

MySQL

Généralités

MySQL est un serveur de base de données opensource. Il constitue l'un des serveurs le plus utilisé au monde. (<http://www.mysql.com/>).

Installation

Dans le cas d'un système non supporté ou si la volonté d'installer une version supérieure est requise, récupérer les RPM correspondants à la distribution et à l'architecture du système sur le site officiel (<http://dev.mysql.com/downloads/mysql/5.0.html#downloads>) et l'installer avec la commande rpm - ivh nompaquetage.

Dans le cas d'un système supporté, l'installation peut être réalisée avec l'une des commandes suivantes. Le seul souci est que nous ne pouvons choisir la version à installer. Sous RHEL ES4.6.

```
up2date mysql-server
up2date mysql-client
```

Sous RHEL5.

```
yum install mysql-server
yum install mysql-client
```

Sous Debian Squeeze.

```
aptitude install mysql-server-5.1
```

Le mot de passe root est demandé à l'installation pour tout de suite le positionner.

Si on installe MySQL à partir de paquets RPM, il faut savoir qu'il existe, le paquet server, le paquet client et il existe deux librairies MySQL : community et compat.

Compat sert à faire le pont avec les anciennes versions de MySQL. Il se trouve que pas mal des logiciels que nous utilisons ont besoin de ces anciennes librairies. On n'installe pas les deux. Elles sont incompatibles entre elles. C'est soit compat, soit la community.

Comment le savoir ? Tout simplement lorsqu'on veut installer un RPM de nos logiciels qui ont besoins de MySQL, il y aura une erreur sur libmysql. libmysqlclient.so.10 is needed by...

Configuration

Configuration générale

A l'installation de la version communautaire de MySQL, la configuration se situe dans `/usr/share/mysql/*.cnf`.

Il y a plusieurs fichiers de configuration qui dimensionne le serveur MySQL en fonction de la configuration matérielle du serveur sur lequel il est.

Emplacement	Fonction
<code>/usr/share/my-huge.cnf</code>	Fichier de configuration à copier dans <code>/etc</code> en fonction notre configuration matériel et de ce qu'on veut faire du serveur.
<code>/usr/share/my-innodb-heavy-4G.cnf</code>	Idem.
<code>/usr/share/my-large.cnf</code>	Idem.
<code>/usr/share/my-medium.cnf</code>	Idem.
<code>/usr/share/my-small.cnf</code>	Idem.

Un commentaire en début de chaque fichier explique pour quel type de configuration cela peut être intéressant.

Configuration spécifique

Lancer le script de sécurisation qui permet au passage d'attribuer un mot de passe à root parce que par défaut, il n'y en a pas. Très clairement, si rien n'est fait MySQL est une passoire par défaut!

```
[root@NOMSERVEUR mysql]# mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
```

```
In order to log into MySQL to secure it, we'll need the current
password for the root user. If you've just installed MySQL, 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):
OK, successfully used password, moving on...
```

```
Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.
```

```
Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

```
By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL 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
... skipping.
```

```
By default, MySQL 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 MySQL
installation should now be secure.
```

```
Thanks for using MySQL!
```

Redémarrer la base de données après (`/etc/init.d/mysqld restart`).

Pour les bases de données, je créé un autre disque que je monte par exemple sur `/data` (Cf. [manipulations_disque_simple_lvm](#)). J'ai donc un point de montage `/data` qui pointe sur le deuxième disque nouvellement ajouté.

J'ai supprimé le fichier `/etc/my.cnf` par défaut (copie de sauvegarde au début quand même) et j'ai copié le fichier `/usr/share/mysql/my-large.cnf` à la place

```
cp /usr/share/mysql/my-large.cnf /etc
```

Plusieurs gabarit sont fournis par défaut pour adapter la configuration en fonction de la taille de la machine (`my-huge.cnf`, `my-large.cnf`, `my-medium.cnf`,...).

Un dossier `/data/sqbd` a été créé pour remplacer le dossier par défaut `/var/lib/mysql`

```
chown mysql:mysql /data/sqbd
```

Arrêter le serveur `/etc/init.d/mysqld stop` Supprimer le dossier `/var/lib/mysql` Créer un lien symbolique nommé `mysql` dans `/var/lib` qui pointe sur le dossier `/data/sqbd`

```
cd /var/lib
ln -s /data/sqbd mysql
```

Options modifiés de `my.cnf` en partant à l'origine de `my-large.cnf`

```
[client]
socket          = /data/sqbd/mysql.sock

# The MySQL server
[mysqld]
datadir         = /data/sqbd
socket          = /data/sqbd/mysql.sock

# Try number of CPU's*2 for thread_concurrency
thread_concurrency = 4
max_connections=200
```

Très important la directive qui active ou désactive les logs binaires de MySQL. L'activation de ces logs est obligatoire dans les cas de serveur maître et esclave ce qui n'est pas le cas pour nous. Les logs binaires contiennent toutes les requêtes qui modifie la BD. Le problème est la taille que prennent ces logs. On ne peut que les purger sur une durée d'un jour ce qui n'est pas assez fin. La rotation est impossible ce qui m'a conduit à les désactiver.

