

Serveur de bases de données

1. Installation et configuration de MySQL



MySQL est un [système de gestion de bases de données](#) relationnelles (SGBDR). Il est distribué sous une double licence [GPL](#) et [propriétaire](#). Il fait partie des logiciels de gestion de [base de données](#) les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec [Oracle](#), [Informix](#) et [Microsoft SQL Server](#).

Pour l'installer, il vous suffit de taper cette commande :

```
apt-get install mysql-server
```

Maintenant pour y accéder, utiliser cette commande :

```
mysql --user=root mysql_
```

```
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.1.26-MariaDB-0+deb9u1 Debian 9.1

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> _
```

Il nous suffit maintenant de créer un utilisateur, nous lui accorderons ici tous les privilèges mais il est déconseillé de faire ça pour les autres utilisateurs, voyez ce compte comme le compte administrateur (remplacer le mot « *test* » par le nom d'utilisateur et la lettre « *a* » par le mot de passe souhaité) :

```
MariaDB [mysql]> CREATE USER 'test'@'%' IDENTIFIED BY 'a'; GRANT ALL PRIVILEGES ON *.* TO 'test'@'%'
WITH GRANT OPTION; FLUSH PRIVILEGES;
```

Vous vous connecterez sur phpMyAdmin (depuis n'importe quel poste grâce au '%') avec cet identifiant et ce mot de passe, mais avant passons à l'installation de **phpMyAdmin** !

2. Installation et configuration de phpMyAdmin

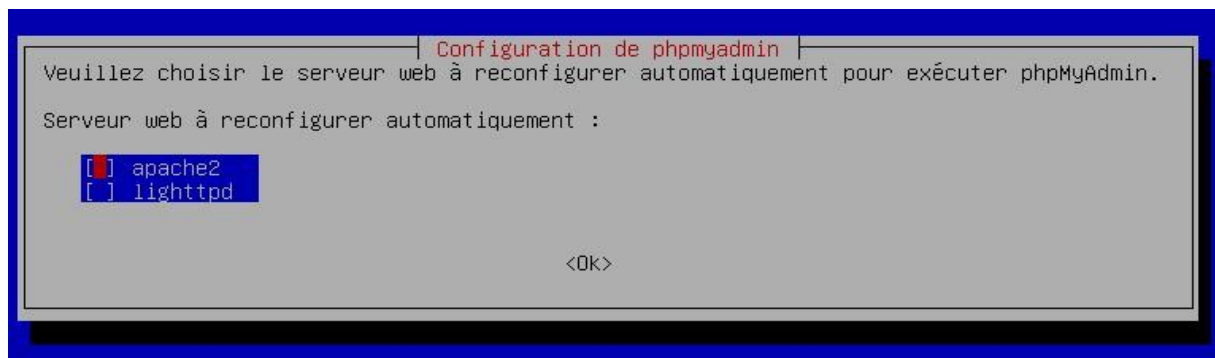


phpMyAdmin (PMA) est une [application Web](#) de gestion pour les [systèmes de gestion de base de données MySQL](#) réalisée principalement en [PHP](#) et distribuée sous licence [GNU GPL](#).

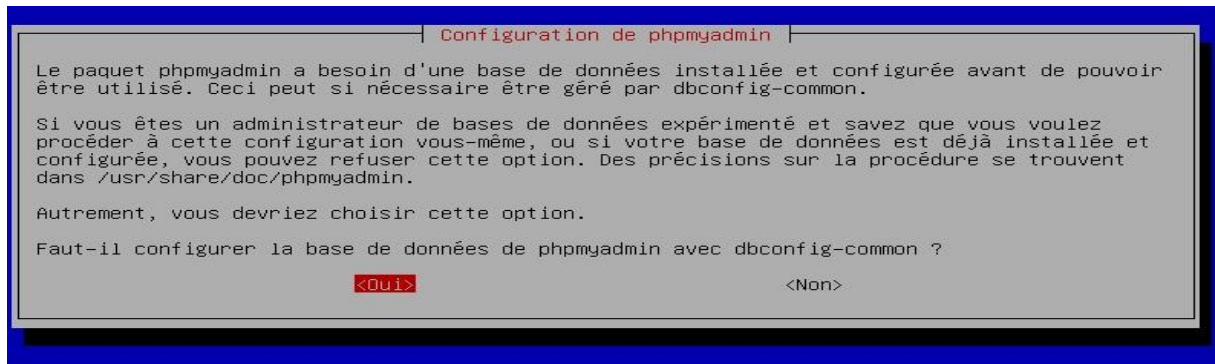
Pour l'installer il vous suffit de taper cette commande :

```
apt-get install phpmyadmin
```

