

Le Serveur Apache

I – Download & Install

Le but de ce guide va être d'installer le serveur APACHE sur la machine. Le serveur Web APACHE a été choisi car c'est le plus utilisé actuellement. Il est de plus gratuit et son code source est disponible sur le site <http://www.apache.org>.

ATTENTION !! Les machines d'OVH ont déjà la configuration. Vous pouvez aussi trouver les scripts standards de compilation sur notre serveur :

`ftp://ftp.ovh.net/made-in-ovh/patch`

1.1. Aller chercher les sources sur le site Web

La version sur laquelle nous allons travailler dans ce TP est la version 1.3.4 . Vous pouvez la trouver sur le site <http://www.apache.org>. Vous devez télécharger le fichier `apache1.3.4.tar.gz`.

1.2. Décompresser les sources dans un répertoire temporaire

Avant la décompression du fichier télécharger, il faut créer un répertoire temporaire à l'aide de la commande : `mkdir nom_du_répertoire`

Il faut ensuite décompresser le fichier à l'aide de la commande suivante : `tar zxvf nom_fichier.tar.gz`. Le fichier est zippé et taré. Il faut bien entendu décompresser les sources dans le répertoire temporaire créé précédemment.

1.3. Lire les fichiers permettant l'installation d'APACHE

Les fichiers permettant l'installation d'apache sont les suivants :

Install et tous les fichiers contenu dans le répertoire `htdocs/manual`.

Grâce à ces fichiers , nous avons pu savoir que la place nécessaire pour l'installation du serveur Apache est de 12 Mo et que l'installation nécessite un compilateur C ANSI.

1.4. Compiler le serveur Web

Pour compiler le serveur Apache, il faut saisir les commandes suivantes :

```
./configure --prefix=PREFIX où PREFIX correspond au chemin d'installation ( Ex : /usr/local/apache )
```

```
make
```

```
make install
```

`PREFIX/sbin/apachectl start` qui permet de démarrer le daemon HTTPD et donc de vérifier si la compilation s'est bien passée. PREFIX correspond aussi au répertoire d'installation.

1.5. Editer les fichiers de configuration

Bien que deux logiciels existent pour configurer le serveur APACHE nous donnerons ici les méthodes de configuration du serveur par modification des fichiers textes contenus dans le répertoire `conf`.

Les fichiers de configuration sont placés dans le répertoire `/etc/httpd/conf`.

Il y a trois fichiers majeurs dans la configuration :

access.conf : contient les instructions permettant de gérer les droits d'accès au serveur.

srm.conf : contient les instructions gérant les noms et les types accessibles aux utilisateurs connectés au moyen des navigateur.

http.conf : contient les directives propres au fonctionnement du serveur lui-même.

1.6. Lancer le serveur Web

La commande suivante permet de démarrer le serveur :

```
apachectl start
```

La commande suivante permet de redémarrer le serveur :

```
apachectl restart
```

1.7. Arrêter le serveur Web

La commande suivante permet d'arrêter le serveur :

```
apachectl stop
```

II – Configuration du serveur Web

2.1. Changer le port et le nom de l'utilisateur

Dans les parties qui suivent il est nécessaire d'avoir créé un utilisateur au préalable. Si ce n'est pas le cas il faut le faire en utilisant la commande : adduser.

Considérons un utilisateur toto.

```
adduser -g users toto
```

Pour changer le nom de l'utilisateur qui va lancer le serveur Web (ici toto), il faut éditer le fichier http.conf.

Trouver l'endroit où il est inscrit User et remplacer nobody par toto. Y'a plus qu'à et ça donne :

```
#User est le nom de d'utilisateur sous lequel le serveur accède aux ressources de l'ordinateur
```

```
User toto
```

Pour changer le port virtuel sur lequel le serveur Web va s'accrocher, il faut aussi éditer le fichier http.conf.

Trouver l'endroit où il est inscrit Port et remplacer la valeur qui se trouve après par le numéro de port que vous voulez. ATTENTION : certains ports sont réservés (FTP, Telnet, ...). Ce qui donne :

```
#Le port du serveur par défaut c'est 80 et on le remplace par 3000
```

```
Port 3000
```

ATTENTION !! si vous changez le port virtuel, lorsque vous vous connectez au site de toto, il faudra spécifier le nouveau port

2.2. Changer le nom du répertoire racine des utilisateurs

Nous voulons dans cette partie changer la configuration pour que l'orsqu'on cherche l'URL

http://monserveur/toto, APACHE cherche le fichier tutu.html (toujours aussi original !!) dans le répertoire WWW de l'utilisateur toto.

Pour se faire, il faut éditer le fichier srm.conf. Trouver (eh oui, encore !) l'endroit où il est inscrit

```
DirectoryIndex?
```

et remplacer ce qui suit par le fichier à charger. Ce qui devrait donner :

nom du ou des fichiers à afficher par défaut lorsqu'aucun fichier n'est donné dans l'URL.

Si vous voulez spécifier plusieurs fichiers ,ils faut les mettre les uns à la suite des autres :

```
DirectoryIndex? chemin/toto/WWW/tutu.html
```

2.3. Changer les messages d'erreur

Il est possible de changer les messages d'erreur en éditant le fichier srm.conf et modifier le contenu suivant le mot ErrorDocument?.

