

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

PROCEDURE E4 : SERVEUR DE REPLICATION MARIADB DEBIAN

SOMMAIRE :

I)	Objectif.....	2
II)	Prérequis.....	2
III)	Définitions.....	2
IV)	Création du serveur de réplication MariaDB sous PROXMOX.....	2-10
V)	Configurations nécessaires pour le service MariaDB.....	10
VI)	Installation du service « mariadb-server ».....	11-12
VII)	Configuration du service MariaDB pour la réplication.....	12-20
	a) Visualisation des données du serveur MariaDB Maître.....	12-15
	b) Configuration du serveur MariaDB Maître.....	15-16
	c) Configuration du serveur de réplication MariaDB (Esclave).....	16-20
VIII)	Conclusion.....	20

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

I) Objectif

Dans cette procédure, nous allons montrer comment installer et configurer un serveur de réplication **MariaDB (Esclave)** pour garder une copie des données du serveur **MariaDB Maître** sous Debian.

II) Prérequis

Pour réaliser cette procédure, nous avons besoin des éléments suivants :

OS	Distribution	Version	C/S	Outil de virtualisation
Debian	Linux	8.5	S	PROXMOX

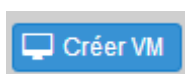
Nom du serveur MariaDB Maître	Nom du serveur MariaDB Esclave	Adresse IP du serveur MariaDB Maître	Adresse IP du serveur MariaDB Esclave
BDMEDLAB	BDMEDLABREP	192.168.1.140/24	192.168.1.133 /24

III) Définitions

- **MariaDB** est un système de gestion de bases de données édité sous licence **GPL (General Public License)**. Il s'agit d'un fork communautaire de **MySQL** : la gouvernance du projet est assuré par la fondation **MariaDB** et sa maintenance par la société Monty Program AB, créateur du projet. Cette gouvernance confère au logiciel l'assurance de rester libre.
- Un serveur de réplication **MariaDB** permet de garder une copie des bases de données du serveur **MariaDB** principal (maître) et de garantir la disponibilité des données.

IV) Création du serveur de réplication MariaDB sous PROXMOX

- Tout d'abord, nous cliquons sur « **Créer VM** » :



ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous saisissons un « VM ID » et un « Nom » et « Suivant » :

Créer: Machine Virtuelle ⊗

Général OS CD/DVD Disque Dur CPU Mémoire Réseau Confirmation

Nœud: Pool de ressource:

VM ID:

Nom:

? Help Retour Suivant

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous choisissons « **Linux 4.X/3.X/2.6 Kernel** » et « **Suivant** » :

Créer: Machine Virtuelle ⊗

Général **OS** CD/DVD Disque Dur CPU Mémoire Réseau Confirmation

Microsoft Windows

- Microsoft Windows 10/2016
- Microsoft Windows 8.x/2012/2012r2
- Microsoft Windows 7/2008r2
- Microsoft Windows Vista/2008
- Microsoft Windows XP/2003
- Microsoft Windows 2000

Linux/Autre types d'OS

- Linux 4.X/3.X/2.6 Kernel
- Linux 2.4 Kernel
- Solaris Kernel
- Autre types d'OS

Help Retour Suivant

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous choisissons l'image ISO de Debian version 8.5 et « **Suivant** » :

Créer: Machine Virtuelle ⊗

Général OS **CD/DVD** Disque Dur CPU Mémoire Réseau Confirmation

Utiliser une image de disque (ISO)

Stockage: ISO ▾

Image ISO: **debian-8.5.0-amd64-CD-1** ▾

Utiliser le lecteur CD/DVD de l'hôte

Aucun media

[Retour](#) [Suivant](#)

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous choisissons « **VirtIO** », « **10** » Go pour le disque et « **Suivant** » :

Créer: Machine Virtuelle ✕

Général OS CD/DVD **Disque Dur** CPU Mémoire Réseau Confirmation

Bus/Device: **VirtIO** 0 Cache: Défait (Désactivé)

Stockage: ISO Aucune sauvegarde:

Taille du disque (GB): **10** Discard:

Format: Image au format QEMU (Image IO thread:

