

Installation complète d'un serveur web sous Debian

Le logiciel libre **Apache HTTP Server (Apache)** est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.

Installation de base

II-B. Préparer son serveur

Il faut se connecter en root sur le serveur puis effectuer une mise à jour :

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

Une fois le serveur à jour, on peut commencer à installer les différents services nécessaires sur notre serveur. Attention, pensez à redémarrer les services après chaque installation ou reconfiguration :

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

II-C. Installer apache 2

Il nous faut installer un serveur HTTP qui va s'occuper d'afficher nos différentes pages. Pour cela, je vous propose d'installer l'un des serveurs les plus utilisés et les plus connus : Apache. Nous installerons ici sa version 2.

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

II-D. Installer php 5

Actuellement, notre serveur peut nous afficher des pages statiques au format HTML. La plupart des sites que vous voudrez installer disposeront d'une partie dynamique. C'est pourquoi nous poursuivons par l'installation de PHP 5 sur le serveur.

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

S'il annonce que le package n'a pas été trouvé, éditez le fichier source d'Apt et ajoutez :

```
# nano /etc/apt/sources.list
```

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

On met à jour la liste des packages :

```
# apt-get update
```

II-E. Installer La base de donnée (Mysql)

PHP est très très souvent couplé à un système de base de données : Mysql. Nous installons ici Mysql-server version 5. Vous verrez plus bas que nous allons également installer phpmyadmin. Il s'agit d'un script php qui permet de gérer ses bases de données Mysql de façon très simple.

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

Définir le mot de passe root de Mysql (« mysql » par exemple). Dans l'écran suivant, il demande s'il faut gérer les connexions d'hôtes qui utilisent Debian Sarge. On répond OUI (répondre non empêchera la configuration de Postfix par la suite !).

On vérifie que Mysql fonctionne bien :

```
# mysql -p  
entrer le mot de passe  
>Exit;
```

Installer les librairies php5-mysql :

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

II-F. Installation de BIND9 (serveur DNS)

Maintenant que nous avons notre serveur web installé, il faut que nos visiteurs puissent nous trouver. Et il n'est guère facile de leur faire retenir une adresse du style: 213.251.175.34/~nom_de_mon_site. Si vous y arrivez, vous êtes un champion ;) . C'est donc le travail de Bind de traduire notre nom de domaine. Il vous suffira de définir votre nom de serveur chez votre registrar, dans la partie DNS1.

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

Une fois fini l'installation il faut configurer les serveur DNS et le Serveur Web pour mon cas j'ai récupéré un ancien serveur DNS que j'avais déjà configuré auparavant donc il faut tout d'abord configurer les 2 fichiers principaux le DB et le REV qui se trouve dans **/var/cache/bind/**

donc pour mon cas je voudrai crée 4 site :

www.steward.fr

support.steward.fr

ventes.steward.fr

catalogue.steward.fr

donc tout d'abord il faut renseigner sur le DNS notre Serveur Web et puis créer les 4 alias dans le fichier de zone direct DB :

```
GNU nano 2.2.6      Fichier : db.steward.fr      Modifié
$TTL 604800
steward.fr. in SOA srv.steward.fr. dnsmaster.steward.fr. (
2015090401
3H
1H
1W
1D )

steward.fr. in NS srv.steward.fr.
steward.fr. in A 192.168.21.1
web.steward.fr. in A 192.168.21.4
www.steward.fr. in CNAME web.steward.fr.
support.steward.fr. in CNAME web.steward.fr.
ventes.steward.fr. in CNAME web.steward.fr.
catalogue.steward.fr. in CNAME web.steward.fr.

;sisr.steward.fr. IN NS dns.sisr.steward.fr.
;dns.sisr.steward.fr. IN A 192.168.21.3
srv.steward.fr. in A 192.168.21.1
client.steward.fr. in A 192.168.21.10
```

puis une fois fini il faudra maintenant renseigner sur le DNS la zone inverse REV :

```
GNU nano 2.2.6      Fichier : rev.steward.fr
$TTL 604800
21.168.192.in-addr.arpa. in SOA srv.steward.fr dnsmaster.steward.fr. (
2015090401
3H
1H
1W
1D )

21.168.192.in-addr.arpa. in NS srv.steward.fr.

4.21.168.192.in-addr.arpa. in PTR web.steward.fr.
1.21.168.192.in-addr.arpa. in PTR srv.steward.fr.
10.21.168.192.in-addr.arpa. in PTR client.steward.fr.
```

Nous avons fini la configuration de notre serveur DNS.

Donc une fois fini maintenant il faudra passer sur le serveur web pour le configurer.

Tout d'abord nous allons commencer par la création de notre VirtualHost, nous allons créer un VirtualHost pour chacun de notre site.

