le serveur apache

- Objet : du tuto : une approche du serveur apache2
- Création par lagrenouille le 04/11/2019
- Niveau requis : Fix Me!
  débutant, avisé
- Commentaires : *Contexte d'utilisation du sujet du tuto.*
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.

Introduction

Internet désigne l'interconnexion mondiale de réseaux, permettant à des milliards d'équipements disséminés dans le monde de communiquer.

Le Web, de son côté, désigne le système hypertexte constitué de l'ensemble des pages desservies par les serveurs HTTP de par le monde.

la doc en français : https://httpd.apache.org/docs/2.4/fr/getting-started.html

Pour communiquer, un serveur et un client ont besoin d'utiliser un protocole commun.

Pour le Web, ce protocole s'appelle HTTP (Hyper Text Transfert Protocol

Le serveur HTTP est le premier et le plus connu des projets, mais la fondation Apache va bien plus loin que cela, et gère d'autres projets d'envergure. serveur LDAP, * OpenOffice.org, SpamAssassin, Subversion,Tomcat….

Apache supporte – bien évidemment – le protocole sécurisé HTTPS, qui permet de chiffrer les communications afin d'en empêcher la lecture par un tiers. Ce support est disponible avec le module mod_ssl .

Un serveur web est un logiciel permettant à des clients, d'accéder à des pages web, c'est-à-dire en réalité des fichiers au format HTML à partir d'un navigateur, ce navigateur est votre client web.

Un serveur web est donc un « simple » logiciel capable d'interpréter les requêtes HTTP arrivant sur le port associé au protocole HTTP (par défaut sur le port 80), et de fournir une réponse avec ce même protocole.

Lorsqu'il démarre, Apache charge les fichiers de configuration et se met en attente de requêtes sur les interfaces réseaux. On dit qu'il écoute (listen en anglais) certains ports.

Le navigateur résout le nom de domaine (il obtient l'adresse IP du serveur. Il envoie une requête HTTP avec la méthode GET à l'IP du serveur sur le port 80 (ou HTTPS sur le port 443) pour lui demander de retourner un contenu particulier.

Le serveur HTTP reçoit la requête, et en fonction de divers paramètres (URL appelée, configuration du serveur, etc.), va chercher un contenu dans un fichier ou lance un script qui va générer un contenu.

Le serveur renvoie ce contenu à l'IP du navigateur depuis le même port (80 ou 443).

Le navigateur traite le contenu et le rend accessible à l'internaute (en l'affichant à l'écran

HTTP (HyperText Transfer Protocol) est un protocole de communication entre un client et un serveur développé pour le Web. L'une de ses fonctions principales est ainsi de récupérer des pages Web.

Quand on ouvre une URL en http://, le navigateur va agir comme un client HTTP. Il va donc envoyer une requête HTTP.

Le serveur HTTP renvoie une réponse HTTP qui contient la page Web demandée.Le navigateur interprète alors la page Web et l'affiche.

Le client : c'est le visiteur d'un site Web. Il demande la page Web au serveur. vous êtes des clients quand vous surfez sur le Web

c'est votre navigateur Web qui est le client, car c'est lui qui demande la page Web.

Les serveurs : ce sont les ordinateurs qui délivrent les sites Web aux internautes (aux clients)

DNS lorsque l'on saisit l'adresse (URL) d'un site dans un navigateur, une requête "DNS"transforme cette adresse textuelle (FQDN) en adresse IP numérique. Cette adresse IP est celle de votre machine où réside votre serveur web Apache

FQDN (Fully Qualified Domain Name)ou (nom de domaine réservé) peut s'appliquer à autant de virtualhost,(d'hôtes virtuels), que vous aurez configuré sous la même ip

Dans les grandes lignes, lorsque l'on saisit l'adresse (URL) d'un site dans un navigateur, une requête DNS transforme cette adresse textuelle (appelée FQDN, pour Fully Qualified Domain Name) en adresse IP numérique. Cette adresse IP est celle de votre machine où réside le serveur Apache (votre serveur web)

Cette association FQDN/adresse IP n'est pas nécessairement exclusive :

un nombre illimité de FQDN peuvent être associés à une seule et même adresse IP. à partir de là, lorsque votre navigateur web interroge un serveur, il précise le FQDN qu'il recherche et le serveur HTTP - Apache en l'occurrence - est capable de desservir des données différentes selon le serveur demandé. L'hôte physique est le même, mais il s'agit virtuellement, pour les navigateurs (clients) qui s'y connectent, de serveurs différents : dans la terminologie 'Apache, on parle alors d'« hôtes virtuels»

Dans le cas de figure le plus courant, on souhaite donc héberger plusieurs sites web ou plus, ayant des adresses (URL) différentes, mais avec une seule interface réseau et une seule adresse IP….."

