

Ajoutez une base de données MySQL à votre site web

Sur votre serveur, vous avez maintenant un serveur web Apache fonctionnel et le langage PHP installé. La dernière chose qui vous manque pour pouvoir installer la plupart des applications web, c'est une base de données pour stocker vos informations dynamiques. Je vais donc maintenant vous guider pour installer la base de données libre de référence : **MySQL**.



En 2008, la société MySQL AB qui centralise le développement de MySQL a été rachetée. Elle est maintenant la propriété d'Oracle. Les désaccords entre une partie des développeurs et cette société autour de la gestion du développement de la version libre de MySQL ont engendré la création de plusieurs projets parallèles (des forks) : **MariaDB**, **Percona Server**, etc. Ils peuvent parfois représenter des alternatives intéressantes en terme de performance ou de fonctionnalités tout en restant compatibles avec MySQL. Vous avez donc intérêt à suivre leur développement.

Installez MySQL

Pour installer MySQL, il vous suffit d'installer le paquet `mysql-server` :

```
$ sudo apt-get install mysql-server
```

La première chose à faire ensuite, c'est **sécuriser l'installation par défaut** de MySQL qui n'est pas adaptée à un environnement de production. MySQL vous fournit l'utilitaire

`mysq_secure_installation` pour vous faciliter la vie :

```
$ sudo mysq_secure_installation
```

Le jeu avec cet utilitaire, c'est de répondre *oui* à toutes les questions. Ça va notamment vous permettre de **définir un mot de passe administrateur** (*root*) pour votre base de données, de n'autoriser que sa connexion en local, d'interdire les connexions anonymes et de supprimer la base de test.

Maintenant que MySQL est installée, vous pouvez faire un premier test de connexion en utilisant le compte root.

Connectez-vous à MySQL

Le compte root n'est maintenant autorisé à se connecter à MySQL qu'en local depuis le serveur. Vous pouvez vous connecter par la commande :

```
$ mysql -h localhost -u root -p
```

mysql	c'est le nom du client texte qui permet de se connecter à MySQL
-h localhost	précise le nom de l'hôte auquel vous voulez vous connecter. localhost est l'hôte par défaut donc vous pouvez vous passer de cette option pour les connexions locales
-u root	indique le nom de l'utilisateur avec lequel vous voulez vous connecter
-p	demande d'afficher un prompt pour entrer le mot de passe de manière interactive

Un fois connecté, vous pouvez par exemple afficher les bases de données existantes par :

