

ANNEXE

Détail d'un fichier de configuration

Nous utiliserons le fichier de configuration par défaut d'un hôte virtuel – */etc/apache2/sites-available/000-default.conf* – voici son contenu (allégé des commentaires) :

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

La directive VirtualHost

`<VirtualHost>` et `</VirtualHost>` sont utilisés pour encapsuler un groupe de directives qui ne seront appliqués qu'à l'hôte virtuel qu'elles définissent. On spécifie l'adresse IP ou le nom de domaine auquel correspond l'hôte virtuel ainsi que le numéro de port. Dans l'exemple ci-dessus, (ainsi que par défaut), l'hôte virtuel est défini pour toutes les IPs (que le serveur http peut lire) et sur le port 80.

La directive ServerAdmin

`ServerAdmin` définit l'adresse de contact que le serveur inclut dans les messages d'erreurs qui seront retournés au client.

La directive DocumentRoot

Cette directive définit le répertoire d'origine à partir duquel httpd distribue les fichiers. À moins que l'url ne corresponde à un `Alias` (voir la directive `Alias`), le serveur ajoute le chemin de l'URL demandée à la racine du document pour faire le chemin au document. La racine (`DocumentRoot`) devrait être spécifiée sans un slash en fin de nom.

La directive Directory

`<Directory>` et `</Directory>` sont utilisés pour encapsuler un groupe de directives qui ne seront appliqués qu'au répertoire (et ces sous-répertoires) spécifié dans `<Directory nom_répertoire>`.

La directive AllowOverride

Quand le serveur trouve un fichier `.htaccess` (comme spécifié par `AccessFileName`) il doit connaître quelles directives déclarées dans ce fichier peuvent ignorer des directives de configuration précédentes. `AllowOverride` est valable seulement dans des sections `<Directory>` définies sans expressions régulières. Quand cette directive est mise à :

- **None**, alors les fichiers `.htaccess` sont complètement ignorés. Dans ce cas, le serveur n'essayera pas même de lire des fichiers de `.htaccess` dans le système de fichiers.
- **All**, le serveur prendra en compte toutes les directives dans le fichier `.htaccess`
 - ...

La directive Order

La directive `Order`, associée avec les directives `Allow` et `Deny`, pilote un système de contrôle d'accès à trois passages.

- Le premier passage traite toutes les directives `Allow` ou `Deny`, comme spécifié selon la directive `Order`.
- Le deuxième passage fait l'analyse syntaxique du reste des directives (`Allow` ou `Deny`).
- Le troisième passage s'applique à toutes les demandes qui ne correspondent à aucun des premiers deux.

Notez que toutes les directives `Allow` et `Deny` sont traitées, à la différence d'un pare-feu classique, où seulement la première directive correspondante est utilisée. De plus, l'ordre dans lequel les lignes apparaissent dans les fichiers de configuration n'est pas significatif - toutes les lignes `Allow` sont traitées comme un groupe et toutes les lignes `Deny` sont considérées comme un autre groupe.

L'ordre de traitement est un des deux suivants :

- **Allow,Deny** : En premier, toutes les directives `Allow` sont évaluées. Au moins une doit correspondre, ou la demande est rejetée. Ensuite, toutes les directives `Deny` sont évaluées. Si une d'elles correspond, la demande est rejetée. En fin, n'importe quelles demandes qui ne correspondent pas à un `Allow` ou une directive `Deny` sont refusées par défaut.
- **Deny,Allow** : En premier, toutes les directives `Deny` sont évaluées. Si une d'elles correspond, la demande est rejetée à moins qu'elle ne corresponde aussi à une directive `Allow`.

La directive Allow

La directive `Allow` définit quels hôtes peuvent avoir accès à un secteur du serveur. L'accès peut être contrôlé par `hostname`, l'adresse IP, la gamme d'adresse IP, ou selon d'autres caractéristiques de la requête du client. Le premier argument à cette directive est toujours `from`. Les arguments suivants peuvent prendre trois formes différentes.