Installation

Quelques installations qui vous seront utils

```
apt install apache2 libapache2-mod-php7.0 libapache2-mod-evasive apachetop
asql debsums php php5-dev php5-gd php-mysql php7.0.sqlite3 php7.0-cli
phpmyadmin mysql sqlite mysql-server mysql-client openssh-client openssh-
```

```
server openssh-sftp-server rsync fail2ban multitail
```

-apache2-utils est installé par defauts ainsi que mariadb

-multitail est un utilitaire pour surveiller les logs apache en directe via un terminal.

```
multitail /var/log/apache2/access.log
```

asql un petit outil qui permet d'interroger vos logs Apache, voir plus loin.

debsums est un utilitaire pour afficher sur votre terminal tout ce qui est installé et concerne un logiciel

```
debsums --all | grep apache2
```

php5-dev : Fichiers de développements (nécessaires pour certains modules) ; php5-gd : La librairie GD, pour manipuler les images.

et ssh, rsync donne accès pour les configurations

Utilisation

J'installe mon ou mes sites en générale dans un répertoire que je nomme WEB, je ferai des liens symbolique vers /var/www/html.

je mets ensuite les droits à 770, (excepté quelques fichiers si c'est du spip à l'installation qui requièrent 777)

Pour que apache puisse écrire sur les fichiers, le groupe doit appartenir à www-data, puis a laide d'adduser je me mets dans le groupe www-data

```
groups
```

```
lagrenouille disk cdrom floppy sudo audio dip www-data video plugdev netdev
scanner lpadmin lufi
```

Il faut configurer un VirtualHost par site et les activer.

La configuration d'Apache utilise des balises sous la forme

```
<...>, / , </...>
<VirtualHost *:80> ou voir <VirtualHost *:8390> suivant le port
DocumentRoot /var/www
[...]
</VirtualHost>
```

Nous définissons donc ici un hôte virtuel. L'argument en ouverture du bloc, « *:80 », ou autre port, correspond à l'interface physique et au port sur lequel écouter.

https://httpd.apache.org/docs/2.4/fr/vhosts/examples.html

Avec Apache, chaque site ou application web correspond en principe à un hôte virtuel (VirtualHost en anglais).

Chaque hôte virtuel est défini par un fichier de configuration indépendant, qu'on trouve ou qu'on créé dans le répertoire /etc/apache2/sites-available/.

La majorité de la configuration d'Apache se fait donc dans les fichiers .conf .

Les répertoires mods-enabled et sites-enabled ne contiennent que des liens symboliques vers leurs équivalents en *-available .

Les liens symboliques sont gérés par les commandes :

```
a2ensite pour activer un site (apache2 enable site) ;
a2dissite pour désactiver un site (apache2 disable site) ;
a2enmod pour activer un module (apache2 enable module) ;
a2dismod pour désactiver un module (apache2 disablemodule).
a2enconf [configuration d'un service à activer]
a2disconf [configuration d'un service à désactiver
```

Sur Debian, l'hôte virtuel de la configuration par défaut d'Apache2 est dans un fichier 000-default : ces trois zéros lui permettent d'être évalué en premier lieu et de servir d'hôte par défaut.ce fichier qui détermine aussi comment afficher la page « Itworks !

Autrement dit, ne placez pas une application critique et privée dans le tout premier hôte virtuel : n'importe qui pourrait y accéder sans même connaître son URL !

le paramètre DocumentRoot , que nous venons de rencontrer, permet de signaler à Apache où il doit chercher les fichiers à desservir.

le fichier php.ini Réglage maximale la taille de vos fichiers uploadés sur vos sites, dans /etc/php/7.3/apache2/php.ini

```
upload_max_filesize = 20000M
```

```
max_file_uploads = 70M
```

Cette valeur peut être défini dans apache2.conf.

Vous aurez parfois ce message:

La limite de mémoire PHP est inférieure à la valeur recommandée de 512 Mo.

```
memory_limit = 512M
```

les fichiers .htaccess

permettent également de définir des contextes.

Avant d'aller plus loin:dans votre conf apache. Lorsque la directive AllowOverrideList est définie à None, les fichiers .htaccess sont totalement ignorés. Dans ce cas, le serveur n'essaiera même pas de lire les ichiers .htaccess du système de fichiers.

```
AllowOverride None
```

Lorsque cette directive est définie à All, toute directive valable dans le Contexte .htaccess sera autorisée dans les fichiers .htaccess.

si il est spécifié AllowOverrideList Redirect RedirectMatch seules les directives Redirect et RedirectMatch sont autorisées. Toutes les autresprovoqueront une erreur interne du serveur.

voir pour les les redirections ici https://httpd.apache.org/docs/2.4/fr/rewrite/remapping.html

ces fichiers permettent de modifier certains aspects de la configuration d'Apache sans avoir accès aux fichiers de configuration ; c'est très utile sur des serveurs où les webmestres n'ont pas accès au serveur en tant que root : ils peuvent alors définir certains points de configuration de manière indépendante.