Par exemple :

ErrorDocument? 500 "Le serveur a commis une erreur" par :

ErrorDocument? 500 "Le serveur n'a pas commis une erreur"

ou :

ErrorDocument? 404 /missing.html

aura pour effet d'afficher la page missing.html lorsque le serveur ne trouvera pas la page désirée (code d'erreur 404).

III – Test du serveur

Il y a deux grandes façons de tester un serveur :

la première assez classique consiste à utiliser le browser pour visualiser la page.

La seconde consiste à effectuer un telnet et à écrire directement en mode texte la requête pour obtenir la réponse.

Voici un exemple d'utilisation avec la commande telnet : on peut récupérer la page de présentation d'un site en tapant telnet vodka 8080 puis get /index.html HTTP/1.0 suivi de deux pressions de la touche ENTER.

La réponse du serveur se fait en deux parties, dans un premier temps :

l'entête qui indique si la question a été comprise en envoyant une valeur de retour (ici 200 OK), si la ressource à été trouvée, le type de la ressource etc...

HTTP/1.0 200 OK

Server : JavaWebServer/1.0.2

Content-Length? : 1721

Content-type : text/html

Last-modified : Fri, 30 Jan 1998 13:55:28 GMT

Connection :Close

Date : Wed, 06 Jan 1999 16:12:28 GMT

puis dans un second temps la ressource elle-même : la page HTML, l'image GIF, etc.

Quelques codes de retour différents :

404 Not Found : la page demandée n'a pas été trouvé

400 Bad Request : la requête n'a pu être interprétée

Les champs Content-Length? et Content-Type? correspondent à la taille et au type du document reçu.

IV – Performance

Pour que le serveur utilise le moins de process possible au démarrage, il faut éditer le fichier httpd.conf et rechercher la chaîne startserveurs . On trouve après cette chaîne le chiffre 5, il suffit alors de mettre 1 à la place pour ne lancer qu'un process.

V – Quelques questions

5.1. Le mécanisme de VirtualHost?

VirtualHost? permet de configurer plusieurs machines virtuelles sur le même serveur. Le serveur doit être configuré pour accepter plusieurs adresses IP ce qui se fait généralement en configurant le noyau Linux de la machine. La configuration se trouve dans le fichier http.conf et à la forme suivante :

ServerAdmin? webmaster@domaine.extension

DocumentRoot? /repertoire
 ServerName? nom.domaine.extension
 ErrorLog? répertoire relatif
 TransferLog? répertoire relatif

5.2. Définition de KeepAlive?, FollowSymLinks? et FancyIndexing?

Définition de KeepAlive? :

si cette variable est placée à ON dans le fichier http.conf, elle autorise les navigateurs à ordonner aux connexions de rester actives. Si cette variable est placée à OFF, elle est désactivée.

Puis on trouve à la suite de cette variable d'autres variables liées à KeepAlive? comme la variable MaxKeepAliveRequests? qui donne le nombre de connexions pouvant rester actives simultanément ou encore KeepAliveTimeout? qui donne le délai maximum en seconde pour que les connexions restées actives attendent sans action.

Définition de FollowSymLinks? :

la variable FollowSymLinks? se trouve dans le fichier access.conf. Si cette variable est positionnée dans le champ OptionsIndexes?, l'exploitation des liens symboliques ou raccourcis sera autorisée.

Définition de FancyIndexing? :

si cette variable est placée à ON dans le fichier srm.conf, des icônes des fichiers accompagnent l'affichage des répertoires. Avec la valeur OFF, les répertoires sont affichés uniquement en HTML.

À la suite de cette variable, on doit trouver une série de noms d'icônes (précédé de AddIcon) qui permet de lier les types de fichiers à des icônes lorsque le mode FancyIndexing? est sur ON. Autrement dit, ça permet de redéfinir les icônes associées aux fichiers lors de l'affichage des répertoires.

VI – Les droits d'accès

Le but est de restreindre les droits d'accès d'un répertoire du site Web. Ceci se fait en deux étapes :

1. Indiquer que le répertoire possède des droits spéciaux dans le fichier access.conf :

La protection d'une page par un mot de passe se fait de façon simple par ajout de la directive ci-dessous dans le fichier access.conf :

AuthName? Libellé affiché dans la fenêtre protégée

AuthType? Basic

AuthUserFile? /etc/users.http

require valid-user

/repertoire/page.html : et la page protégée

AuthName? : est le libellé qui est affiché dans la fenêtre ainsi protégée

AuthType? est en général positionné avec la valeur Basic.

AuthUserFile? est le fichier contenant le nom des utilisateurs qui ont le droit d'accéder à la page ainsi que leur mot de passe associé.

require est une directive qui peut prendre la valeur valid-user.

2. Créer un utilisateur et un mot de passe :

La fabrication d'un mot de passe se fait par le programme htpasswd qui prend la syntaxe suivante :

htpasswd [-c] passwordfile username

avec l'option -c le fichier est créé ; sans l'option -c, le fichier est complété par la nouvelle entrée.

OVH

passwordfile est le nom du fichier contenant les mots de passe.
username est le nom de l'utilisateur.