Help Retour **Suivant**

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous cliquons directement sur « **Suivant** » :

Créer: Machine Virtuelle ⊗

Général OS CD/DVD Disque Dur **CPU** Mémoire Réseau Confirmation

Sockets: ⬆ ⬇ Type: ⬇

Cœurs: ⬆ ⬇ Total cœurs: 1

Enable NUMA:

🔍 Help Retour Suivant

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous mettons « **1024** » pour 1 Go en mémoire et « **Suivant** » :

Créer: Machine Virtuelle ✕

Général OS CD/DVD Disque Dur CPU **Mémoire** Réseau Confirmation

Utiliser une taille de mémoire fixe

Mémoire (MB):

Ballooning:

Allouer automatiquement la mémoire dans cette page

Mémoire maximum (MB):

Mémoire minimum (MB):

Partages:

Help Retour Suivant

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous cliquons directement sur « **Suivant** » :

Créer: Machine Virtuelle ✕

Général OS CD/DVD Disque Dur CPU Mémoire **Réseau** Confirmation

Accès par pont
 Modèle: VirtIO (paravirtualisé) ▾

Tag VLAN: no VLAN ▾
 Adresse MAC: auto

Pont: vubr0 ▾
 Limite de débit (MB/s): unlimited ▾

Firewall:
Multiqueues: ▾

NAT
 Déconnecter:

Aucun périphérique réseau

Help
Retour **Suivant**

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Après la confirmation des données de la VM, nous cliquons sur « **Terminé** » :

Créer: Machine Virtuelle ✕

Général OS CD/DVD Disque Dur CPU Mémoire Réseau **Confirmation**

Paramètres

Key ↑	Value
cores	1
ide2	ISO:iso/debian-8.5.0-amd64-CD-1.iso,media=cdrom
memory	1024
name	BDMEDLABREP
net0	virtio,bridge=vibr0
nodename	Prox3
numa	0
ostype	l26
sockets	1
virtio0	ISO:10,format=qcow2

Retour **Terminé**

V) Configurations nécessaires pour le service MariaDB

- Une fois la machine créée et installée, nous donnons son nom et son adresse IP (Voir prérequis).
- Nous éditons les fichiers « **/etc/hosts** » sur les 2 serveurs **MariaDB** en ajoutant leurs adresses IP et leurs noms correspondants :

⇒ Contenu du fichier « **/etc/hosts** » du serveur **MariaDB Maître** :

```
GNU nano 2.2.6 Fichier : /etc/hosts
127.0.0.1 localhost
127.0.1.1 BDMEDLAB
192.168.1.140 BDMEDLAB
192.168.1.133 BDMEDLABREP
```

⇒ Contenu du fichier « **/etc/hosts** » du serveur de réplication **MariaDB (Esclave)** :

```
GNU nano 2.2.6 Fichier : /etc/hosts
127.0.0.1 localhost
127.0.1.1 BDMEDLABREP
192.168.1.133 BDMEDLABREP
192.168.1.140 BDMEDLAB
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

VI) Installation du service « mariadb-server »

- Tout d'abord, nous mettons à jour les paquets :

```
root@BDMEDLABREP:~# apt-get update
```

- Ensuite, nous installons le service « **mariadb-server** » :

```
root@BDMEDLABREP:~# apt-get install mariadb-server
```

- Nous saisissons un mot de passe pour **MariaDB** :

```

Configuration de mariadb-server-10.0
Il est très fortement recommandé d'établir un mot de passe pour le
compte d'administration de MariaDB (« root »).

Si ce champ est laissé vide, le mot de passe ne sera pas changé.