Cependant les fichiers .htacces sons sensible à la casse et parfois il vaut mieux mettre certaines configurations directement dans le fichier apache.

Les fichiers .htaccess ne doivent être utilisés que si vous n'avez pas accès au fichier de configuration du serveur principal. L'utilisation des fichiers .htaccess ralentit le fonctionnement de votre serveur .

* Il est toujours préférable de définir les directives que vous pouvez inclure dans un fichier .htaccess dans une section Directory, car elles produiront le même effet avec de meilleures performances.

Les fichiers .htaccess fournissent une méthode pour modifier la configuration du serveur au niveau d'un répertoire. Un fichier, contenant une ou plusieurs directives de configuration, est placé dans un répertoire de documents particulier, et ses directives s'appliquent à ce répertoire et à tous ses sous-répertoires.

* Si je regarde les" .htaccess "de mon installation "galette" en local, seul localhost est autorisé.(127.0.0.1), le module mod_authz_host. restreint l'accès à certaines parties de votre site web en fonction de l'adresse de l'hôte de vos visiteurs.

En général, les fichiers .htaccess utilisent la même syntaxe que les fichiers de configuration principaux. Ce que vous pouvez mettre dans ces fichier est déterminé par la directive AllowOverride.

Lire les descriptions des options de vos htaccess

-AllowOverride AuthConfig, -AllowOverride FileInfo -AllowOverride Indexes -AllowOverride Limit

le fichier /etc/apache2/ports.conf définit les ports en écoute.

on retrouve les directives Listen et NameVirtualHost

* Comme éxpliqué dans la doc d'apache, la directive Listen spécifiée dans le fichier de configuration est à sa valeur par défaut de 80 (ou tout autre port inférieur à 1024), il est nécessaire de posséder les privilèges root pour pouvoir démarrer apache, et lui permettre d'être associé à ce port privilégié.

* Lorsque le serveur est démarré, il effectue quelques opérations préliminaires comme ouvrir ses

fichiers de log, puis il lance plusieurs processus enfants qui ont pour rôle d'écouter et de répondre aux requêtes des clients. Le processus httpd principal continue à s'exécuter sous l'utilisateur root, tandis que les processus enfants s'exécutent sous un utilisateur aux privilèges restreints. Ceci s'effectue par la voie du Module Multi-Processus (MPM)

les processus d'apache, sont visible avec la commande.

```
ps -edf | grep apache
```

Pour voir tous les modules apache chargés, faites la commande :

```
apachectl -t -D DUMP_MODULES
```

ou

```
apache2ctl -M
```

Pour qu'apache puisse ecrire sur mes confs, il faut activer le module rewrite.

On active le module rewrite en ajoutant ces lignes dans le fichier /etc/apache2/apache2.conf

```
<ifModule mod_rewrite.c>
RewriteEngine On
</ifModule>
```

Puis il faut l'activé.

```
a2enmod rewrite
```

```
systemctl restart apache2
```

Pour vérifier la bonne écriture dans la conf de votre vhost

```
apachectl configtest
```

ou

```
apache2ctl -t
```

Pour vérifier et analyser la configuration de vos serveurs virtuels, vous pouvez utiliser

```
apachectl -S
```

Pour savoir à quelle date le serveur a était installé.

```
ls -lct /etc | tail -1 | awk '{print $6, $7,$8}'
```

Pour relancer apache2 :

```
systemctl restart apache2
```

Pour recharger la configuration d'apache2 :

```
systemctl reload apache2
```

virtualhost	extrait court des directives de configuration
NameVirtualHost	indique au serveur web apache que sur le port 1280 on a l'ip 192.168.1.12
soit :	Name Virtualhost 192.168.1.12:1280
ServerName	indique l'URL du site hébergé, ServerName localhost
ServerAdmin	webmaster@localhost
DocumentRoot	permet de signaler à Apache où il doit chercher les fichiers à desservir, En d'autres termes, c'est le répertoire racine du serveur, le répertoire de référence
ServerAlias	permet de donner un ou plusieurs noms complémentaires,par lesquels les visiteurs pourront également accéder au site concerné
Timeout 120	c'est, en secondes le temps mort maximal requis entre une émission et une réception, ici 120s.

