

Sommaire :

Sommaire :	1
Introduction.....	1
Phase 1 – Installation et configuration du serveur maître :.....	2
Phase 2 – Restauration des données sur le serveur maître :.....	7
Phase 3 – Installation et configuration du serveur esclave :.....	8
Phase 4 – Test de la réplication :.....	11
Conclusion :	11

Introduction

Objectif : Cette procédure va nous permettre de mettre en place de la réplication de base de données. Un serveur de base de données maître va être utilisé pour accueillir la création, la modification et la suppression d'informations dans une base. Un serveur esclave sera automatiquement mis à jour grâce à la configuration appliquée avec cette procédure.

Rappels : Il ne faut pas confondre la réplication et répartition ! En effet, la répartition permet l'utilisation simultanée de plusieurs serveurs maîtres, tous enregistrant des informations différentes. La réplication n'autorise qu'un serveur maître et plusieurs serveurs esclaves qui partagent les mêmes informations.

Prérequis : Nous devons avoir une Debian 8.5 ainsi que des documents sur l'installation, l'utilisation et la configuration de MySQL.

Norme adoptée : Toutes les commandes et les noms utilisés sur notre machine Debian seront écrits en ***gras et en italique***.

E6 :

Elaboration de documents relatifs à la production et à la fourniture de services

A1.1.1 , Analyse du cahier des charges d'un service à produire

A1.2.4 , Détermination des tests nécessaires à la validation d'un service

A1.3.4 , Déploiement d'un service

A4.1.9 , Rédaction d'une documentation technique

Phase 1 – Installation et configuration du serveur maître :

Nous commençons par installer un serveur MySQL en version 5.6 sur notre machine Debian. Pour cela, il faut préalablement télécharger le paquet : **wget http://dev.mysql.com/get/mysql-apt-config_0.8.0-1_all.deb** :

```
root@stmysqlmaster:~# wget http://dev.mysql.com/get/mysql-apt-config_0.8.0-1_all
.deb
--2016-10-04 09:51:17-- http://dev.mysql.com/get/mysql-apt-config_0.8.0-1_all.d
eb
Résolution de dev.mysql.com (dev.mysql.com)... 137.254.60.11
Connexion à dev.mysql.com (dev.mysql.com)|137.254.60.11|:80... connecté.
requête HTTP transmise, en attente de la réponse... 302 Found
Emplacement : http://repo.mysql.com//mysql-apt-config_0.8.0-1_all.deb [suivant]
--2016-10-04 09:51:17-- http://repo.mysql.com//mysql-apt-config_0.8.0-1_all.deb
Résolution de repo.mysql.com (repo.mysql.com)... 2.20.134.142
Connexion à repo.mysql.com (repo.mysql.com)|2.20.134.142|:80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 19202 (19K) [application/x-debian-package]
Sauvegarde en : « mysql-apt-config_0.8.0-1_all.deb »

mysql-apt-config_0. 100%[=====] 18,75K --.-KB/s ds 0,03s
2016-10-04 09:51:17 (610 KB/s) - « mysql-apt-config_0.8.0-1_all.deb » sauvegardé
[19202/19202]
```

