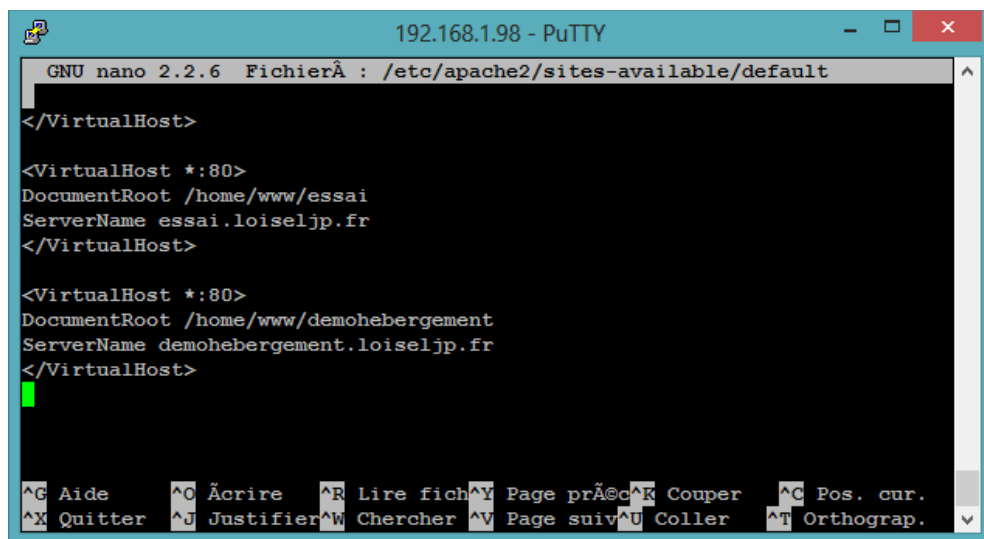
```
rm -rf /home/demohebergement
```

```
rm -rf /home/www/demohebergement
```

Mais il sera alors nécessaire d'éditer les fichiers de DNS :

```
nano /etc/apache2/sites-available/default
```

Puis supprimer les lignes ajoutées...



```
GNU nano 2.2.6 Fichier : /etc/apache2/sites-available/default
</VirtualHost>

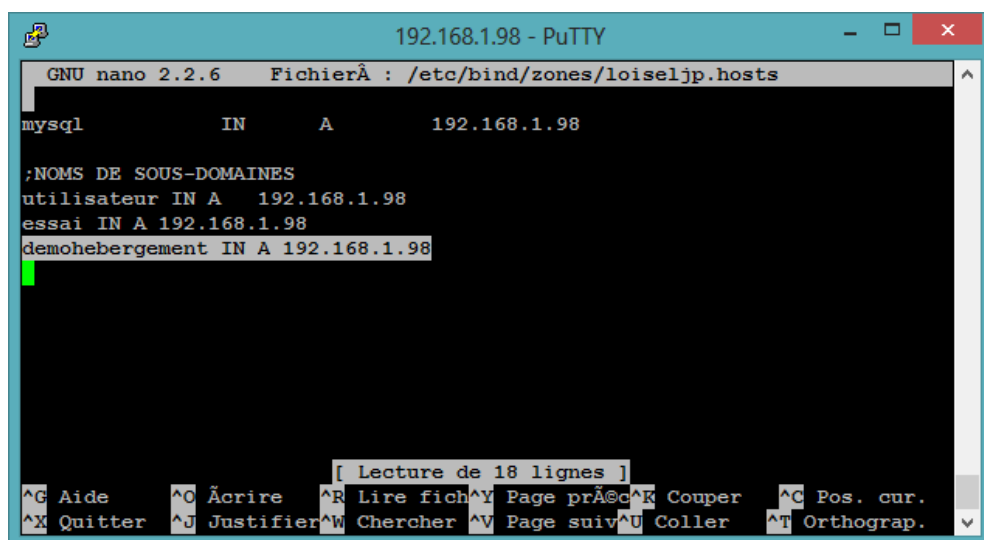
<VirtualHost *:80>
DocumentRoot /home/www/essai
ServerName essai.loiseljp.fr
</VirtualHost>

<VirtualHost *:80>
DocumentRoot /home/www/demohebergement
ServerName demohebergement.loiseljp.fr
</VirtualHost>

```

Idem pour le fichier de zone :

```
nano /etc/bind/zones/loiseljp.hosts
```



```
GNU nano 2.2.6 Fichier : /etc/bind/zones/loiseljp.hosts
mysql      IN      A      192.168.1.98

;NOMS DE SOUS-DOMAINES
utilisateur IN A      192.168.1.98
essai      IN A      192.168.1.98
demohebergement IN A 192.168.1.98

```

Après modification, il sera nécessaire de redémarrer les services :

```
service bind9 restart
```

```
service apache2 restart
```

## 14 Petite page web

Ce chapitre va vous proposer une petite page web, qui n'a pas pour but de présenter ou créer un site Web vous donne quelques explications sur la marche à suivre.

