

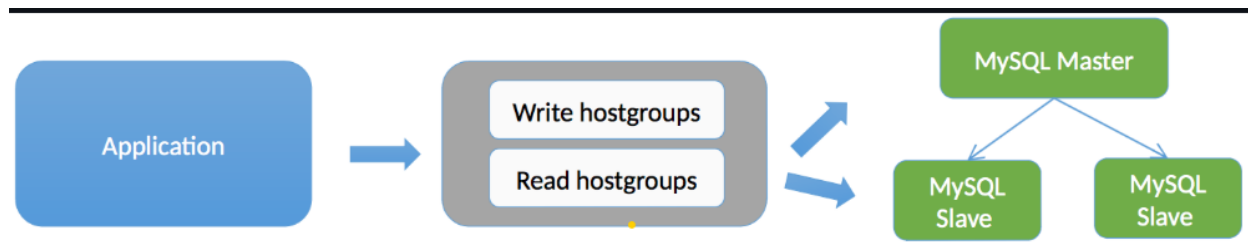
# Gestion d'hôtel

16/03/2023

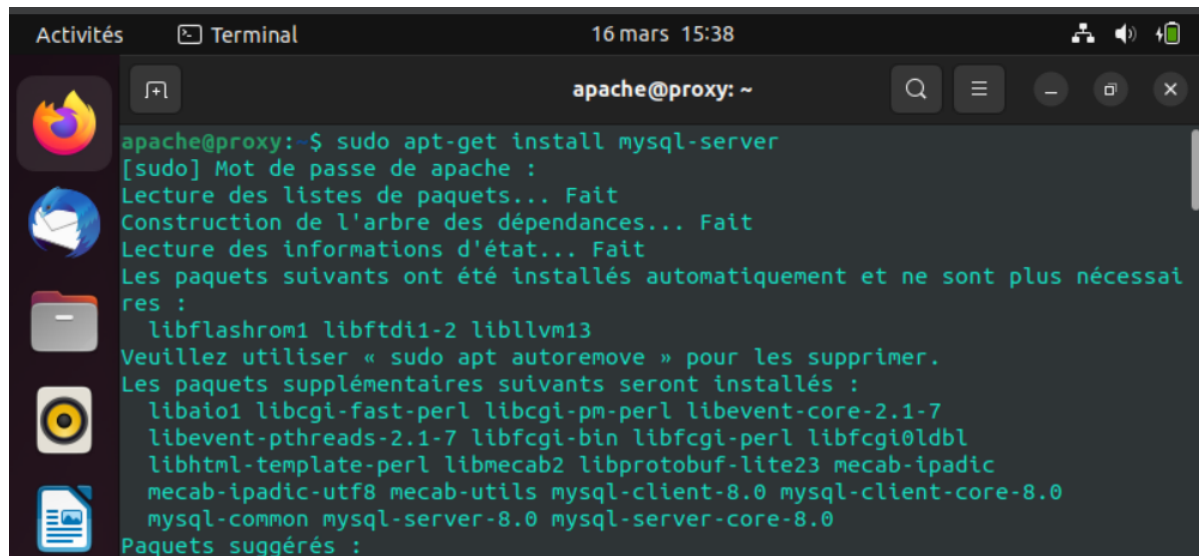
Mise en place d'une base de données MySQL répartie sur 3 nœuds en utilisant ProxySQL.

Nous allons effectuer ce travail en utilisant trois machines virtuelles dont l'une sera le master qui contiendra également le Proxy et les deux autres seront les slaves.

L'architecture que nous cherchons à avoir est la suivante:



1. Installation de MySQL-server et configuration sur chaque machine:



A terminal window titled "Terminal" with the date and time "16 mars 15:38". The user is logged in as "apache@proxy: ~". The command executed is "sudo apt-get install mysql-server". The output shows the password prompt, package list reading, dependency tree construction, and state information reading. It lists automatically installed packages (libflashrom1, libftdi1-2, libllvm13) and suggests removing them with "sudo apt autoremove". It then lists additional packages to be installed, including various Perl modules, libevent, libfcgi, libhtml-template-perl, libmecab, and MySQL client and server packages. The terminal window has a sidebar with icons for Firefox, a mail client, a file manager, a media player, and a document viewer.

```
apache@proxy:~$ sudo apt-get install mysql-server
[sudo] Mot de passe de apache :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessai
res :
  libflashrom1 libftdi1-2 libllvm13
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
  libaio1 libcgi-fast-perl libcgi-pm-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl
  libhtml-template-perl libmecab2 libprotobuf-lite23 mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server-8.0 mysql-server-core-8.0
Paquets suggérés :
```