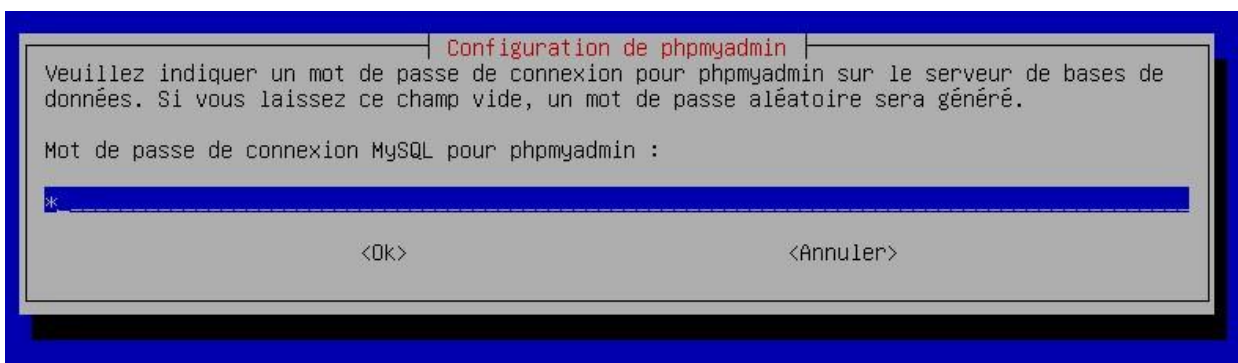
Une fois lancée, l'installation de phpMyAdmin va directement vous demander d'être configurée :



Sélectionner « *apache2* » (avec la touche espace) puis validez



Sélectionnez « Oui » pour configurer la base de données phpMyAdmin avec
« *dbconfig-common* ».



Choisissez un mot de passe de connexion MySQL pour phpMyAdmin puis dans la fenêtre suivante confirmez-le.

Voilà phpMyAdmin est installé et configuré il ne vous reste plus qu'à tester l'interface graphique permettant de créer/administrer des bases de données.

Pour vérifier que phpMyAdmin est bien fonctionnel, rendez-vous sur votre serveur avec un navigateur internet grâce à votre adresse IP et ajoutez seulement « */phpMyAdmin* » à la fin, par exemple : « *10.0.109.08/phpMyAdmin* »

S'il n'est pas accessible, utilisez cette commande :

```
ln -s /usr/share/phpmyadmin /var/www/html/
```



Bienvenue dans phpMyAdmin

Langue - *Language*

Français - French ▼

Connexion ⓘ

Utilisateur :

Mot de passe :

Exécuter

Bravo ! Vous avez réussi à l'installer et à le configurer, passons maintenant à la sécurisation !

3. Sécurisation de MySQL

Pour sécuriser MySQL il vous suffit seulement de la commande :

```
mysql_secure_installation
```

Il faut ensuite effectuer les changements suivants : (*y = yes et n = no*)

```
Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

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

Pas besoin de changer le mot de passe root si vous en avez déjà mis un, sinon mettez un mot de passe (*Attention à ne pas prendre un mot de passe trop simple*)

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!
```

Supprimer la connexion des utilisateurs anonyme (sans compte)

```
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!
```

Désactiver la connexion avec le compte root (compte par défaut)

```
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

Supprimer la base de données « test », qui est la base de données par défaut et qui peut être utilisée par tout le monde.

```
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!
```

Pour terminer tapez « y » qui permet de confirmer les changements et l'appliquer pour tous les utilisateurs.

```
Cleaning up...  
  
All done! If you've completed all of the above steps, your MariaDB  
installation should now be secure.  
  
