Apache 2 : Serveur Web

Présentation

Apache

Le serveur Web le plus utilisé dans le monde (source journal du net ici) est un outil libre développé dans l'environnement Linux/UNIX : le serveur Apache de l'*Apache Software Foundation* (http://www.apache.org) qui se décline en d'innombrables versions (dont *Tomcat* pour l'exécution de script côté serveur en Java : *servlet*). Il permet la mise en place d'un serveur web (pages HTML) avec exécution de scripts PHP, JSP ou ASP. Il peut sécuriser limiter l'accès aux pages (*.htaccess*), créer des **hôtes virtuels** (plusieurs noms de site différents sur une seule exécution du service) ou des sites indépendants (sur des ports différents).

Apache peut aussi être utilisé comme *reverse proxy*, étant le point d'entrée unique vers l'ensemble des serveurs Web internes.

L'installation du service standard ne permet pas la sécurisation *SSL* (voir la fiche correspondante).

Procédures

Installation du service

1. Installer les paquetages apache2
    1. pour les versions d'Ubuntu avant 10.04 et antérieures, installer manuellement les paquetages ***apache2.2-common*** et ***apache2-utils*** pour la version de base
2. Pour ajouter des fonctions d'authentification, la gestion du *multi-processing* et la possibilité de changer le service en mode *root* : installer les paquetages ***apache2-mpm-prefork***, ***libapache2-mod-chroot***, ***libapache2-mod-auth-pam***, ***libapache2-mod-auth-sys-group***
3. Tester depuis une machine cliente (http://adresseIPServeur)

Installation de l'interpréteur PHP

Pour prendre en charge PHP, Apache doit être complété par un interpréteur.

1. Installer le module ***php***, qui installera les dépendances ***php-common***, ***php-gd***, ***php5-cli*** et ***libapache2-mod-php***
2. Redémarrer le serveur *apache*

3. Tester en déposant ou créant un fichier avec du code PHP dans le répertoire /***var/www/html***

Prise en charge MySQL par Apache/PHP

En complément du serveur Web, on installe la capacité pour *Apache* de gérer les échanges avec *MySQL*. Cela ne nécessite pas nécessairement que *MySQL* soit présent sur la machine hébergeant Apache (*MySQL* peut être sur un serveur Windows sous la forme d'une installation *EasyPHP* par exemple)

1. Installer le paquetage ***libapache2-mod-auth-mysql*** pour la gestion de l'authentification *MySQL*
2. Installer éventuellement le paquetage ***php-mysql*** pour permettre les échanges entre PHP et la base *MySQL*
3. Redémarrer *Apache*
4. Tester en insérant une connexion à une base *MySQL* dans un fichier PHP sur /***var/www***

Administration de MySql via PhpMyAdmin

Pour administrer MySQL via PhpMyAdmin, on doit installer les paquetages *apache2, php5* (ou *php* pour avoir la dernière version en date [actuellement php7]), *mysql-server* et *phpmyadmin*.

Paramétrages de Apache et PHP

Activer le débogage PHP

Pour permettre à Apache d'afficher les erreurs d'exécution PHP (sur un serveur de test, mais pas en *production*), on activera le paramètre ***display_errors*** (valeur *on*) dans le fichier ***/etc/php/<version>/apache2/php.ini*** .

Installer PHP7.3

Voir sur https://geekmag.fr/2019/10/05/installer-php-7-3-sur-debian-9-ou-debian-8/

Augmenter le temps d'exécution des requêtes

Certaines requêtes peuvent durer un temps très long, notamment lorsqu'on réimplante une base de données volumineuse.

Il est alors possible d'augmenter le temps maximum alloué à l'exécution d'une requête. **Ceci crée un potentiel de malveillance sur des requêtes qui mobiliseraient indéfiniment le serveur ou pour des actions de détournement ou de destruction**.

On modifiera le paramètre ***max_execution_time*** (valeur en secondes) dans le fichier ***/etc/php/apache2/php.ini*** .

Mode opératoire

Serveur Web, répertoires virtuels, sites multiples

1 - Installation et répertoires

Un serveur web correspond à un service en écoute sur le port 80 (port standard pour le protocole HTTP). L'installation d'Apache (***apt-get install apache2***) entraîne l'installation du service ***/etc/init.d/apache2***.

Le fichier principal de la configuration est ***/etc/apache2/apache2.conf*** qui fait lui-même appel à d'autres fichiers externalisés. On trouve aussi le fichier avec toutes les configurations dans ***/etc/apache2/sites-enabled/000-default.conf***.