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
| sys               |
+-----+
mysql> quit;
```

Ici les 4 bases affichées sont toutes des bases de données gérées par MySQL pour assurer son fonctionnement.



Je ne vais pas rentrer dans les détails de ce qu'est un SGBD ni du langage SQL utilisé dans le client mysql pour interagir avec vos bases mais **en tant qu'administrateurs je vous conseille vivement d'être à l'aise pour faire quelques opérations de base** : recherche simple, lecture des informations, modification et suppression de lignes, gestion des utilisateurs, etc. Pour cela, je vous conseille la lecture du cours "[Administrez vos bases de données avec MySQL](#)".

Pour comprendre un peu mieux le fonctionnement de MySQL et de ses options de configuration, vous devez connaître les différents **moteurs de stockage**.

Découvrez les deux moteurs de stockage MyISAM et InnoDB



Dans MySQL, la manière dont les données sont gérées et accédées est définie au niveau de chaque table. Cette gestion est assurée par des moteurs de stockage, chacun ayant ses

caractéristiques propres. Les deux types de moteurs principaux que vous devez connaître avec MySQL sont :

- **MyISAM** : c'est le moteur le plus ancien et le plus "basique". Il est très rapide pour effectuer des recherches ou des opérations en lecture mais ne gère pas les transactions (cf encadré) ni les clés étrangères.
- **InnoDB** : c'est un moteur plus récent. Il est plus robuste dans le sens où il gère les transactions mais il est un peu plus lent pour les opérations de lecture. Il gère aussi les clés étrangères.



Une **transaction** est une opération dite **ACID** pour atomique, cohérente, isolée et durable. En gros, ça veut dire qu'une opération est soit réalisée, soit non-réalisée mais n'est jamais quelque part entre les deux. C'est un critère important pour certaines opérations comme les opérations bancaires par exemple : si un transfert d'argent est réalisé, le compte de l'émetteur est prélevé ET l'argent est versé au bénéficiaire mais jamais l'un sans l'autre.

Une même base de données peut contenir à la fois des tables s'appuyant sur un moteur et des tables s'appuyant sur un autre. Rassurez-vous, en général ce sont les développeurs qui définissent le type de moteur à utiliser pour chaque table. Malgré tout, ces choix ont des impacts sur la gestion de la base de données et il est important pour un administrateur de comprendre le principe général (et plus si affinité si vous voulez vraiment optimiser les performances).

Maintenant que vous en savez un peu plus sur le fonctionnement de MySQL, vous pouvez partir à la découverte de ses fichiers de configuration et de stockage.

Explorez les fichiers utilisés par MySQL



Les fichiers de configuration utilisés par MySQL sont dans `/etc/mysql` . Vous y découvrirez :

- les fichiers `my.cnf` et `mysql.cnf` : fichiers généraux par défaut, ils ne contiennent maintenant que des lignes pour inclure les fichiers dans `conf.d` et `mysql.conf.d` car la configuration a été découpée en différents fichiers dans ces répertoires
- un fichier `debian.cnf` : sur les systèmes Debian (dont Ubuntu), un utilisateur `debian-sys-maint` ayant les droits administrateur est créé pour effectuer les opérations de maintenance de MySQL. Ce fichier contient les informations de connexion de cet utilisateur et ne doit pas être modifié.
- un fichier `mysql.conf.d/mysql.d.cnf` : il contient la plupart des options de configuration générale du serveur.

Si vous ouvrez ce dernier fichier `/etc/mysql/mysql.conf.d/mysql.d.cnf` , vous verrez que les fichiers de configuration de MySQL sont découpés en sections dont le titre est entre crochets. La configuration générale est dans la section `[mysqld]` . Vous y trouverez quelques variables notables :

<code>user = mysql</code>	Définit l'utilisateur système utilisé par MySQL.
<code>port = 3306</code>	Définit le port TCP sur lequel écoute MySQL
<code>datadir = /var/lib/mysql</code>	Définit le répertoire dans lequel MySQL stocke ses fichiers
<code>bind-address = 127.0.0.1</code>	Définit les IP sur lesquelles écoute MySQL. Sur Ubuntu, par défaut MySQL n'écoute qu'en local ce qui est une bonne chose en terme de sécurité. Comme votre serveur est sur la même machine que MySQL, vous pouvez laisser comme ça pour l'instant mais c'est à adapter si vous voulez vous connecter à MySQL depuis un autre serveur.
<code>log_error = /var/log/mysql/error.log</code>	Définit le log d'erreurs de MySQL

Au delà de ces quelques options de base, vous y trouverez un grand nombre de paramètres d'optimisation. La modification de ces options demande d'avoir une bonne connaissance du fonctionnement de MySQL mais peut changer radicalement les performances de MySQL.

MySQL a beau être une base de données relationnelle, elle utilise bien sûr des fichiers pour enregistrer ses informations de manière pérenne. Comme vous avez pu le voir dans la configuration, ces fichiers sont dans `/var/lib/mysql`. Assurez-vous donc d'avoir assez d'espace disponible dans ce répertoire pour stocker vos bases de données.

Allez regarder de plus près le contenu du répertoire `/var/lib/mysql`. Dans ce répertoire de stockage, vous y trouverez un sous-répertoire pour chacune de vos bases de données. Dans ces répertoires de base de données, vous trouverez pour chaque table un fichier `NOM_DE_LA_TABLE.frm` qui décrit la structure de la table en question. Les autres fichiers dépendent du moteur de stockage que vous utilisez pour vos tables.

Pour les tables MyISAM, vous trouverez notamment des fichiers contenant les données avec une extension `.myd` et des fichiers contenant les données d'index avec une extension `.myi`.

Pour les tables InnoDB, les données de toutes les tables sont stockées dans le fichier `/var/lib/mysql/ibdata1`. Ce fichier peut donc devenir assez gros et il est possible de définir dans la configuration MySQL comment créer automatiquement de nouveaux fichiers au delà d'une certaine taille. Dans le même répertoire, les fichiers `ib_logfile0` et `ib_logfile1` sont des fichiers tampon utilisés pour gérer les transactions et améliorer les performances.



Pour sauvegarder vos bases MySQL, il pourrait être tentant de faire une sauvegarde de votre répertoire `/var/lib/mysql`. Malheureusement, si des écritures sont effectuées sur la base pendant votre sauvegarde, vous risquez d'avoir une sauvegarde incohérente. Pour effectuer vos sauvegardes ou vos extraits de base, il est donc obligatoire de passer

par un client prévu pour cette tâche. Le plus courant est le client `mysqldump` qui exporte les données des bases au format SQL.



Pour pouvoir utiliser MySQL depuis vos applications PHP, vous aurez besoin d'installer les modules PHP permettant d'accéder à MySQL. Pour cela, sous Ubuntu, il vous suffit d'installer le paquet `php-mysql` .

Vous disposez maintenant d'un serveur LAMP complet ce qui vous permettra de pouvoir installer les nombreuses applications PHP disponibles dont le célèbre système de gestion de contenu Wordpress.

Dans la partie suivante, vous verrez comment bénéficier également des applications basées sur le langage Java et vous apprendrez à améliorer les performances et la sécurité de votre infrastructure par l'utilisation de proxies.

En résumé



- MySQL est le serveur de base de données libre de référence.
- vous pouvez définir un mot de passe administrateur et sécuriser votre installation grâce à l'utilitaire `mysql_secure_installation` .
- différents moteurs de stockage permettent de gérer la manière dont les données sont stockées et accédées.
- les deux moteurs de stockage les plus courants sont **MyISAM** et **InnoDB**.

Le professeur



Etienne Lavanant

Étienne Lavanant, Ingénieur Systèmes diplômé de Télécom Sud-Paris, travaille en tant que Freelance sur Paris