eBook Gratuit

APPRENEZ apache

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#apache

Table des matières

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: apache

It is an unofficial and free apache ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official apache.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec apache

Remarques

Cette section fournit une vue d'ensemble de ce qu'est apache et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans Apache, et établir un lien avec les sujets connexes. La documentation pour Apache étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Versions

Diverses versions d'Apache httpd

Version	Version actuelle	Libération
1.3	1.3.42	1998-06-06
2.0	2.0.65	2002-04-06
2.2	2.2.32	2005-12-01
2.4	2.4.25	2012-02-21

Examples

Installation ou configuration

Instructions détaillées sur la configuration ou l'installation d'Apache.

Installation d'Ubuntu

```
sudo apt-get install apache2
```

Installation de Windows

Découvrez la pile WAMP . WAMP signifie Windows, Apache, MySQL, PhpMyAdmin.

Installation CentOS

Apache 2.2 est fourni avec CentOS6, tandis que 2.4 est fourni avec CentOS7, pour s'installer sur les deux systèmes d'exploitation, exécuter

```
yum -y install httpd
```

Installation de macOS

macOS est livré avec Apache pré-installé, cependant, peut installer Apache via Homebrew

Si vous avez déjà Apache intégré en cours d'exécution, vous devez d'abord l'éteindre et supprimer tous les scripts à chargement automatique.

```
$ sudo apachectl stop
$ sudo launchctl unload -w /System/Library/LaunchDaemons/org.apache.httpd.plist 2>/dev/null
$ brew install httpd24 --with-privileged-ports --with-http2
```

[Ubuntu] Exemple simple de Hello World

Cet exemple vous guidera dans la configuration d'un serveur principal servant de page HTML Hello World.

Installation des exigences

La commande compte pour cette étape!

- `sudo apt-get install apache2`

Mise en place du HTML

Les fichiers Apache vivent dans `/var/www/html/` . Permet d'y arriver rapidement. Assurez-vous que vous êtes dans votre répertoire racine en premier, `cd` , puis `cd /var/www/html/` .

Ce répertoire `html` est l'endroit où tous les fichiers de votre site Web vivront. Permet de créer rapidement un simple fichier Hello World.

En utilisant votre éditeur de texte préféré, tapez ce qui suit dans

```
<!DOCTYPE html>
<html>
<head>
    <title>Hello World!</title>
</head>
<body>
    <h1>Hello World!</h1>
</body>
</html>
```

Enregistrez ce fichier sous le nom `index.html` dans le répertoire en cours et vous êtes prêt à partir!

Visiter votre page Web

Pour visiter la page que vous venez de créer, dans le navigateur de votre choix, rendez-vous sur `localhost` . Si cela ne fonctionne pas, essayez `127.0.0.1` . Vous devriez voir "Hello World!" comme un `h1` . Vous avez terminé!

Pour vous assurer que le serveur est opérationnel.

Si vous recevez un message indiquant que le navigateur ne peut pas se connecter au serveur, vérifiez d'abord que le serveur est opérationnel.

```
$ ps -aef | grep httpd
```

Vous devriez voir quelques processus httpd si Apache est opérationnel.

Lire Démarrer avec apache en ligne: https://riptutorial.com/fr/apache/topic/964/demarrer-avec-apache

Chapitre 2: Apache Flume

Introduction

`Apache Flume` est un mécanisme d'ingestion d'outils, de services et de données permettant de collecter et de transporter de grandes quantités de données en continu telles que des fichiers journaux, des événements (etc ...) provenant de diverses sources vers un **magasin de données centralisé** .

Flume est un outil extrêmement fiable, distribué et configurable. Il est principalement conçu pour copier des données en continu (données de journal) de divers serveurs Web vers `HDFS` .

Examples

Streaming / Log Data

Généralement, la plupart des données à analyser seront produites par diverses sources de données telles que des serveurs d'applications, des sites de réseaux sociaux, des serveurs cloud et des serveurs d'entreprise. Ces données seront sous la forme de fichiers journaux et d'événements.

Fichier journal - En général, un fichier journal est un fichier répertoriant les événements / actions qui se produisent dans un système d'exploitation. Par exemple, les serveurs Web répertorient toutes les requêtes effectuées sur le serveur dans les fichiers journaux.

Lors de la collecte de ces données de journal, nous pouvons obtenir des informations sur -

les performances de l'application et localiser les diverses défaillances logicielles et matérielles. le comportement des utilisateurs et obtenir de meilleures perspectives commerciales. La méthode traditionnelle de transfert de données dans le système HDFS consiste à utiliser la commande put. Voyons comment utiliser la commande put.

