

# CHAPITRE 0

## PREAMBULE, INSTALLATION DE L'OS

Mettons-nous d'abord dans le contexte, nous souhaitons installer notre serveur (web, email, fichier) sur notre réseau local. Pour ceci nous allons déjà commencer par donner un nom de domaine à notre réseau local :

```
bebenet.local
```

Ensuite donnons un nom à notre futur serveur :

```
bebeserv
```

Notre réseau est le suivant : Plusieurs PC sous Windows et Linux sont relié à un switch lui même brancher sur un modem router ADSL (Livebox, Freebox, Netgear ...)

Nos PCs se connectent par câble ou WIFI, le réseau est en DHCP (pas d'adresse IP fixe) et à l'heure actuelle tout fonctionne bien. Notre réseau local est sous l'adresse IP 10.0.0.0/24. Voici donc la configuration de notre réseau à l'heure actuelle :

```
network                10.0.0.0
netmask                255.255.255.0
gateway ou passerelle  10.0.0.1    (le routeur modem ADSL)
broadcast              10.0.0.255
serveur DNS            10.0.0.1    (le routeur modem ADSL)
serveur DHCP           10.0.0.1    (le routeur modem ADSL)
la plage DHCP est de 10.0.0.10 à 10.0.0.100
```

Voici ce que ne fera pas notre serveur : pare-feu pour le réseau, passerelle, DNS, DHCP  
Nous pourrions optionnellement rajouter le DNS et le DHCP si le routeur modem ADSL n'est pas satisfaisant dans ces services.

Maintenant que le contexte est posé, nous allons installer notre Debian Etch.

Téléchargez l'image ISO de Debian (par exemple celle ci : [http://cdimage.debian.org/debian-cd/4.0\\_r0/i386/iso-cd/debian-40r0-i386-netinst.iso](http://cdimage.debian.org/debian-cd/4.0_r0/i386/iso-cd/debian-40r0-i386-netinst.iso)) sur un poste capable de graver une image ISO sur CD.

Une fois l'image téléchargée, gravée, ce joli CD tout beau tout neuf va directement aller dans le poste qui va servir de serveur et qui est prêt à booter sur le CD (à régler dans le BIOS le cas échéant).

Vous arrivez sur la page de boot Debian (sobri et efficace), faites « [Entrer] » pour lancer l'installation. Assurez vous simplement que la machine est branchée sur le réseau local par un câble relié au switch ou directement sur le routeur.

Suivez l'installation tranquillement, répondez aux questions avec sérieux, si vous avez bien lu et avez fais la correspondance avec votre propre réseau et matériel par rapport au contexte que j'ai

donné, rien ne devrait vous poser de soucis.

Au moment de la configuration du réseau, après la détection automatique, choisissez « configurer manuellement » : nous donneront une adresse IP fixe au serveur. Par exemple ici l'adresse sera « 10.0.0.2 » (puisque'elle est disponible et hors de la plage DHCP de notre modem routeur ADSL).

Nous choisirons « ALBAN » comme nom d'utilisateur local.

A la dernière question, il vous sera demandé d'installer des paquets supplémentaires, environnement de bureau, serveur web etc... n'ajoutez rien (utilisez la barre espace pour décocher les options pré-sélectionnées).

Votre Debian Etch est maintenant toute fraîche installée et redémarrée, vous possédez donc la Debian la plus minimaliste possible avec un compte utilisateur et un compte « root » (super administrateur).

Nous allons ensemble installer, logiciel par logiciel, tout ce dont vous aurez besoin pour transformer votre Debian en véritable serveur.

Connectez vous sous votre compte utilisateur local « ALBAN », jamais en « root ».

```
Debian GNU/Linux 4.0 bebeserv tty1
bebeserv login : ALBAN
Password:
[...]
ALBAN@bebeserv:~$
```

La première étape consiste par commencer à ne jamais se connecter en « root ». Afin de pouvoir continuer à administrer votre serveur vous aurez besoin de l'utilitaire « sudo » qui vous permettra d'utiliser les droits « root » quand c'est nécessaire.

Donc pour cette fois-ci uniquement, passez sous l'utilisateur « root » :

```
ALBAN@bebeserv:~$ su root
Password:
[...]
bebeserv:/home/ALBAN#
```

Passons à l'installation de « sudo » :

```
bebeserv:/home/ALBAN# apt-get update
bebeserv:/home/ALBAN# apt-get install sudo
[...]
```

Paramétrez « sudo » afin que « ALBAN » puisse, quand c'est nécessaire, avoir les droits « root ». Pour ceci nous allons éditer le fichier « /etc/sudoers » à l'aide de la commande :

```
bebeserv:/home/ALBAN# visudo
```

Sous la dernière ligne « root ALL=(ALL) ALL » rentrez la ligne suivante :

```
ALBAN ALL=(ALL) ALL
```

Enregistrez « [CTRL + O] » puis tapez « [Entrer] » et quittez « [CTRL + X] »  
Quittez l'utilisateur « root » pour revenir sous « ALBAN » :

```
bebeserv:/home/ALBAN# exit  
exit  
ALBAN@bebeserv:~$
```

Dorénavant à chaque commande nécessitant les droits d'administration vous devrez la faire précéder de « sudo ».

