tutoriel

Exemples d'utilisation de VirtualHost

Voici quelques scénarios dans lesquels plusieurs serveurs Web doivent tourner sur une seule et même machine au moyen de serveurs virtuels par nom ou par IP.

1. **Plusieurs serveurs virtuels par nom sur une seule adresse IP** : Votre serveur possède plusieurs noms d'hôte pour une seule adresse IP, et vous souhaitez des réponses différentes si on demande www.exemple.com ou www.exemple.org.

   Les noms des serveurs virtuels doivent être définis dans le DNS (leur configuration dans Apache ne provoque pas leur apparition magique dans la configuration du DNS) et être résolus sur l'adresse IP du serveur, faute de quoi personne ne pourra visiter votre site Web.

   On peut ajouter des entrées dans le fichier hosts pour des tests locaux, mais qui ne fonctionneront que sur la machine possédant ces entrées.

```
# Apache doit écouter sur le port 80
Listen 80
<VirtualHost *:80>
    DocumentRoot "/www/exemple1"
    ServerName www.exemple.com
    # Autres directives ici
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "/www/exemple2"
    ServerName www.exemple.org

    # Autres directives ici
</VirtualHost>
```

   Les astérisques correspondent à toutes les adresses, si bien que le serveur principal ne répondra jamais à aucune requête.
   Le serveur virtuel **ServerName www.example.com** est en premier dans le fichier de configuration : il a la plus grande priorité et peut être vu comme **serveur par défaut**. Toute requête ne correspondant à aucune des directives ServerName sera servie par ce premier <VirtualHost>.
   Il faudra utiliser une configuration différente pour servir un contenu différent selon l'adresse IP ou le port.

   Vous pouvez remplacer * par une adresse IP du système.
   Le serveur virtuel concerné ne sera alors sélectionné que

pour les requêtes HTTP vers cette adresse IP.

En général, il est commode d'utiliser * sur les systèmes dont l'adresse IP n'est pas constante - par exemple, pour des serveurs dont l'adresse IP est attribuée dynamiquement par le FAI, et où le DNS est géré au moyen d'un DNS dynamique quelconque.

Comme * signifie n'importe quelle adresse, cette configuration fonctionne sans devoir être modifiée quand l'adresse IP du système est modifiée.

2. **Serveurs virtuels par nom sur plusieurs adresses IP**

   Toutes les techniques présentées ici peuvent être étendues à un plus grand nombre d'adresses IP.

   Le serveur a deux adresses IP. Sur 172.20.30.40, le serveur “principal” server.exemple.com doit répondre, et sur 172.20.30.50, deux serveurs virtuels (ou plus) répondront.

```
Listen 80

<VirtualHost 172.20.30.40>
# Serveur "principal" sur 172.20.30.40
    ServerName server.exemple.com
    DocumentRoot "/var/html/exemple"
    # D'autres directives ici ...
</VirtualHost>

<VirtualHost 172.20.30.50>
    DocumentRoot "/var/html/exemple1"
    ServerName www.exemple.com
    # D'autres directives ici ...
</VirtualHost>

<VirtualHost 172.20.30.50>
    DocumentRoot "/var/html/exemple2"
    ServerName www.exemple.org
    # D'autres directives ici ...
</VirtualHost>
```

   Toute requête arrivant sur une autre adresse que 172.20.30.50 sera servie par le serveur principal. Les requêtes vers 172.20.30.50 avec un nom de serveur inconnu, ou sans en-tête Host:, seront servies par var/html.exemple.com.

3. **Servir le même contenu sur des adresses IP différentes** (telle qu'une adresse interne et une externe) : La machine serveur dispose de deux adresses IP (192.168.1.1 et 172.20.30.40). Cette machine est placée à la fois sur le réseau interne (l'Intranet) et le réseau externe (Internet) : **Sur Internet**, le nom server.example.com pointe vers l'adresse externe

(172.20.30.40) ; **sur le réseau interne**, ce même nom pointe vers l'adresse interne (192.168.1.1).
Le serveur peut être configuré pour répondre de la même manière aux requêtes internes et externes, au moyen d'une seule section <VirtualHost>

```
<VirtualHost 192.168.1.1 172.20.30.40>
    DocumentRoot "/var/www/html/server1"
    ServerName server.example.com
    ServerAlias server
</VirtualHost>
```

Les requêtes en provenance de chacun des deux réseaux seront servies par le même <VirtualHost>.

