

Configuration de VHOST hybride

Adama Tilly Diepkilé, PhD



Un server Apache peut à la fois écouter plusieurs IP/ports et différencier les noms des servers. Il s'agit alors d'un virtual host par IP et par noms.

Adama Telly Diepkilé, PhD



Exemple

Listen 192.168.101.1:80

Listen 192.168.101.2:80

ServerName DefaultServer

DocumentRoot /var/www

NameVirtualHost 192.168.101.1:80

<VirtualHost 192.168.101.1:80>

ServerName thegeekstuff.com

DocumentRoot /var/www1

</VirtualHost>

<VirtualHost 192.168.101.1:80>

ServerName top5freeware.com

DocumentRoot /var/www2

</VirtualHost>

NameVirtualHost 192.168.101.2:80

<VirtualHost 192.168.101.2:80>

ServerName ns.server.net

DocumentRoot /var/www3

</VirtualHost>

Dans cet exemple le serveur écoute 192.168.101.1:80 et 192.168.101.2:80.

Il doit servir les noms **thegeekstuff.com**, **top5freeware.com**,
ns.server.net.

En jouant avec *NameVirtualHost*, il est possible d'indiquer à Apache le premier aiguillage : celui de l'IP/port. Une fois cet aiguillage effectué, Apache va analyser le nom dans l'en-tête *Host*, et l'aiguiller à nouveau dans le bloc *<VirtualHost>* qui le précise avec la directive **ServerName**

Cas 1

Dans cet exemple, le vhost `http://thegeekstuff.com` est supposé être routé dans le DNS vers `192.168.101.1`.

La requête arrive donc vers `192.168.101.1:80` et Apache entre dans la première section *NameVirtualHost*. Il analyse l'hôte demandé dans la requête : il s'agit de `thegeekstuff.com`. Cet dernier étant bien précisé dans le premier bloc *<VirtualHost>*, il sera alors choisi comme second aiguillage, et `/var/www1` sera servi.

Cas 2

En imaginant que `www.server.com` soit routé dans le DNS vers `192.168.101.2`.

Comme dans la section *NameVirtualHost 192.168.101.2:80* il n'existe aucun bloc dont *ServerName* vaut "server4", une telle requête sera aiguillée vers le serveur par défaut qui sera ns.server.net.

Principe de base

- Dans le cas d'un virtual host par IP et par noms,
- La valeur de *NameVirtualHost* doit être systématiquement la même que celle de tous les blocs *<VirtualHost>* qui vont suivre.
- la valeur de *NameVirtualHost* doit être un couple IP et port.
- l'ensemble des directives *NameVirtualHost* se comportent comme dans le cas de l'hébergement par IP : il ne doit pas y avoir ambiguïté pour Apache.

Règles de choix du serveur par défaut lorsque des noms interviennent

- dans le cas d'un serveur faisant intervenir des noms avec (*NameVirtualHost*), le serveur par défaut n'est pas celui déclaré hors des blocs *<VirtualHost>* (comme c'est le cas pour l'hébergement virtuel par IP);
 - le serveur par défaut dans un hébergement virtuel par noms est le premier bloc *<VirtualHost>* qui suit le premier filtre : *NameVirtualHost*;
 - le serveur déclaré hors des blocs *<VirtualHost>* est le serveur par défaut si aucun couple IP/port n'a trouvé de correspondance dans les directives *NameVirtualHost* déclarées.
-
-

Remarques :

- Il a été dit que dans le cas de "Virtual Host par réseau (IP/ports)" qu'il ne faut pas créer de situation ambiguë.
- Dans les virtual hosts par noms, plusieurs blocs *<VirtualHost>* peuvent posséder exactement la même syntaxe, sans que cela ne crée aucune confusion à Apache. Car ces blocs ne seront interrogés que lorsque la directive *NameVirtualHost* du même nom aura trouvé une correspondance vis à vis de la requête en entrée. Cette directive est donc indispensable dans le cas d'un hébergement virtuel à base de nom et elle ne doit pas amener à des situations ambiguës car elle est testée dès l'entrée de la requête.

- Une ou plusieurs IP, un ou plusieurs ports, peu importe : si un vhost basé sur le nom d'hôte est utilisé, *NameVirtualHost* est indispensable.
- Même avec une seule IP, on peut aussi parler "de vhost par IP et par nom".

Adama Tilly Diepkilé, PhD