Voici un exemple, réessayez sous « ALBAN » d'éditer « /etc/sudoers » :

```
ALBAN@bebeserv:~$ visudo  
- bash: visuo: command not found
```

Vous obtenez cette erreur car « visudo » n'est pas accessible sans les droits super utilisateur. Essayer ceci :

```
ALBAN@bebeserv:~$ sudo visudo  
Password:
```

Un mot de passe vous est demandé. Entrez le mot de passe de « ALBAN » (pas celui de « root »). Et voilà vous pouvez éditer de nouveau « /etc/sudoers ».

Quittez sans modifier le fichier « [CTRL + X] ».

Si vous retentez « sudo visudo » le mot de passe ne vous sera pas redemandé, il est enregistré pendant 15 minutes. Pour que le mot de passe vous soit de nouveau redemandé avant les 15 minutes (pour stopper la session « sudo » en d'autres termes) faites :

```
ALBAN@bebeserv:~$ sudo -k
```

Il se peut aussi que vous ayez besoin d'utiliser en permanence la commande « sudo » quand vous installez un logiciel par exemple ou que vous administrez votre serveur. Plutôt que de faire un « su root » ou de devoir faire précéder chaque commande de « sudo » faites ceci :

```
ALBAN@bebeserv:~$ sudo -i  
Password:  
bebeserv:~#
```

Entrez toujours le mot de passe de « ALBAN ». Ceci aura le même effet qu'un « su root ».

Pourquoi est-ce mieux alors ? C'est mieux car « sudo » enregistre dans un fichier de log « /var/log/auth.log » toutes les commandes exécutées. Ainsi vous pouvez retracer tout ce qui a été manipulé par vous même ou un autre administrateur.

Faites « exit » de la même manière que pour « su root » pour quitter et revenir sous « ALBAN » :

```
bebeserv:~# exit
```

```
logout
ALBAN@bebeserv:~$
```

Maintenant que nous sommes bien dans le bain, avant d'aller plus loin nous allons rendre le « bash » (le « shell » ou éditeur en ligne de commande) un peu plus confortable. Éditez votre configuration personnelle de « bash » ou créer un fichier des alias (ce que je vais faire ici) :

```
ALBAN@bebeserv:~$ nano ~/.bash_aliases
```

Entrez ceci :

```
alias cp='cp -i'
alias mv='mv -i'
alias rm='rm -i'

alias df='df -hT'
alias du='du -h'
alias grep='grep -color'

alias ls='ls --color -F'
alias ll='ls --color -lF'
alias l='ls --color -lahF'
```

Enregistrez et quittez.

Pour activer les nouveaux alias il faut maintenant éditer le fichier « .bashrc »

```
ALBAN@bebeserv:~$ nano ~/.bashrc
```

Décommentez ou écrivez les lignes suivantes :

```
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
```

Enregistrez et quittez.

Il ne reste plus qu'à vous déloguer et vous reloguer pour que les modifications prennent effet :

```
ALBAN@bebeserv:~$ exit
exit
Debian GNU/Linux 4.0 bebeserv tty1

bebeserv login : ALBAN
Password:
[...]
ALBAN@bebeserv:~$
```

Voilà.

Maintenant pour finir de vous roder avant d'attaquer les choses sérieuses nous allons installer « ntpdate » qui va nous permettre de mettre à jour l'heure de notre serveur en synchronisant votre serveur avec une horloge atomique distante.

Commençons par installer « ntpdate » :

```
ALBAN@bebeserv:~$ sudo apt-get install ntpdate  
[...]
```

Pour synchroniser votre horloge vous aurez besoin donc d'un serveur NTP. Voici la liste des serveurs NTP français [http://www.cru.fr/NTP/serveurs\\_francais.html](http://www.cru.fr/NTP/serveurs_francais.html).  
Maintenant lançon une synchronisation :

```
ALBAN@bebeserv:~$ sudo ntpdate ntp.univ-lyon1.fr  
[...]
```

ATTENTION : il se peut que « sudo » n'apprécie guère le changement d'heure de votre serveur surtout si celui-ci était vraiment désynchroniser. Le changement d'heure perturbe la gestion des session de « sudo ». J'ai réglé le problème en faisant un « sudo -i » puis un « exit » et tout est rentré dans l'ordre.

Maintenant l'heure est synchrone.

C'est bien mais si mon serveur se désynchronise de nouveau, devrai-je faire cette manipulation à la main à chaque fois ?

Non bien-sûr nous allons automatiser ceci. Éditer le fichier « /etc/crontab ».

```
ALBAN@bebeserv:~$ sudo nano /etc/crontab
```

Rentrez ceci à la fin du fichier :

```
33 3 * * * root ntpdate ntp.univ-lyon1.fr 2>&1 > /dev/null
```

Enregistrez et quittez. Maintenant tous les jours à 3 H 33 votre serveur se remettra à l'heure tout seul. Notons que « > 2>&1 > /dev/null » est facultatif, il a pour effet de redirigé la sortie standard et la sortie d'erreur vers rien afin d'éviter que des e-mails soient envoyés à l'administrateur en cas d'erreur.

Voilà c'est terminé pour ce chapitre.

```
ALBAN@bebeserv:~$
```