Sur le réseau interne, il est possible d'utiliser le nom raccourci server au lieu du nom complet server.example.com.

Notez également que dans l'exemple précédent, vous pouvez remplacer la liste des adresses IP par des * afin que le serveur réponde de la même manière sur toutes ses adresses

4. **Servir différents sites sur différents ports** : Vous disposez de plusieurs domaines pointant sur la même adresse IP et vous voulez également servir de multiples ports.
Cet exemple montre que la sélection en fonction du nom intervient après la sélection de la meilleure correspondance du point de vue adresse IP/port.

```
Listen 80
Listen 8080

<VirtualHost 172.20.30.40:80>
    ServerName www.example.com
    DocumentRoot "/www/domain-80"
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
    ServerName www.example.com
    DocumentRoot "/www/domain-8080"
</VirtualHost>

<VirtualHost 172.20.30.40:80>
    ServerName www.example.org
    DocumentRoot "/www/otherdomain-80"
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
    ServerName www.example.org
    DocumentRoot "/www/otherdomain-8080"
```

```
</VirtualHost>
```

5. **Hébergement virtuel basé sur IP** : Le serveur dispose de deux adresses IP (172.20.30.40 et 172.20.30.50) correspondant respectivement aux noms www.example.com et www.example.org.

```
Listen 80

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/example1"
    ServerName www.example.com
</VirtualHost>

<VirtualHost 172.20.30.50>
    DocumentRoot "/www/example2"
    ServerName www.example.org
</VirtualHost>
```

   Les requêtes provenant d'adresses non spécifiées dans l'une des directives <VirtualHost> (comme pour localhost par exemple) seront dirigées vers le serveur principal, s'il en existe un.

6. **Hébergements virtuels mixtes basés sur les ports et sur les IP** : Le serveur dispose de deux adresses IP (172.20.30.40 et 172.20.30.50) correspondant respectivement aux noms www.example.com et www.example.org. Pour chacun d'eux, nous voulons un hébergement sur les ports 80 et 8080.

```
Listen 172.20.30.40:80
Listen 172.20.30.40:8080
Listen 172.20.30.50:80
Listen 172.20.30.50:8080

<VirtualHost 172.20.30.40:80>
    DocumentRoot "/www/example1-80"
    ServerName www.example.com
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
    DocumentRoot "/www/example1-8080"
    ServerName www.example.com
</VirtualHost>

<VirtualHost 172.20.30.50:80>
    DocumentRoot "/www/example2-80"
    ServerName www.example.org
</VirtualHost>

<VirtualHost 172.20.30.50:8080>
    DocumentRoot "/www/example2-8080"
    ServerName www.example.org
```

```
</VirtualHost>
```

7. **Hébergements virtuels mixtes basé sur les noms et sur IP** : Toute adresse indiquée comme argument d'une section VirtualHost et n'apparaissant dans aucun autre serveur virtuel, fait de cette section un serveur virtuel sélectionnable uniquement en fonction de son adresse IP.

```
Listen 80
<VirtualHost 172.20.30.40>
    DocumentRoot "/www/example1"
    ServerName www.example.com
</VirtualHost>

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/example2"
    ServerName www.example.org
</VirtualHost>

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/example3"
    ServerName www.example.net
</VirtualHost>

# IP-based
<VirtualHost 172.20.30.50>
    DocumentRoot "/www/example4"
    ServerName www.example.edu
</VirtualHost>

<VirtualHost 172.20.30.60>
    DocumentRoot "/www/example5"
    ServerName www.example.gov
</VirtualHost>
```

8. **Utilisation simultanée de Virtual_host et de mod_proxy** : L'exemple suivant montre comment une machine peut mandater un serveur virtuel fonctionnant sur le serveur d'une autre machine.
   Dans cet exemple, un serveur virtuel de même nom est configuré sur une machine à l'adresse 192.168.111.2.
   La directive ProxyPreserveHost On est employée pour permette au nom de domaine d'être préservé lors du transfert, au cas où plusieurs noms de domaines cohabitent sur une même machine

```
<VirtualHost *:*>
    ProxyPreserveHost On
    ProxyPass        "/" "http://192.168.111.2/"
    ProxyPassReverse "/" "http://192.168.111.2/"
    ServerName hostname.example.com
</VirtualHost>
```