Nouveau mot de passe du superutilisateur de MariaDB :
****_-----
<Ok>

```

- Nous le confirmons :

```

Configuration de mariadb-server-10.0

Confirmation du mot de passe du superutilisateur de MariaDB :
****_-----
<Ok>

```

- Nous téléchargeons l'archive contenant le fichier de **MariaDB** :

```
root@BDMEDLABREP:~# wget https://www.adminer.org/static/download/4.2.5/adminer-4.2.5-mysql-en.php
```

- Nous allons sécuriser le serveur **MariaDB** :

```
root@BDMEDLABREP:~# mysql_secure_installation
```

- Nous tapons le mot de passe du super utilisateur « **root** » (non visible) :

```
Enter current password for root (enter for none):
```

- Nous ne changeons pas le mot de passe « **root** » :

```
Change the root password? [Y/n] n
```

- Nous enlevons les utilisateurs anonymes :

```
Remove anonymous users? [Y/n] Y
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous n'interdisons pas la connexion à distance « root » :

```
Disallow root login remotely? [Y/n] n
```

- Nous supprimons la base de données de test et l'accès :

```
Remove test database and access to it? [Y/n] Y
```

- Nous rechargeons les tables de privilèges :

```
Reload privilege tables now? [Y/n] Y
```

VII) Configuration du service MariaDB pour la réplication

a) Visualisation des données du serveur MariaDB Maître

- Tout d'abord, nous nous connectons sur le serveur **MariaDB Maître** et visualisons les BDD créées et à répliquer :

```
root@BDMEDLAB:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.
Your MariaDB connection id is 356
Server version: 10.0.30-MariaDB-0

Copyright (c) 2000, 2016, Oracle,
Type 'help;' or '\h' for help. Ty
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| BDMED    |
| BDMEDOCLAB |
| BDPHARMA |
| information_schema |
| mysql    |
| performance_schema |
+-----+
6 rows in set (0.00 sec)
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous pouvons également vérifier le contenu des BDD (tables) détaillé dans le fichier de sauvegarde « `/home/sio/autosave.sql` » en l'éditant :

⇒ Données de la BDD « **BDMED** » :

```
GNU nano 2.2.6      Fichier : /home/sio/autosave.sql
-- MySQL dump 10.15  Distrib 10.0.30-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: localhost      Database: BDMED
--
-- Server version      10.0.30-MariaDB-0+deb8u1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Current Database: `BDMED`

CREATE DATABASE /*!32312 IF NOT EXISTS*/ `BDMED` /*!40100 DEFAULT CHARACTER SET$
USE `BDMED`;

--
-- Table structure for table `medecins`
--
DROP TABLE IF EXISTS `medecins`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `medecins` (
  `Nom` varchar(20) NOT NULL,
  `Prenom` varchar(20) NOT NULL,
  `Specialite` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `medecins`
--

LOCK TABLES `medecins` WRITE;
/*!40000 ALTER TABLE `medecins` DISABLE KEYS */;
/*!40000 ALTER TABLE `medecins` ENABLE KEYS */;
UNLOCK TABLES;
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

⇒ Données de la BDD « BDMEDOCLAB » :

```
-- Current Database: `BDMEDOCLAB`
--
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `BDMEDOCLAB` /*!40100 DEFAULT CHARACTERSET=
USE `BDMEDOCLAB`;
--
-- Table structure for table `produits`
--
DROP TABLE IF EXISTS `produits`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `produits` (
  `Nom_produit` varchar(30) NOT NULL,
  `Description` varchar(50) NOT NULL,
  `id_produit` int(100) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
--
-- Dumping data for table `produits`
--
LOCK TABLES `produits` WRITE;
/*!40000 ALTER TABLE `produits` DISABLE KEYS */;
/*!40000 ALTER TABLE `produits` ENABLE KEYS */;
UNLOCK TABLES;
--
-- Current Database: `BDPHARMA`
```

⇒ Données de la BDD « BDPHARMA » :

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `BDPHARMA` /*!40100 DEFAULT CHARACTER
USE `BDPHARMA`;
--
-- Table structure for table `medicaments`
--
DROP TABLE IF EXISTS `medicaments`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `medicaments` (
  `Nom_medicament` varchar(30) NOT NULL,
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

```

`Description` varchar(50) NOT NULL,
`id_medicament` int(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `medicaments`
--

LOCK TABLES `medicaments` WRITE;
/*!40000 ALTER TABLE `medicaments` DISABLE KEYS */;
/*!40000 ALTER TABLE `medicaments` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

```

```

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2017-05-30 15:22:01