Pour la configuration, nous allons modifier le fichier “/etc/mysql/mysql.conf.d/mysqld.cnf”:

```

GNU nano 6.2 /etc/mysql/mysql.conf.d/mysqld.cnf
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram
[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql

# If MySQL is running as a replication slave, this should be

```

Lecture de 78 lignes

^G Aide    ^O Écrire    ^W Chercher    ^K Couper    ^T Exécuter  
 ^X Quitter    ^R Lire fich.    ^\ Remplacer    ^U Coller    ^J Justifier

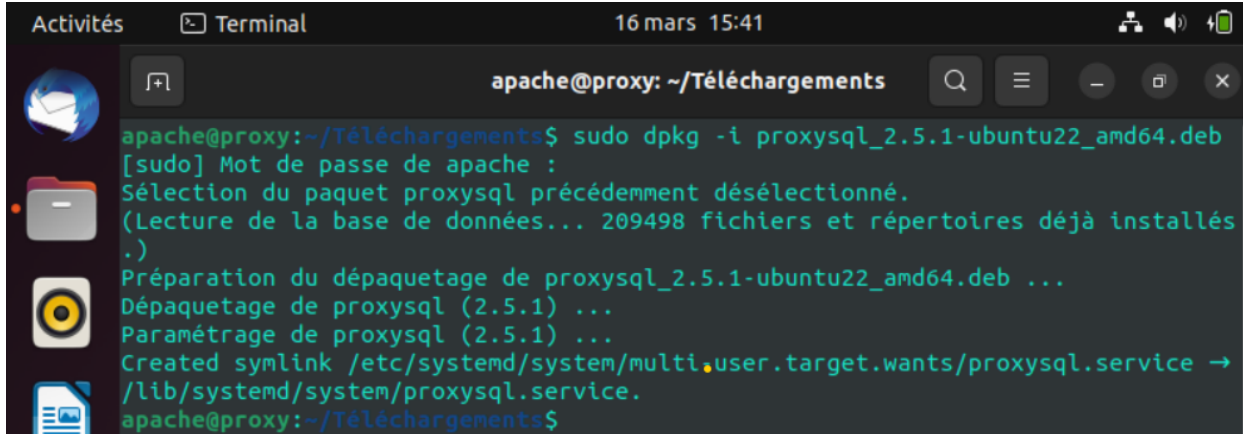
```

GNU nano 6.2 /etc/mysql/mysql.conf.d/mysqld.cnf
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address        = 0.0.0.0
mysqlx-bind-address = 127.0.0.1
#
server-id            = 1
log_bin              = /var/log/mysql/mysql-bin.log

```

Veiller à avoir le numéro de server-id différent pour chaque VM et exécuter la commande “sudo service mysql restart” pour redémarrer le serveur mysql pour qu’il prenne en compte les changements effectués.