Avec l'aide de **dpkg -i mysql-apt-config_0.8.0-1_all.deb**, nous pouvons préparer le fichier ainsi téléchargé pour l'installation :

```
Outil de configuration des paquets

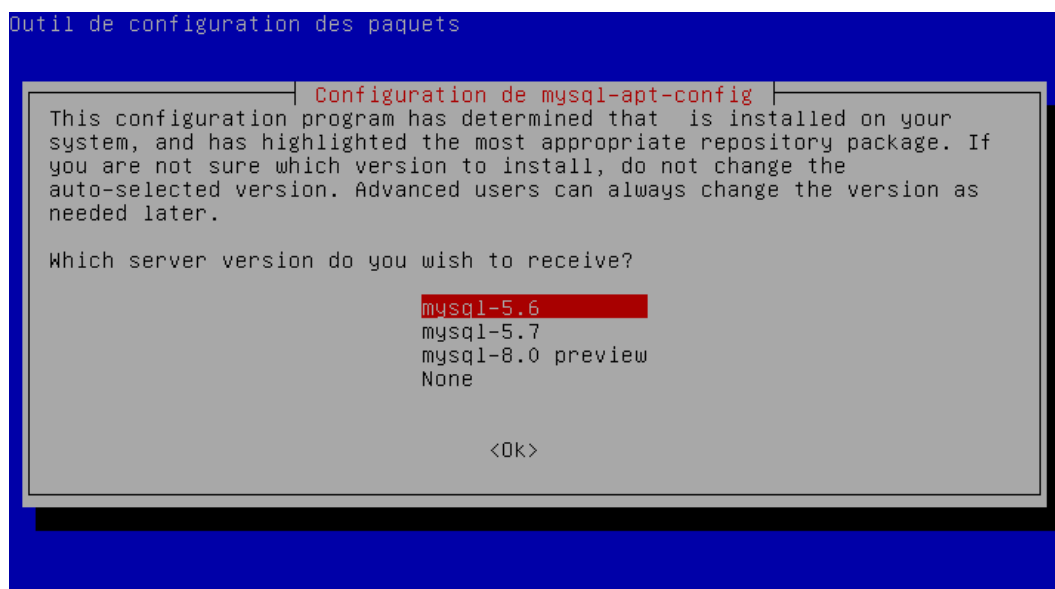
Configuration de mysql-apt-config
MySQL APT Repo features MySQL Server along with a variety of MySQL
components. You may select the appropriate product to choose the version
that you wish to receive.

Once you are satisfied with the configuration then select last option
'Apply' to save the configuration. Advanced users can always change the
configurations later, depending on their own needs.

Which MySQL product do you wish to configure?

MySQL Server (Currently selected: mysql-5.6)
MySQL Tools & Connectors (Currently selected: Enabled)
MySQL Preview Packages (Currently selected: Disabled)
OK

<Ok>
```



La commande ***apt-cache policy mysql-server*** nous permet de vérifier si la bonne version de MySQL est effectivement prête à être installée :

```
root@stmysqlmaster:~# apt-cache policy mysql-server
mysql-server:
  Installé : 5.6.33-1debian8
  Candidat : 5.6.33-1debian8
  Table de version :
  *** 5.6.33-1debian8 0
      100 /var/lib/dpkg/status
  5.5.52-0+deb8u1 0
      500 http://security.debian.org/ jessie/updates/main amd64 Packages
  5.5.50-0+deb8u1 0
      500 http://ftp.fr.debian.org/debian/ jessie/main amd64 Packages
```

Il ne nous reste plus qu'à faire un ***apt update***, suivi d'un ***apt install mysql-server*** pour lancer l'installation (et optionnellement un ***apt -f install*** si certaines dépendances n'ont pas été installés).

Répéter la commande ***apt install mysql-server*** permet de vérifier si l'installation s'est bien déroulée. Un ***mysql --version*** nous donne le numéro de version de notre serveur :

```
root@stmysqlmaster:~# mysql --version
mysql Ver 14.14 Distrib 5.6.33, for Linux (x86_64) using EditLine wrapper
```

Nous devons maintenant configurer le serveur MySQL pour autoriser la connexion à distance. Nous commençons par éditer le fichier `/etc/mysql/my.cnf`, en appliquant l'adresse IP de notre serveur sur la ligne `bind-address = 192.168.1.116`, afin que les serveurs esclaves puissent « écouter » cette adresse :

```
GNU nano 2.2.6      Fichier : /etc/mysql/my.cnf      Modifié
socket             = /var/run/mysqld/mysqld.sock
port               = 3306
basedir            = /usr
datadir            = /var/lib/mysql
tmpdir             = /tmp
lc-messages-dir   = /usr/share/mysql
explicit_defaults_for_timestamp

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address       = 192.168.1.116_

log-error          = /var/log/mysql/error.log

# Recommended in standard MySQL setup
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES

# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
```

Nous redémarrons ensuite le service : `systemctl restart mysql`. Puis on s'y connecte : `mysql -u root -p`. L'option `-u` permet de demander un utilisateur, tandis que l'option `-p` demande le mot de passe.

