



Anthony GARCIA  
2008

# SOMMAIRE

## ***I - Introduction***

- Qu'est-ce que Apache ?
- Qu'entend-t-on par **serveur HTTP** et **serveur Web** ?

## ***II - Historique***

## ***III - Configuration d'Apache... ...par des liens***

- Documentation Apache v2.0
- Configuration Apache
- Les fichiers de configurations
- Un peu de sécurité
- Windows
- Linux
- FreeBSD

## ***IV - Notion de « serveurs virtuels »***

- Qu'entend-t-on par « serveurs virtuels » ?
- Pourquoi faire ?
- Le principe

## ***V - Configuration d'Apache... ...par l'exemple***

- Conditions de réalisation
- Situation initiale
- Situation finale
- Déroulement
  - Installation
  - Configurations
  - Premières vérifications
  - Un peu de sécurité...
  - Vérifications finales

# Introduction

## ***Qu'est-ce que Apache ?***

**Apache HTTP Server**, plus souvent appelé **Apache**, est un logiciel de serveur HTTP produit par *Apache Software Foundation*.

C'est le serveur HTTP le plus populaire du Web.

## ***Qu'entend-t-on par serveur HTTP et serveur Web ?***

- Un *serveur HTTP* est un logiciel servant des requêtes qui respectent le protocole de communication client/serveur **Hypertext Transfer Protocol** (HTTP). Plus d'informations sur ce protocole développé pour le Web ici :  
<http://tools.ietf.org/html/rfc2616>
- Un *serveur Web* est une machine sur laquelle fonctionne un serveur HTTP. On confond souvent les deux termes et on appelle, par serveur Web, un serveur HTTP car le protocole HTTP a été développé pour le Web et, en pratique, les pages Web sont toujours servies avec ce protocole. La plupart du temps un serveur Web combine : serveur HTTP + serveur de base de données + PHP. Pour des raisons de gratuité et d'Open Source, les plus utilisés pour cela sont : Apache + MySQL + PHP. Pour plus d'informations voir les serveurs LAMP (Linux Apache MySQL PHP), WAMP (Windows ...) et MAMP (Macintosh ...).

## Historique

Apache est apparu en 1995. Au début il s'agissait d'une collection de correctifs et d'additions au **serveur NCSA HTTPd 1.3**, qui était dans le domaine public et le *serveur HTTP* alors le plus répandu. Pour plus d'informations sur **NCSA HTTPd** :

<http://hoohoo.ncsa.uiuc.edu/>

De cette origine, de nombreuses personnes affirment que le nom *Apache* vient de « *a patchy server* », soit « un serveur rafistolé ». Par la suite Apache a été complètement ré écrit, de sorte que, dans la version 2, il ne reste aucune trace de **NCSA HTTPd**.

Au début, Apache était la seule alternative sérieuse et libre au serveur HTTP de Netscape (Sun ONE). Depuis avril 1996, selon l'étude permanente de Netcraft, Apache est devenu le serveur HTTP le plus répandu sur Internet. En mai 1999, il faisait tourner 57% des serveurs Web, et si, en début 2004, il était à environ 69% de parts de marché, il ne détient plus que 50,61% du marché à la fin du mois de janvier 2008. En février 2008, Apache représente 50,93% des parts de marché.

Apache supporte plusieurs plates-formes et fonctionne principalement sur les systèmes d'exploitation Unix et Windows.

## Configuration d'Apache... ...par des liens

- Documentation Apache v2.0 :  
<http://httpd.apache.org/docs/2.0/fr/>
- Introduction à la configuration d'Apache (1.3.23) :  
<http://www.ac-creteil.fr/reseaux/systemes/linux/config-apache.html>
- Les fichiers de configuration :  
<http://httpd.apache.org/docs/2.0/fr/configuring.html>
- Le fichier httpd.conf :  
<http://linux.com.hk/penguin/man/8/fr/httpd.html>
- Un peu de sécurité avec les fichiers .htaccess :  
<http://www.commentcamarche.net/apache/apacht.php3>
- Apache et hôtes virtuels sous Windows :  
<http://apache.developpez.com/cours/virtual-host/>
- Apache et hôtes virtuels sous Linux :  
<http://cern91.tuxfamily.org/linux/indexnet.php?page=virtual>
- Apache et hôtes virtuels sous FreeBSD :  
<http://www.freebsdidiary.org/virtualhosts.php>

## Notion de « serveurs virtuels »

### ***Qu'entend-t-on par « serveurs virtuels » ?***

Le principe de l'hébergement virtuel est devenu incontournable dans la version 2 d'Apache, à tel point que l'on imagine plus l'un sans l'autre. Cette technique propose *plusieurs sites sur un même serveur Web*.