Il faut aller dans le fichier **000-default-conf** qui se trouve dans :

/etc/apache2/sites-enabled/000-default.conf

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www/html/htdocs

  ServerName www.steward.fr

<Directory />
  Options FollowSymLinks
  AllowOverride all
</Directory>
<Directory /var/www/html/htdocs>
  AuthType Basic
  AuthName steward
  AuthUserFile /var/www/html/etc/passwords
  Options Indexes FollowSymLinks MultiViews
  AllowOverride all
  Require all granted
  Order allow,deny
  allow from www.steward.fr
</Directory>

  ErrorLog /var/www/html/logs/error.log
  CustomLog /var/www/html/logs/access.log combined
</VirtualHost>
```

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/support/html/htdocs

  ServerName support.steward.fr

<Directory />
  Options FollowSymLinks
  AllowOverride all
</Directory>
<Directory /var/www/support/htdocs>
  AuthType Basic
  AuthName steward
  AuthUserFile /var/www/support/etc/passwords
  Options Indexes FollowSymLinks MultiViews
  AllowOverride all
  Require all granted
  Order allow,deny
  allow from support.steward.fr
</Directory>

  ErrorLog /var/www/support/logs/error.log
  CustomLog /var/www/support/logs/access.log combined
</VirtualHost>
```

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/ventes/html/htdocs

  ServerName ventes.steward.fr
<Directory />
  Options FollowSymLinks
  AllowOverride all
</Directory>
<Directory /var/www/ventes/htdocs>
  AuthType Basic
  AuthName steward
  AuthUserFile /var/www/ventes/etc/passwords
  Options Indexes FollowSymLinks MultiViews
  AllowOverride all
  Require all granted
  Order allow,deny
  allow from ventes.steward.fr
</Directory>

  ErrorLog /var/www/ventes/logs/error.log
  CustomLog /var/www/ventes/logs/access.log combined
</VirtualHost>
```

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/catalogue/html/htdocs

  ServerName catalogue.steward.fr
<Directory />
  Options FollowSymLinks
  AllowOverride all
</Directory>
<Directory /var/www/catalogue/htdocs>
  AuthType Basic
  AuthName steward
  AuthUserFile /var/www/catalogue/etc/passwords
  Options Indexes FollowSymLinks MultiViews
  AllowOverride all
  Require all granted
  Order allow,deny
  allow from catalogue.steward.fr
</Directory>

  ErrorLog /var/www/catalogue/logs/error.log
  CustomLog /var/www/catalogue/logs/access.log combined
</VirtualHost>
```

Après avoir fini de créer les VirtualHost maintenant nous allons créer les répertoires qui contiennent les fichiers Web et les utilisateurs et leur mot de passe et le fichier de log.

Pour cela, il faudra aller dans `/var/www/` et ainsi créer les différents répertoires de notre site grâce à la commande **mkdir** :

/html →

- **etc** (ce répertoire contient les utilisateurs, leur mot de passe et le fichier d'authentification)
- **htdocs** (ce répertoire contient les pages de notre site index.html)
- **logs** (ce répertoire contient les fichiers logs il faudra rien écrire à l'intérieur de ce répertoire les fichiers vont être créés AUTO)

/support

- **etc** (ce répertoire contient les utilisateurs, leur mot de passe et le fichier d'authentification)
- **htdocs** (ce répertoire contient les pages de notre site index.html)
- **logs** (ce répertoire contient les fichiers logs il faudra rien écrire à l'intérieur de ce répertoire les fichiers vont être créés AUTO)

/ventes

- **etc** (ce répertoire contient les utilisateurs, leur mot de passe et le fichier d'authentification)
- **htdocs** (ce répertoire contient les pages de notre site index.html)
- **logs** (ce répertoire contient les fichiers logs il faudra rien écrire à l'intérieur de ce répertoire les fichiers vont être créés AUTO)

/catalogue

- **etc** (ce répertoire contient les utilisateurs, leur mot de passe et le fichier d'authentification)
- **htdocs** (ce répertoire contient les pages de notre site index.html)
- **logs** (ce répertoire contient les fichiers logs il faudra rien écrire à l'intérieur de ce répertoire les fichiers vont être créés AUTO)

il faudra créer la page index.html puis la placer dans /htdocs et maintenant nous allons créer les utilisateurs dans /etc

tout d'abord il faudra créer les utilisateurs et leur fichier grâce à la commande :

```
# htpasswd -c /var/www/html/etc/passwords steward
# htpasswd -c /var/www/support/etc/passwords steward
# htpasswd -c /var/www/ventes/etc/passwords steward
# htpasswd -c /var/www/catalogue/etc/passwords steward
```

une fois fini il faudra maintenant créer le répertoire d'authentification dans /**htdocs** grâce à la commande # **nano .htaccess** puis il faudra configurer ce fichier comme ceci :

```
AuthType Basic
AuthName steward
AuthBasicProvider file
AuthUserFile /var/www/html/etc/passwords
Require valid-user
```

une fois avoir fini il faudra redémarrer le serveur apache et le serveur DNS puis tester le fonctionnement en tapant sur une station client le URL de notre site.

SRV DNS → `/etc/init.d/bind9 restart`

SRV WEB → `/etc/init.d/apache2 restart`