



Réplication de Base de données

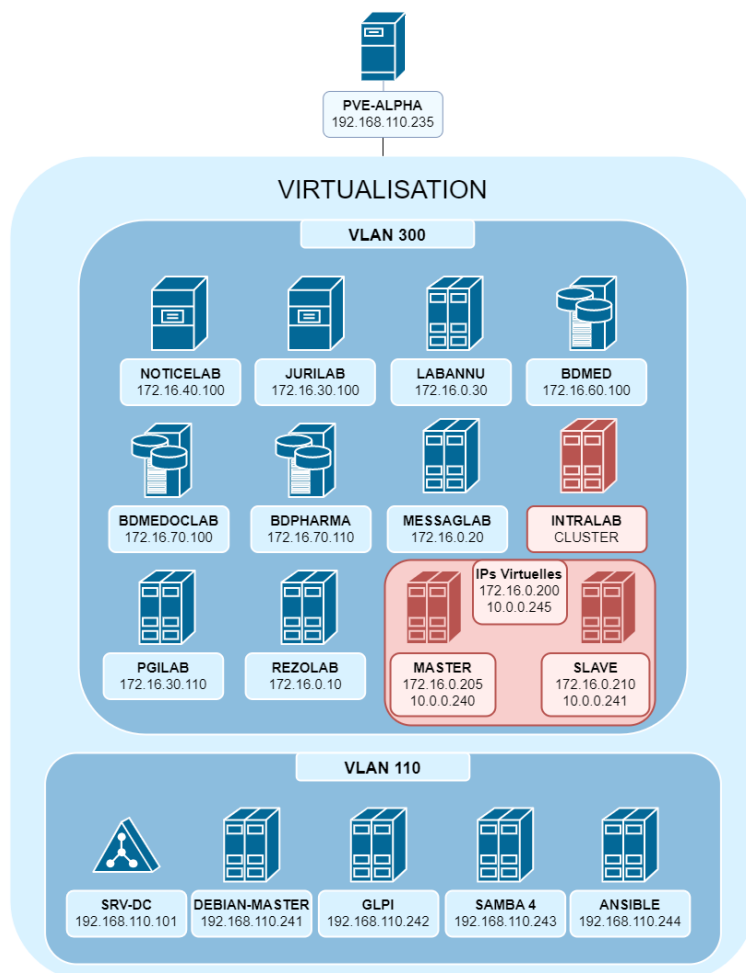
MELNOTTE Hugo
BTS SIO

Introduction	1
Configuration du Serveur maître	2
Configuration du Serveur Esclave :	4
Teste de fonctionnement	5

Introduction

Afin d'assurer la haute disponibilité du serveur INTRALAB, le service technique de l'entreprise GSB est chargé de créer un cluster composé de deux serveurs, INTRALAB qui prendra le rôle *master* et un clone qui prendra le rôle *slave*. Les deux serveurs se nomment respectivement INTRALAB-MASTER et INTRALAB-SLAVE.

Afin d'effectuer les tests de fonctionnement, j'ai à ma disposition une machine de test sous Windows 10 et connecté au réseau GSB depuis le VLAN 30 (RH, comptabilité)





Réplication de Base de données

MELNOTTE Hugo
BTS SIO

Nous allons voir comment mettre en place un service MySQL de réplication de base de données de type maître-Esclave. Le but est de pouvoir sauvegarder, sur un second serveur, toutes les données que nous ajoutons sur un serveur MySQL afin de lutter contre de potentielles pertes.

Nous utiliserons ici deux serveurs distincts, l'un maître, l'autre esclave :

- Le serveur INTRALAB-MASTER (maître) avec pour @IP : 172.16.0.205
- Le serveur INTRALAB-SLAVE (esclave) avec pour @IP : 172.16.0.210

Configuration du Serveur maître

Tout d'abord il faut éditer le fichier `/etc/mysql/my.cnf` pour permettre au service d'accepter les connexions entrantes depuis l'extérieur de la machine :

```
nano /etc/mysql/my.cnf
```

Et configurer les lignes suivantes ainsi :

```
#bind-address = 127.0.0.1
server-id=1
log-bin=/var/log/mysql/mysql-bin.log
log-error=/var/log/mysql/mysql-error.log
expire_logs_days = 10
max_binlog_size = 100M
binlog-do-db = db_glpi
bind-address = 172.16.0.205
```

Nous redémarrons MySQL afin de prendre en compte nos modifications.

```
service mysql restart
```

Nous allons maintenant créer l'utilisateur « replication_user » qui sera dédié au service de réplication des données :

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

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'replication_user'@'172.16.0.210'
IDENTIFIED BY 'motdepasse';
mysql> FLUSH PRIVILEGES;
mysql> USE db_glpi;
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SHOW MASTER STATUS;
mysql> exit
```



Réplication de Base de données

MELNOTTE Hugo
BTS SIO