Lire Apache Flume en ligne: https://riptutorial.com/fr/apache/topic/9630/apache-flume

Chapitre 3: Comment créer un hôte virtuel dans Apache

Remarques

Le principal point d'accès à `VirtualHost` d'Apache est la documentation d'Apache Virtual Host . De là, vous disposez d'une documentation générale sur la configuration de l'hôte virtuel et de la documentation de référence sur `VirtualHost` et les directives associées.

Examples

Configuration de l'hôte virtuel basé sur le nom

L'hébergement virtuel basé sur nom sur Apache est décrit sur le site Web Apache en tant que tel:

> Avec l'hébergement virtuel basé sur le nom, le serveur s'appuie sur le client pour signaler le nom d'hôte dans le cadre des en-têtes HTTP. Grâce à cette technique, de nombreux hôtes différents peuvent partager la même adresse IP.

Par conséquent, plusieurs sites Web peuvent être hébergés sur un serveur via cette méthode. Sur Ubuntu, les fichiers de configuration se trouvent dans / etc / apache2 / sites-available. Dans ce répertoire, vous trouverez 000-default.conf. C'est la configuration par défaut, toutes les demandes seront envoyées à ce fichier de configuration jusqu'à ce que d'autres soient configurées.

Pour configurer un hôte virtuel, ici **example.com** sera utilisé, mais vous devez le remplacer par votre **domaine.com** . Copiez le fichier par défaut:

```
cp 000-default.conf example.com.conf
```

Le fichier de configuration peut avoir les directives suivantes:

```
<VirtualHost *:80>
        ServerAdmin admin@example.com
        ServerName example.com
        ServerAlias www.example.com

        DocumentRoot /var/www/example.com/html

        ErrorLog /var/log/apache/logs/error.log
        # Possible values include: debug, info, notice, warn, error, crit,
        # alert, emerg.
        LogLevel warn

        CustomLog /var/log/apache/logs/access.log combined
</VirtualHost>
```

- La première ligne indique que toutes les requêtes sur le port 80 (port http par défaut) doivent

correspondre. Vous pouvez également avoir une adresse IP au lieu de *, qui est l'adresse IP du serveur.
- `ServerAdmin` est les coordonnées de l'administrateur du site Web utilisé pour afficher les messages d'erreur http.
- `ServerName` est le nom de domaine du site Web.
- `ServerAlias` est un nom secondaire du site, généralement www.domain.com
- `DocumentRoot` est le dossier racine chargé lorsque nous parcourons un site Web.
- `ErrorLog` est le fichier dans lequel les erreurs sont dirigées
- `LogLevel` . est le niveau d'erreurs à envoyer au journal
- `CustomLog` est le fichier où les informations d'accès sont dirigées

Modifiez le fichier en remplaçant example.com par le nom de domaine de votre site Web et le répertoire approprié pour les fichiers du site Web.

Enregistrez le fichier et activez le site avec la commande Apache suivante:

```
sudo a2ensite example.com.conf
```

Recharger apache

```
sudo service apache2 reload
```

Quelques autres choses à vérifier:

- Assurez-vous que votre DNS pour votre domaine est configuré pour la bonne adresse IP (cela peut prendre du temps à se propager)
- Assurez-vous que votre port 80 est ouvert sur le pare-feu
- Assurez-vous que vos autorisations de fichiers sont correctement configurées sur les fichiers du serveur - la propriété doit être www-data: les autorisations de données et de répertoire de www doivent être 750 et les autorisations de fichier doivent être 640.

Votre hôte virtuel devrait être opérationnel! Vous pouvez répéter ceci pour d'autres sites Web sur le même serveur, avec un fichier de configuration différent (utilisant la même convention de dénomination) et différents répertoires sous / var / www.

PHP Virtual Host Host

Ceci est un exemple sur la façon de contrôler la journalisation des erreurs PHP sur un site hôte virtuel pour le développement et le débogage. Hypothèses

- Le module PHP a été installé.
- L'environnement de développement n'est pas destiné à la production.

```
<VirtualHost *:80>
    ServerName example.com
    DocumentRoot /var/www/domains/example.com/html
    ErrorLog /var/www/domains/example.com/apache.error.log
    CustomLog /var/www/domains/example.com/apache.access.log common
    php_flag log_errors on
```

```
    php_flag display_errors on
    php_value error_reporting 2147483647
    php_value error_log /var/www/domains/example.com/php.error.log
</VirtualHost>
```

Remarque: la configuration de l'hôte virtuel est destinée au développement uniquement parce que display_errors est activé et que vous ne le souhaitez pas en production.