## ***Pourquoi faire ?***

Cette technique se rencontre par nature chez les fournisseurs d'accès Internet qui proposent à leurs clients, outre l'accès, un espace d'hébergement pour les pages Web personnelles. Dans le cadre d'une entreprise, la segmentation par site logiquement les services : service support, administration, ventes, etc. et nécessite des droits adaptés pour chacun.

## ***Le principe***

Un même serveur peut héberger plusieurs sites Web et peut se faire de deux façons :

- soit la machine possède plusieurs adresses IP (on parle alors d'*IP aliasing*) et deux techniques sont possibles :
  - on attache une instance de serveur HTTP à chaque adresse IP, il s'agit alors d'*hébergement sur des serveurs séparés*.
  - le serveur HTTP est capable de distinguer entre les différentes adresses IP sur lesquelles arrivent les requêtes des navigateurs, c'est l'*hébergement virtuel basé sur l'IP*.
- soit la machine possède une seule adresse IP et c'est le serveur HTTP qui essaie de décoder le nom du service demandé par le navigateur distant, on parle alors d'*hébergement virtuel basé sur le nom* ; cette technique nécessite un navigateur qui expédie un en-tête *HOST* optionnel avec **HTTP/1.0** et obligatoire avec **HTTP/1.1**.

Quelle que soit la méthode choisie, il y a un lien entre l'hébergement de plusieurs serveurs virtuels et le **DNS**, dans tous les cas les serveurs virtuels sont référencés dans la base de données **DNS** mais :

- pour l'hébergement basé sur les adresses IP, chaque *serveur* est référencé par un enregistrement de type **A**.
- pour l'hébergement basé sur les noms, on préférera la création d'un enregistrement de type CNAME pour chacun des hôtes virtuels.

Un peu plus :

<http://httpd.apache.org/docs/2.2/fr/vhosts/>

# Configuration d'Apache... ...par l'exemple

## ***Conditions de réalisation***

On dispose, pour notre exemple, de trois machines virtuelles (émulée par Vmware Workstation). Deux machines sous Ubuntu 7.10 en mode serveur et une machine sous Ubuntu 7.10 avec interface graphique.

Nous aurons besoin du package **APACHE2** pour l'installation du serveur HTTP.

## ***Situation initiale***

Les trois machines sont les suivantes :

- un serveur DNS Ubuntu en console (IP : 192.168.1.2)
- un client avec Ubuntu en mode graphique
- une machine Ubuntu en console pour le serveur HTTP (IP : 192.168.1.3)

Les trois machines sont connectées sur un même réseau (192.168.1.0/24 dans notre exemple). Les paramètres réseaux sont ok (IP, DNS,...).

Sur la machine cliente, des comptes utilisateurs existent déjà (toto, tutu, titi, tata) ainsi que des groupe d'utilisateurs :

- support (contient toto et titi)
- compta (contient tutu)
- paye (contient tata)

## ***Situation finale***

Notre réseau possède désormais un serveur Web hébergeant deux sites :

- un site ouvert à tous
- un site sécurisé dont l'accès est interdit aux personnes n'appartenant pas au groupe « support ».

## ***Déroulement***

### Installation

Il nous faut commencer par installer le package « apache2 » :

```
# sudo apt-get install apache2
```

### Configurations

Configurons notre serveur HTTP, pour qu'il sache « qui fait office de serveur HTTP », en ajoutant au fichier de configuration ***/etc/apache2/httpd.conf*** la

ligne suivante :

```
ServerName 192.168.1.3
```

Nous créerons ensuite les répertoires qui contiendront les différents sites (ainsi que les journaux de « log ») :

```
# sudo mkdir /var/www/web
# sudo mkdir /var/www/web/logs
# sudo mkdir /var/www/support
# sudo mkdir /var/www/support/logs
```

On aura donc au final deux sites :

- le premier sera contenu dans **/var/www/web/**
- le deuxième sera contenu dans **/var/www/support/**

Pour des questions pratiques lors de notre exemple, des fichiers « index.html » différents ont été placés dans chacun de ces emplacements.

