

# Bases De Données

*TP MySQL/MariaDB*

*par*

*julien.breyault@u-pec.fr*

## Introduction

---

Le but de ce TP est :

- Mettre en place une **plateforme DEBIAN GNU/LINUX** servant de base à un **serveur LAMP** (manuellement pour la démarche générique, pas par paquet DEBIAN ici -qui serait préféré en production pourtant-)
- Y ajouter les composants, notamment **MySQL**
- Installer un **CMS**
- Administrer, dépanner et optimiser la base de données à l'aide d'**outils en ligne de commande et graphiques**
- Migrer vers **MariaDB (si besoin, DEBIAN<9 par ex)**, la nouvelle version communautaire de MySQL
- Monter un **cluster de serveurs** MariaDB
- Sauvegarder, restaurer automatiquement et de manière incrémentale

N'hésitez pas à créer des « snapshots » **à chaque étape-clé** afin de pouvoir revenir en arrière.

## Table de matières

Introduction.....	2
Système « de base ».....	4
Installation de la plateforme « LAMP ».....	5
Installation du moteur de blog « wordpress ».....	7
Actions sur la base.....	8
Sécurisation du service MySQL.....	10
Récupération de mot de passe root MySQL.....	11
Supervision et « tuning » MySQL.....	12
Sauvegardes, complètes et partielles.....	14
Archivage et journalisation automatique.....	15
Migration de MySQL vers MariaDB (uniquement pour DEBIAN<9).....	16
Cluster de serveurs MariaDB / Galera.....	17
Changement du répertoire de logs MySQL.....	18



debian

## **Systeme « de base »**

---

- Mettre les machines en NAT avec réseau activé entre machines virtuelles (Réseau NAT).
- Utiliser **la machine debian serveur de base avec** les composants : **SSH et composants usuels** (pas poste de travail et serveur d'impression!)



debian

## Installation de la plateforme « LAMP »

---

```
# apt-get update
# apt-get install mysql-server mysql-client
```

Remarquez que vous installez en fait directement MariaDB si vous êtes sur DEBIAN $\geq$ 9.  
Spécifier le mot de passe MySQL (root MySQL et non pas root système)

Pour le TP uniquement (pas en production) :

Dans /etc/mysql/mariadb.conf.d/50-server.cnf remplacer le bind-address 127.0.0.1 par 0.0.0.0

Redémarrer mysql :  
# service mysql restart

Ce n'est pas le sujet direct du TP, mais on aura besoin aussi d'un serveur web :  
# apt-get install apache2

Allez sur votre navigateur avec une autre machine virtuelle et tapez l'adresse de votre serveur pour vérifier qu'il est bien lancé.

Nous ajoutons les composants PHP :  
# apt-get install php libapache2-mod-php

Et rechargeons la nouvelle configuration apache :  
# /etc/init.d/apache2 restart

On teste la configuration PHP :  
Créez le fichier test.php  
# nano /var/www/test.php (par exemple ici, peu importe)  
et ajoutez-y la ligne suivante :  
<?php phpinfo(); ?>

Après ceci, faites un tour sur <http://<SERVERIP>/test.php> avec votre navigateur préféré pour commencer à vous en servir.  
Et on ouvre notre navigateur sur cette page (<http://ipduserveur/test.php>)

Nous allons avoir besoin du support de MySQL par PHP :  
apt-get install php-mysql

Et pourrions rajouter suivant ce que nous avons à installer ensuite (ne pas tout installer ici) :  
php-curl php-pear php-gd php-intl php-imagick php-imap php-mcrypt php-memcache php-pspell  
php-recode php-snmp php-sqlite php-tidy php-xmlrpc php-xsl ...

On redémarre apache :  
/etc/init.d/apache2 restart

Vérifier dans le test PHP que le(s) nouveaux modules PHP est (sont) bien chargé(s).

