

TP n°2: Mise en place d'un serveur Web avec PHP et MySQL

Le but de ce TP est de vous apprendre comment installer et configurer un serveur Web avec PHP et MySQL sous Linux. Cela requiert plusieurs étapes :

1. installer et configurer un serveur Web (ici Apache)
2. installer le support pour le langage PHP (on peut aussi utiliser d'autres langages pour la conception de sites Web, comme Perl, Python, Ruby, ...)
3. installer et configurer un serveur de bases de données (ici MySQL),
4. éventuellement, installer une interface Web pour l'administration du serveur de bases de données, comme `PhpMyAdmin`.

Exercice 1 - Installation et configuration de Apache.

Un serveur Web est un programme qui va fournir des fichiers (généralement des pages HTML et des fichiers multimédias) à tous les clients (navigateurs Web sur la même machine ou une machine distante) qui en font la demande. Nous utiliserons ici la version 2 du logiciel Apache, qui est actuellement le serveur Web le plus utilisé.

1.1 La première étape consiste à installer la version 2 de Apache. Vous pouvez le faire depuis le gestionnaire de paquets de votre choix (`synaptic` ou commande `sudo apt-get` sous Ubuntu, ...). Le paquet se nomme généralement `apache` ou `apache2`. Installez le.

1.2 Si l'installation est réussie, vous devriez être en mesure d'accéder au site Web hébergé par votre machine en tapant l'adresse `http://localhost`. Faites le test. Si ça marche, ajoutez votre propre fichier `index.html` dans le répertoire `/var/www` et rechargez la page.

note : par défaut, il vous faudra avoir les droits administrateurs pour ajouter un fichier dans `/var/www`.

1.3 Vous pouvez aussi accéder à votre site en remplaçant `localhost` par `127.0.0.1`, ou par l'adresse IP de votre machine (donnée par la commande `ip addr`), ou encore par le nom de votre machine (il figure dans le fichier `/etc/hosts`). Essayez ces différentes possibilités. Comment faire pour accéder au site Web hébergé sur la machine de votre voisin ?

1.4 La configuration de Apache se fait principalement dans les fichiers `/etc/apache2/apache2.conf` et `/etc/apache2/ports.conf`. Ouvrez ces fichiers, et répondez aux questions suivantes :

- (a) quel utilisateur est utilisé pour lancer le serveur Apache ?
- (b) quel groupe est utilisé pour lancer le serveur ?
- (c) quels sont les ports utilisés par le serveur ?
- (d) quel est le nom choisi pour les fichiers de configuration s'appliquant aux répertoires du site Web hébergé par le serveur ?

1.5 Modifier le contenu du dossier `/var/www` nécessite des droits que votre compte utilisateur ne possède pas. Pour remédier à ce problème, on pourrait ajouter des droits à votre compte. Au lieu de cela, nous allons activer un module de Apache, appelé `userdir`.

- (a) Vérifier que ce module est bien disponible en regardant si le dossier `/etc/apache2/mods-available` contient le fichier `userdir.load`.

- (b) Le dossier `/etc/apache2/mods-enabled` contient des liens symboliques vers les fichiers du dossier `/etc/apache2/mods-available`. La présence d'un tel lien signifie que le module en question est activé. Regarder si le module `userdir` est déjà activé.
- (c) S'il n'est pas activé, vous devez soit créer vous même les liens symboliques avec la commande `ln -sv` (il faut aussi un lien vers le fichier `userdir.conf` s'il existe). Plus simplement, vous pouvez lancer en root la commande `a2enmod userdir`.
- (d) Après avoir activé un module, il faut relancer Apache, par exemple avec la commande `apache2ctl restart` en root.

1.6 Une fois le module `userdir` activé, chaque utilisateur peut créer son propre site Web dans le dossier `~/public_html` (dossier `public_html` dans le home de l'utilisateur). Créez ce répertoire et mettez y un fichier `index.html`. Vérifier que ce fichier est bien accessible via l'url `http://localhost/~toto`.

note : vous remplacerez bien entendu `toto` par votre login.

Exercice 2 - Ajout du support de PHP.

Afin de faire un site Web dynamique, on a besoin d'utiliser un langage de programmation. On utilisera ce langage pour générer en outre du code HTML qui sera utilisé dans les pages envoyés aux clients. C'est donc le serveur Web qui se charge d'exécuter le code avant d'envoyer le résultat aux clients. Nous utiliseront ici le langage PHP dans sa version 5.

2.1 Pour ajouter le support du PHP au serveur Apache, on peut par exemple installer le paquet `libapache2-mod-php5` (ou nom similaire). Faites le, et vérifiez que le module `php5` a bien été activé par défaut en consultant le contenu du dossier `/etc/apache2/mods-enabled`.

2.2 Pour tester que PHP est bien supporté, créez le fichier `info.php` dans le répertoire `/var/www` avec le contenu suivant :

```
<?php phpinfo(); ?>
```

Rendez vous alors sur la page `http://localhost/info.php`.

2.3 La configuration de PHP se fait principalement via le fichier `php.ini` (souvent situé dans le dossier `/etc/php5/apache2`). Ouvrez ce fichier et répondez aux questions suivantes :

- (a) le mode Safe de PHP est-il activé ?
- (b) l'affichage des erreurs est-il activé ? (si non, ça peut être une bonne idée de l'activer)
- (c) Avec combien de chiffres de précision PHP effectue-t-il les calculs avec des nombres flottants ?

note : Si vous modifiez le fichier `php.ini`, n'oubliez pas de relancer Apache afin que les modifications soient prisent en compte.

2.4 Créez maintenant un fichier `info.php` (avec le même contenu que précédemment) dans votre dossier `~/public_html`. Vérifiez que vous pouvez utiliser PHP pour le site Web de votre compte utilisateur.

