Installation de base d'un serveur web sous Debian

par Olivier Lange

Date de publication :

Dernière mise à jour :

Installer les logiciels de base pour gérer un serveur Web.

I - Installation de base

I-A - Préparer son serveur

se connecter en root sur le serveur Effectuer une mise a jour

```
apt-get update
apt-get upgrade
```

Une fois le serveur a jour, on peut commencer a installer les différents services nécessaires sur notre serveur. Attention, pensez a redémarrer les services après chaque installation / configuration

```
# /etc/init.d/nom_service restart
```

I-B - Installer apache 2

```
# apt-get install apache2
Tester l'installation d'apache : http://xxx.xxx.xxx.xxx/ (IP du serveur)
```

Vous pouvez supprimer la redirection sur /apache2-default/

```
Editer /etc/apache2/sites-available/default et commenter RedirectMatch ^/$ /apache2-default/
```

I-C - Installer php 5

```
# apt-get install php5
```

Si il annonce que le package n'a pas été trouvé, éditez /etc/apt/sources.list et ajouter :

```
deb http://packages.dotdeb.org stable all
```

On mets a jour la liste des package

```
# apt-get update
```

Et on installe php5

Installer libapache2-mod-php5 (il ne le fait pas d'office !!!)

```
# apt-get install libapache2-mod-php5
```

I-D - Installer La base de donnée (Mysql)

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

Définir le mot de passe root : « mysql » (par exemple) Tester que Mysql fonctionne :

```
# mysql -p
entrer le mot de passe
Exit pour quitter
```

Installer les librairies php5-mysql

```
# apt-get install php5-mysql
```

Installer PhpMyAdmin

```
# apt-get install phpmyadmin
Choix du serveur a paramétrer : Apache2
On redémarre apache quand proposé
```

On se connecte par http://xxx.xxx.xxx.xxx/phpmyadmin

I-E - Installation de BIND9 (serveur DNS)

```
# apt-get install bind9
```

I-F - Installation du serveur mail

```
# apt-get install postfix
Choix de distribution : site internet
Choix suivants : par défaut
```

I-G - Installation du FTP (VSFTPD)

```
# apt-get install vsftpd
```

II - Configuration des virtualhost d'Apache 2

II-A - Pré-requis

Une fois que le serveur est installé de base, nous allons créer et configurer nos espaces d'hébergements. Tout d'abord, ce tutorial part des principes suivants:

- Vous désirez pouvoir accéder a vos sites par ip_du_server/~nom_user
- Vous n'avez qu'une seule IP pour tout vos sites
- Vous allez configurer BIND (voir tuto un peu plus loin)

Dans cette première partie, nous allons modifier 1 fichier: /etc/apache2/apache2.conf, et créer des fichiers dans les répertoires /etc/apache2/sites-available et /etc/apache2/sites-enabled. Mais allons-y par étapes.

II-B - Configuration d'apache2.conf

- ON édite le fichier de configuration principal d'apache2

A ce niveau-là, un /etc/init.d/apache2 restart permet déjà d'accéder a ses répertoires privés de chacun de ses users (penser à rajouter un dossier public_html dans ceux-ci pour voir qqch!)

II-C - Installation des virtualhost

On va maintenant créer nos hôtes virtuels. Par défaut, je les appellerai test1.com et test2.com. A vous de mettre vos noms que vous désirez. Mais avant de s'attaquer aux users, on commence par modifier le squelette de la création des nouveaux users. L'avantage? Ne pas avoir besoin a chaques fois de devoir créer le répertoire public_html et logs quand on crée un nouvel user, mais aussi d'avoir directement une page d'accueil.

```
# mkdir /etc/skel/public_html
# mkdir /etc/skel/logs
# echo « <h1>Nouvel espace web créer</h1> » >> /etc/skel/public_html/index.html
```

Une fois le squelette créer, on peut créer un nouvel user

```
# useradd -g www-data -m test1
```

```
# nano /etc/apache2/sites-available/test1.com
```