Hôte virtuel dans WAMP

En supposant que vous travaillez avec Windows 7 PC

Étape 1: GOTO -> C: \ Windows \ System32 \ drivers \ etc Où vous trouverez un fichier nommé «hosts», copiez-le et collez-le au même endroit. Un fichier de copie des hôtes y sera créé.

Maintenant, nous devons apporter des modifications à ce fichier, mais si vous essayez de le modifier avec un éditeur tel que notepad ou notepad ++, cela ne vous permettra pas d'enregistrer le fichier.

Maintenant, copiez à nouveau le même fichier et collez-le sur votre bureau, vous pouvez maintenant modifier ce fichier facilement.

Vous trouverez une ou plusieurs entrées comme: 127.0.0.1 localhost Dans ce fichier. Ajoutez maintenant une autre ligne au-dessous de cette ligne, par exemple: 127.0.0.1 myproject1.local De cette manière, vous avez défini un nouveau sous-domaine «myproject1.local» qui peut remplacer «localhost / myproject1».

Étape 2: OK, il est maintenant temps de définir le chemin d'accès racine pour accéder à ce domaine nouvellement créé, n'est-ce pas? GOTO: C: \ wamp \ bin \ apache \ Votre-Apache-Version \ conf \ extra Vous trouverez ici un fichier nommé "httpd-vhosts". Ouvrez-le dans l'éditeur et collez-y les lignes ci-dessous.

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy.example.com
    DocumentRoot "c:/wamp/www/myproject1/”
    ServerName myproject1.local
    ErrorLog "logs/myproject1.local-error.log"
    CustomLog "logs/myproject1.local.log" common
</VirtualHost>
```

Maintenant, vous êtes presque là pour accéder au projet qui réside à «c: / wamp / www / myproject1 /»

Etape 3: GOTO: C: \ wamp \ bin \ apache \ votre-Apache-Version \ conf

Recherchez un fichier nommé **"httpd.conf"** , copiez-le et collez-le au même endroit pour plus de sécurité. Ouvrez le fichier dans l'éditeur et trouvez un mot **«# Hôtes virtuels»** . Vous trouverez ci-dessous une ligne **«Inclure conf / extra / httpd-vhosts.conf»** .

Accédez à votre navigateur Web et écrivez myproject1.local, vous pouvez voir le projet en cours

d'exécution maintenant.

Maintenant, vous pourriez rencontrer un problème que votre localhost ne fonctionnera pas en utilisant localhost comme URL. Pas de soucis… collez ce code dans le fichier **“httpd-vhosts”** .

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy.example.com
    DocumentRoot "c:/wamp/www"
    ServerName localhost
    ErrorLog "logs/localhost-error.log"
    CustomLog "logs/localhost.log" common
</VirtualHost>
```

Redémarrez tous les services de WAMP, le travail est terminé.

Merci et acclamations **Chintan Gor**

1) Vhosts basés sur IP 2) Plusieurs vhosts avec le même port 3) Définition de vhosts avec Macro (Apache2.4)

1) Vhosts basés sur IP

```
 <VirtualHost 192.168.13.37>
     ServerName example.com
     DocumentRoot /var/www/domains/example.com/html
     ErrorLog /var/log/example.com/error.log
     CustomLog /var/log/example.com/access.log common
 </VirtualHost>

 <VirtualHost 192.168.47.11>
     ServerName otherurl.com
     DocumentRoot /srv/www/htdocs/otherurl.com/html
     ErrorLog /var/log/otherurl.com/error.log
     CustomLog /var/log/otherurl.com/access.log common
 </VirtualHost>
```

Il suffit de changer le port pour votre adresse IP donnée. Le port n'est pas pertinent pour la décision de choisir vhost.

2) plusieurs hôtes virtuels avec le même port

Comme NameVirtualHost n'est plus nécessaire, vous pouvez simplement écrire plusieurs hôtes virtuels avec le même port.

```
<VirtualHost *:80>
     DocumentRoot /srv/www/htdocs/otherurl.com/html
     ErrorLog /var/log/otherurl.com/error.log
     CustomLog /var/log/otherurl.com/access.log common
</VirtualHost>

<VirtualHost *:80>
     ServerName example.com
     ServerAlias ex1.com ex2.com
     DocumentRoot /var/www/domains/example.com/html
```

```
    ErrorLog /var/log/example.com/error.log
    CustomLog /var/log/example.com/access.log common
 </VirtualHost>
```

Ici, l'inverse s'applique: l'adresse IP n'est pas pertinente, mais si la demande est reçue sur le port 80, le nom que vous avez entré est évalué. Avez-vous appelé ex1.com le 2e vhost est choisi. Et si vous appelez une autre URL (comme otherurl.com, mais aussi example3.com), la première sera choisie. Vous pouvez utiliser ce vhost comme un «repli» si vous voulez.