Thanks for using MariaDB!  
root@ServeurWeb:~# _
```

Félicitation, MySQL est maintenant sécurisé !

Pour Aller plus loin :

Requis : Serveur Supplémentaire

Dans une entreprise, il est important de s'assurer qu'aucune donnée ne sera perdue. Il est donc nécessaire de faire des sauvegardes régulières et de mettre en place tous les mécanismes qui participeront à minimiser le risque de perte.

Voici l'une des manières possibles pour éviter les risques de pertes, mettre en place une réplication Maître/Esclave entre deux serveurs MySQL.

Vous avez donc besoin d'un autre serveur en plus de celui que vous venez de configurer, prenez par exemple votre serveur DHCP, pour ma part ce sera mon serveur DNS, les manipulations restent exactement les mêmes.

4. Répliquations des données : Configuration de base

Du coup, nous allons commencer par configurer le serveur « Maître » (serveur de bases de données principal, vous les ajouterez et modifierez sur celui-ci), utilisez donc le serveur que vous venez de sécuriser.

Commencez par modifier ce fichier : **/etc/mysql/mariadb.conf.d/50-server.cnf**.

- Modifiez la ligne : **bind-address = localhost** par **bind-address = 192.168.3.10** (IP de votre serveur)
- Décommentez (*supprimer le #*) et Modifiez la ligne : **serveur-id = 1** en **server-id = 10** (Le numéro doit être différents de 1, comme par exemple 10)
- Décommentez la ligne : **log_bin = /var/log/mysql/mysql-bin.log**.

Vous pouvez enregistrer (**CTRL + O**) et quitter le document (**CTRL + X**) !

Redémarrez maintenant les services MySQL : `root@ServeurWeb:~# service mysql restart`

Nous allons maintenant créer l'utilisateur qui servira à répliquer les données, commencez par accéder à MySQL (en utilisant la commande « **mysql -p** ») :

- Création de l'utilisateur :

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'replication'@'%' IDENTIFIED BY 'mdp';
```

Ici, « replication » est le nom d'utilisateur, vous pouvez le changer puis remplacer « mdp » par un mot de passe complexe. L'utilisateur peut se connecter via phpMyAdmin mais n'a aucun droit.

- Mettre à jour les modifications :

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

Vous avez configuré le fichier de base pour la répllication et créé l'utilisateur « Esclave », passons maintenant sur le second serveur, celui qui va nous servir d'Esclave. Installez donc MySQL sur ce dernier, nul besoin de créer d'utilisateur (Tu as oublié comment on fait ? [Clique ici !](#)).

Commencez aussi par modifier ce fichier : `/etc/mysql/mariadb.conf.d/50-server.cnf`.

- Modifiez la ligne : **bind-address** = `localhost` par **bind-address** = `192.168.3.11` (IP de votre serveur)
- Décommentez (*supprimer le #*) et Modifiez la ligne : **serveur-id** = `1` en **server-id** = `11` (Le numéro doit être différents de celui du serveur « Maître »)
- Décommentez la ligne : **log_bin** = `/var/log/mysql/mysql-bin.log`.
- Il vous faut maintenant ajouter des lignes justes en dessous :

```
GNU nano 2.7.4      Fichier : /etc/mysql/mariadb.conf.d/50-server.cnf
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#      other settings you may need to change.
server-id            = 11
log_bin              = /var/log/mysql/mysql-bin.log
expire_logs_days     = 10
max_binlog_size      = 100M
#binlog_do_db        = include_database_name
#binlog_ignore_db     = exclude_database_name
read_only             = 1
report-host          = ns1.vanbeek.loc
```

read_only = 1 permet de mettre ce serveur juste en lecture pour les bases de données, il ne pourra donc pas les modifier ou les supprimer.

report-host = ns1.vanbeek.loc permet de définir le nom de la machine actuelle (disponible dans : `/etc/hostname`)

Vous pouvez maintenant enregistrer (`CTRL + O`) et quitter le document (`CTRL + X`) !

Redémarrez maintenant les services MySQL : `root@ServeurWeb:~# service mysql restart`

5. Répliquations des données : Mise en Place

Les configurations de base sont maintenant terminées, retournons sur le serveur « Maître », et mettons en place cette répllication.