```

b) Configuration du serveur MariaDB Maître

- Tout d'abord, sur le serveur **MariaDB Maître**, nous éditons et configurons le fichier « **/etc/mysql/my.cnf** » en modifiant la ligne « **bind-address** » en mettant l'adresse IP correspond à ce serveur :

```

GNU nano 2.2.6      Fichier : /etc/mysql/my.cnf
socket             = /var/run/mysqld/mysqld.sock
port               = 3306
basedir            = /usr
datadir            = /var/lib/mysql
tmpdir             = /tmp
lc-messages-dir   = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to li
# localhost which is more compatible and is not less
bind-address       = 192.168.1.140

```

- Nous modifions l'identifiant du serveur **Esclave** au niveau de la ligne « **server-id** » et décommentons la ligne « **log_bin** » concernant les logs binaires de **MySQL** qui représente le journal de transaction :

```

# other settings you may need to change.
server-id          = 101
log_bin            = /var/log/mysql/mysql-bin.log

```

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous redémarrons le service **MySQL** pour prendre en compte les modifications :

```
root@BDMEDLAB:~# systemctl restart mysql.service
root@BDMEDLAB:~#
```

- Maintenant, nous nous connectons de nouveau au service **MariaDB** :

```
root@BDMEDLAB:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.
Your MariaDB connection id is 38
Server version: 10.0.30-MariaDB-0

Copyright (c) 2000, 2016, Oracle,

Type 'help;' or '\h' for help. Ty
MariaDB [(none)]>
```

- Nous attribuons tous les droits aux bases de données (BDD) créées :

```
MariaDB [(none)]> grant replication slave on *.* to replica@%' identified by 'r
oot';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

- Nous nous déconnectons du service **MariaDB** :

```
MariaDB [(none)]> exit
Bye
root@BDMEDLAB:~#
```

c) Configuration du serveur de réplication MariaDB (Esclave)

- Maintenant, sur le serveur de réplication **MariaDB**, nous éditons et configurons le fichier « **/etc/mysql/my.cnf** » en modifiant la ligne « **bind-address** » en mettant l'adresse IP correspondant à ce serveur :

```
GNU nano 2.2.6      Fichier : /etc/mysql/my.cnf
socket              = /var/run/mysqld/mysqld.sock
port                = 3306
basedir             = /usr
datadir             = /var/lib/mysql
tmpdir              = /tmp
lc-messages-dir    = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to li
# localhost which is more compatible and is not less
bind-address        = 192.168.1.133
```


ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Ensuite, au niveau de la zone « **[mysqld]** », nous modifions l'identifiant du serveur **Esclave** au niveau de la ligne « **server-id** » et décommentons la ligne « **log_bin** » pour le journal de transaction :

```
# other settings you may need to change.
server-id          = 102
log_bin           = /var/log/mysql/mysql-bin.log
```

NB : Cette configuration est similaire à celle du serveur **Maître** au niveau des paramètres mais les valeurs à saisir sont différentes.

- Nous redémarrons le service **MySQL** pour prendre en compte les modifications :

```
root@BDMEDLABREP:~# systemctl restart mysql.service
root@BDMEDLABREP:~#
```

- Ensuite, nous retournons sur le serveur **Maître** et nous nous connectons de nouveau à **MariaDB** sur le serveur **Maître** :

```
root@BDMEDLAB:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.
Your MariaDB connection id is 38
Server version: 10.0.30-MariaDB-0

Copyright (c) 2000, 2016, Oracle,
Type 'help;' or '\h' for help. Ty
MariaDB [(none)]>
```

- Nous verrouillons les tables en cours d'exécution :

```
MariaDB [(none)]> flush tables with read lock;
Query OK, 0 rows affected (0.00 sec)
```

- Nous visualisons l'état du serveur **Maître** :

```
MariaDB [(none)]> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 |        604 |              |                   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- Nous nous déconnectons du service **MariaDB** :

```
MariaDB [(none)]> exit
Bye
root@BDMEDLAB:~#
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous effectuons une sauvegarde de toutes les BDD vers un fichier nommé « **mysql_dump.sql** » :

```
root@BDMEDLAB:~# mysqldump -u root -p --all-databases --lock-all-tables --events
> mysql_dump.sql
Enter password:
root@BDMEDLAB:~# _
```

- Nous nous reconnectons à **MariaDB** et déverrouillons les tables :