On va installer une interface graphique d'administration MySQL :

apt-get install phpmyadmin

On y accède ici : <http://votreip/phpmyadmin/> avec l'utilisateur « phpmyadmin » créé lors de l'installation du paquet (pour éviter la connexion directe avec le root mysql, depuis debian9) (si ce n'est pas le cas, créer un [lien symbolique](#) de /usr/share/phpmyadmin vers /var/www/phpmyadmin)

#### Informations :

Toutes les informations des utilisateurs et des privilèges sont stockées dans la base de données mysql. :

Les utilisateurs sont stockés dans la table user, avec leurs privilèges globaux (c'est-à-dire valables au niveau du serveur, sur toutes les bases de données).

La base mysql possède par ailleurs quatre tables permettant de stocker les privilèges des utilisateurs à différents niveaux.

- db : privilèges au niveau des bases de données.
- tables\_priv : privilèges au niveau des tables.
- columns\_priv : privilèges au niveau des colonnes.
- procs\_priv : privilèges au niveau des routines (procédures et fonctions stockées).

On peut aussi se connecter avec un **client « lourd »** comme **MySQLWorkbench** :

Il faut cependant créer un compte pour ce client afin de l'autoriser à se connecter.

Soit avec PMA, ou bien en ligne de commande:

```
CREATE USER 'root'@'%' IDENTIFIED BY 'mot_de_passe';  
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;
```

## Installation du moteur de blog « wordpress »

---

Installation type LAMP « générique », pour wordpress vous pourriez, pour de la production et sans objectif d'avoir la dernière version, utiliser le paquet DEBIAN natif.

wget <http://wordpress.org/latest.zip>

mysql -u root -p # On se connecte en root.

mysql>CREATE DATABASE wordpress ; # Création d'une base de données "wordpress".

mysql>CREATE USER 'wordpressuser'; # Création d'un utilisateur "wordpressuser".

mysql>SET PASSWORD FOR 'wordpressuser' = PASSWORD('Votremdp'); # Définir un mot de passe

mysql>GRANT ALL PRIVILEGES ON wordpress.\* TO wordpressuser@localhost IDENTIFIED by 'Votremdp'; #Attribution de tous les droits pour "wordpressuser" sur la base de données "wordpress"

mysql>exit # Quitter la console d'administration MySQL

apt-get install zip

unzip latest.zip -d /var/www/html/

chown -Rf www-data:www-data /var/www/html/wordpress

chmod 775 -R /var/www/html/wordpress

Lancer l'assistant d'installation en pointant avec un navigateur sur <http://ipserveur/wordpress>



debian

## Actions sur la base

Si vous saisissez une commande et que vous obtenez le prompt « -> » c'est que vous n'avez pas saisi la commande entièrement cela peut être dû notamment au fait que vous avez oublié de mettre un « ; » à la fin de la commande.

Si c'est le cas, saisissez « ; » au prompt « -> » pour terminer votre commande :

### Informations :

Il y a deux commandes principales pour gérer les **droits/les privilèges** en SQL :

- **GRANT** : permet d'attribuer un ou plusieurs droits/privilèges à un utilisateur sur une ou des tables.
- Syntaxe du **GRANT** :

```
GRANT nom_action ON nom_table TO nom_utilisateur;
```

- **REVOKE** : permet de retirer un ou plusieurs droits/privilèges à un utilisateur sur une ou des tables.
- Syntaxe du **REVOKE** :

```
REVOKE nom_action ON nom_table FROM nom_utilisateur;
```

### Tests pour le TP :

Ajouter le droit de faire un **INSERT** sur toutes les tables de la **BDD** utilisée pour l'utilisateur « **invite** » (par exemple):

```
mysql>GRANT INSERT ON * TO invite;
```

- Ajouter le droit de faire un **INSERT** sur la table « **table1** » pour l'utilisateur “**invite**” :

```
mysql>GRANT INSERT ON table1 TO invite;
```

- Attribuer tous les droits à l'utilisateur “**invite**” sur la base de données utilisée :

```
mysql>GRANT ALL ON * TO invite;
```

- Retirer le droit de faire un **INSERT** sur toutes les tables de la **BDD** utilisée pour l'utilisateur « **invite** » :

```
mysql> REVOKE INSERT ON * FROM invite;
```

