

apache2 : serveur Web multi-site

- Objet : Mettre en place un serveur apache2 hébergeant plusieurs sites
- Niveau requis :
[débutant, avisé](#)
- Commentaires : *Vous pouvez héberger différents sites relatifs à différents noms de domaine sur un même serveur apache2, trop facile !*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[obsolète](#)
 - Création par [freddec](#) le 22/03/2011
 - Mis à jour par [bundy](#) le 25/10/2012 (sous squeeze)
 - Mis à jour par [bendia](#) le 11/10/2015 (pour Jessie)
 - Mis à jour par [mercredi](#) le 19/10/2015 (pour Jessie)
- Commentaires sur le forum : [ici](#)¹⁾

Préambule et pré-requis

Ce bref tutorial a deux objectifs :

1. Me servir de pense-bêtes ! (et vous en faire profiter) 😊
2. Expliquer comment mettre en œuvre la gestion multi-site d'Apache2

Tout est parti du besoin que j'ai eu de devoir tester des outils de "création de site" au sein de l'intranet de mon travail.

Il me fallait donc un serveur Apache2 et la possibilité de tester plusieurs sites, un par outil (Divers CMS, Divers Wiki...). Le tout en local.

Or, un coup de Google m'a montré beaucoup des liens dans la langue de Shakespeare, mais bien peu dans celle de Molière et en tous les cas rien qui m'ait beaucoup avancé.

Je vais donc, avec ce tutoriel, essayer de vous (me) faire gagner du temps.

"hère we Go..."

Pré-requis

- Un serveur Apache 2 (+/- PHP, MySQL suivant le type de site à réaliser)

Pour ceux n'ayant pas de serveur LAMP installé :



```
apt-get install apache2 mysql-server libapache2-mod-php5 php5-mysql
```

Et voilà.

- Au moins 2 sites à tester !



Sous Debian 8 (Jessie), Apache2 passe de la version 2.2 à la version 2.4. Il existe des différences de configuration entre ces 2 versions indiquées [dans la documentation d'Apache2](#)

Déclaration des sites existants

Dans Apache2, les sites disponibles (on parle de virtualhosts, ou vhost) sont déclarés à l'aide de fichier dans le répertoire : `/etc/apache2/sites-available/`

Partons du principe que vous souhaitez mettre en ligne 2 sites que nous nommerons **site-1** et **site-2** (pourquoi faire compliqué ? 😊)

Apache vous livre lors de son installation dans ce même répertoire un fichier “default” qui va nous servir de base pour nos 2 sites
On va donc en faire une copie :

Pour Apache2.2 dans Wheezy

```
cp /etc/apache2/sites-available/default /etc/apache2/sites-available/site-1
```

```
cp /etc/apache2/sites-available/default /etc/apache2/sites-available/site-2
```

Pour Apache2.4 dans Jessie

```
cp /etc/apache2/sites-available/default /etc/apache2/sites-available/site-1.conf
```

```
cp /etc/apache2/sites-available/default /etc/apache2/sites-available/site-2.conf
```

Contenu du fichier de vhost

Dans votre éditeur de texte favori (pas un traitement texte attention !) ouvrez le fichier site-1 (ou site-1.conf sous Jessie

Faisons un rapide tour d'horizon des sections qui nous intéressent:

[site-1 ou site-1.conf](#)

```
<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

# Possible values include: debug, info, notice, warn, error,
crit,
# alert, emerg.
LogLevel warn

CustomLog ${APACHE_LOG_DIR}/access.log combined

Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>

</VirtualHost>

```

- `<VirtualHost *:80>` et `</VirtualHost>`: Signalent le début et la fin de la section du vhost, en écoute pour toutes les interfaces/adresses IP (*) sur le port **80**.
- `ServerAdmin webmaster@localhost` : adresse mail où envoyer les messages d'erreur. Devra donc être remplacée par la vôtre
- `ServerName www.blablabla.xxx` : Nom utilisé par le vhost, remplacez-le par le nom de votre site (ici **site-1**)
- `ServerAlias site-1.xxx*.site-1.xxx` : `ServerAlias` définit les autres sous domaines pour lesquels le vhost répondra.
- `DocumentRoot /var/www/htdocs` : Répertoire de stockage du site (sa racine). Vous devrez donc modifier cette directive pour qu'elle pointe sur votre racine (ex: `/home/www/site-1`)

- `DirectoryIndex index.php` : En l'absence de page spécifique demandée par l'internaute, c'est cette page que sera affichée, c'est donc en quelque sorte votre page d'accueil. A modifier donc si votre page d'accueil porte un nom différent (`default.html`, `accueil.php`, etc.)

Voilà pour les directives qui nous préoccupent.



Comme vous le constatez vous même, il en reste d'autre que je n'ai pas détaillées, mais qui sont importantes notamment pour la sécurité. Je vous invite donc à consulter cette page pour savoir plus : <http://doc.ubuntu-fr.org/apache2>

Fichier vhost adapté

Voyons voir a quoi pourrait ressembler notre fichier une fois adapté à nos besoins:

Pour Apache2.2 dans Wheezy

site-1

```
<VirtualHost *:80>
    ServerAdmin MonMail@gmail.com
    ServerName      www.site-1.fr
    ServerAlias     site-1.fr
    ServerAlias     *.site-1.fr

    DocumentRoot /home/www/site-1
    DirectoryIndex  index.php
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>

    <Directory /home/www/site-1>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ErrorLog /home/www/apache_log/error_site-1.log

    # Possible values include: debug, info, notice, warn, error,
crit,
    # alert, emerg.
    LogLevel warn

    CustomLog /home/www/apache_log/access_site-1.log combined
```

```
</VirtualHost>
```

Pour Apache2.4 dans Jessie

site-1.conf

```
<VirtualHost *:80>
    ServerAdmin MonMail@gmail.com
    ServerName      www.site-1.fr
    ServerAlias     site-1.fr
    ServerAlias     *.site-1.fr

    DocumentRoot /home/www/site-1
    DirectoryIndex index.php
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>

    <Directory /home/www/site-1>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Require all granted
    </Directory>

    ErrorLog /home/www/apache_log/error_site-1.log

    # Possible values include: debug, info, notice, warn, error,
crit,
    # alert, emerg.
    LogLevel warn

    CustomLog /home/www/apache_log/access_site-1.log combined
</VirtualHost>
```

Renouveler les opérations pour créer un fichier de vhost pour site-2

Activation/Désactivation des sites

Une fois les fichiers de vhost créés, il reste à les activer pour qu'Apache2 les prenne en compte.

Ce mécanisme permet d'avoir tout les fichiers de vhosts que l'on veut et de n'activer que ceux nécessaires.

Deux méthodes possible:

- *Méthode Classique*: Création/suppression d'un lien symbolique dans `/etc/apache2/sites-enabled` pointant vers `/etc/apache2/sites-available/site-1`

```
ln -s /etc/apache2/sites-available/site-1 /etc/apache2/sites-enabled
```

- *Méthode Debian* : les Développeurs Debian, qui font bien les choses (sinon, ils ne seraient pas DD !), ont mis en place un outil très simple:

```
a2ensite site-1
```

...et comme ils ne font pas les choses à moitié, ils ont aussi prévu la désactivation

```
a2dissite site-1
```

Adaptation du fichier `/etc/hosts`



Cette opération n'est nécessaire qu'à fin de tester votre installation en local, si votre serveur et votre client sont sur la même machine, ou si vous ne disposez pas de plusieurs noms de domaine.

Éditez votre fichier `/etc/hosts` et modifiez le comme suit pour que votre machine fasse le rapport entre vos vhost et le localhost.

[/etc/hosts](#)

```
127.0.0.1    localhost
127.0.0.1    site-1
127.0.0.1    site-2
```

Relancer le serveur apache2 et tester

Relancer apache2

Voir : [invoke-rc.d](#)

```
service apache2 restart
```

Testez vos sites

Ouvrez votre navigateur et testez en tapant dans la barre d'adresse:

- <http://site-1/> puis <http://site-2/>

Vous devriez voir apparaître les pages d'accueil des site-1 et site-2.

Astuce - Apache en local

Dans le cas d'une utilisation du serveur à des fins de test ou d'un développement en local, il peut-être pratique de pouvoir modifier tous les fichiers de son site sans s'empêtrer avec les problèmes de droits Root/User.

Pour cela on crée un [lien symbolique](#) depuis le répertoire du serveur vers le répertoire de son dossier personnel.

Par exemple on place les fichiers de site-1 dans : **/home/utilisateur/dev/site-1**, puis on crée un lien symbolique comme ceci :

```
ln -s /home/utilisateur/dev/site-1 /var/www/site-1
```

Liens

Pour approfondir les hôtes virtuels sur un serveur apache :

- http://doc.ubuntu-fr.org/tutoriel/virtualhosts_avec_apache2

Mettre en place une redirection avec apache :

- [La redirection d'URL pour les débutants](#)

¹⁾

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:reseau:apache2:multisite>

Last update: **02/06/2023 23:31**