On copie le contenu ci-dessous (Description complète sous peu, pour compléter ce fichier):

```
<VirtualHost *>
        ServerAdmin postmaster@test1.com
        ServerName www.test1.com
        ServerAlias test1.com *.test1.com
        DocumentRoot /home/test1/public_html/
        <Directory /home/test1/public_html/>
                Options -Indexes FollowSymLinks MultiViews
                AllowOverride All
        </Directory>
        ErrorLog /home/test1/logs/error.log
        LogLevel warn
        CustomLog /home/test1/logs/access.log combined
        ServerSignature Off
</VirtualHost>
```

On valide et on ferme le fichier. On rends le domaine créer disponible

```
# ln -s /etc/apache2/sites-available/test1 /etc/apache2/sites-enabled/test1.com
```

On redémare apache2

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

Et on peut accéder a notre répertoire

```
http://xxx.xxx.xxx.xxx/~test1
```

Et on devrait voir une page web! Il ne reste plus qu'à informer les visiteurs de la présence de ce site sur ce serveur. Et cela, c'est Bind qui s'en charge! Rendez-vous sur le éprochain tutoriel ;)

III - Configuration de Bind 9

III-A - Introduction

Tout d'abord, qu'est-ce que bind? Bind est le petit programme qui va informer vos visiteurs de la présence et la validité d'une url sur ce serveur. Il n'est pas indispensable, sachant que chaque registar correcte permet de les gérer depuis leurs interfaces. Pour ma part, je préfère tout gérer depuis mon dédié. En conséquence, je ne vous indiquerai que cette méthode pour rendre vos sites disponibles.

III-B - Named.conf

Pour configurer Bind, nous allons modifier un ficher /etc/bind/named.conf et en créer un pour chaque domaine que nous désirons héberger.

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

On ne touche pas aux données par défaut, mais on rajoute, après la dernière zone (1 entrée pour chaque domaine, évidemment. Et test1.com est a changer par votre nom de domaine, cela va de soit!):

```
zone "test1.com" {
        type master;
        file "/etc/bind/db.test1.com";
};
```

Une petite modification à faire, pour éviter que notre serveur ne serve de relay DNS ouvert. On ajoute ces lignes dans le fichier (juste sous le premier include):

```
options {
        allow-recursion { localhost; };
};
```

On valide et on sort du fichier

III-C - Création du fichier de configuration de zone

On crée le fichier ce description de notre zone

```
# nano  /etc/bind/db.test1.com
```

On entre les valeurs de notre domaine (ceci est un exemple. A vous de l'adapter a vos besoins/envies):

```
$ttl 86400
test1.com.           IN      SOA     ksXXXXX.kimsufi.com. webmaster.test1.com. (
                                              2006121903
                                               21600
                                          3600
                                          604800
                                          86400 )
test1.com. IN      NS                    ksXXXXX.kimsufi.com.
test1.com. IN      NS                    ns.kimsufi.com.
test1.com. IN      MX                10 mail.test1.com.
test1.com. IN A              xxx.xxx.xxx.xxx
Server         IN A         xxx.xxx.xxx.xxx
www         IN A              xxx.xxx.xxx.xxx
mail         IN A              xxx.xxx.xxx.xxx
```

```
smtp         IN A               xxx.xxx.xxx.xxx
pop                IN A               xxx.xxx.xxx.xxx
pop3         IN CNAME         Server
imap         IN A               xxx.xxx.xxx.xxx
sql         IN A               xxx.xxx.xxx.xxx
mysql         IN A               xxx.xxx.xxx.xxx
```

Quelques explications:

- 2006121903: est à modifier à chaque édition du fichier. Par convention, on l'écrit: année-mois-jour-numéro a 2 chiffres
- 21600: Temps que le serveur esclave doit attendre avant de questionner à nouveau le serveur maitre
- 3600:Temps à attendre avant d'effectuer une nouvelle demande au serveur maitre en cas de non réponse
- 604800: Temps d'expiration du serveur principal en cas de non réponse
- 86400: Temps de mise en cache minium par d'autres serveur DNS
- CNAME permet de faire le liens entre une entrée à un alias (ou plusieurs).

On sauvegarde, on redémarre bind9

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

Et voila, il ne nous reste plus qu'à tester notre redirection, avec nos noms de domaines. Une fois que bind est configuré, il est possible de tester notre serveur DNS avec un petit site internet: http://dnsreport.com/tools/dnsreport.ch?domain=test1.com Il est également possible d'avoir une description complète des différentes options de bind sous google. Une simple recherche vous donnera une foule de site pour cela!