

# MySQL : Répliquer en mode actif/actif une base de données MySQL



## Introduction

Pour des raisons de haute disponibilité, il peut être utile de mettre en place une réplication de bases de données.

Cet article montre et décrit comment mettre en place cela.

Ces manipulations ont été réalisées avec **MySQL 5.5.40**

## Les serveurs

Pour la suite de cet article, voici ma configuration :

Cluster1 : **192.168.1.190**

Cluster2 : **192.168.1.10**

Remarque 1 : Le Cluster 2 est une machine qui était configurée sur un autre réseau (1.x.x.x) et a donc fait l'objet d'une modification de son fichier INTERFACES (/etc/network/interfaces) afin qu'elle puisse communiquer avec le Cluster 1. Pour ce faire :

- Se mettre en super utilisateur (sudo -s pour Ubuntu)
- Vérifier qu'une sauvegarde d'interfaces existe (cd /etc/network puis ls /etc/network)
- Si une sauvegarde du fichier interfaces n'existe pas encore → cp interfaces interfaces.old
- Vim /etc/network/interfaces
- Modifier les lignes Address, Netmask, Network, Broadcast, Gateway,

Moteur de base de données : **MySQL 5.5**

## La mise en œuvre

Avant de commencer vérifier si les deux machines peuvent communiquer :

### ***Sur le cluster1 :***

On ping le cluster2

Code BASH :

```
ping 192.168.1.10
```

### ***Sur le cluster2 :***

On ping le cluster1

Code BASH :

```
ping 192.168.1.190
```

### ***Sur le cluster1 :***

On se connecte

Code BASH :

```
mysql -u root -p
```

On crée la base :

Code SQL :

```
CREATE DATABASE base;
```

On vérifie que la base est bien créée :

Code SQL :

```
SHOW DATABASES;
```

On rajoute dans le fichier **/etc/mysql/my.cnf** section **[mysqld]** :

Code BASH :

```
log-bin
server-id=1
binlog-do-db=base
binlog-ignore-db=mysql
```

*#base à répliquer*  
*#base à ne pas répliquer*

S'assurer que mysql n'écoute pas que sur le réseau concerné :

**bind-address=0.0.0.0**

On crée un utilisateur de réplication (IP du cluster2):

Code SQL :

```
GRANT REPLICATION SLAVE ON *.* TO 'replication'@192.168.1.10 IDENTIFIED BY 'slave';
```

On relance mysql

Code BASH :

```
/etc/init.d/mysql restart
```

On se reconnecte et taper les commandes :

Code SQL :

```
FLUSH TABLES WITH READ LOCK;  
SHOW MASTER STATUS;
```

```
+-----+-----+-----+-----+  
| File           | POSITION | Binlog_Do_DB   | Binlog_Ignore_DB |  
+-----+-----+-----+-----+  
| mysqld-bin.000011 | 107    | base           | mysql              |  
+-----+-----+-----+-----+  
1 ROW IN SET (0.00 sec)
```

Les informations importantes sont : **File** et **Position** il faut les noter

Puis libérer les tables par

Code SQL :

```
UNLOCK TABLES;
```

Sauvegarder la base à répliquer :

Code BASH :

```
mysqldump -u root -p --create-options --master-data -B base >  
/root/base.sql
```

Transférer le fichier sur le cluster2.

Code BASH :

```
scp /root/base.sql 192.168.1.10:/root/
```

## ***Sur le cluster2***

Se connecter à MySQL puis

Code SQL :

```
STOP SLAVE;
```

On restaure la base:

Code BASH :

```
mysql -u root -p < /root/base.sql
```

Et on vérifie la présence de la base :

Code SQL :

```
SHOW DATABASES;
```

On édite le fichier `/etc/mysql/my.cnf` et on ajoute dans la rubrique

**[mysqld]** :

Code BASH :

```
log-bin
```

```
server-id=2
```

```
binlog-do-db=base
```

```
# à répliquer
```

```
binlog-ignore-db=mysql
```

```
# à ne pas répliquer
```

S'assurer que mysql n'écoute pas que sur le réseau concerné :

**bind-address=0.0.0.0**

On redémarre mysql

Code BASH :

```
/etc/init.d/mysql restart
```

Se connecter au serveur Mysql puis :

Code SQL :

```
STOP SLAVE;
```

Puis :

Code SQL :

```
CHANGE MASTER TO MASTER_HOST='192.168.1.190', MASTER_USER='replication',  
MASTER_PASSWORD='slave', MASTER_LOG_FILE='mysqld-bin.000011',  
MASTER_LOG_POS=107;
```

Les valeurs **MASTER\_LOG\_FILE** et **MASTER\_LOG\_POS** correspondent aux valeurs relevées tout à l'heure avec la commande **SHOW MASTER STATUS**;

Relancer par :

Code SQL :

```
START SLAVE;
```

On ajoute un utilisateur pour la réplication :

Code SQL :

```
GRANT REPLICATION SLAVE ON *.* TO 'replication'@192.168.1.190 IDENTIFIED BY 'slave';
```

## ***Sur le cluster1***

Les commandes à saisir dans MySQL parlent d'elle même :