Nous devons ensuite créer un utilisateur qui sera autorisé à se connecter à distance à la base de données :

```
grant all privileges on *.* to 'distance'@'%' identified by 'distance' ;  
flush privileges ;
```

Avec cette commande, nous donnons tous les privilèges de contrôle en MySQL (select, insert, etc.) à l'utilisateur qui sera nommé « **distance** », et l'argument « % » lui permettra d'accéder à distance à la base, depuis n'importe quel IP du réseau. Le dernier « **distance** » de la commande représente le mot de passe.

Nous créons ensuite la base **vehicules** :

```
mysql -u root -p // On se connecte au serveur MySQL.  
CREATE DATABASE vehicules ; // On créer la base vehicules.  
SHOW DATABASES ; // On vérifie la création de la base.
```

```
mysql> create database vehicules;
Query OK, 1 row affected (0,00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| vehicules |
+-----+
4 rows in set (0,00 sec)
```

Enfin, il faut procéder à l'installation d'une machine cliente pour vérifier la connectivité à distance. Pour cela, il faut installer le paquet **mysql-client** en version 5.6, en suivant les commandes suivantes :

```
wget http://dev.mysql.com/get/mysql-apt-config_0.8.0-1_all.deb // On télécharge le paquet.
dpkg -i mysql-apt-config_0.8.0-1_all.deb // On prépare l'installation de
la bonne version du paquet.
apt update // On met à jour la machine.
apt-cache policy mysql-client // On vérifie la version qui s'installera.
apt install mysql-client // On installe le client.
```

Un nouveau **apt-cache policy mysql-client** permet de vérifier la version du client installée :

```
root@stmysqlslave1:/home/anthony# apt-cache policy mysql-client
mysql-client:
  Installé : 5.6.33-1debian8
  Candidat : 5.6.33-1debian8
Table de version :
*** 5.6.33-1debian8 0
   500 http://repo.mysql.com/apt/debian/ jessie/mysql-5.6 amd64 Packages
   100 /var/lib/dpkg/status
 5.5.52-0+deb8u1 0
   500 http://security.debian.org/ jessie/updates/main amd64 Packages
 5.5.50-0+deb8u1 0
   500 http://ftp.fr.debian.org/debian/ jessie/main amd64 Packages
```

Nous nous connectons ensuite sur le serveur depuis notre machine cliente, avec l'utilisateur « **distance** » :

```
mysql -u distance -p -h 192.168.1.116
SHOW DATABASES ;
```

```
root@stmysqlslavel:/home/anthony# mysql -u distance -p -h 192.168.1.116
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.6.33 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| vehicules |
+-----+
4 rows in set (0,00 sec)
```

La connectivité à distance fonctionne donc parfaitement. Nous devons ensuite faire quelques modifications pour que notre serveur maître soit considéré comme tel. Pour cela, nous accédons au fichier `/etc/mysql/my.cnf`, pour rajouter les deux dernières lignes **`log-bin=/var/log/mysql/mysql-bin.log`** et **`server-id=1`** :

```
server-id      = 1
log-bin        = /var/log/mysql/mysql-bin.log
binlog-do-db   = vehicules

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.

bind-address   = 192.168.1.116
log-error      = /var/log/mysql/error.log
```

Le **server-id** doit être unique ! Puis on redémarre le service. Pour préparer la suite, il faut autoriser le futur serveur esclave distant à répliquer la base de données, avec les commandes SQL suivantes :

```
grant replication slave on *.* to 'replication'@'192.168.1.128' identified by 'replication';
flush privileges ;
flush tables with read lock ;
```

Nous pouvons vérifier la création de cet utilisateur dans la table « *user* » de la base « *mysql* » :

```
mysql> select user,host from user;
+-----+-----+
| user      | host      |
+-----+-----+
| distance  | %         |
| root      | 127.0.0.1 |
| replication | 192.168.1.128 |
| root      | ::1       |
| root      | localhost |
| root      | stmysqlmaster |
+-----+-----+
6 rows in set (0,00 sec)
```

Nous pouvons maintenant procéder à l'étape suivante.