Les pages Web sont stockées par défaut dans ***/var/www*** ou ***/var/www/html/***.

2 - Fichier apache2.conf

Généralités

Ce fichier comporte dans sa version standard plus de 200 lignes (dont beaucoup de commentaires). Il est conseillé de travailler sur une copie du fichier d'installation dans lequel on supprimera tous les éléments inutilisés. Bien entendu, il est prudent de savoir ce que contiennent les **directives** avant de les activer, supprimer ou modifier, comme cela est rappelé dans les premiers commentaires :

```
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
```

Remarque : Les commentaires du fichier source présenté plus loin ont été supprimés, seules les directives principales et basiques sur lesquelles on intervient en général sont présentées.

Structure du fichier

Il y a trois grandes sections dans le fichier :

1. **globale** : définit les paramètres du service (durée de session, connexions simultanées, etc)
2. **site standard** (fichiers dans *conf/*) : paramétrage manuel du service standard (répertoires, droits d'accès, etc)

3. **hôtes virtuels** (fichiers dans *sites-enabled/*) : paramétrage de sites accessibles sous des noms ou adresse IP virtuelles

Section 1 : Paramétrage du service

Dans cette première section, on paramètre la façon dont le service HTTP prendra en charge les demandes de connexion. On y précise aussi les inclusions pour ce qui est du reste de la configuration (cela évite d'avoir un fichier unique de 900 lignes comme le *httpd.conf* de la version 1 de Apache).

Contenu du fichier (Extraits)

```
Timeout 300             # durée de vie d'une connexion en secondes (temps
avant la déconnexion automatique)
KeepAlive On            # On ou Off : permet pour une même connexion de
faire plusieurs requêtes simultanément.
            # améliore les temps de réponse, mais oblige le serveur à garder
des sessions en mémoire
KeepAliveTimeout 15     # temps en seconde entre deux actions avant que la
session soit déconnectée
AccessFileName .htaccess # indique la façon dont on restreint l'accès aux
dossiers
            # (ici, fichiers .htaccess dans le dossier à sécuriser
            # à commenter si on ne veut pas appliquer de restrictions
LogLevel warn           # indique le type d'événement enregistré dans les
journaux
            # valeurs : debug, info, notice, warn, error, crit, alert,
emerg.
Include mods-enabled/*.load   #modules appelés au chargement du service,
            # (authentification, PHP, transferts de fichiers riches type
PDF, MP3 et autres...)
Include mods-enabled/*.conf   # configurations spécifiques pour les modules
            # Include all the user configurations:
Include httpd.conf      # appel aux configurations spécifiques (section 2 :
« service standard »)
Include ports.conf      # appel aux paramétrages sur les ports d'écoute
Include conf.d/         # appel au dossier contenant les paramétrages
spécifiques des sites
                # par exemple, on y trouve généralement phpmyadmin.conf
Include sites-enabled/  # sites gérés, construit lors du lancement du
service à partir des fichiers  de configuration
```

Fichier ports.conf

```
NameVirtualHost *:80  # Gestion du serveur avec prise en charge d'hôtes
virtuels sur le port 80
Listen 80             # port d'écoute (80 est la valeur par défaut).
<IfModule mod_ssl.c>  # paramétrage si prise en charge de SSL (remplacé par
TLS)
```

```
    Listen 443
</IfModule>
<IfModule mod_gnutls.c> # paramétrage si prise en charge de TLS
    Listen 443
</IfModule>
```

Section 2 : Configuration du site Web

On trouvera dans cette section les directives pour le site principal (c'est à dire pour une installation sans hôte virtuel). À cet endroit, on définit les caractéristiques du site, les sous répertoires accessibles, et les options de sécurité qui s'y appliquent. On peut mettre ces directives dans le fichier *httpd.conf*, ou dans le fichier *apache2.conf* (**non recommandé**). En cas d'utilisation d'hôtes virtuels, il est préférable de renseigner ces informations dans le fichier *default-000* du sous-dossier *sites-enabled*.