Elle vous permettra de tester le serveur web proposé dans ce tutoriel...

Créer une nouvelle page index.php à la racine du serveur :

```
touch /var/www/index.php
```

Modifier la page créée :

```
nano /var/www/index.php
```

Ajouter à l'entête du fichier les métas données nécessaires à la reconnaissance du site sur le web :

```
<html>

<head>

<title>Mon serveur web</title>


<!--si vous avez une icone

<link rel="icon" type="image/x-icon" HREF="http://loiseljp.fr/favicon.ico" >

<link rel="shortcut icon" type="image/x-icon" HREF="http://loiseljp.fr/favicon.ico" >

<LINK REL="SHORTCUT ICON" HREF="http://loiseljp.fr/favicon.ico">

<link href='http://loiseljp.fr/favicon.ico' rel='icon' type='image/x-icon'/>

-->

<meta name="title" content="Mon site Web">

<meta name="titre" content="Mon site Web">

<!--si vous etes inscrit sur google -->

<meta name="google-site-verification" content="kjRw5Zy_jrsNC0dqLsLumS8vXApge7oE3xVwE9f1hUs" />

<meta name="description" content="Mon hébergement blablabla.">

<meta name="abstract" content=" Mon hébergement blablabla.">


<!-- jusqu'à concurrence de 1000 caractères environ -->

<meta name="keywords" content="tuto,cours,siteweb,hénergement ">

<meta name="mots clef" content=" tuto,cours,siteweb,hénergement ">


<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<meta http-equiv="Content-Language" content="fr-FX">

<META NAME="LANGUAGE" CONTENT="FR">


<meta name="category" content="informatique">
```

```

<meta name="Classification" content=" web, pages htm, html,php,serveur">

<meta name="subject" content=" web, pages htm, html,php,serveur ">


<!--suivant le besoin-->

<meta name="rating" content=" web, pages htm, html,php,serveur ">


<meta name="author" content="LOISEL Jean Paul">

<META NAME="COPYRIGHT" CONTENT="LJProgrammation">

<meta name="conception" content="loiseljp">


<META HTTP-EQUIV="PRAGMA" CONTENT="NO-CACHE">


<meta name="distribution" content="Global">


<meta name="owner" content="loiseljp@club-internet.fr">

<meta name="reply-to" content="loiseljp@club-internet.fr">


<meta name="Date-Creation-yyyymmdd" content="20130615">

<meta name="Date-Revision-yyyymmdd" content="20130615">

<META NAME="REVISIT-AFTER" CONTENT="30 DAYS">

<META NAME="ROBOTS" CONTENT="All">

</head>

```

(Cette liste est loin d’être exhaustive)

Le fichier de métas données pourra être mis dans un fichier séparé puis inclus par une balise « include » php.

Passer ensuite au corps de la page (chacun y mettre ce qu’il y veut...)

```

<p align="center"><font size="4"><b><font size="5">Serveur Web Debian</font></b></font></p>

<p align="center">&quot;Un serveur d’hébergement sur mesure&quot;</p>

```

On pourra au travers d’une nouvelle base de données crée pour la circonstance récupérer une liste (des clients par exemple) qui aura été insérée dans une table (ici la table s’appelle clients)

```

<?php

```



```
// base de donnée locale

$mysqlhost="localhost";

// Base de donnée commune (on charge les différents paramètres)

$mysqllogin="essai";

$mysqlDB= "essai";

$mysqlpasswd="azerty";

$mysqllink = mysql_pconnect($mysqlhost,$mysqllogin,$mysqlpasswd);

// on teste la base de données

@mysql_select_db("$mysqlDB",$mysqllink) or die ("CONNEXION à MYSQL IMPOSSIBLE. ");

?>

<hr>

<p>Voici une petite liste des clients qui ont choisi notre hébergement :</p><br><br>

<?php

$sql = 'SELECT COUNT(*) FROM `clients`';

$resultat = mysql_query($sql);

while ($enr = mysql_fetch_array($resultat))

{

echo '<b>'. $enr['client']. '</b><br> ';

echo $enr['description']. '<br><br>';

echo "<small><i>Cliquer <a href=http://". $enr['client']. ".loiseljp.fr>ici</a>

pour visiter le site. </i></small><br><br>" ;

}

?>

<hr>
```

***On notera « ' » pour délimiter les champs et « ` » pour délimiter les tables...***

Accéder au dossier « download » créé à la racine serveur web :