- Retirer le droit de faire un **INSERT** sur la table « **table1** » pour l'utilisateur “**invite**” :

```
mysql> REVOKE INSERT ON table1 FROM invite;
```

- Retirer tous les droits à l'utilisateur “**invite**” sur toutes les tables de la BDD utilisée :

```
mysql> REVOKE ALL PRIVILEGES ON * FROM invite;
```

- **Si vous avez ce message : “Table is marked as crashed and should be repaired”**

Il faut se rendre dans le PhpMyAdmin pour réparer la table (l'opération est aussi faisable en ligne de commande sur le serveur MySQL). On doit ensuite se loguer puis se rendre dans la base de données qui contient la table qui doit être réparée. On la sélectionne, puis on se rend en bas de page pour cliquer sur la liste déroulante « Pour la sélection » et sélectionner « Réparer la table »

**On peut également le faire en ligne de commande, il faut pour cela savoir le nom de la base de données et de la table en question :**

```
repair table nombase.nomtable;
```



debian

## Sécurisation du service MySQL

---

#mysql\_secure\_installation

Cela permet de :

- s'assurer de bien avoir un mot de passe root MySQL
- empêcher d'accéder en root MySQL depuis le réseau (*par le socket MySQL ; par contre on accède à distance par SSH et ensuite on se connecte à SQL une fois rentré dans la machine donc en « local »*).
- empêcher les connexions anonymes
- supprimer la base de test (accessible par défaut par tous, même anonymes) et les privilèges qui autorisent tout le monde à accéder à toutes les bases commençant par **test\_**



debian

## Récupération de mot de passe root MySQL

---

On stoppe d'abord le service :

```
#!/etc/init.d/mysql stop
```

On le redémarre sans mot de passe (« mode de récupération ») :

```
# mysqld_safe --skip-grant-tables &
```

À l'aide du client MySQL on se connecte au serveur :

```
# mysql -u root
```

On arrive à l'invite de commande MySQL :

```
mysql> use mysql;  
mysql> update user set password=PASSWORD("nouveaumotdepasse") where user='root';  
mysql> flush privileges;  
mysql> quit
```

On recharge avec le nouveaux paramètres

On stoppe notre service « de dépannage »

```
#!/etc/init.d/mysql stop
```

On redémarre le service normalement

```
#!/etc/init.d/mysql start
```

On essaie de se connecter avec le nouveau mot de passe :

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

**OUF !!!**



debian

## Supervision et « tuning » MySQL

- [MySQLTuner](#) est un script PERL (**voir module PASR**) qui permet d'optimiser les performances d'un serveur de bases de données MySQL en faisant un diagnostic :

<https://raw.githubusercontent.com/major/MySQLTuner-perl/master/mysqltuner.pl>

Type de résultat obtenu après lancement du script :

```
----- Performance Metrics -----
[--] Total buffers: 34.0M global + 2.6M per thread (1000 max threads)
[!!] Allocating > 2GB RAM on 32-bit systems can cause system instability
[!!] Maximum possible memory usage: 2.6G (16% of installed RAM)
[OK] Slow queries: 0% (141/99M)
[!!] Highest connection usage: 100% (1001/1000)
```

- Mytop est un outil du type de « top » mais pour les bases de données. Il est aussi écrit en PERL (tiens, encore?) et a une interface shell permettant de voir en temps réel tâches, requêtes, performances,...

Il est **intégré à MariaDB client depuis debian 9** (un **apt-get install mytop** suffit sinon ;- ) et ensuite :

*mytop --prompt -d test* (pour la base « test »)

**Ensuite on peut copier un de ces fichiers-modèles vers le fichier de conf courant my.cnf (et modifier selon les résultats ci-dessus+conseils ci-dessous) :**

Nous avons des exemples pré-installés pour des types de machines différents dans /usr/share/doc/mysql-server-5.X.XX/ my-small.cnf,... my.large.cnf, my-huge.cnf et my-innodb-heavy-4G.cnf.

Attention aux noms, on est loin de configurations de machines avec 16GB RAM (large est pour 1à2Go de RAM, heavy 4G,...) : faire une règle de 3 (proportionnalité) sur toutes les valeurs mémoire .