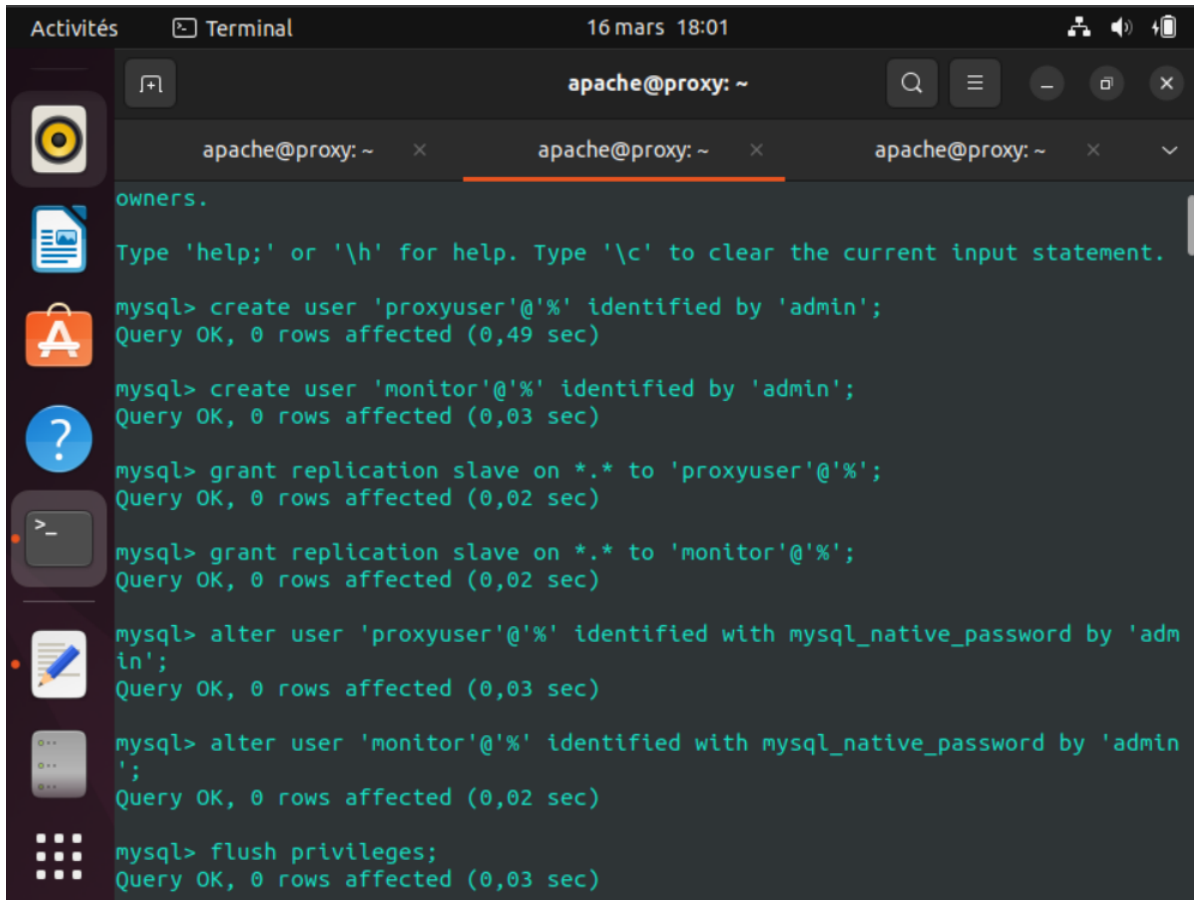
## 2. Installation de ProxySQL:



```
Activités Terminal 16 mars 15:41
apache@proxy: ~/Téléchargements
apache@proxy:~/Téléchargements$ sudo dpkg -i proxysql_2.5.1-ubuntu22_amd64.deb
[sudo] Mot de passe de apache :
Sélection du paquet proxysql précédemment désélectionné.
(Lecture de la base de données... 209498 fichiers et répertoires déjà installés
.)
Préparation du dépaquetage de proxysql_2.5.1-ubuntu22_amd64.deb ...
Dépaquetage de proxysql (2.5.1) ...
Paramétrage de proxysql (2.5.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/proxysql.service →
/lib/systemd/system/proxysql.service.
apache@proxy:~/Téléchargements$
```

## 3. Création des utilisateurs:

ProxySQL a besoin du nom d'utilisateur et du mot de passe de l'utilisateur qui peut se connecter aux serveurs principaux (la base de données MySQL) et également du moniteur qui supervise les paramètres de performance. Nous allons donner à ces utilisateur la possibilité de faire la réplication sur les noeuds.



The screenshot shows a terminal window titled 'Terminal' with the date '16 mars 18:01'. The prompt is 'apache@proxy: ~'. The terminal displays the following commands and their outputs:

```
owners.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> create user 'proxyuser'@'%' identified by 'admin';  
Query OK, 0 rows affected (0,49 sec)  
mysql> create user 'monitor'@'%' identified by 'admin';  
Query OK, 0 rows affected (0,03 sec)  
mysql> grant replication slave on *.* to 'proxyuser'@'%';  
Query OK, 0 rows affected (0,02 sec)  
mysql> grant replication slave on *.* to 'monitor'@'%';  
Query OK, 0 rows affected (0,02 sec)  
mysql> alter user 'proxyuser'@'%' identified with mysql_native_password by 'admin';  
Query OK, 0 rows affected (0,03 sec)  
mysql> alter user 'monitor'@'%' identified with mysql_native_password by 'admin';  
Query OK, 0 rows affected (0,02 sec)  
mysql> flush privileges;  
Query OK, 0 rows affected (0,03 sec)
```

