Serveur Web Apache

Installation du service LAMP (Linux Apache MySQL PHP)

Installation de MariaDB (remplace MySQL)

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.

Remarque : La distribution MySQL a été remplacé par MariaDB

Installer les paquets MariaDB :

```
apt-get install mariadb-server mariadb-client
```

Il faut ensuite sécuriser MariaDB avec la commande mysql_secure_installation comme suit (les modifications sont en rouge) :

Lancer mysql_secure_installation dans un terminal :

```
mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.
Enter current password for root (enter for none): <-- Appuyez sur la touche Entrée
OK, successfully used password, moving on...
Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.
Set root password? [Y/n] <-- y
New password: <-- Entrez votre mot de passe « root user »
Re-enter new password: <-- Entrez de nouveau votre mot de passe
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
```

```
go a bit smoother. You should remove them before moving into a
production environment.
Remove anonymous users? [Y/n] <-- y
... Success!
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
Disallow root login remotely? [Y/n] <-- n si vous voulez accéder au serveur en « root
user » depuis un autre poste, y sinon
... Success!
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.
Remove test database and access to it? [Y/n] <-- y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
Reload privilege tables now? [Y/n] <-- y
... Success!
Cleaning up...
All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.
Thanks for using MariaDB!
```

Installation d'Apache

Installer des paquetages apache2 pour la version de base :

```
apt-get install apache2
```

Tester depuis une machine cliente (http://adresseIPServeur)

Le répertoire par défaut des fichiers HTML et PHP est /var/www/html.

Les fichiers de configuration

Ils se situent dans le répertoire /etc/apache2

Fichiers	Explication

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)
envvars	Utilisé pour définir des variables d'environnement propres à Apache
conf.d	Contient plusieurs petits fichiers qui seront analysés par apache. Le seul fichier pour le moment est charset, qui spécifie l'encodage à utiliser par défaut
mods-available	Contient la liste des modules d'apache installés
mods-enabled	Contient la liste (liens symboliques) des modules d'apache activés (activer : `a2enmod`, désactiver : `a2dismod`)
sites-available	Contient la liste des vhosts installés
sites-enabled	Contient la liste (liens symboliques) des vhosts activés (activer : `a2ensite`, désactiver : `a2dissite`)

Installation du PHP

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

Installer les paquets PHP :

```
apt-get install php7.0 libapache2-mod-php7.0
```

Redémarrer le serveur apache :

```
systemctl restart apache2
```

Tester en créant un fichier avec du code PHP, nommé test.php, dans le répertoire /var/www/html. Le contenu est le suivant : `<?php phpinfo(); ?>`

Complément PHP pour MySQL

Installer les paquets suivants :

```
apt-get -y install php7.0-mysql php7.0-curl php7.0-gd php7.0-intl php-pear php-imagick
php7.0-mcrypt php-memcache
```