Les autorisations d'accès

Les directives Order , Allow et Deny permettent de définir les autorisations d'accès, selon les éléments fournis par les requêtes (adresses IP,nom d'hôte,user-agent...) :

allow définit les clients autorisés

deny Définit quels hôtes ne sont pas autorisés à accéder au serveur

order Définit le statut d'accès par défaut et l'ordre dans lequel les directives Allow et Deny sont évaluées.

https://httpd.apache.org/docs/2.4/fr/mod/mod_alias.html

-La doc apache dit:

Les directives fournies par le module mod_access_compat sont devenues obsolètes depuis la refonte du module mod_authz_host. Mélanger d'anciennes directives comme Order, Allow ou Deny avec des nouvelles comme Require est techniquement possible mais déconseillé. En effet, mod_access_compat a été conçu pour supporter des configurations ne contenant que des anciennes directives afin de faciliter le passage à la version 2.4. Voir le document upgrading pour plus de détails.

- Les directives Allow et Deny permettent de spécifier quels clients sont ou ne sont pas autorisés à accéder au serveur, alors que la directive Order définit le statut d'accès par défaut, et détermine la manière dont les directives Allow et Deny interagissent entre elles.

Quels outils pour la surveillance d'apache et les logs d'apache ?

Tous les services réseau peuvent faire l'objet d'attaques de type: "Déni de service" (Denial of Service

- DoS) * Pour empêcher aux services réseaux de répondre correctement aux demandes, les pirates saturent leurs ressource par l'envoie d'énormes envoies de données, difficile de se prémunir de ces attaques, sinon par des pare-feu qui limitent les connexions d'ip ou de réseaux. (voir iptables et ntables)

Des outils de surveillances et d'audit peuvent faciliter la.source des attaques, ainsi qu des configurations propres au système d'exploitation.

La directive RequestReadTimeout permet de limiter le temps que met le client pour envoyer sa requête, ce qui peut être à double tranchant.

Voir dans les nombreux modules d'apache sil existe des conf qui peuvent minimiser les problèmes de DoS.

Assurez vous que les utilisateurs n'ait aucun droit pour modifier les configurations, apache c'est à root et à lui seul.

Apache est démarré par l'utilisateur root, puis il devient la propriété de l'utilisateur défini par la directive User afin de répondre aux demandes.

-Enfin, maintenez votre serveur à jour, et lisez la doc ici :
https://httpd.apache.org/docs/2.4/fr/misc/security_tips.html

-Surveillez vos journaux, même si les fichiers journaux ne consignent que des évènements qui se sont déjà produits, ils vous informeront sur la nature des attaques qui sont lancées contre le serveur et vous permettront de vérifier si le niveau de sécurité nécessaire est atteint.

contrôler le fonctionnement du démon Apache avec l'interface de contrôle du serveur HTTP Apache: **apachectl**

```
apachectl status
```

D' autres options start: démarre apache stop: arrête apache restart redémarre apache

apachetop Outil de surveillance de Apache en temps réel, apache top est un utilitaire en temps réel basé sur curses qui affiche des informations à propos d'une copie d'Apache en cours d'exécution.

Il est conçu sur le modèle de l'utilitaire standard « top » et affiche des informations telles que le nombre de requêtes par seconde, le nombre d'octets par seconde et les adresses les plus souvent demandées.

Il doit être lancé à partir d'une machine exécutant Apache car il fonctionne en traitant les fichiers de journaux situés dans /var/log/apache

la doc apache spécifie:

"La première chose à faire lors de la mise en production d'un serveur Apache est de faire en sorte qu'il n'affiche pas l'ensemble de ses informations de versions afin d'éviter à tout attaquant potentiel de connaître trop d'informations sur sa machine cible."

-Pour cela deux directives sont à modifier. -Ces deux directives se trouvent dans le fichier /etc/apache2/conf.d/security, que l'on édite : soit dans /etc/apache2/conf.d/security ou dans

etc/apache2/conf-available/security.conf

-si l'on a ServerTokens OS * mis par défaut en général : le serveur affichera Apache, son numéro de version complet et son système d’exploitation Apache/2.0.55 (Debian) mettre la plus restrictive, Apache ne devrait afficher que Apache ; ServerTokens Prod

-si l'on a ServerTokens OS * mis par défaut en général : le serveur affichera Apache, son numéro de version complet et son système d’exploitation Apache/2.0.55 (Debian) mettre ServerSignature Off et l'on n’affichera pas ces informations.

On n’oublie pas de recharger la configuration d’Apache

```
service apache2 reload
```

Voir plus de détails ici: https://httpd.apache.org/docs/2.4/fr/sections.html

c'est loin, loin d’être complet, reportez vous à la doc d'apache, très complète et volumineuse.