#### 4. Configuration du master/slave:

Nous allons nous positionner sur les machines esclaves et nous allons taper les commandes ci-dessus:

```
mysql> change master to master_host='192.168.1.16', master_user='proxyuser', master_password='admin', master_log_file='mysql-bin.000001', master_log_pos=1681;

Query OK, 0 rows affected, 8 warnings (0,07 sec)

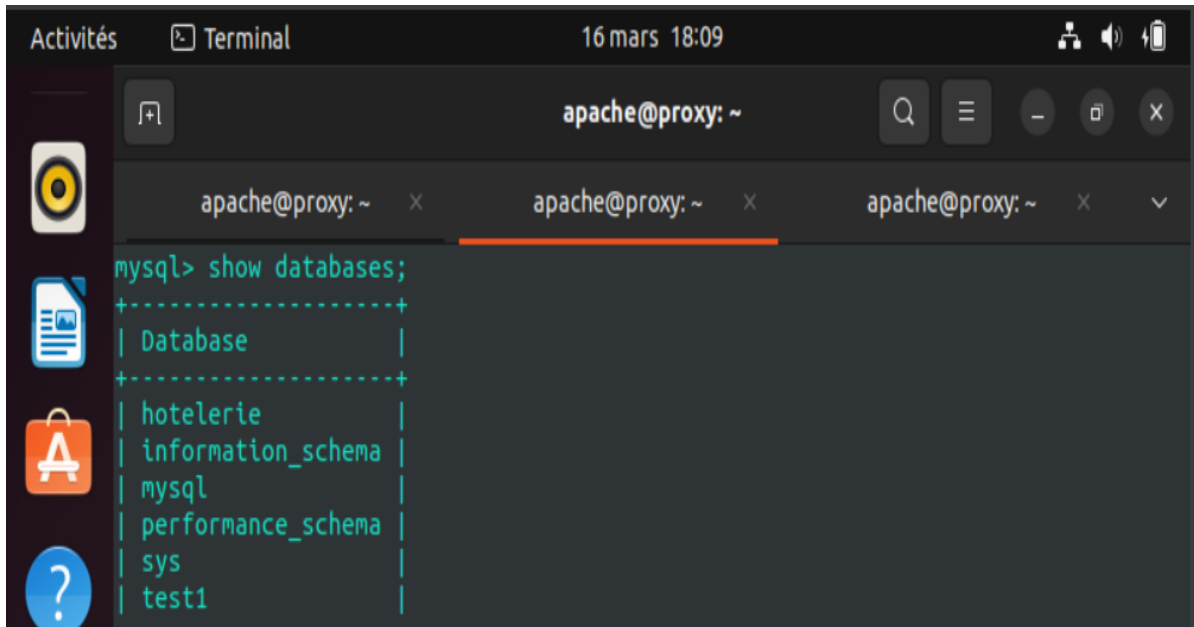
mysql> start slave;

Query OK, 0 rows affected, 1 warning (0,08 sec)

mysql> show slave status \G;
***** 1. row *****
      Slave_IO_State: Waiting for source to send event
        Master_Host: 192.168.1.16
        Master_User: proxyuser
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: mysql-bin.000001
    Read_Master_Log_Pos: 1681
        Relay_Log_File: proxy-relay-bin.000002
        Relay_Log_Pos: 326
    Relay_Master_Log_File: mysql-bin.000001
      Slave_IO_Running: Yes
     Slave_SQL_Running: Yes
```

On voit que les modifications ont été bien faites et le slave attend un événement du master. On répète cela pour le deuxième slave également.

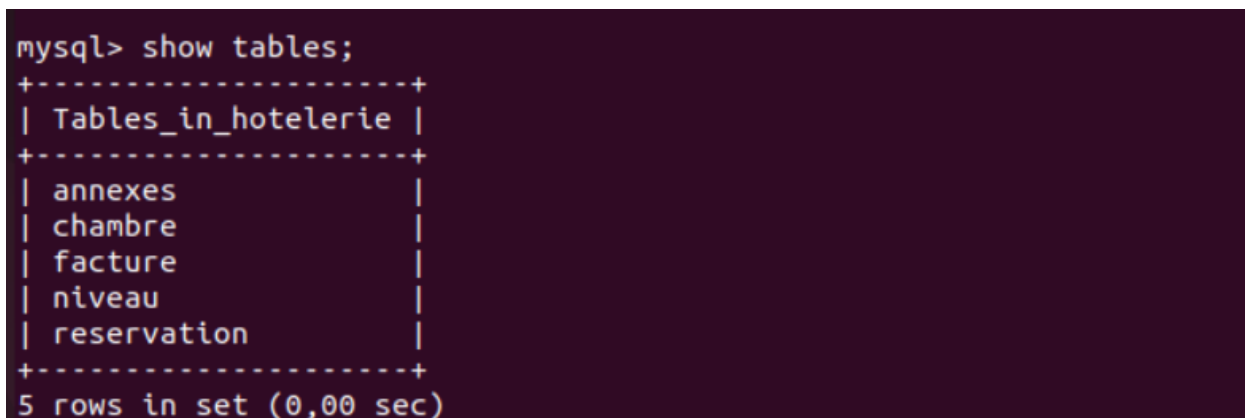
Maintenant, lorsqu'on fait des modifications sur le SGBD du master, on voit cela également sur les slaves. On a créé la base de données hôtellerie et on voit que cela a été répliqué sur chaque slave, de même que les tables et les insertions qui ont été faites.