- **Ajouter ou modifier si déjà présent :**

1. key\_buffer:

Regarder dans SHOW GLOBAL STATUS si

key\_reads/key\_read\_requests ratio est inférieur à 0.01 pour éviter l'accès disque

Si non, augmenter le key\_buffer

2. thread\_concurrency (uniquement si machine multicœurs):

2X le nombre de cœurs

3. Activer le log des requêtes lentes dans my.cnf (normalement juste à dé-commenter):

slow\_query\_log=1

slow\_query\_log\_file=/var/log/mysql/mysql-slow.log

**Et demander d'indiquer le temps le plus lent :**

long\_query\_time = 2

**Analyser les résultats:**

```
tail -n 1000 /var/log/mysql/mysql-slow.log
```

4. Vérifier qu'il n'y a pas de "connection leakage": there are not too many open connections in the database.

Utiliser SHOW PROCESSLIST;

5. Analyser les verous avec :

SHOW STATUS LIKE 'Table%';

et SHOW ENGINE INNODB STATUS (si vous utilisez INNODB comme moteur)



debian

## Sauvegardes, complètes et partielles

---

NB : On sauvegarde souvent une (ou plusieurs) base en omettant de sauvegarder également les utilisateurs et leurs privilèges. Ces informations sont dans la table users de la base mysql.

- **Procédures :**

### A chaud

Un backup à chaud est réalisé sans arrêter le serveur de données et éventuellement pendant la production.

■ mysqldump : Outil en ligne de commande livré avec mysqld.

**MYSQL\_PWD=mot-de-passe** (si vous voulez utiliser cette command dans un script, définir cette variable plutôt que mettre le mot de passe dans le script, c'est plus sûr)

**mysqldump --user=root --all-databases | gzip > save.mysql.sql.gz**

(ou -p mot-de-passe pour un lancement manuel)

Le fichier généré contiendra les commandes sql pour recréer bases, tables et données.

■ Les différents outils graphiques (mysqlworkbench, phpmyadmin, ...) proposent des utilitaire de sauvegarde, voire de restauration.

### A froid

Elle consiste à sauvegarder les fichiers “physiques” du serveur de données. Elle nécessite, par définition l'arrêt du serveur. Aucune activité n'est donc possible. C'est un avantage pour le DBA mais il est parfois difficile voire impossible de stopper l'activité en cours. En revanche si l'on dispose d'un réplica on peut tout à fait l'envisager. Celui ci se “re synchronisera” à son redémarrage.

La sauvegarde à froid consiste à archiver certains fichiers du serveur.

/var/lib/mysql/ et /var/log/mysql.log

Pour les tables MyISAM il s'agit plus précisément des fichiers .FRM, .MYI et .MYD

Pour les tables InnoDB : .FRM, les fichiers de données (ibdata), les fichiers des tables .ibd (si vous êtes en innodb\_file\_per\_table) et les fichiers journaux (iblogfile) intermédiaires.

## **2. Restauration**

### a. A chaud

**Sur un serveur en service** on va réutiliser un “dump” de la base sauvegardée et l'injecter. Un dump est un fichier contenant une succession de commandes sql dont l'exécution recréera l'intégralité de la base : les tables, les données, les relations ...

```
mysql -u user -p
```

```
mysql> use database votrebase ;
```

```
mysql > source /home/user/mysql/db.sql
```

OU directement pour inclure dans un script par exemple :

```
mysql -u myuser -p < /home/user/mysql/db.sql ( automatiser totale avec --password = < mot de passe > )
```

### b. A froid

Consiste à redéposer dans leurs répertoires les fichiers archivés (en ayant arrêté le service)

## Archivage et journalisation automatique

---

- Dans le module stockage, voir BackupPC qui sauvegarde très bien les bases de données : un outil pour tout c'est pratique ;-)

### - Un script « qui fait tout », en plus packagé par DEBIAN :

- Notification par mel
- Compression et chiffrement des sauvegardes
- Rotation des sauvegardes configurable
- Sauvegardes incrémentales