- **Allow from all** : Pour permettre à tous les hôtes l'accès, soumis à la configuration des directives `Order` et `Deny`.
- **Allow from 'host'** : Pour permettre seulement aux hôtes particuliers ou les groupes d'hôtes d'avoir accès au serveur, l'hôte peut être spécifié dans n'importe lequel des formats suivants :
 - **Un nom de domaine partiel ou non** : `Allow from apache.org`
 - **Une adresse IP partielle ou non** : `Allow from 10.1.2.3` ou `Allow from 10.1`
 - **Un couple réseaux/netmask** : `Allow from 10.1.0.0/255.255.0.0`
 - **Un couple réseaux/nnn CIDR** (nnn= nombre de bit à 1) : `Allow from 10.1.0.0/16`

La directive Deny

Cette directive permet de limiter l'accès au serveur en fonction du `hostname`, de l'adresse IP, ou des variables d'environnement. Les arguments pour la directive `Deny` sont identiques à ceux de la directive `Allow`.

La directive Alias

La directive `Alias` permet aux documents d'être stockés dans le système de fichiers local ailleurs qu'à la racine définie par la directive `DocumentRoot`. Imaginons que nous souhaitons donner accès au répertoire `/home/student/public_html/secret` en utilisant l'url <http://www.bts1.sio/secret/>. Il suffirait pour cela d'ajouter les lignes suivantes dans le fichier `000-default` (qui contiendrait la configuration d'un *virtual host*).

```
Alias /secret/ "/home/student/public_html/secret/"
<Directory "/home/student/public_html/secret/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

La directive ScriptAlias

La directive `ScriptAlias` a le même comportement que la directive `Alias`, mais en plus elle indique la liste des répertoires contenant les scripts CGI qui seront exécutés par le module `mod_cgi`. Il est plus sûr d'éviter de placer des scripts CGI sous la racine `DocumentRoot`. Cela évitera de révéler accidentellement leur code source si jamais la configuration du serveur est changée.

La directive Options

La directive `Options` contrôle que les fonctions du serveur sont disponibles dans un répertoire particulier. `Options` peut être mise à une des valeurs suivantes :

- **None** : dans ce cas on ne permet aucune fonction supplémentaire,
- **All** : toutes les options à part `MultiViews`. C'est la valeur par défaut.
- **ExecCGI** : On permet l'exécution de scripts CGI utilisant `mod_cgi`.
- **FollowSymLinks** : le serveur suivra des liens symboliques dans ce répertoire.
- **Includes** : autorise les inclusions du côté du serveur à l'aide du module `mod_include`.
- **IncludesNOEXEC** : les inclusions du côté du serveur, mais les `#exec cmd` et `#exec cgi` sont désactivés.
- **Indexes** : si une URL correspondant à un répertoire est demandée et qu'il n'y a pas de `DirectoryIndex` (e.g., `index.html`) dans le répertoire, alors le module `mod_autoindex` pourra renvoyer une liste de répertoire. •...

Normalement, si plusieurs directives `Options` sont appliquées à un répertoire, alors la plus spécifique est utilisée et les autres sont ignorées. Cependant si toutes les options de la directive `Options` sont précédées par le symbole `+` ou `-` les options sont fusionnées. N'importe quelle option précédée par un `+` est ajoutée aux options actuellement actives et n'importe quelle option précédée par un `-` est enlevée des options actuellement actives.

- **Attention**, le mélange d'options avec un `+` ou `-` avec des sans symbole n'est pas une syntaxe valable. Il est probable que vous obteniez des résultats inattendus.

Par exemple, sans aucun `+` et `-` symboles où seul l'option `Includes` sera active pour le répertoire `/web/docs/spec`.

```
<Directory /web/docs>
  Options Indexes FollowSymLinks
</Directory>
<Directory /web/docs/spec>
  Options Includes
</Directory>
```

Cependant si la deuxième directive `Options` utilise des symboles `+` et `-` alors les options `FollowSymLinks` et `Includes` seront actives pour le répertoire `/web/docs/spec`.

```
<Directory /web/docs>
  Options Indexes FollowSymLinks
</Directory>
<Directory /web/docs/spec>
  Options +Includes -Indexes
</Directory>
```