Redémarrer le serveur apache :

```
systemctl restart apache2
```

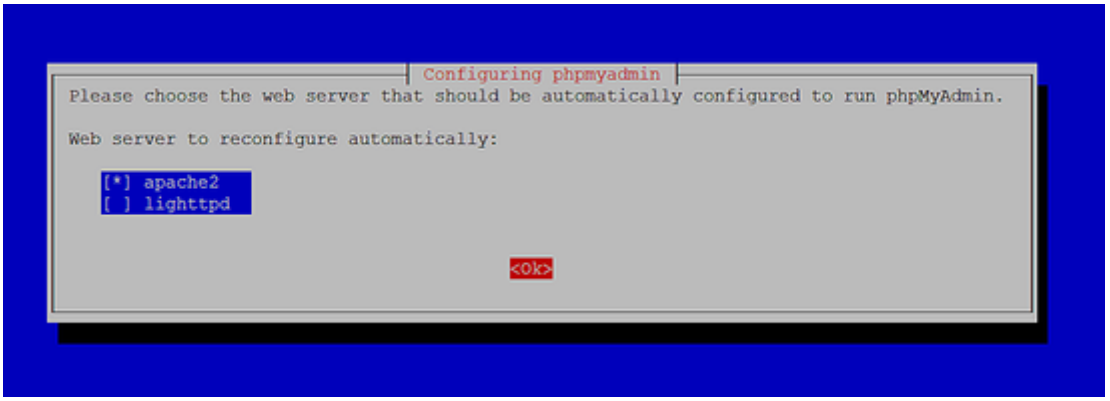
Installer PHPMyAdmin

En complément du serveur MySQL, vous pouvez installer la gestion de celui-ci via une interface Web. Cette installation pourrait se faire sur un autre serveur.

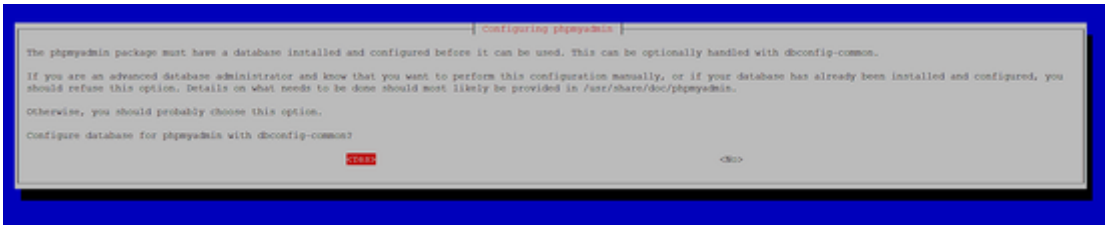
Installer phpMyAdmin

```
apt-get -y install phpmyadmin
```

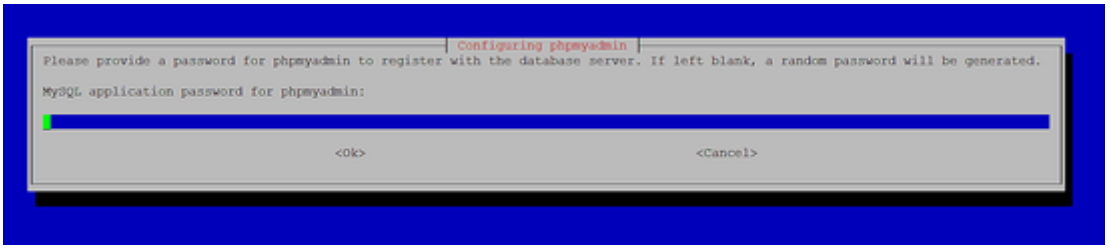
Il faut ensuite réponde aux questions :




```
Web server to reconfigure automatically: <-- apache2
```



```
Configure database for phpmyadmin with dbconfig-common? <-- Yes
```




```
MySQL application password for phpmyadmin: <-- Appuyer sur la touche Entrée, un mot de passe aléatoire sera généré.
```

Vous pouvez ensuite accéder à PHPMyAdmin via l'URL http://<adresse ip de la machine>/phpmyadmin/

Autoriser l'accès « root user » de MySQL pour PHPMyAdmin

Bien que vous puissiez vous loguer à MySQL en « root user » dans un terminal, vous ne pouvez pas le faire depuis PHPMyAdmin. Pour activer l'accès, entrez le commande suivante dans un terminal :

```
echo "UPDATE mysql.user SET plugin = 'mysql_native_password' WHERE user = 'root' AND plugin = 'unix_socket';FLUSH PRIVILEGES;" | mysql -u root -p
```

Le serveur web

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 ce fichier :

1. **globale** : définit les paramètres du service (durée de session, connexions simultanées, etc)
2. **site standard** (fichier ***sites-available/default***) : paramétrage manuel du service standard (répertoires, droits d'accès, etc)
3. **hôtes virtuels** (fichiers dans ***sites- available/*** et ***sites-enabled/***) : paramétrage de sites accessibles sous des noms ou adresse IP virtuelles

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).