# `apt-get install automysqlbackup`

et affiner la configuration (si nécessaire) dans `/etc/default/automysqlbackup`

### - Un outil pas mal non plus, toujours packagé par DEBIAN :

`apt-get install rsnapshot`

`nano /etc/rsnapshot.conf`

```
snapshot_root /var/cache/rsnapshot/  
backup /etc/ localhost/  
retain hourly 6  
retain daily 7  
retain weekly 4
```

`cp /usr/share/doc/rsnapshot/examples/utils/backup_mysql.sh /usr/local/bin/`

`chown root.root /usr/local/bin/backup_mysql.sh`

`chmod o-w /usr/local/bin/backup_mysql.sh`

`nano /usr/local/bin/backup_mysql.sh`

```
/usr/bin/mysqldump --all-databases > mysqldump_all_databases.sql
```

ou pour base par base :

```
/usr/bin/mysqldump --defaults-file=/etc/mysql/debian.cnf DATABASE >  
DATABASE.SQL
```

`nano /etc/rsnapshot.conf`

```
backup_script /usr/local/bin/backup_mysql.sh localhost/mysqldump
```

TAB

`rsnapshot configtest`

Et ensuite utiliser l'outil cron pour programmer la sauvegarde.

### - Un outil permettant de sauvegarder à chaud, spécifique :

<https://www.percona.com/doc/percona-xtrabackup/LATEST/index.html>



debian

## Migration de MySQL vers MariaDB (uniquement pour DEBIAN<9)

---

Le SGBD(Système de gestion de base de données) MariaDB est un fork sous licence GPL de Oracle Mysql, qui offre la garantie de rester libre contrairement à l'avenir incertain de Oracle Mysql.

### Attention!!!

Sur un serveur en production, avant de procéder à la migration, faire une sauvegarde des bases de données!

- Mettre à jour le serveur Debian :  
apt-get update && apt-get upgrade  
apt-get install python-software-properties software-properties-common

MariaDB 10 .1 n'est pas encore disponible dans les dépôts officiels de Debian 8 (problème de compatibilité de licences Galera), pour pouvoir installer MariaDB 10.1 (avec Galera intégré) il faut ajouter le dépôt externe au sources.list :

liste à jour dispo sur <https://downloads.mariadb.org/mariadb/repositories/>

- Vous pouvez passer à 10.1 directement depuis MySQL>5.5 (notre cas ici), sinon PRÉFÉRER PASSER d'abord par MariaDB5.5 pour MySQL <=5.5.

```
apt-key adv --recv-keys --keyserver keyserver.ubuntu.com 0xc9cb082a1bb943db
(si problème, remplacer keyserver.ubuntu.com par hkp://keyserver.ubuntu.com:80/ ou
gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv 0xc9cb082a1bb943db
gpg --export --armor 0xc9cb082a1bb943db | apt-key add -) et vérifier (si proxy) que le proxy https est
bien défini.
```

```
nano /etc/apt/sources.list.d/mariadb.list
```

Y mettre (selon votre version/distribution au moment de l'installation) :

```
deb http://fr.mirror.babylon.network/mariadb/repo/10.1/debian jessie main
```

Mise à jour

```
apt-get update
```

Installation de MariaDB

```
apt-get install mariadb-server
```

Des messages de conflit vont s'afficher (normal on fait une migration) et le système va demander d'accepter la suppression de Mysql:

Lors de l'installation de MariaDB, il sera demandé un nouveau mot de passe pour le compte root, laisser vide pour conserver le même mot de passe.

Vérification de l'installation de MariaDB en lieu et place de Mysql :