Nous allons d'abord bloquer toutes les tables, pour empêcher toutes modifications pendant les manœuvres :

```
MariaDB [(none)]> FLUSH TABLES WITH READ LOCK;_
```

Nous allons maintenant voir les **fichiers importants**, les informations à retenir sont encadré en rouge ci-dessous et ne sont pas forcément les mêmes pour tout le monde :

```
MariaDB [(none)]> SHOW MASTER STATUS;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB
mysql-bin.000004	327		

1 row in set (0.00 sec)

Ouvrez maintenant une nouvelle fenêtre (*ALT + F2* sous Debian 9), *cela nous évitera de quitter MySQL pour y retourner juste ensuite*, puis exécuter la commande suivante :

```
root@ServeurWeb:~# mysqldump -u root -p --all-databases --lock-all-tables --events > mysql_dump.sql
```

Cette commande permet d'effectuer un « dump », c'est-à-dire une sauvegarde, de toute les bases de données.

Retournons maintenant sur l'autre fenêtre (*ALT + F1* sous Debian 9), puis débloquons les tables :

```
MariaDB [(none)]> UNLOCK TABLES;
```

Nous allons maintenant envoyer nos données vers la machine Esclave en utilisant cette commande :

```
root@ServeurWeb:~# scp mysql_dump.sql thomas@ns1.vanbeek.loc:/tmp/  
thomas@ns1.vanbeek.loc's password:  
mysql_dump.sql 100% 497KB 43.2MB/s 00:00  
root@ServeurWeb:~#
```

thomas@ns1.vanbeek.loc est le nom de notre machine Esclave.

thomas est le nom d'utilisateur (*ne pas prendre l'utilisateur root car le transfert se fait en SSH (port 22) et pour plus de sécurité, le compte root ne peut pas s'y connecter*).

ns1.vanbeek.loc est le nom de la machine Esclave (*disponible dans le fichier `/etc/hostname` de cette dernière*).

Le mot de passe demandé correspond donc à l'utilisateur **thomas** de la machine **ns1.vanbeek.loc**.

Si vous avez un message d'erreur, c'est que votre machine esclave ne dispose pas de connexion SSH, il vous suffit donc de vous rendre sur cette dernière et de l'installer via cette commande :

```
root@ServeurWeb:~# apt-get install ssh
```

Vous n'avez rien à configurer, il vous suffit de réessayer d'envoyer vos données avec le Maître !

Il nous faut ensuite récupérer ces informations sur le serveur Esclave, rendez-vous sur ce dernier et exécuter la commande ci-dessous :

```
root@ns1:~# mysql -u root -p < /tmp/mysql_dump.sql
```

Une fois terminé, utilisez la commande « `mysql -p` » pour vous terminer la mise en place de la réplication, puis faite comme ci-dessous :

```
MariaDB [(none)]> CHANGE MASTER TO  
-> master_host='192.168.3.10',      —————> Adresse IP du Maître  
-> master_user='replication',      —————> Utilisateur MySQL pour la réplication  
-> master_password='2312',         —————> Mot de passe de l'utilisateur MySQL ci-dessus  
-> master_log_file='mysql-bin.000004', ————> Correspond aux informations du tableau, colonne "File" et "Position"  
-> master_log_pos=327;  
Query OK, 0 rows affected (0.40 sec)
```

Attention de bien respecter la syntaxe !

Démarrez la réplication :

```
MariaDB [(none)]> START SLAVE;
```

La réplication n'est pas instantanée, attendez quelques minutes !

6. Répliquations des données : Vérification

Vérifions maintenant que la réplication est bien fonctionnelle, exécutez la commande suivante sur chaque serveur dans MySQL :

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| Dev      |
| SISR     |
| information_schema |
| mysql    |
| performance_schema |
| phpmyadmin |
+-----+

7 rows in set (0.11 sec)
```

Le tableau doit être identique sur les 2 serveurs !

Ajouter maintenant une base de données sur le serveur Maître :

```
MariaDB [(none)]> CREATE DATABASE Web;
```

Puis vérifier le tableau :

```
+-----+
| Database |
+-----+
| Dev      |
| SISR     |
| Web      |
| information_schema |
| mysql    |
| performance_schema |
| phpmyadmin |
+-----+

7 rows in set (0.11 sec)
```

Félicitations, la réplication est fonctionnelle ! N'oubliez pas, seul le « Maître » peut créer, modifier ou supprimer des bases de données. En cas de perte des données, il vous suffit de passer l'esclave en maître et le maître en esclave pour pouvoir les récupérer.