Fichier apache.conf (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 demandes.
# améliore les temps de réponse, 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 listing
Include ports.conf
#appel aux paramétrages sur les ports d'écoute
# Include generic snippets of statements
Include conf.d/
#appel au dossier contenant les paramétrages spécifiques des sites
# par exemple, on y trouve phpmyadmin.conf
# Include the virtual host configurations:
Include sites-enabled/
#sites gérés, construit lors du lancement du service à partir des fichiers
# de configuration
```

Fichier ports.conf

```
# 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>
```

Si votre serveur génère ce type d'erreur : *... Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName*

Il suffit d'ajouter `ServerName localhost` dans votre fichier /etc/apache2/apache2.conf

Configuration du site Web

Rappel : Les directives se trouvent dans le fichier ***site-available/default*** et il est activé.

Ces directives définissent les caractéristiques du site, les sous répertoires accessibles, et les options de sécurité qui s'y appliquent.

Les hôtes virtuels

Intérêts

Pourquoi utiliser un hôte virtuel ?

- Afin d'utiliser un serveur physique différent (ou autant de machines virtuelles) pour chaque site hébergé.
- Pour gérer des sites accessibles par des noms différents (noms FQDN ou adresse IP) depuis un même serveur Apache

Le fichier ***site-available/default*** permet de créer des *hôtes virtuels* qui seront accessibles par http://nom_hote_Virtuel. Mais il est préférable de créer un fichier spécifique dans ***site-available*** puis de l'activer.

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

Définition de l'hôte

```
# définition d'un hôte virtuel depuis n'importe quelle adresse IP du serveur,
# sur le port 80
<VirtualHost *:80>
        # chemin où sont stockées les pages du site
        DocumentRoot /app/www/sitevitrine
        # nom du site virtuel
        ServerName www.sitevitrine.fr

    <Directory /app/www/sitevitrine>
        # Gestion de la sécurité spécifique à l'hôte virtuel
        Require All Granted
    </Directory>

    # Log file locations
    LogLevel warn
    ErrorLog  /app/www/sitevitrine/log/error.log
    CustomLog /app/www/sitevitrine/log/access.log combined
</VirtualHost>
```

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.

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)
<VirtualHost *:8800>
        # chemin où sont stockées les pages du site
        DocumentRoot /var/www/intranet
        # nom du site virtuel
        ServerName intra.entreprise.fr
</VirtualHost>
```

SECURISATION DES ACCES (.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.

Déclaration pour un hôte virtuel

```
<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 *:80>
        DocumentRoot /app/www/SiteCommercial
        ServerName commerce.online.fr ; nom du site virtuel
        <Directory /app/www/SiteCommercial>
                # Gestion de la sécurité spécifique à l'hôte virtuel
                AllowOverride AuthConfig
        </Directory>
</VirtualHost>
```

Ecriture du fichier .htaccess

```
# type d'authentification standard (mots de passe en clair)
AuthType Basic
# chemin et nom du fichier des utilisateurs (fichier à créer)
AuthUserFile /etc/apache2/httpUtilisateurs
# commentaire pour les utilisateurs refusés
AuthName "Accès réservé"
# utilisateurs autorisés
require user paul
require user jacques
```

Création des utilisateurs

L'outil htpasswd permet de créer des utilisateurs :

```
# La première création de compte nécessite aussi la création du fichier
# htpasswd -c /chemin/nom_fichier_utilisateurs nom_utilisateur
htpasswd -c /etc/apache2/httpUtilisateurs pierre
# Le système demandera alors les mots de passe
# Pour les suivants
htpasswd /etc/apache2/httpUtilisateurs paul
htpasswd /etc/apache2/httpUtilisateurs jacques
```

La commande `htpasswd` peut aussi utiliser un mot de passe :

```
htpasswd –b /etc/apache2/httpUtilisateurs henry passHenry
```

Table des matières