3) Définition de vhosts en utilisant la macro (Apache2.4)

```
<Macro VHost $port $host>
    <VirtualHost *:$port>
        Servername $host
        DocumentRoot /srv/www/htdocs/$host
        ErrorLog /var/log/$host/error.log
    </VirtualHost>
</Macro>

Use VHost 80 example.com
Use VHost 443 secure_example.com
```

Crée deux hôtes virtuels, un pour le port 80, un pour 443, et définit les variables utilisées en conséquence.

Forcer HTTPS en utilisant un hôte virtuel

Utilisez la **redirection** pour forcer les utilisateurs à se connecter à l'URL sécurisée.

```
<VirtualHost *:80>
    ServerName example.com
    SSLProxyEngine on
    Redirect permanent / https://secure_example.com/
</VirtualHost>
```

Le reste de la configuration peut être placé dans l'hôte virtuel ssl (port 443) car tout est redirigé.

```
<VirtualHost _default_:443>
    ServerName secure_example.com
    ServerAdmin webmaster@example.com
    DocumentRoot /var/www/domains/secure_example.com/html
    ErrorLog /var/log/secure_example.com/error.log
    CustomLog /var/log/secure_example.com/access.log common
    SSLEngine On
    ...
</VirtualHost>
```

Lire Comment créer un hôte virtuel dans Apache en ligne:
https://riptutorial.com/fr/apache/topic/4856/comment-creer-un-hote-virtuel-dans-apache

Chapitre 4: Fichiers .htaccess dans Apache

Examples

Réécrire le moteur

Le module RewriteEngine d'Apache est utilisé pour réécrire dynamiquement les URL et les chemins en fonction des différentes expressions fournies:

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [END]
</IfModule>
```

Les règles ci-dessus réécrivent les fichiers PHP pour ne plus afficher leur extension, et pour que index.php apparaisse simplement comme un domaine nu (similaire au comportement normalement vu dans index.html). La règle ci-dessus est fournie avec WordPress.

Notez que dans Apache httpd 2.2.16 et versions ultérieures, ce bloc entier peut être remplacé par une seule ligne à l'aide de la directive FallbackResource:

```
FallbackResource /index.php
```

Force HTTPS

.htaccess peut être utilisé pour forcer votre site HTTP à rediriger vers HTTPS.

Voici un moyen rapide de ne pas modifier le code de votre domaine:

```
RewriteEngine On
RewriteCond %{HTTPS} =off
RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

> **Avertissement:** le code ci-dessus suppose que vous pouvez faire confiance à `%{HTTP_HOST}` pour qu'il pointe vers votre domaine.
>
> Si vous devez vous assurer que l'emplacement de la redirection est votre domaine, remplacez `%{HTTP_HOST}` par votre domaine.

Le code ci-dessus fait ceci:

1. Activer RewriteEngine .
2. Continuer si la demande en cours n'utilise pas HTTPS.
3. Effectuez une redirection HTTP 301 vers `https://%{HTTP_HOST}%{REQUEST_URI}` , où

- `%{HTTP_HOST}` est l'hôte demandé par le navigateur et
- `%{REQUEST_URI}` est l'URI demandé par le navigateur (tout après le domaine).

Avertissement: Votre application Web doit être capable de gérer les demandes HTTPS et Apache pour votre hôte doit être configuré avec un certificat de site valide.

Notez qu'il est beaucoup plus efficace de simplement faire une `Redirect` dans http vhost que de faire ces multiples comparaisons par demande dans un fichier .htaccess. Voir http://wiki.apache.org/httpd/RedirectSSL pour plus de détails sur cette technique.

Activer CORS

Pour activer le ***partage de ressources inter-origines*** (**CORS**) dans Apache, vous devez définir au moins un en-tête HTTP qui le modifie (le comportement par défaut consiste à bloquer CORS). Dans l'exemple suivant, nous allons définir cet en-tête HTTP dans `.htaccess` , mais il peut également être défini dans votre fichier site `your-site.conf` ou dans le fichier de configuration Apache. Quelle que soit l'apparence de votre configuration, vous pouvez définir les en-têtes HTTP appropriés dans tout bloc de configuration Apache, à savoir `<VirtualHost>` , `<Directory>` , `<Location>` et `<Files>` .

