

MySQL en mode Multi Instances sur un même serveur

I► Installer plusieurs instances de MySQL sur le même serveur peut être intéressante pour optimiser l'accès au serveur et gérer les bases de données de plusieurs clients en hébergement mutualisé :

Cela peut être aussi intéressant de pouvoir ajuster finement des paramètres tels que `max_connections` pour chacune des instances de MySQL en fonction des applications et/ou des clients.

On peut également faire cohabiter plusieurs serveurs avec plusieurs versions différentes.

? Méthode sous LINUX avec MYSQLD_MULTI (script PERL)

1. Le serveur MySQL doit être correctement installé et bien configuré
2. Stopper le service : `/etc/init.d/mysql stop`
3. Il va falloir créer les répertoires qui vont stocker toutes les informations propres à chaque instance de serveur(ces infos par défaut sont stockés sous `/var/lib/mysql`)

```
mkdir /var/lib/mysql1 /var/lib/mysql2 /var/lib/mysql3
```

Mise à jour des propriétaires et des droits

```
chown -R mysql:mysql /var/lib/mysql1 /var/lib/mysql2 /var/lib/mysql3
```

```
chmod 700 /var/lib/mysql1 /var/lib/mysql2 /var/lib/mysql3
```

Copie des fichiers (R=réursive et p=parent)

```
cp -Rp /var/lib/mysql/mysql/ var/lib/mysql1/  
cp -Rp /var/lib/mysql/mysql/ var/lib/mysql2/  
cp -Rp /var/lib/mysql/mysql/ var/lib/mysql3/
```

4. Nous allons maintenant générer le fichier de conf de mysql_multi : (avec l'utilitaire mysql_multi)

```
mysqld_multi -example > /etc/mysql/multi_my.cnf
```

```
#-----
```

[mysqld_multi]

```
mysqld = /usr/bin/mysqld_safe  
mysqladmin = /usr/bin/mysqladmin  
user = root  
password = debian  
log = /var/log/mysqld_multi.log
```

[mysqld1]

```
socket = /tmp/mysql.sock1  
#chemin du socket (différent pour chaque serveur)  
port = 3309  
pid-file = /var/lib/mysql1/hostname.pid1  
datadir = /var/lib/mysql1  
#répertoire de stockage pour chaque serveur  
language = /usr/share/mysql/english  
user = tux  
#propriétaire du processus
```

[mysqld2]

```
socket = /tmp/mysql.sock2  
port = 3307  
pid-file = /var/lib/mysql2/hostname.pid2  
datadir = /var/lib/mysql2  
language = /usr/share/mysql/english  
user = tux
```

[mysqld3]

```
socket = /tmp/mysql.sock6
```

```
port = 3311
pid-file = /var/lib/mysql3/hostname.pid3
datadir = /var/lib/mysql3
language = /usr/share/mysql/french
user = tux
#_____
```

1. Nous allons maintenant démarrer ou arrêter les instances de MySQL

```
mysqld_multi -defaults-file=/etc/mysql/multi_my.cnf
report
```

Reporting MySQL servers

MySQL server from group: mysqld1 is not running

MySQL server from group: mysqld2 is not running

MySQL server from group: mysqld3 is not running

2. **mysqld_multi -defaults-file=/etc/mysql/multi_my.cnf
start 1,2,3**

```
mysqld_multi -defaults-file=/etc/mysql/multi_my.cnf
start 1-3
```

```
mysqld_multi -defaults-file=/etc/mysql/multi_my.cnf stop
1,3
```

3. Se connecter à ces instances de MySQL : Pour se connecter à MySQL par défaut, nous utilisons la commande :

```
mysql -uLogin -p (ex mysql -uroot -p)
```

Pour se connecter aux différentes instances du Server MySQL,

nous devons spécifier le **socket** précis : ex: **mysql -uroot**

```
-socket=/tmp/mysql.sock1 -p Ce socket est bien sûr défini
```

```
dans /etc/mysql/multi_my.cnf
```

? Méthode sous WINDOWS

On crée deux répertoires data pouvant contenir les databases respectivement sous /MYSQL/MY1 et /MYSQL/MY2.

ces répertoires contiennent également un fichier my.ini qui

sera propre au démarrage de l'occurrence du serveur concerné.
le port doit obligatoirement être différent ...



On peut ensuite se connecter aux multiples instances du serveur MySQL en utilisant le client de son choix.



Michel BOCCIOLESI