Principales directives

Directive	Usage
<Directory…> </Directory>	Définitions de conditions et comportements particuliers pour un répertoire (chemin d'accès, restrictions d'accès, etc)
Alias	Définit un raccourci utilisateur pour un chemin d'accès complexe à un répertoire
Allow	Autorise l'accès à des machines ou des adresses de réseau
Deny	Interdit l'accès à des machines ou des adresses de réseau
DocumentRoot	Chemin d'accès au dossier racine contenant des pages Web
DirectoryIndex	Définition de la page d'accueil par défaut
LogLevel	Niveau d'enregistrement des problèmes dans le journal d'événement
Order	Ordre dans lequel sont appliquées les règles de sécurité (Deny et Allow)
ServerAdmin	Nom ou adresse de l'administrateur du serveur
ServerName	Nom du serveur tel qu'il devra être tapé dans la barre d'adresse du navigateur N'a d'intérêt au niveau général que si on gère des hôtes virtuels sur le serveur

Exemple de fichier (Extraits)

```
ServerAdmin webmaster@gsb.net    # pour joindre l'administrateur
ServerName public.gsb.net        # nom FQDN auquel répond le serveur
DocumentRoot /var/www            # localisation du fichier contenant les
pages web
DirectoryIndex  accueil.php      # localisation de la première page à ouvrir
<Directory />                    # droits s'appliquant à la racine (définie
dans DocumentRoot) du site
  Options FollowSymLinks         # autorise l'usage de liens symboliques
(non maîtrisé à ce jour)
  AllowOverride None             # les droits ne peuvent être redéfinis
ailleurs (dans fichiers .htaccess)
  Deny from All                  # interdit le parcours du répertoire
</Directory>
<Directory /var/www/>            # paramétrage pour le répertoire /var/www
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
```

```
  Order allow,deny                  # ordre de consultation des droits
  allow from all                    # autorise l'accès à toute machine
</Directory>
Alias /doc/ "/usr/share/doc/"     # raccourci pour le répertoire
<Directory "/usr/share/doc/">     # paramètres spécifiques
  Order deny,allow
  Deny from all
  Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
```

Dans une telle configuration où l'accès au site est limité à un usage local (*Allow from 127.0.0.0/255.0.0.0 ::1/128*), l'accès au site depuis une machine distante donnerait le résultat ci-contre.

Section 3 : Hôtes virtuels

Principes

La configuration standard (dans *httpd.conf*) ne permet d'avoir, pour un serveur apache, qu'un seul site web. On peut choisir d'utiliser un serveur physique différent (ou autant de machines virtuelles) pour chaque site hébergé. On peut aussi décider de gérer des sites accessibles par des noms différents (noms FQDN ou adresse IP) depuis un même serveur Apache. On aura alors recours à la définition d'hôtes virtuels.

Remarque : il ne faut pas confondre un *hôte virtuel* et un *sous-répertoire* (ou répertoire virtuel dans IIS) : ce dernier est accessible par *http:/nom_serveur/nom_Repertoire_virtuel*, comme cela est utilisé pour les “pages perso” des hébergeurs gratuits par exemple, ou pour l'accès à *www.etab.ac-caen.fr/rostand//. Le répertoire virtuel est géré par la directive* Alias *dans la configuration Apache.* Fichier default dans le dossier sites-enabled *Le fichier* default-000 *permet de créer des hôtes virtuels qui seront accessibles par* http://nom_hote_Virtuel*. Un hôte virtuel se comporte comme un nouveau serveur web et peut donc reprendre les directives indiquées plus haut (*DocumentRoot, Listen, Directory, *etc.). **Exemple de déclaration d'hôtes** <code apache> #définition d'un hôte virtuel depuis n'importe quelle adresse IP du serveur, sur le port 80 <VirtualHost *:80> NameVirtualHost 192.168.0.152 # Déclaration d'un hôte virtuel sur une adresse IP déterminée DocumentRoot /var/www </VitualHost> #Définition d'un hôte virtuel sur une adresse précise du serveur, avec pour nom virtuel « www.SiteVitrine.fr » <VirtualHost 192.168.0.152:80> DocumentRoot /var/www/SiteVitrine # chemin où sont stockées les pages du site ServerName www.SiteVitrine.fr # nom du site virtuel #Déclaration des options de sécurité du répertoire virtuel #Choisir la gestion par .htaccess, limiter les droits de lecture/modification, etc </VirtualHost> #Définition d'un hôte virtuel sur la même adresse pour un second site de nom « commerce.online.fr » <VirtualHost 192.168.0.152:80> DocumentRoot /var/www/SiteCommercial # chemin où sont stockées les pages du site ServerName commerce.online.fr # nom du site virtuel #Déclaration des options de sécurité du répertoire virtuel </VirtualHost> </code> L'accès à l'hôte virtuel nécessite bien entendu de pouvoir l'atteindre grâce à un FQDN, donc grâce à un enregistrement DNS ou un renseignement dans le fichier* hosts *ou* lmhosts *(Voir fiche DNS).* Écoute sur plusieurs ports *Pour permettre la mise en place de sites spécifiques ou sécurisés (parce qu'ils n'écoutent pas sur le port standard), on pourra aussi créer des hôtes virtuels en écoute sur un port spécifique. #Définition d'un hôte virtuel sur une adresse précise du serveur, sur un autre port (exemple ici : port 8800) <code apache> <VirtualHost 192.168.0.152:8800> DocumentRoot /var/www/intranet # chemin où sont stockées les pages du site ServerName intra.entreprise.fr # nom du site virtuel </VirtualHost> </code> ==== Sécurisation des*