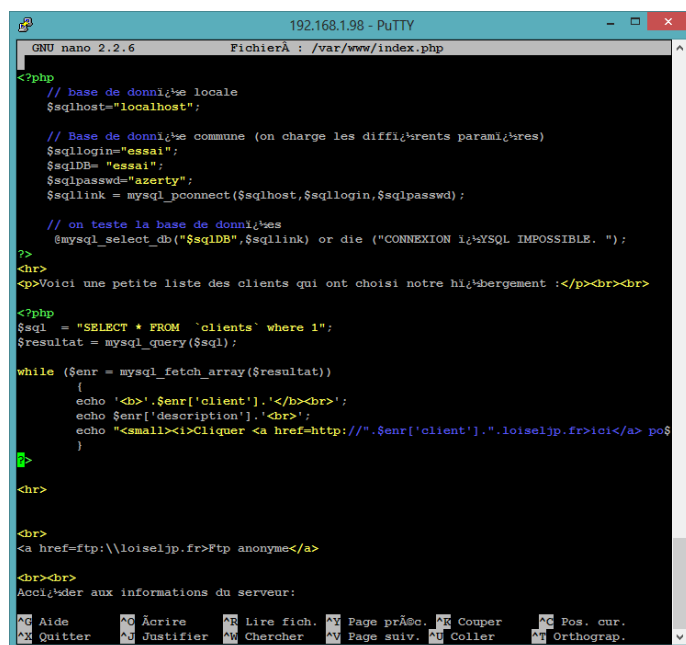
```
<br>

<a href =ftp:\\loiseljp.fr>Ftp anonyme</a>
```

Quelques informations...

<br><br>Accéder aux informations du serveur : <a href =phpinfo.php >Cliquez ici</a>

En général le phpinfo() ne se donne pas, les personnes mal intentionnées y chercheront les erreurs de configuration, pas forcément celles qui mettront en péril le serveur, mais celles qui permettront des actions indésirables (envoi de mail, redirection, proxy, ...).



```
GNU nano 2.2.6 FichierA : /var/www/index.php
<?php
// base de données locale
$sqlhost="localhost";

// Base de données commune (on charge les différents paramètres)
$sqllogin="essai";
$sqlDB= "essai";
$sqlpasswd="azerty";
$sqllink = mysql_pconnect($sqlhost,$sqllogin,$sqlpasswd);

// on teste la base de données
if(mysql_select_db($sqlDB,$sqllink) or die ("CONNEXION à MySQL IMPOSSIBLE. "));
?>
<hr>
<p>Voici une petite liste des clients qui ont choisi notre hébergement :</p><br><br>
<?php
$sql = "SELECT * FROM `clients` where 1";
$resultat = mysql_query($sql);

while ($enr = mysql_fetch_array($resultat))
{
    echo '<b>'. $enr['client']. '</b><br>';
    echo $enr['description']. '<br>';
    echo "<small><i>Cliquez <a href=http://". $enr['client']. ".loiseljp.fr>ici</a> pour"
}

<br>

<br>
<a href=ftp:\\loiseljp.fr>Ftp anonyme</a>

<br><br>
Accéder aux informations du serveur:
```

Après saisie du code et ajustements (il arrive souvent d'oublier les «.», « ; » en fin de ligne, et il arrive au contraire souvent de mettre des espaces où il n'en faut pas... Ce n'est plus dans le domaine de l'administration de serveur.

Il sera possible d'accéder à une page web sensiblement identique à celle-ci.



Avec un accès au serveur FTP (dossier « download » définit en cours d'installation)



## 15 Conclusion

Dans la globalité, la création d'un serveur Web ne demande aucune connaissance particulière, mais chaque point ajouté l'un derrière l'autre rend le sujet assez complexe car il va être question de faire intervenir un grand nombre de compétences (Réseau, administration Serveur, Sécurité, Programmation, Design, Base De Données...) ce qui fait que l'on pourra très vite s'y perdre. Il est alors nécessaire de faire preuve de méthode et de rigueur.

Ce tutoriel qui se veut suffisant pour créer un serveur Web, n'en est pas pour autant complet et exhaustif dans les moyens disponibles, il faudrait y ajouter un système de sauvegarde, un suivi des logs, une rotation de ces logs, un système de statistiques, une « fabrique » avec une bibliothèque de données,...

---

---

# # init 0



# ©Propriété

## Installation d'un serveur WEB sous Linux.

Jean Paul Loisel

56 Rue Philippe de Girard

59160 Lomme

✉ [loiseljp@club-internet.fr](mailto:loiseljp@club-internet.fr)

☎ 07 70 32 35 67

## Licence

Ce document est distribué en "Public Documentation License".

The contents of this Documentation are subject to the Public Documentation License. You may only use this Documentation if you comply with the terms of this License.

A copy of the License is available at this mail [loiseljp@club-internet.fr](mailto:loiseljp@club-internet.fr).

The Original Documentation is " **Installation d'un serveur WEB sous Linux**".

The Initial Writer of the Original Documentation is Jean Paul LOISEL © 2013. All Rights Reserved.

Contributor(s): \_\_\_\_\_.

Portions created by \_\_\_\_\_ are Copyright © \_\_\_\_\_ [Insert year(s)]. All Rights Reserved.

(Contributor contact(s): \_\_\_\_\_ [Insert hyperlink/alias]).

📖 The text of this chapter may differ slightly from the text of the notices in the files of the Original Documentation.

You should use the text of this chapter rather than the text found in the Original Documentation for Your Modifications.