Phase 2 – Restauration des données sur le serveur maître :

Afin de préparer la réplication, nous devons importer la base de données **BDD_vehicules** de notre PC jusqu'au serveur de notre **Debian** maître. Pour cela, on se procure le script via **FTP**, puis on l'importe sur le serveur **MySQL** avec la commande **mysql -u root -p vehicules < /root/bdd_vehicules.sql** :

```
root@stmysqlmaster:~# mysql -u root -p vehicules < /root/bdd_vehicules.sql
Enter password:
```

Nous pouvons vérifier que l'importation a bien été effectuée :

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| vehicules  |
+-----+
4 rows in set (0,00 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_vehicules |
+-----+
| t_agents             |
| t_attribuer          |
| t_reparations        |
| t_vehicules          |
+-----+
4 rows in set (0,00 sec)
```

Pour que seule la base **vehicules** soit répliquée, on ajoute la ligne **binlog-do-db = vehicules** dans le fichier **/etc/mysql/my.cnf** :

```
server-id      = 1
log-bin        = /var/log/mysql/mysql-bin.log
binlog-do-db   = vehicules
```

On peut vérifier le fonctionnement de cette configuration avec la commande **show master status** :

```
mysql> show master status;
+-----+-----+-----+
----+
| File           | Position | Binlog_Do_DB |
Set |
+-----+-----+-----+
----+
| mysql-bin.000003 |      120 | vehicules    |
|                |         |              |
+-----+-----+-----+
----+
1 row in set (0,00 sec)
```

Phase 3 – Installation et configuration du serveur esclave :

Nous allons maintenant pouvoir utiliser notre machine cliente précédemment installée. Nous pouvons vérifier que le serveur maître et la machine cliente puissent communiquer à l'aide d'un **ping** par exemple :

```
root@stmysqlslave1:/home/anthony# ping 192.168.1.116
PING 192.168.1.116 (192.168.1.116) 56(84) bytes of data.
64 bytes from 192.168.1.116: icmp_seq=1 ttl=64 time=0.652 ms
64 bytes from 192.168.1.116: icmp_seq=2 ttl=64 time=0.696 ms
64 bytes from 192.168.1.116: icmp_seq=3 ttl=64 time=0.963 ms
64 bytes from 192.168.1.116: icmp_seq=4 ttl=64 time=0.734 ms
64 bytes from 192.168.1.116: icmp_seq=5 ttl=64 time=0.919 ms
64 bytes from 192.168.1.116: icmp_seq=6 ttl=64 time=0.511 ms
64 bytes from 192.168.1.116: icmp_seq=7 ttl=64 time=0.640 ms
64 bytes from 192.168.1.116: icmp_seq=8 ttl=64 time=1.14 ms
64 bytes from 192.168.1.116: icmp_seq=9 ttl=64 time=0.688 ms
```

Les deux machines peuvent donc communiquer. Nous allons maintenant réutiliser la procédure précédente afin d'installer un serveur MySQL sur notre machine esclave, soit les commandes :

```
wget http://dev.mysql.com/get/mysql-apt-config_0.8.0-1_all.deb // On télécharge le paquet.
dpkg -i mysql-apt-config_0.8.0-1_all.deb // On prépare l'installation de
la bonne version du paquet.
apt update // On met à jour la machine.
apt-cache policy mysql-client // On vérifie la version qui s'installera.
apt install mysql-client // On installe le client.
```

Un **systemctl status mysql** permet de valider l'installation du service :


```
root@stmysqlslave1:/home/anthony# systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled)
   Active: active (running) since ven. 2016-10-07 14:46:40 CEST; 31s ago
 Main PID: 2049 (mysqld_safe)
    CGroup: /system.slice/mysql.service
            └─2049 /bin/sh /usr/bin/mysqld_safe
               └─2324 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql
```

Nous devons maintenant à nouveau importer la base de données **BDD_vehicules** de notre PC jusqu'au serveur de notre **Debian** esclave. Pour cela, on se procure le script via **FTP**, puis on l'importe sur le serveur **MySQL** avec la commande **mysql -u root -p vehicules < /home/anthony/bdd_vehicules.sql** :