accès (.htaccess) ==== Par défaut, un serveur Web ne permet pas la sécurisation des informations autrement que par les droits d'accès accordés à l'utilisateur qui exécute le processus. Il n'est donc pas possible de réaliser de sécurité en fonction des utilisateurs. Avec Apache (mais aussi avec IIS), une connexion authentifiée va autoriser d'affiner les accès en fonction des utilisateurs grâce à un fichier nommé .htaccess *situé dans chaque répertoire pour lequel on voudra limiter l'accès spécifiquement. Ces fichiers définissent les utilisateurs autorisés pour le répertoire courant et tous les répertoires, jusqu'à rencontrer un nouveau fichier* .htaccess *plus bas dans l'arborescence. Pour cela, chaque répertoire (éventuellement dans les hôtes virtuels) sera paramétré pour prendre en compte ces fichiers grâce à la directive* Exemple de fichier de recours aux fichiers .htaccess *<code apache> <Directory "MonRepertoire"> #Déclaration des options de sécurité du répertoire AllowOverride AuthConfig # Les fichiers d'authentification .htaccess s'ils # existent remplacent les droits du dossier </Directory> <VirtualHost 192.168.0.152:80> DocumentRoot /var/www/SiteCommercial ServerName commerce.online.fr # nom du site virtuel <Directory /var/www/SiteCommercial> # Gestion de la sécurité spécifique à l'hôte virtuel AllowOverride AuthConfig </Directory> </VirtualHost> </code> Chaque fichier* .htaccess *est constitué comme suit: <code apache> AuthType Basic # type d'authentification standard (mots de passe en clair) AuthUserFile /etc/apache2/conf/httpUtilisateurs # chemin et nom du fichier des utilisateurs (fichier à créer) AuthName "Accès réservé" # commentaire pour les utilisateurs refusés require user nom_utilisateur1 # utilisateurs autorisés require user nom_utilisateur2 </code> La création des utilisateurs se fera par l'outil* **htpasswd** *comme suit : <code lscript>* La première création de compte nécessite aussi la création du fichier *htpasswd -c /chemin/nom_fichier_utilisateurs nom_utilisateur htpasswd -c /etc/apache2/conf/httpUtilisateurs pierre* Le système demandera alors les mots de passe *Pour les utilisateurs suivants htpasswd /etc/apache2/conf/httpUtilisateurs paul htpasswd /etc/apache2/conf/httpUtilisateurs jacques </code> ==== Environnement et fichiers ==== Le répertoire par défaut des fichiers HTML et PHP est* /var/www/html*. Les principaux fichiers de configuration (dans* /etc/apache2*) ^ Fichier ^ Usage ^ |apache2.conf |Définit les principales caractéristiques techniques du service HTTP : mode de transfert de l'information, durée avant la fermeture d'une session, gestion de la sécurité, etc. Fait appel aux fichiers externes pour le reste de la configuration | |ports.conf |Indique les ports sur lesquels le serveur écoute (défaut HTTP:80, HTTPS:443).*
Permet la création d'hôtes virtuels (un seul serveur apache, plusieurs sites sous des noms différents)| |httpd.conf |C'est l'ancien fichier de configuration. Il peut être intéressant d'y insérer les paramétrages spécifiques pour améliorer la lisibilité (vérifier qu'il est appelé par include *dans* apache2.conf*) | ===== Sources internet ===== * http://public.loligrub.be/contrib/tlepoint/BASE/node605.html : Site (en français) très détaillé sur beaucoup de configurations dans le monde de l'administration système et réseau (ici pour Apache, avec explication assez claire des différentes options et directives) * http://doc.ubuntu-fr.org/lamp : en français, installer pas à pas un LAMP sous Ubuntu*