

HTTP Serveur



Serveur

Un **serveur** est un **logiciel** qui, dans le modèle **client/serveur**, **attend** des **requêtes** via le réseau.

Par extension on appelle aussi serveur la machine qui fait tourner ce logiciel.

Dans les **systèmes d'exploitation** ces logiciels sont appelés **services** car il sont lancés automatiquement par le système.

Serveurs WEB

CERN httpd 1990 , puis
NCSA httpd 1993, remplacé par
Apache httpd 1995 25 à 30 %

Microsoft **IIS** 1994 5 à 10 %
Nginx 2004 25 à 30 %

Le d à la fin de httpd veut dire **daemon**

Type d'Hébergement

L'hébergement mutualisé représente près de 40 % du marché.

Dans ce type d'hébergement Apache Httpd est quasiment seul.

Nginx domine légèrement dans les hébergements dédié.

Nous allons voir la configuration de Apache httpd

Apache httpd

C'est un logiciel **modulaire** :

Vous **choisissez** quelles fonctionnalités sont exécuter.

Cela permet de **limiter** la consommation de ressources système (**mémoire**) au strict **nécessaire**.

Il faut donc bien **penser** à **activer** les **modules** voulus.

L'**ajout** de nouveaux modules **nécessite** le **redémarrage** complet du serveur (changements du programme).

Configuration httpd

La configuration de httpd se fait dans un fichier texte. Celui-ci est écrit en pseudo **xml**.

À chaque fois que vous **modifiez** ce fichier il faut **demandeur** à apache de venir le relire pour se **mettre à jour**.

Ceci n'interrompt pas le serveur.

Une erreur de **syntaxe** dans le fichier peut faire **crasher** le serveur à ce moment là !

Configuration httpd

Dans les distributions "Debian" ce fichier est **découpé** en **plusieurs** fichiers pour en faciliter la maintenance et bien **isoler** les différentes parties.

Chaque **site** a son propre fichier.

Chaque **module** a son propre fichier.

Configuration httpd

Les **Balises** xml sont utilisées pour organiser les directives et définir leurs **portées** :

- Globale
- Pour un site <virtualhost ...>
- Pour un répertoire* <Directory ...>
- Pour une url <Location ...>

* : et ses sous-répertoires

Principes fondamentaux

Un serveur web va gérer plusieurs sites web. Chacun de ces sites sera configuré dans une section

```
<virtualHost IP:port></virtualHost>
```

IP et port permettent d'identifier le site par l'adresse IP ou/et le port de la requête.

Valeurs par défaut :

Pour un site en http : `<virtualHost *:80>`

Pour un site en https : `<vitrtualHost *:443>`

Principes fondamentaux

Un **site** web est constitué de **fichiers** organisés dans une **arborescence**.

Le serveur web fait la **correspondance directe** entre les chemins de l'URL et le chemin sur le **disque** dur.

Dans la configuration du serveur il **faut** indiquer où se trouve le **répertoire racine** du site avec la directive :

DocumentRoot `chemin_absolu`

Principes fondamentaux

La distinction entre les différents sites se fait grâce à :

- L'adresse **IP** dans `<virtualHost >`
- Le **port** dans `<virtualHost >`
- La **directive** `serverName`

Cette dernière utilise le champs **Host** de la **requête** pour identifier le **site**.

`serverName` `nom_dns_du_site`

Exemple minimaliste

Site <http://groupe12.gremmi.fr/>
déposé dans le répertoire
</home/groupe12/www> du serveur

```
<virtualHost *:80>  
    serverName groupe12.gremmi.fr  
    documentRoot /home/groupe12/www  
</virtualHost>
```

Droits d'accès

Par défaut le serveur interdit la publication du contenu du disque du serveur.

Il faut donc autoriser explicitement la publication pour le documentRoot

En utilisant une directive Require dans une balise <directory>

Droits d'accès

La directive **require** permet de donner ou restreindre l'accès :

- À certaines adresses **IP** du client
- À certains **types** de requête (GET, POST)
- Aux clients **authentifiés**

Exemple

```
<virtualHost *:80>  
    serverName groupe12.gremmi.fr  
    documentRoot /home/groupe12/www  
    <directory /home/groupe12/www>  
        Require all granted  
    </directory>  
</virtualHost>
```

Journaux

L'activité du serveur est enregistré dans des **fichiers** journaux (**logs**) aussi appelé fichiers de **trace**.

Il existe 2 types de journaux :

- Le journal des **erreurs**
- Le journal des **accès**

Journal des accès

Par défaut ce journal est dans le fichier : `/var/log/apache2/access.log`

Il est possible de créer un journal par virtualHost avec les directives :

CustomLog : nom du fichier et format utilisé

LogFormat : définition du format

En France la loi impose de conserver le journal des accès pendant 1 an

Journal des erreurs

Par défaut ce journal est dans le fichier : `/var/log/apache2/error.log`

Il est possible de créer un journal par virtualHost avec les directives :

`errorLog` : nom du fichier

`LogLevel` : niveau de verbosité

La **verbosité** peut être **ajustée** module par module

Démocratisation

Les fichiers de **configuration** du serveur ne sont **éditables** que par **l'administrateur** du système.

Pourtant beaucoup de **directives** dépendent plus du **site** que du système.

Apache propose un mécanisme pour pouvoir mettre les directives dans les sites :

Les fichiers **.htaccess**

.htaccess

Les **directives** qui sont dans un fichier **.htaccess** s'appliquent au **répertoire** - et ses sous-répertoires - dans lequel se **trouve** ce fichier.

L'utilisation des **.htaccess** est configurée par la directive :

allowOverride

Elle permet de restreindre le jeu de directives autorisées par famille de directive.

.htaccess

Toutes les **directives** ne sont pas possibles dans un fichier **".htaccess"**. La **documentation** vous indique si une directive est **utilisable** dans le contexte des fichier **".htaccess"**.

La **documentation** vous indique aussi à quelle **famille** la directive appartient, i.e. si elle est **autorisée** par **allowOverride**.

HTTPS

Les sites en **HTTPS** (HTTP over **TLS**) en plus du port **443**, ont besoin d'avoir un **certificat** et sa clé privée. Les **directives** sont :

SSLEngine //activation https

SSLCertificateFile //le certificat

SSLCertificateKeyFile //la clé privée

Documentation

La documentation de **Apache** est un modèle du genre :

Détaillée, complète, avec des exemples et des **tutoriaux**.

Cerise sur le gâteau : elle est **traduite** en de nombreuses langues.

Conclusion

RTFM

documentation [Apache httpd](http://httpd.apache.org/docs/)