Créons maintenant un fichier d'hôtes virtuels. Pour nous simplifier la vie, nous reprendrons le fichier par défaut de Apache2 que nous modifierons à notre convenance : **/etc/apache2/sites-available/default**

```
NameVirtualHost 192.168.1.3
```

```
<VirtualHost 192.168.1.3>
```

```
    DocumentRoot /var/www/web          //là où sont les fichiers du site
    ServerName www.web-garcia.fr       //le site sera appelé par www.web-garcia.fr
    ServerAdmin webmaster@web-garcia.fr //adresse mail du webmaster
    ErrorLog /var/www/web/logs         //emplacement des fichiers logs
    TransferLog /var/www/web/logs      //      "      "      "      "
    <Directory /var/www/web>
        Option Indexes FollowSymlinks MultiViews
        Order allow,deny
        allow from all
    </Directory>
```

```
</VirtualHost>
```

```
<VirtualHost 192.168.1.3>
```

```
    DocumentRoot /var/www/support
    ServerName support.web-garcia.fr
    ServerAdmin webmaster@web-garcia.fr
    ErrorLog /var/www/support/logs
    TransferLog /var/www/support/logs
    <Directory /var/www/support>
```

```
    Option Indexes FollowSymlinks MultiViews
    Order allow,deny
    allow from all
</Directory>
</VirtualHost>
```

Notons qu'il est possible de ne pas utiliser le fichier par défaut mais plutôt de créer d'autres fichiers pour les hôtes virtuels (un fichier par hôte par exemple). La seule chose qu'il faut penser à faire c'est de créer un lien symbolique vers **/etc/apache2/sites-enabled** car seuls les fichiers d'hôte étant répertoriés ici seront pris en compte par le serveur HTTP.

```
# ln -s /etc/apache2/sites-available/leFichier /etc/apache2/sites-enabled
```

Dans notre cas le lien symbolique est créé de base avec le fichier par défaut donc inutile d'exécuter cette commande.

Ajouter au serveur DNS cette fois, deux alias dans le fichier de zone direct : (**srv-http** correspondant au nom de notre serveur HTTP)

```
www          IN CNAME  srv-http    //pour www.web-garcia.fr
support      IN CNAME  srv-http    //pour support.web-garcia.fr
```

Une fois tout cela fait, il nous faut relancer les services HTTP et DNS : (dans notre exemple apache2 et bind9 pour le DNS)

```
# /etc/init.d/apache2 restart
# /etc/init.d/bind9 restart
```

## Premières vérifications

A ce niveau il nous est possible de vérifier notre travail. Sur la station cliente, ouvrons un navigateur Web et constatons que lorsque l'on écrit :

- « **www.web-garcia.fr** » on est dirigé vers le site WEB
- « **support.web-garcia.fr** » on est dirigé vers le site SUPPORT

## Un peu de sécurité...

Petit rappel sur les utilisateurs :

- le groupe support contient uniquement toto et titi
- tutu et tata ne font pas parti de l'équipe support

Nous souhaitons interdire l'accès aux personnes qui n'appartiennent pas au groupe support.

Nous allons protéger l'accès à « **support.web-garcia.fr** » à l'aide du fichier ***/var/www/support/.htaccess*** :

```
AuthName « Zone réservée à l'équipe support » //texte de la fenêtre password
AuthType Basic //méthode supportée par toutes les versions des navigateurs
AuthUserFile /var/www/support/pass/pass.txt //fichier password utilisateur
AuthGroupFile /var/www/support/pass/group.txt //fichier de groupe
Require valid-user
```

Créons maintenant le fichiers cryptés de mots de passe utilisateurs :

```
# mkdir /var/www/support/pass
# htpasswd -c /var/www/support/pass/pass.txt toto // -c pour la création
# htpasswd /var/www/support/pass/pass.txt titi
...
```

Puis le fichier de groupe ***/var/www/support/pass/group.txt*** :

```
support: toto titi
```

Un peu plus encore :

Créons un fichier **.htaccess** à la base du répertoire de mot de passe pour interdire toute navigation dans ce répertoire précieux :

```
deny from all
```

Indiquons à apache2 qu'il doit gérer ce niveau de sécurité pour le site « **support.web-garcia.fr** ». Il faut donc ajouter la ligne suivante au fichier d'hôtes virtuels ***/etc/apache2/sites-available/default*** :  
(entre <Directory /var/www/supprot> et </Directory>)

```
AllowOverride AuthConfig
```

Enfin redémarrons apache2 pour que cette dernière ligne soit prise en compte :

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

A savoir ; toutes modifications concernant les fichiers **.htaccess** n'entraînent pas la nécessité du redémarrage du service apache2 !

## Vérifications finales

Il est temps de tester si notre stratégie de sécurisation est fonctionnelle ou non en testant l'accès au site « **support.web-garcia.fr** » par les différents membres du groupe support et les autres...

...