Sous Debian et Ubuntu, ce n'est pas toujours le cas par défaut. Il vous faudra alors modifier le fichier `/etc/apache2/mods-available/php5.conf` en suivant les consignes données en commentaire, puis relancer Apache.

Exercice 3 - Installation et configuration de MySQL.

Il nous faut encore un système de gestion de bases de données (SGBD) afin de pouvoir stocker les données dynamiques. Nous allons ici installer MySQL.

3.1 Installez le paquet `mysql-server`. On vous demandera normalement de choisir un mot de passe administrateur. Choisissez en un et **notez le** pour ne pas l'oublier.

3.2 Le fichier de configuration principal de MySQL est `/etc/mysql/my.cnf`. Ouvrez le et répondez aux questions suivantes :

- (a) Quel port de votre machine le serveur MySQL utilise-t-il ?
- (b) Dans quel dossier les données seront-elles stockées ?
- (c) Depuis quelles machines a t'on accès au serveur MySQL ?

3.3 Pour vérifier que MySQL est bien installé, lancez (en tant qu'utilisateur normal) la commande `mysql -u root -p`. Saisissez votre mot de passe administrateur pour MySQL. Vous devriez obtenir un nouveau prompt. Essayez alors successivement les commandes suivantes :

```
> help;
> show databases;
> use mysql;
> show tables;
> select Host,User,Password from user;
```

Qu'obtenez vous ?

note : ne pas oublier le ";" à la fin de chaque commande.

3.4 On peut maintenant créer un compte utilisateur grâce à la commande

```
> CREATE USER 'toto'@'%' IDENTIFIED BY 'motdepasse';
```

Créez ainsi un compte MySQL correspondant à votre compte UNIX. Vérifier que la table `user` de la base de données `mysql` contient bien une nouvelle entrée. Que signifie '%' dans cette commande ?

3.5 Créez une nouvelle base de données appelée `test` grâce à la commande

```
> CREATE DATABASE test;
```

et donner les droits complets sur cette base de données à votre compte utilisateur via

```
> GRANT ALL ON test.* TO 'toto'@'%';
```

3.6 Quittez MySQL, et reconnecter vous en utilisant cette fois ci votre compte utilisateur grâce à la commande `mysql -u <login> -p`. Créez la table `notes` correspondant à :

<u>numID</u>	nom	prenom	noteBDD	noteComSys
10001	Ullman	Jeffrey	20	17
10002	Doe	John	7	9
20001	Doe	John	11	10
20002	Palsistaime	James	10	-3
20003	Broggs	Joe	14	11
20004	Knuth	Donald	99	99

On utilisera les commandes MySQL `CREATE TABLE` et `INSERT`, ainsi que `SELECT * FROM notes;` pour vérifier que tout est ok.

3.7 Supprimez la première ligne sur John Doe, ainsi que les lignes contenant des notes non comprises entre 0 et 20.

3.8 Maintenant, créez un compte MySQL pour votre voisin. Donnez lui le droit de faire des requêtes de type `SELECT` dans la base de données `test` (changer `ALL` en `SELECT`). Votre voisin peut-il accéder à votre serveur MySQL via la commande `mysql -h <ip> -u <login> -p?` (si non, modifier la configuration de MySQL pour rendre cela possible). Vérifiez que votre voisin ne peut effectivement faire que des requêtes de type `SELECT` dans la base de données `test`.

Exercice 4 - Accès à la base de données en PHP.

Vous avez maintenant tout ce qu'il faut pour créer votre site Web dynamique. . .

4.0 . . . ou presque. Vérifier dans la sortie de `phpinfo()` que le support de MySQL est bien installé (il faut pour cela que le driver `mysql` de PDO soit présent). Dans le cas contraire, installez le paquet `php5-mysql` et relancez Apache.

4.1 Écrivez un code PHP qui permet d'afficher le contenu de la table `notes` dans la base de données `test` dans une page Web.

rappels :

– pour accéder à la base de données `test`, faire

```
$pdo = new PDO('mysql:host=localhost;dbname=test','login','motdepasse');
```

– pour récupérer le résultat d'une requête, utiliser une variable temporaire et une boucle `foreach` comme suit :

```
$res = $pdo->query('insérer votre requete SQL ici');  
foreach ($res as $row) {  
    ... $row['nom'] ...  
}
```

4.2 Écrivez un code PHP qui permet d'ajouter une entrée à la table `notes`. On vérifiera que les notes sont entre 0 et 20 avant de faire l'ajout.

4.3 Faites une page HTML avec un formulaire de façon à pouvoir saisir une nouvelle entrée destinée à la table `notes`. L'action associée au formulaire sera d'appeler votre code pour la question précédente.

rappel : pour faire un formulaire HTML, compléter/adapter le code

```
<form action="myform.php" method="post">  
...  
<input type="submit" value="ok">  
</form>
```

4.4 Utiliser ce que vous avez fait lors du premier TD pour afficher un histogramme représentant le nombre d'étudiants en fonction de la note au cours « bases de données ».

Exercice 5 - Installation et utilisation de PhpMyAdmin.

PhpMyAdmin est une interface Web pour administrer un serveur MySQL.

5.1 Installez le paquet `phpmyadmin`. Ce programme aura besoin du mot de passe administrateur pour MySQL, fournissez le quand on vous le demande. Une fois l'installation terminée, vérifiez que la page `http://localhost/phpmyadmin` existe, et que vous pouvez vous connecter soit avec votre compte administrateur, soit avec votre compte utilisateur.

5.2 Reprenez les questions 3.4 à 3.8 (en changeant `test` en `test2`, et en changeant éventuellement de voisin).