The screenshot shows a terminal window titled 'Terminal' with the date and time '16 mars 18:09'. The prompt is 'apache@proxy: ~'. The command 'mysql> show databases;' has been executed, resulting in a table of databases. The table has one column named 'Database' and lists the following databases: 'hotelerie', 'information\_schema', 'mysql', 'performance\_schema', 'sys', and 'test1'.

```
mysql> show databases;
+-----+
| Database |
+-----+
| hotelerie |
| information_schema |
| mysql |
| performance_schema |
| sys |
| test1 |
+-----+
```

Cette capture d'écran ci-dessus montre les bases de données contenues dans le master. On peut voir les tables contenues dans le master au niveau du MySQL de chaque slave:



The screenshot shows a terminal window with the prompt 'mysql> show tables;'. The command has been executed, resulting in a table of tables. The table has one column named 'Tables\_in\_hotelerie' and lists the following tables: 'annexes', 'chambre', 'facture', 'niveau', and 'reservation'. The output also includes '5 rows in set (0,00 sec)'.

```
mysql> show tables;
+-----+
| Tables_in_hotelerie |
+-----+
| annexes |
| chambre |
| facture |
| niveau |
| reservation |
+-----+
5 rows in set (0,00 sec)
```

## 5. Configuration de ProxySQL:

Nous devons nous assurer que ProxySQL est conscient de ces choses :

- Les serveurs MySQL;
- L'utilisateur qui peut se connecter aux serveurs (à la fois celui de surveillance et celui de "requêtes");
- Qui est le Maître et qui sont les esclaves;

Nous allons mettre à jour les valeurs des variables "mysql-monitor\_username" et "mysql-monitor\_password" :

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
Admin> SET mysql-monitor_username='monitor';  
Query OK, 1 row affected (0,02 sec)  
  
Admin> SET mysql-monitor_password='admin';  
Query OK, 1 row affected (0,00 sec)
```

Après ceci, nous allons déplacer la configuration de la mémoire vers la couche Runtime et conserver la configuration sur le disque. Pour ce faire nous allons exécuter:

```
Admin> LOAD MYSQL VARIABLES TO RUNTIME;  
Query OK, 0 rows affected (0,00 sec)  
  
Admin> SAVE MYSQL VARIABLES TO DISK;  
Query OK, 155 rows affected (0,02 sec)
```

Et maintenant, nous feront connaître à ProxySQL les données et nous allons sauvegarder:

```
Admin> INSERT INTO mysql_users (username,password, default_hostgroup) VALUES ('  
proxyuser','admin',2);  
Query OK, 1 row affected (0,00 sec)  
  
Admin> LOAD MYSQL USERS TO RUNTIME;  
Query OK, 0 rows affected (0,00 sec)  
  
Admin> SAVE MYSQL USERS TO DISK;  
Query OK, 0 rows affected (0,03 sec)
```



## 6. Configuration des serveurs:

Vous pouvez ajouter un serveur à tous les groupes d'hôtes que vous souhaitez. Cela aidera sur le routage des requêtes et pour avoir une distribution de charge contrôlée, entre autres choses. Cependant, ProxySQL, dans un effort pour simplifier les choses, a le type spécial "mysql\_replication\_hostgroups" qui n'est rien d'autre qu'une façon de dire quel groupe d'hôtes détient le maître et lequel détient les esclaves. En quoi est-ce différent d'un groupe d'hôtes régulier ? Simplement : La tâche de déplacer les serveurs entre les groupes d'hôtes devient une opération automatique et ne dépend que d'une seule chose : la valeur de la variable `read_only`. Si un serveur a "`read_only = 1`", il fera partie du groupe `reader_host`. Sinon, est le maître et fait partie du groupe `writer_host`. Cela signifie que vous devez être très prudent avec cette variable. Une bonne pratique consistera à appliquer `read_only = 1` sur le fichier `my.cnf` et à le modifier à la volée dans le Master. Pour définir le groupe d'hôtes de réplication, nous allons simplement faire une insertion dans la table `mysql_replication_hostgroups`:

```
Admin> INSERT INTO mysql_replication_hostgroups (writer_hostgroup, reader_hostgroup) VALUES(1,2);
Query OK, 1 row affected (0,00 sec)
```

## 7. Ajout des serveurs Mysql à ProxySQL:

Ajoutons le serveur, mais avant, il faut noter: Si nous souhaitons que le maître reçoive non seulement le trafic "en écriture", mais également le trafic en "lecture", il doit appartenir aux deux groupes d'hôtes. Un moyen d'y parvenir consiste à définir `read_only=1` sur le maître avant de l'insérer, puis après l'annulation de l'insertion, la valeur à `readn_only=0`. Ou vous pouvez simplement l'ajouter directement aux deux groupes d'hôtes.

Maintenant, les inserts :

```
Admin> INSERT INTO mysql_servers (hostgroup_id, hostname, port, weight) VALUES  
(1, '192.168.1.16', 3306, 1);  
Query OK, 1 row affected (0,00 sec)  
  
Admin> INSERT INTO mysql_servers (hostgroup_id, hostname, port, weight) VALUES  
(2, '192.168.1.16', 3306, 1);  
Query OK, 1 row affected (0,00 sec)  
  
Admin> INSERT INTO mysql_servers (hostgroup_id, hostname, port, weight) VALUES  
(2, '192.168.1.17', 3306, 1);
```

Bien Sûr, il ne faut pas oublier d'effectuer la sauvegarde pour appliquer les changements:

```
Admin> LOAD MYSQL SERVERS TO RUNTIME;  
Query OK, 0 rows affected (0,08 sec)  
  
Admin> SAVE MYSQL SERVERS TO DISK;  
Query OK, 0 rows affected (0,21 sec)
```

## 8. Définition des règles de requête:

Au moins, dans la configuration la plus basique, nous devons définir 2 règles de requête:

- Tout le SELECT ... FOR UPDATE va au groupe d'hôtes de l'écrivain
- Les SELECT restants vont au groupe d'hôtes du lecteur

Pour définir les règles, nous devons ajouter des lignes à la table `mysql_query_rules`.

```

Admin> INSERT INTO mysql_query_rules (active, match_pattern, destination_hostgroup, cache_ttl) VALUES (1, '^SELECT .* FOR UPDATE', 1, NULL);
Query OK, 1 row affected (0,01 sec)

Admin> INSERT INTO mysql_query_rules (active, match_pattern, destination_hostgroup, cache_ttl) VALUES (1, '^SELECT .*', 2, NULL);
Query OK, 1 row affected (0,00 sec)

Admin> LOAD MYSQL QUERY RULES TO RUNTIME;
Query OK, 0 rows affected (0,02 sec)

Admin> SAVE MYSQL QUERY RULES TO DISK;
Query OK, 0 rows affected (0,06 sec)

```

## 9. Vérifiez l'état de mysql\_server:

Vérifiez que le maître et les esclaves font partie des groupes d'hôtes auxquels ils appartiennent. Pour cette requête, la table runtime\_mysql\_servers :

```

Admin> SELECT hostgroup_id,hostname,port,status FROM runtime_mysql_servers;
+-----+-----+-----+-----+
| hostgroup_id | hostname      | port | status |
+-----+-----+-----+-----+
| 1            | 192.168.1.16 | 3306 | ONLINE |
| 2            | 192.168.1.16 | 3306 | ONLINE |
| 2            | 192.168.1.17 | 3306 | ONLINE |
| 2            | 192.168.1.18 | 3306 | ONLINE |
+-----+-----+-----+-----+
4 rows in set (0,00 sec)

```

Parfait, tout fonctionne à merveille. Nous avons un master et deux slaves qui communiquent parfaitement. Nous avons fait le travail demandé qui était de mettre en place une base de données mysql et de le répartir sur 3 nœuds avec ProxySQL.