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

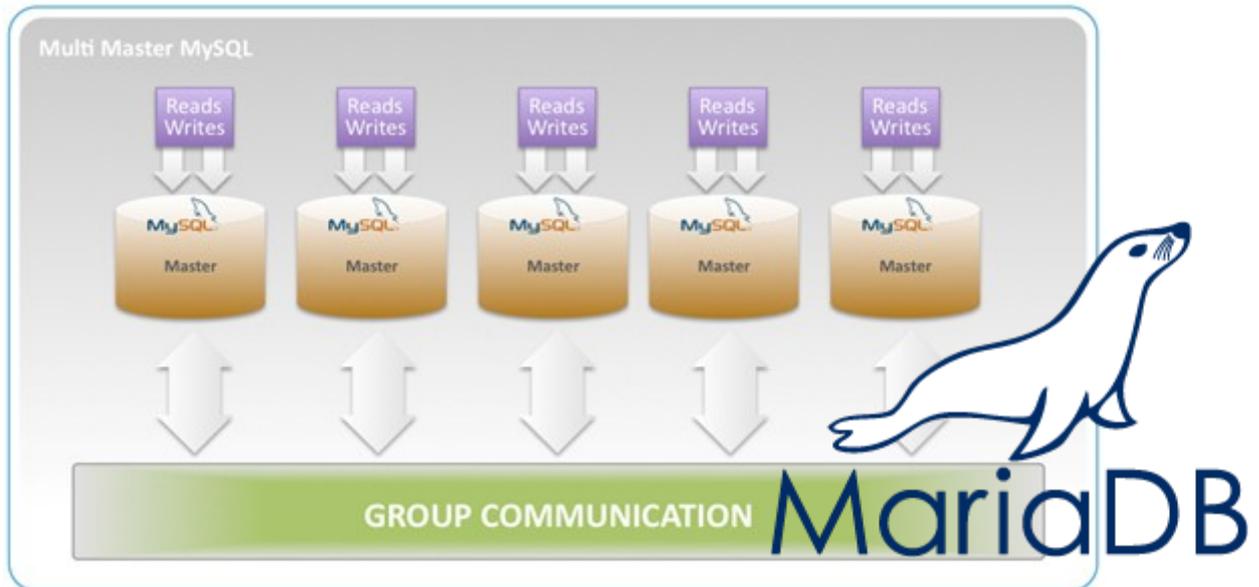


debian

## Cluster de serveurs MariaDB / Galera

Réutiliser une machine de base (Debian+outils usuels+SSH) et installer mariadb directement (sans MySQL auparavant)

Nous allons choisir une synchro maître-maître, bonne base pour un cluster :



En production il faut **3 serveurs au minimum** (pour gérer les conflits) mais nous pouvons commencer par tester avec 2.

### Création du cluster :

```
service mysql stop  
mysqld --wsrep-new-cluster  
ou bien :  
galera_new_cluster (sur systemes avec systemd)
```

### Sur les autres nœuds :

```
service mysql stop (normalement pas obligatoire mais sert parfois)  
mysqld --wsrep_cluster_address=gcomm://ip (ou nom DNS du serveur utilisé à la creation du cluster)
```

Vous pouvez ensuite voir les statuts de synchro avec cette commande sql :  
SHOW STATUS LIKE 'wsrep\_%';

Une page vous permet de générer une configuration « affinée » :  
<http://www.severalnines.com/New-Galera-Configurator/index.html>



debian

## Changement du répertoire de logs MySQL

Nécessaire lorsque il y a beaucoup de transactions, car on écrit à la fois les données ET les logs sur le même disque... d'où un grand nombre d'IO (voir cours stockage dans le module gestion de parc) pénalisant la réactivité du cluster de bases de données.

Dans my.cnf, changer :

```
log-bin=/chemindunouveaurépertoire/mysql-bin
```

Purger un maximum de logs (PURGE BINLOG...) pour ne pas avoir trop de fichiers à transférer.

Arrêter le service :

```
service mysql stop
```

Déplacer les fichiers vers le nouveau répertoire :

```
mv /var/lib/mysql /mysql-bin.* /chemindunouveaurépertoire/mysql-bin
```

On peut redémarrer le service :

```
service mysql start
```



Ce document est publié sous la licence libre **Creative Commons-BY-SA** :

<http://creativecommons.org/licenses/by-sa/3.0/deed.fr>

**BY : Paternité.** Vous devez citer le nom de l'auteur original.

**SA : Partage des Conditions Initiales à l'Identique.** Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

En outre, à chaque réutilisation ou distribution, vous devez faire apparaître clairement aux autres les conditions contractuelles de mise à disposition de cette création. Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits.