

Trusty, BROUILLOON

Apache 2 : un serveur web HTTP

Pré-requis

- Disposer d'un utilisateur non root
- Quelques connaissances de base des protocoles HTTP, DNS et TCP/IP (adressage, sous-réseaux, etc.) ;
- Un réseau local en état de marche (IP attribuées, connexion à internet) ;
- Un serveur DNS configuré (pour le domaine).

L'arborescence du site se présente comme suit :

RACINE

└ www

 └ html

 └ site 1

 └ site 2

 └ ...

Installation

[Mise à jour de la version 2.2 vers la version 2.4](#)

Configuration

Apache2 se fonde sur les hôtes virtuels qui sont indispensables.

Il faut créer un hôte virtuel (*VirtualHost*) pour chaque projet.



Il ne faut pas lancer les sites locaux par `http://localhost/projet1/` ou `http://localhost/projet2/` mais par un hôte virtuel pour chacun : **http://projet1/** ou **http://projet2/**

[Apache : configurer les hôtes virtuels](#)

[Apache 2 : sous-domaines \(serveurs virtuels\)](#)

VirtualHost

Les fichiers de configuration sont maintenant des hôtes virtuels placés dans **/etc/apache2/sites-available** et activés par **a2ensite**.

 Les fichiers `.htaccess` doivent être si possible remplacés par des blocs `<directory>...</directory>` dans les fichiers de configuration des vhosts.

Ne les utiliser que si on n'a aucun accès à la config du serveur (par exemple sur un hébergement mutualisé)

Le fichier de configuration principal (anciennement `httpd.conf`) est remplacé par le fichier **/etc/apache2/sites-available/000-default.conf**

Pour les options et les directives, voir la page [Configuration de Apache 2 Apache 2 : options et directives de configuration](#)

Préparation du fichier `/etc/hosts`

Le fichier **/etc/hosts** donne les correspondances entre les adresses IP et les noms de site.

Une petite modification est nécessaire pour un bon fonctionnement de Apache.

Ouvrez avec les droits d'administration le fichier **/etc/hosts** pour le modifier comme ceci :

| changez les deux lignes | en la ligne |
|----------------------------|---|
| 127.0.0.1 localhost | 127.0.0.1 localhost.localdomain localhost |
| ... | |
| 127.0.0.1 nom_réseau_du_pc | nom_réseau_du_pc |

Relancez le serveur apache par :

- `sudo service apache2 restart`

Les hôtes virtuels

Activez les hôtes virtuels en lançant en ligne de commande :

```
$ sudo a2enmod vhost_alias
Enabling module vhost_alias.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Pour créer un **VirtualHost** Apache, deux étapes :

Créez et activez la définition du **VirtualHost**

Ajoutez votre nouveau nom de domaine dans le fichier **/etc/hosts**.



Dans nos exemples, nous parlerons du site **monsite.com** et pour les sous-domaines du site **doc.monsite.com**

Préalable : l'hôte localhost

Pour des raisons de sécurité, il ne faut pas laisser les visiteurs non enregistrés circuler dans toute la machine.

Créez avec les droits d'administration le fichier **/etc/apache2/sites-available/000-localhost.conf** pour y écrire ceci :

</etc/apache2/sites-available/000-localhost.conf>

```
<VirtualHost *:80>
    ServerName localhost
    ServerAlias localhost
    DocumentRoot /var/www/html
    <Directory /var/www/html>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride all
        Require local
    </Directory>
</VirtualHost>
```

Options

Indexes

Génère automatiquement un index si aucun n'est défini pour ce répertoire

FollowSymLinks

Le serveur va suivre les liens symboliques dans le répertoire concerné

MultiViews

vues multiples autorisées

Require local

n'autorise que les connexions en provenance de l'hôte local

Cette option remplace les anciennes options *order* et *allow*





AllowOverride all

Active le mécanisme .htaccess

? Directive Require

Require ip

contrôle l'accès au serveur en fonction de l'adresse IP du client distant
L'adresse ip peut être une ip complète, partielle ou une paire réseau/masque de sous-réseau
exemple : Require ip 10 172.20 192.168.2

Require host

contrôle l'accès au serveur en fonction du nom d'hôte du client distant\\Ce nom peut être un nom de domaine (éventuellement partiel)
Exemples :
Require host example.org
Require host .net example.edu

Require local



ne permet l'accès au serveur qu'aux clients locaux (l'adresse IP du client correspond à 127.0.0.0/8 ou ::1 ou les adresses IP du client et du serveur sont identiques)

Require all [granted|denied]

Accès autorisé ou refusé dans tous les cas

Utilisateurs

Require user identifiant utilisateur [identifiant utilisateur] ...

Seuls les utilisateurs spécifiés sont autorisés!!

Require group nom groupe [nom groupe] ...

Seuls les utilisateurs appartenant aux groupes spécifiés sont autorisés

Require valid-user

Tous les utilisateurs valides sont autorisés

Pour activer le vhost :

```
...@...:~ $ sudo a2enmod 000-localhost
Enabling module 000-localhost
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Nous allons maintenant décrire les étapes pour créer l'hôte virtuel du site **monsite1**.

Première étape : création et activation de la définition du VirtualHost

Créez avec les droits d'administration le fichier “**/etc/apache2/sites-available/monsite.com.conf**” pour y écrire ceci :

</etc/apache2/sites-available/monsite.com.conf>

```
<VirtualHost *:80>
    ServerName www.monsite.com
    ServerAlias monsite.com *.monsite.com
    ServerAdmin administrateur.web@domain.tld
    DocumentRoot /srv/www/html/monsite1
    <Directory /srv/www/html/monsite1>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride all
        Order allow,deny
        allow from all
    </Directory>
    ErrorLog /var/log/apache2/error.monsite1.conf.log
    CustomLog /var/log/apache2/access.monsite1.conf.log combined
</VirtualHost>
```

 ? ServerName :: Nom du vhost = nom du domaine !!
 ? ServerAlias :: Autres sous domaines pour lesquels le serveur répondra !!
 ? ServerAdmin :: email à utiliser en cas de problème !!
 ? DocumentRoot :: Répertoire racine où seront stockés les fichiers du site !!
 ? <Directory /srv/www/html/monsite1> :: options par défaut du répertoire !!

Pour activer le vhost, lancez la commande :

- `sudo a2ensite monsite1`

Deuxième étape : ajout dans le fichier **/etc/hosts**

Nous devons maintenant ajouter dans le fichier **/etc/hosts** les noms de domaine des hôtes virtuels que nous avons définis.

Ouvrez avec les droits d'administration le fichier **"/etc/hosts"**.

Pour chaque hôte virtuel, insérez une ligne avec :

IP nom_de_l_hote



On peut regrouper jusqu'à 8 noms de serveurs maximum sur une même ligne pour la même IP.

On obtient quelque-chose comme :

[/etc/hosts](#)

```
# pour localhost
127.0.0.1 localhost projet1
::1 localhost projet1

" pour les sites sur localhost
127.0.0.1 monsite1.com doc.monsite1.com monsite2

# sites sur une autre machine du réseau
192.0.0.12 site12.tld

# machines extérieures
123.456.7.8 site.fr
```

Les hôtes virtuels dynamiques

les Virtualhosts dynamiques permettent une gestion beaucoup plus simple que les virtualhosts simples quand il faut gérer une grande quantité de sites.

De plus, il n'est pas nécessaire de relancer Apache après l'ajout d'un site web par exemple.

Avec cette méthode, pour créer un domaine ou un sous-domaine, il suffit de créer dans la racine un répertoire racine portant le même nom que le domaine ou le sous-domaine voulu.

Par exemple, pour créer le site « la-maman-de.marcel.com », il suffit de créer un répertoire **la-maman-de.marcel.com** dans **/var/www/**.

Il n'y a rien d'autre à faire !! En tapant <http://la-maman-de.marcel.com> dans un navigateur vous y accédez directement.

Voici un exemple de VirtualHost dynamique :

```
<VirtualHost *:80>
    ServerName www.loicg.dev.com
    ServerAlias *.*.dev.com
    VirtualDocumentRoot /home/web/workspace/%2/%1
```

```

<Directory />
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
</Directory>
</VirtualHost>

```

```

NameVirtualHost *:80
<VirtualHost *:80>
    VirtualDocumentRoot /var/www/%0
    VirtualScriptAlias /var/www/%0/cgi-bin

    LogFormat "%V %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i
\" \"%{forensic-id}n\" %v" combinedvhost
    CustomLog "/var/log/apache2/access.log" combined
    CustomLog "| /usr/sbin/split-logfile" combinedvhost
</VirtualHost>

```

Explications

?LogFormat "%V %h %l %u %t \"%r\" %>s %b
 \"%{Referer}i\" \"%{User-Agent}i\" \"%{forensic-id}n\" %v"
 combinedvhost ::Format de log pour les virtualhost
 dynamiques, l'option la plus importante ici est le %V écrit
 l'adresse d'accès au site en début de ligne et qui permet
 donc de découper les logs par site web. Nous verrons ça un
 peu plus bas. Le %V doit être placé au début.!!

?CustomLog "/var/log/apache2/access.log" combined
 ::Permet d'avoir un référencement de tous les accès au
 serveur web dans un seul fichier, pratique pour des
 statistiques globales par exemple. Attention ce fichier peut
 devenir assez gros.!!



?CustomLog "| /usr/sbin/split-logfile" combinedvhost ::Ici
 c'est une directive un peu spéciale, elle fait appelle à la
 commande split-logfile pour séparer les logs.!!

?VirtualDocumentRoot /var/www/%0 ::C'est ici que
 commence le vrai pouvoir des virtualhosts dynamiques,
 cette variable indique le chemin vers les répertoires
 contenant les sites web. Jusque-là rien de bien nouveau mais
 le c'est le %0 qui change toute la donne, en effet %0 sera
 égal l'adresse du site tapée dans la barre d'adresse de votre
 navigateur. Si le répertoire égal à l'adresse du site n'existe
 pas alors une erreur 404 apparaîtra.!!

?VirtualScriptAlias /var/www/%0/cgi-bin ::Si vos souhaitez
 utiliser des scripts .py .pl .cgi alors il vous faut cette ligne.

Les scripts devront se trouver dans le répertoire
/var/www/marcel.com/cgi-bin/ pour qu'ils puissent être
exécutés.!!

Interpolation du nom de répertoire

Toutes les directives de ce module insèrent une chaîne dans un nom de chemin. La chaîne insérée (que nous appellerons maintenant le "nom") peut être soit le nom du serveur (voir la directive UseCanonicalName pour les détails sur la manière dont il est déterminé), soit l'adresse IP du serveur virtuel hébergé par le serveur sous la forme d'un quadruplet d'octets séparés par des points. L'insertion est contrôlée par des spécificateurs inspirés de printf et possèdant de nombreux formats :

?%:::insère un %!!
?%p:::insère le numéro de port du serveur virtuel!!
?%N.M:::insère le nom (en partie)!! N et M permettent de spécifier des sous-chaînes du nom. N sélectionne un des composants du nom séparés par des points, et M sélectionne des caractères à l'intérieur de ce que N a sélectionné. M est optionnel et sa valeur par défaut est 0 s'il n'est pas spécifié ; le point doit être présent si et seulement si M l'est aussi. Les modes d'insertion sont les suivants :

?0::le nom en entier!!
?1::la première partie!!
?2::la seconde partie!!
?-1::la dernière partie!!
?-2::l'avant-dernière partie!!
?2+::toutes les parties à partir de la seconde!!
?-2+::toutes les parties jusqu'à l'avant-dernière!!
?1+ et -1+::identique à 0!! Si N ou M est plus grand que le nombre de parties disponibles, seul un caractère de soulignement est inséré.



DNS : voir
<http://mayakron.altervista.org/wikibase/show.php?id=AcrylicHome> et la mise en place sous windows 10.

et la réponse
<http://stackoverflow.com/questions/138162/wildcards-in-a-windows-hosts-file>



Relancez Apache par la commande :

- `sudo service apache2 restart`



Utilisation

[apachectl - Interface de contrôle du serveur HTTP](#)

[Apache](#)

Lancement :

```
...@...:~ $ sudo apachectl start
```

Arrêt en douceur :

```
...@...:~ $ sudo apachectl graceful-stop
```

Arrêt immédiat :

```
...@...:~ $ sudo apachectl stop
```

Redémarrage en douceur :

```
...@...:~ $ sudo apachectl graceful
```

Redémarrage immédiat :

```
...@...:~ $ sudo apachectl restart
```

Vérification de la syntaxe du fichier de configuration :

```
...@...:~ $ sudo apachectl configtest
```

Rapport d'état :

```
...@...:~ $ sudo systemctl status apache2
```

Désinstallation

**Voir aussi**

- [\(fr\) http://httpd.apache.org/docs/2.4/](http://httpd.apache.org/docs/2.4/)
- [\(fr\) https://www.vincentliefoghe.net/content/configuration-apache-24-php-fpm](https://www.vincentliefoghe.net/content/configuration-apache-24-php-fpm) (très clair, à jour)
- [\(fr\) https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04](https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04)

Basé sur <[Titre original de l'article](#)> par <[Auteur Original](#)>.

From:
<https://nfrappe.fr/doc-0/> - Documentation du Dr Nicolas Frappé

Permanent link:
<https://nfrappe.fr/doc-0/doku.php?id=logiciel:internet:apache:start>

Last update: **2022/08/13 21:57**