```
MariaDB [(none)]> unlock tables;
Query OK, 0 rows affected (0.00 sec)
```

- Nous nous déconnectons de nouveau du service **MariaDB** :

```
MariaDB [(none)]> exit
Bye
root@BDMEDLAB:~#
```

- Maintenant, nous envoyons le fichier de sauvegarde des BDD sur le serveur de réplication et saisissons le mot de passe « **root** » :

```
root@BDMEDLAB:~# scp mysql_dump.sql root@192.168.1.133:/var/tmp/
root@192.168.1.133's password:
mysql_dump.sql 100% 469KB 468.8KB/s 00:00
root@BDMEDLAB:~# _
```

- Ensuite, nous retournons sur le serveur **Esclave**, sauvegardons le fichier de sauvegarde « **mysql_dump.sql** » et saisissons le mot de passe « **root** » :

```
root@BDMEDLABREP:~# mysql -u root -p < /var/tmp/mysql_dump.sql
Enter password:
root@BDMEDLABREP:~#
```

- Nous nous connectons à **MariaDB** :

```
root@BDMEDLABREP:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Com
Your MariaDB connection id is 35
Server version: 10.0.30-MariaDB-0+de
Copyright (c) 2000, 2016, Oracle, Ma
Type 'help;' or '\h' for help. Type
MariaDB [(none)]>
```

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous modifions le statut « **master** » en renseignant les informations ci-dessous :

```

MariaDB [(none)]> change master to
-> master_host='192.168.1.140',
-> master_user='replica',
-> master_password='root',
-> master_log_file='mysql-bin.000001',
-> master_log_pos=604;
Query OK, 0 rows affected (0.02 sec)

```

- Nous démarrons l'esclave :

```

MariaDB [(none)]> start slave;
Query OK, 0 rows affected (0.00 sec)

```

- Nous ressaisissons cette commande pour démarrer la réplication sur le serveur **Esclave** car nous voyons un avertissement (« **1 warning** ») :

```

MariaDB [(none)]> start slave;
Query OK, 0 rows affected, 1 warning (0.00 sec)

```

- Enfin, nous visualisons l'état du serveur **Esclave** et constatons que la réplication a fonctionné car nous voyons l'adresse IP du serveur **Maître**, son fichier de log nommé « **mysql-bin.000001** », son numéro, le fichier secondaire de log du serveur **Esclave** nommé « **mysql-relay-bin.000002** » ainsi que la réplication en exécution :

```

MariaDB [(none)]> show slave status\G
***** 1. row *****
Slave_IO_State: Waiting for master to s
Master_Host: 192.168.1.140
Master_User: replica
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000001
Read_Master_Log_Pos: 604
Relay_Log_File: mysql-relay-bin.000002
Relay_Log_Pos: 535
Relay_Master_Log_File: mysql-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 604

```

ETTORI Bastien	BTS SIO 2 ^{ème} année
30 Mai 2017	Année scolaire : 2016/2017
Option : SISR	Version 2

- Nous visualisons l'emplacement du fichier de sauvegarde envoyé par le serveur **Maître** :

```
root@BDMEDLABREP:~# ls /var/tmp/  
mysql_dump.sql  
root@BDMEDLABREP:~# _
```

- Nous pouvons également afficher les BDD en nous connectant à **MariaDB**, constatons que celles-ci sont toutes présentes et donc répliquées et nous nous déconnectons de **MariaDB** :

```
MariaDB [(none)]> show databases;  
+-----+  
| Database  
+-----+  
|  BDMED  
|  BDMEDOCLAB  
|  BDPHARMA  
|  information_schema  
|  mysql  
|  performance_schema  
+-----+  
6 rows in set (0.00 sec)  
  
MariaDB [(none)]> exit  
Bye  
root@BDMEDLABREP:~# _
```

VIII) Conclusion

En conclusion, nous pouvons dire que le serveur de réplication **MariaDB (Esclave)** est fonctionnel car celui-ci a copié et récupéré toutes les données (BDD) du serveur **MariaDB Maître**.