```
MariaDB [db_glpi]> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000010 |      342 | db_glpi      |                   |
+-----+-----+-----+-----+
1 row in set (0,000 sec)

MariaDB [db_glpi]> █
```

Les valeurs des colonnes 'File' et 'Position' sont à garder puisqu'elles nous seront nécessaires plus tard.

Nous pouvons désormais faire un dump de la base données 'db_glpi' :

```
mysqldump -u root -p db_glpi > db_glpi.sql
```

On peut alors désactiver le mode de lecture seule :

```
mysql -u root -p
mysql> USE db_glpi;
mysql> UNLOCK TABLES;
mysql> exit
```

Puis on envoie notre fichier contenant les données de notre base vers notre serveur INTRALAB-SLAVE qui est notre serveur esclave :

```
$ scp db_glpi.sql root@172.16.0.210:/root/
```

```
root@INTRALAB-MASTER:~# scp db_glpi.sql root@172.16.0.210:/root/
root@172.16.0.210's password:
db_glpi.sql                                100% 715KB 60.7MB/s 00:00
root@INTRALAB-MASTER:~# █
```



Configuration du Serveur Esclave :

On se connecte maintenant sur notre serveur esclave afin d'y importer le dump de la base de données 'db_glpi'.

Et on y importe les données du maître grâce au fichier dump :

```
mysql -u root -p db_glpi < db_glpi.sql
```

On configure ensuite comme pour notre serveur maître le fichier /etc/mysql/my.cnf de la manière suivante :

```
nano /etc/mysql/my.cnf
```

```
#bind-address = 127.0.0.1  
server-id = 2  
log_bin = /var/log/mysql/mysql-bin.log  
expire_logs_days = 10  
max_binlog_size = 100M  
binlog_do_db = db_glpi
```

On redémarre le service MySQL pour prendre en compte les modifications :

```
$ service mysql restart
```

On se connecte au service MySQL afin de configurer la réplication :

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

```
mysql> CHANGE MASTER TO MASTER_HOST='172.16.0.205',  
MASTER_USER='replication_user', MASTER_PASSWORD='motdepasse',  
MASTER_LOG_FILE='mysql-bin.000011', MASTER_LOG_POS= 342;
```

```
mysql> START SLAVE;
```

N'oubliez pas de modifier les valeurs des colonnes 'File' et 'Position' que nous avons notées précédemment puisqu'elles ne sont pas les mêmes en fonction des installations.

Nous pouvons ensuite vérifier que notre service de réplication est bien démarré à l'aide la commande suivante :



```
mysql> SHOW SLAVE STATUS\G
```

```
MariaDB [(none)]> SHOW SLAVE STATUS\G;
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 172.16.0.205
      Master_User: replication_user
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000011
      Read_Master_Log_Pos: 342
      Relay_Log_File: mysqld-relay-bin.000002
      Relay_Log_Pos: 555
      Relay_Master_Log_File: mysql-bin.000011
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
```

La ligne 'Slave_IO_State: Waiting for master to send event' nous indique que notre serveur est bien configuré comme esclave et qu'il attend maintenant des changements du côté maître.

Teste de fonctionnement

Nous allons maintenant tester notre service de réplication afin d'être sûr qu'il soit fonctionnel.

Sur notre serveur maître, nous créons une nouvelle table 'exempletable' dans notre base de données 'db_glpi' qui est normalement répliquée.

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

```
mysql> USE db_glpi;
mysql> CREATE TABLE exempletable (
  -> ID INT NOT NULL AUTO_INCREMENT,
  -> TEST VARCHAR(100) NOT NULL,
  -> PRIMARY KEY (ID));
```

```
MariaDB [db_glpi]> CREATE TABLE exempletable (
  -> ID INT NOT NULL AUTO_INCREMENT,
  -> TEST VARCHAR(100) NOT NULL,
  -> PRIMARY KEY (ID));
Query OK, 0 rows affected (0,067 sec)

MariaDB [db_glpi]> █
```



Réplication de Base de données

MELNOTTE Hugo
BTS SIO

Puis on se rend sur notre serveur esclave afin de vérifier que la table ait bien été ajoutée dans la base de données :

```
$ mysql -u root -p  
mysql> USE db_glpi;  
mysql> SHOW TABLES;
```

```
Database changed  
MariaDB [db_glpi]> SHOW TABLES;  
+-----+  
| Tables_in_db_glpi |  
+-----+  
| exempletable      |
```

La table 'exempletable' que nous avons créée sur notre serveur maître a été ajoutée automatiquement dans la base de données 'db_glpi' de notre serveur esclave. Notre service est donc fonctionnel.