```
mysql> show databases
-> ;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| vehicules |
+-----+
4 rows in set (0,00 sec)

mysql> use vehicules
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_vehicules |
+-----+
| t_agents |
| t_attribuer |
| t_reparations |
| t_vehicules |
+-----+
4 rows in set (0,00 sec)
```

Nous devons maintenant paramétrer le serveur esclave afin qu'il soit considéré comme tel. Pour cela, il faut, dans le fichier **/etc/mysql/my.cnf** ajouter un **server-id** unique :

```
[mysqld]
user      = mysql
pid-file  = /var/run/mysqld/mysqld.pid
socket    = /var/run/mysqld/mysqld.sock
port      = 3306
basedir   = /usr
datadir   = /var/lib/mysql
tmpdir    = /tmp
lc-messages-dir = /usr/share/mysql
explicit_defaults_for_timestamp
server-id = 24
```

Ensuite, il faut entrer les informations du serveur maître sur notre serveur esclave afin que la réplication puisse s'effectuer. La plupart de ces informations se retrouvent avec un **show master status** sur le serveur maître.

```
stop slave ;           // On arrête le serveur esclave.
show slave status ;   // On vérifie qu'il est bien arrêté.

                        // On entre les informations adéquates :
change master to master_host='192.168.1.116', master_user='replication',
master_password='replication', master_log_file='mysql-bin.000003', master_log_pos=120 ;

                        // On démarre le serveur esclave :
start slave ;
```

Enfin, nous redémarrons le serveur esclave avec un **systemctl restart mysql**, nous débloquons les tables précédemment bloquées sur le serveur maître grâce à **unlock tables**, puis nous vérifions le fonctionnement du serveur esclave, avec un **show slave status \G**, afin de constater que toutes les informations entrées précédemment sont correctes :

```
mysql> show slave status \G;
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 192.168.1.116
      Master_User: replication
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000004
      Read_Master_Log_Pos: 365
      Relay_Log_File: mysqld-relay-bin.000006
      Relay_Log_Pos: 528
      Relay_Master_Log_File: mysql-bin.000004
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
```

Toutes les informations ont donc bien été prises en compte.

Phase 4 – Test de la réplication :

Pour tester la réplication, nous ajoutons un enregistrement, sur le serveur maître dans la table « **t_agents** » de la base « **vehicules** » :

```
use vehicules ;  
insert into t_agents values ('ABCD', 'TERRIEUR', 'Alain', 1, 2, '');
```

```
mysql> insert into t_agents values('ABCD', 'TERRIEUR', 'Alain', 1, 2, '');  
Query OK, 1 row affected (0,03 sec)
```

Finalement, sur le serveur esclave, nous pouvons vérifier le contenu de la table « **t_agents** » de la base « **vehicules** » :

```
use vehicules ;  
select * from t_agents ;
```

```
mysql> select * from t_agents  
-> ;  
+-----+-----+-----+-----+-----+  
| MatA | NomA      | PrenomA | PB  | PC  | PhotoA |  
+-----+-----+-----+-----+-----+  
| ABCD | TERRIEUR  | Alain   | 1   | 2   |         |  
| B125 | CHECHUR   | Macha   | 1   | 1   | objet OLE non gr |  
| E540 | BIENUNEPOZ | Humphrey | 1   | 0   | objet OLE non gr |  
| E578 | ZEBLUES   | Agathe  | 1   | 1   | objet OLE non gr |  
| e712 | fotohoukoi | Thomas  | 1   | 1   | objet OLE non gr |  
| G124 | MORFLAIS  | Ella    | 1   | 0   | objet OLE non gr |  
| H540 | MANSOIF   | Grard   | 0   | 0   | objet OLE non gr |  
+-----+-----+-----+-----+-----+  
7 rows in set (0,00 sec)
```

Nous constatons que l'enregistrement inséré dans le serveur maître a bien été instantanément répliqué sur le serveur esclave. La réplication fonctionne donc parfaitement.

Conclusion :

La réplication peut être difficile à mettre en place car il existe plusieurs façons de procéder. Par exemple, ici, nous avons entré les informations du serveur maître grâce à des fonctions SQL dans le serveur esclave. Mais il est également possible d'ajouter ces informations directement dans le fichier de configuration du serveur esclave. Une bonne maîtrise de MySQL est également requise.