9. **Utilisation de serveurs virtuels _default_**

1. **Serveurs virtuels _default_ pour tous les ports** : Exemple de capture de toutes les requêtes émanant d'adresses IP ou de ports non connus, c'est-à-dire, d'un couple adresse/port non traité par aucun autre serveur virtuel

```
<VirtualHost _default_:*>
    DocumentRoot "/www/default"
</VirtualHost>
```

L'utilisation d'un tel serveur virtuel avec un joker pour le port empêche de manière efficace qu'une requête n'atteigne le serveur principal.
Un serveur virtuel par défaut ne servira jamais une requête qui est envoyée vers un couple adresse/port utilisée par un serveur virtuel par nom.
Si la requête contient un en-tête Host: inconnu, ou si celui-ci est absent, elle sera toujours servie par le serveur virtuel primaire par nom (celui correspondant à ce couple adresse/port trouvé en premier dans le fichier de configuration).
Vous pouvez utiliser une directive AliasMatch ou RewriteRule afin de réécrire une requête pour une unique page d'information (ou pour un script).

2. **Serveurs virtuels _default_ pour des ports différents** : configuration similaire à l'exemple précédent, mais le serveur écoute sur plusieurs ports et un second serveur virtuel _default_ pour le port 80 est ajouté.

```
<VirtualHost _default_:80>
    DocumentRoot "/www/default80"
    # ...
</VirtualHost>

<VirtualHost _default_:*>
    DocumentRoot "/www/default"
    # ...
</VirtualHost>
```

Le serveur virtuel par défaut défini pour le port 80 (il doit impérativement être placé avant un autre serveur virtuel par défaut traitant tous les ports grâce au joker *) capture toutes les requêtes envoyées sur une adresse IP non spécifiée.
Le serveur principal n'est jamais utilisé pour servir une requête.

3. **Serveurs virtuels _default_ pour un seul port** : Nous voulons créer un serveur virtuel par défaut seulement pour le port 80.

```
<VirtualHost _default_:80>
    DocumentRoot "/www/default"
...
</VirtualHost>
```

Une requête vers une adresse non spécifiée sur le port 80 sera servie par le serveur virtuel par défaut, et toute autre requête vers une adresse et un port non spécifiés sera servie par le serveur principal.
L'utilisation du caractère générique * dans la déclaration d'un serveur virtuel l'emporte sur _default_.

4. **Migration d'un serveur virtuel par nom en un serveur virtuel par IP** : Le serveur virtuel par nom avec le nom de domaine www.example.org (de notre exemple par nom) devrait obtenir sa propre adresse IP. Pendant la phase de migration, il est possible d'éviter les problèmes avec les noms de serveurs et autres serveurs mandataires qui mémorisent les vielles adresses IP pour les serveurs virtuels par nom.
La solution est simple, car il suffit d'ajouter la nouvelle adresse IP (172.20.30.50) dans la directive VirtualHost.

```
Listen 80
ServerName www.example.com
DocumentRoot "/www/example1"

<VirtualHost 172.20.30.40 172.20.30.50>
    DocumentRoot "/www/example2"
    ServerName www.example.org
    # ...
</VirtualHost>

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/example3"
    ServerName www.example.net
    ServerAlias *.example.net
    # ...
</VirtualHost>
```

Le serveur virtuel peut maintenant être joint par la nouvelle adresse (comme un serveur virtuel par IP) et par l'ancienne adresse (comme un serveur virtuel par nom).

Voir aussi

- **(fr)** Exemples d'utilisations de VirtualHost

Basé sur « Exemples d'utilisations de VirtualHost » par apache.org.

From:
https://doc.wikis.frapp.fr/ - **doc**

Permanent link:
https://doc.wikis.frapp.fr/doku.php?id=tutoriel:reseau:http:serveur:apache:vhost:exemples:start

Last update: **2024/09/21 10:12**