Il existe quelques en-têtes HTTP liés à CORS que vous pouvez renvoyer dans la réponse:

```
Access-Control-Allow-Origin

Access-Control-Allow-Credentials
Access-Control-Allow-Methods
Access-Control-Max-Age
Access-Control-Allow-Headers
Access-Control-Expose-Headers
```

Certains des éléments ci-dessus sont requis pour les demandes de contrôle en amont. Certains clients HTTP (à savoir, les navigateurs modernes) effectuent une requête **avant** la requête souhaitée simplement pour voir s'ils ont l'autorisation de faire la requête réelle sur le serveur. Voir https://en.wikipedia.org/wiki/Cross-origin_resource_sharing pour plus d'informations sur la demande de contrôle en amont.

L'en-tête HTTP principal dont nous avons besoin est `Access-Control-Allow-Origin` et c'est ce que nous allons définir. Cependant, le même principe s'applique à tous (il suffit de savoir quoi retourner).

L'exemple suivant définit l'en-tête HTTP requis dans un bloc de configuration `<Directory>` pour activer un nom de domaine complet (FQDN) client protégé par SSL:

```
<Directory /path/to/your/site/>
        Header set Access-Control-Allow-Origin "https://my.CLIENT.domain"
</Directory>
```

Une fois que nous avons défini cela sur le **serveur** , nous pouvons maintenant effectuer une requête depuis https: //my.client.domain vers notre serveur et il doit répondre.

Remarque: de nombreuses personnes utilisent `Access-Control-Allow-Origin: "*"` qui est un caractère générique, pour signifier que les demandes provenant de **TOUS les** domaines doivent être acceptées. Cela est généralement déconseillé, sauf si vous utilisez une API ou un référentiel de fichiers public. Notez également le contexte de votre en-tête HTTP. Vous voudrez peut-être autoriser les requêtes HTTP pour une API, mais pas pour les images "hotlinking", etc. Vous pouvez définir cet en-tête où vous voulez dans votre flux de configuration Apache pour le définir **uniquement** dans des situations spécifiques. Par exemple, les éléments suivants définiraient **uniquement** l'en-tête HTTP CORS lorsque le chemin demandé n'est **pas** un fichier ou un répertoire (convient à une API publique qui interdit le lien dynamique des images):

```
<Directory /path/to/your/site/>
    Options +FollowSymlinks
    Options +Indexes
    RewriteEngine On

    #Make sure it's not a specific file or directory that they're trying to reach
    RewriteCond %{SCRIPT_FILENAME} !-f
    RewriteCond %{SCRIPT_FILENAME} !-d
    Header set Access-Control-Allow-Origin "*"
    RewriteRule ^(.*)$ index.php/$1 [L]
</Directory>
```

Conditions préalables

Vous devez avoir mod_headers installé et activé: `a2enmod headers`

301 Redirection par Htaccess

Le code d'état de réponse HTTP 301 Moved Permanently est utilisé pour la redirection d'URL permanente, ce qui signifie que les liens ou les enregistrements actuels utilisant l'URL pour laquelle la réponse est reçue doivent être mis à jour. La nouvelle URL doit être fournie dans le champ Emplacement fourni avec la réponse. La redirection 301 est considérée comme une meilleure pratique pour mettre à niveau les utilisateurs de HTTP à HTTPS. écrire ce code dans le fichier htaccess pour PHP-APACHE

```
Redirect 301 /oldpage/ /newpage/
```

Voici un exemple d'utilisation d'un fichier htaccess pour rediriger vers un non www avec un SSL attaché au domaine.

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ http://%1/$1 [R=301,L]

RewriteCond %{HTTPS} on
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ https://%1/$1 [R=301,L]

RewriteEngine On
RewriteCond %{SERVER_PORT} 80
```

```
RewriteRule ^(.*)$ https://example.com/$1 [R,L]
```

Lire Fichiers .htaccess dans Apache en ligne: https://riptutorial.com/fr/apache/topic/2089/fichiers--htaccess-dans-apache

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec apache	Community, fab, Flamewires, hjpotter92, James, Katie, Kuhan, Nicholas Qiao, Rich Bowen
2	Apache Flume	Vinod Kumar
3	Comment créer un hôte virtuel dans Apache	Chintan Gor, Clutch, fab, Harikrishnan, Hello Fishy, Katie, Olaf Dietsche
4	Fichiers .htaccess dans Apache	Chintan Gor, Deltik, ezra-s, Luke Bearl, Rich Bowen, SimpleAnecdote