```
# binary logging - not required for slaves, but recommended
#log-bin=mysql-bin
```

Toujours dans la section `mysqld`, j'ai décommenté et modifié la configuration concernant `InnoDB` car c'est le type de table que nous allons utiliser.

```
# Uncomment the following if you are using InnoDB tables
#innodb_data_home_dir = /data/sqbd/
#innodb_data_file_path = ibdata1:10M:autoextend

#innodb_log_group_home_dir = /data/sqbd/
#innodb_log_arch_dir = /data/sqbd/

# You can set .._buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
innodb_buffer_pool_size = 256M
innodb_additional_mem_pool_size = 20M
# Set .._log_file_size to 25 % of buffer pool size
innodb_log_file_size = 64M
#innodb_log_buffer_size = 8M
#innodb_flush_log_at_trx_commit = 1
#innodb_lock_wait_timeout = 50

# on peut créer un fichier par table innodb
```

```
#innodb_file_per_table

# Gestion des logs binaires (mysql-bin.0000X avec index mysql-bin.index)
#Default 1G = 1073741824
#150Mo * 1024 * 1024 = 157286400 octets
#60Mo * 1024 * 1024 = 62914560 octets

#max_binlog_size=62914560

#Default 4Go = 4294967295 octets
#1073741824 = 1Go
#max_binlog_cache_size=1073741824
#expire_logs_days=1
```

Redémarrer la base de données (/etc/init.d/mysqld restart) et MySQL est prêt à être utilisé.

Création initiale base et utilisateur

Création d'une base de données avec un utilisateur qui ne peut que se connecter en local et qui dispose de tous les droits sur la base mais pas la possibilité de donner des permissions à d'autres comptes.

```
CREATE DATABASE <databasename>
CREATE USER '<user>'@'localhost' IDENTIFIED BY 'mot_de_passe';
GRANT ALL PRIVILEGES ON <databasename>.* TO '<user>'@'localhost';
FLUSH PRIVILEGES;
```

La clause WITH GRANT OPTION de la commande GRANT donne à l'utilisateur le droit de donner les droits qu'il possède à d'autres utilisateurs. La plus grande prudence est recommandée pour cette commande, c'est qu'on ne veut surtout pas.

```
show grants for '<user>'@'localhost';
```

Commande inverse de suppression de droits.

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM '<user>'@'localhost';
```

Dump

Création de dump à chaud avec la commande mysqldump. Elle convient pour les tables MyISAM et InnoDB.

```
mysqldump --databases OPTIONS DB1 DB2...
mysqldump --all-databases OPTIONS DB1 DB2...

mysqldump -u root -p nombd > /tmp/nombd.sql
```

Sauvegarde

Sauvegarde directe des fichiers.

Commande à exécuter connecté avec le client mysql.

- backup table : convient uniquement aux tables myisam.
- restore table : convient uniquement aux tables myisam.
- mysqlhotcopy copie les fichiers de tables myisam à chaud.

Pour effectuer la même chose avec des bases innodb, il n'existe pas d'outil gratuit stable pour le moment. Il faut suivre cet outil lorsqu'il aura bien mûri :

<http://www.percona.com/docs/wiki/percona-xtrabackup:start>.

Maintenance des bases

Il existe 4 maintenances.

- check
- analyze
- repair
- optimise

En MyISAM, les 4 fonctionnent. En Innodb, seul le check et l'optimize ont une utilité.

En Innodb, analyze n'a pas d'intérêt et repair n'est pas supporté et peut entraîner des pertes de données.

Exemple avec le check.

```
mysqlcheck -u root -pPASSWORD --all-databases -s
```

Exemple avec l'optimize

```
mysqlcheck -u root -pPASSWORD --all-databases -optimize
```

Suppression manuelle des logs binaires Voir les logs en cours.

```
mysql> show master logs;
```

```
mysql> PURGE MASTER LOGS TO 'mysql-bin.000012';
```

Signifie supprimer tout les logs précédents le log 12. Cela supprime les logs de 1 à 11.

From:
<https://wiki.ouiehoutca.eu/> - **kilsufi de noter**

Permanent link:
<https://wiki.ouiehoutca.eu/mysql>

Last update: **2021/01/21 21:42**