Code SQL :

```
STOP SLAVE;  
CHANGE MASTER TO MASTER_HOST='192.168.1.10', MASTER_USER='replication',  
MASTER_PASSWORD='slave';  
START SLAVE;
```

Remarque : Vous pouvez voir les informations du master avec :

## ***On teste :***

### **Sur le cluster1 :**

On crée une table

Code SQL :

```
USE base;  
  
CREATE TABLE matable(  
    texte VARCHAR(256)  
);
```

Vérifier que la table est créée:

Code SQL :

```
SHOW TABLES;
```

Insérer une ligne :

Code SQL :

```
INSERT INTO matable VALUES ('Un texte');
```

On vérifie que le texte est rentré :

Code SQL :

```
SELECT * FROM matable;
```

## On se connecte sur cluster2 :

Code SQL :

```
USE base;
```

On vérifie que la table a bien été répliquée :

Code SQL :

```
SHOW TABLES;
```

Et que l'enregistrement saisi sur cluster1 ait bien été inséré aussi :

Code SQL :

```
SELECT * FROM matable;
```

On insère un deuxième enregistrement :

Code SQL :

```
INSERT INTO matable VALUES ('Un deuxieme texte');
```

On vérifie qu'il a été inséré :

Code SQL :

```
SELECT * FROM matable;
```

## Retour sur cluster1

On vérifie que les 2 enregistrements sont là aussi:

Code SQL :

```
SELECT * FROM matable;
```

Si ça ne marche pas, vérifier le statut via :

Code SQL :

```
SHOW SLAVE STATUS;
```



## ***Exemple : En cas de problème, pour pouvoir remettre à zéro les machines :***

Pour la suite de cet article, voici ma configuration :

Cluster1 : **1.12.100.1**

Cluster2 : **1.20.0.4**

(Adresses à modifier en fonction de celles des serveurs)

En prérequis vous devrez avoir :

- Le logiciel Putty
- Les identifiants des machines, pour pouvoir s'y connecter, exemple ici :

Pour le cluster 1 :

- Nom d'utilisateur = bddelastix
- Mot de passe = Denys2001

Pour le cluster 2 :

- Nom d'utilisateur = root
- Mot de passe = Denys2001

### **Situation :**

Nous avons deux bases de données configurées pour le mirroring mais la SLAVE ne reçoit pas les entrées de la MASTER, il faut donc les réinitialiser.

### **La mise en œuvre (à faire sur les deux machines) :**

On ouvre Putty pour chaque machine, dans le champ « Host name » on écrit l'adresse du serveur.

**Remarque : Appuyer sur la touche « Entrée » pour valider la saisie d'une commande/ligne.**

Cela vous demande le nom d'utilisateur de la machine (login : *nom d'utilisateur*).

Il vous demande ensuite le mot de passe de la machine (password : *mot de passe* (Attention, aucun caractère n'apparaît à l'écran)).

S'il vous demande à nouveau le mot de passe c'est que celui que vous avez entré est mauvais.

Vous voilà connecté à la machine.

On se connecte à mysql :

Code BASH :

```
mysql -u root -p
```

On arrête les slaves :

Code SQL :

```
STOP SLAVE;
```

On supprime les slaves :

Code SQL :

```
RESET SLAVE ALL;
```

Si la dernière commande ne fonctionne pas :

Code SQL :

```
RESET SLAVE;
```

On supprime les master :

Code SQL :

```
RESET MASTER;
```

On recherche les utilisateurs qui s'appellent « replication » :

Code SQL :

```
SELECT user,host FROM mysql.user;
```

```
mysql> select user,host from mysql.user;
```

user	host
root	%
replication	1.20.0.4
root	127.0.0.1
root	VME1
root	VME1
	localhost
a2billinguser	localhost
asterisk	localhost
asteriskuser	localhost
root	localhost
roundcube	localhost

```
11 rows in set (0.01 sec)
```

On supprime ces utilisateurs un par un, exemple :

Code SQL :

```
DROP USER 'replication'@1.12.100.1;
```

On supprime le cache :

Code SQL :

```
RESET QUERY CACHE;
```

```
FLUSH QUERY CACHE;
```

On quitte mysql :

Code SQL :

```
quit
```

On redémarre mysql :

Code BASH :

```
/etc/init.d/mysql restart
```

Ou si ça ne fonctionne pas :

Code BASH :

```
/etc/init.d/mysqld restart
```

Revenir au début du document et refaite la mise en œuvre du tutoriel.



## **La mise en œuvre (à faire sur les deux machines) :**

Si cela ne fonctionne toujours pas il faut purger les binary logs, ils se trouvent dans le dossier /var/lib/mysql/, vous pouvez les voir avec la commande (connecté dans MySQL) :

**SHOW BINARY LOGS ;**

Récupérer le nom du fichier avec le plus haut nombre et effectuer cette commande pour supprimer tous les fichiers plus anciens (les X représentent le nombre du fichier) :

**PURGE BINARY LOGS TO 'mysqld-bin.XXXXXX';**

Pour purger les binaires avant la date de création actuelle, :

**PURGE BINARY LOGS before now() ;**

## **Problèmes :**