



Epreuve de Pratique des Techniques Informatiques

Nom et prénom : ALO David

Nature de l'activité

Synchronisation d'une base de données sous MySQL pour plateforme Ganesha

Contexte : Un centre de formation s'agrandi, il nous faut monter un nouveau serveur de base de données secondaire pour faciliter les modifications.

Objectifs : La base de données sera dupliquée et les modifications seront synchronisées entre les 2 serveurs.

Compétences mises en œuvre pour la réalisation de cette activité

C21	Installer et configurer un ordinateur
C22	Installer et configurer un réseau
C24	Installer un système de gestion de bases de données
C37	Administrer et maintenir une base de données

Conditions de réalisations

<u>Matériels</u> :	<u>Logiciels</u> :
- 2 PC	- serveur LAMP - MySQL

Durée : 45 mn

Autres contraintes et difficultés :

Description de l'activité réalisée

Situation initiale : 1 serveur LAMP en production et un deuxième serveur où il reste MySQL à installer

Situation finale : 2 serveurs sont identiques et se synchronisent

Installation de logiciels

```
> Aptitude install mysql-server
```

Sécurisation du serveur

```
> mysql_secure_installation
```

Il s'agit de :

- choisir un mot de passe pour l'utilisateur « root ».
- de verrouiller l'accès aux utilisateurs anonymes.
- d'autoriser ou pas la connexion de root à distance.

Un dernier paramètre de sécurité doit être renseigné dans `/etc/mysql/my.conf`, afin de déterminer si des connexions extérieures sont acceptées :

```
bind address =  
  #connexion interne seulement (adresse loopback)  
  #127.0.0.1  
  #connexions externes autorisées (adresse IP)  
  192.168.201.52
```

Cette configuration est à exécuter sur les 2 serveurs.

Pour prendre en compte ces modifications:

```
> /etc/init.d/mysql restart
```

Récupération de la base de données

Une fois la configuration de Mysql effectuée sur le nouveau serveur, il nous faut importer la base de données qui nous intéresse, se trouvant sur l'autre serveur.

- Extraction du code sql :

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

- Envoi du code :

```
> scp /chemin/bruz.sql usrafpi@192.168.201.65:/chemin/
```

- Import de la base :

Il faut avoir créé au préalable une base vierge pour pouvoir y importer la base distante nouvellement créée. sous mysql :

```
> create database bruz ; use  
> use bruz ;  
> source /chemin/bruz.sql
```

Enfin on doit déclarer quel utilisateur dispose de tel ou tel pouvoir à partir de quelle(s) machine(s) et sur quelle(s) base(s) voire quelle(s) table(s).

```
> GRANT update ON bruz.* TO 'ganesha'@'localhost' IDENTIFIED BY 'password'
```

Pour prendre en compte le nouvel utilisateur, nous devons rafraichir les privilèges MySQL :

```
> flush privileges;
```

Nous devons renseigner la configuration de Ganesha afin qu'elle puisse interroger la base de données avec ce compte:

```
> vim /var/www/ganeshaifti/ganesha/inc/config.inc.php
```

```
// Gestion de la connexion a la base de donnees
$BDD['name'] ='bruz';
$BDD['user'] ='ganesha';
$BDD['pass'] ='jslpdl';
$BDD['server']='localhost';
define('HTTP_MAIN',      'http://192.168.201.52/ganeshaifti/');
```

On peut à présent tester le site Ganesha situé sur le nouveau serveur.

Avant de mettre la réplication, nous devons déclarer les comptes SQL autorisés à répliquer :

Dans MySQL sur le serveur 1 :

```
> grant replication slave on *.* to replic1@192.168.201.65 identified by <password>;
```

Dans MySQL sur le serveur 2 :

```
> grant replication slave on *.* to replic2@192.168.201.52 identified by <password>;
```

Et pour prendre en compte les privilèges:

```
> flush privileges;
```

Nous allons maintenant configurer MySQL pour définir les paramètres de réplication. Toutes ces modifications sont à faire dans le fichier my.cnf:

```
> vim /etc/mysql/my.cnf
```

On y ajoute les lignes suivantes (ici la configuration du serveur 1):

-->skip-slave-start	pour que le serveur démarre normalement et ne démarre pas la réplication
-->auto_increment_increment=10	l'auto-incrémentation de toutes les clés se fait 10 en 10
-->auto_increment_offset=1	l'incrément des clés se termine par le chiffre 1
-->#paramétrage du maitre	
-->server-id=1	l'id du serveur est 1
-->log-bin	active la création de log pour le maitre
-->#paramétrage de l'esclave	

-->master-host=192.168.201.65	nom de son maitre
-->relay-log=slave1	log des fichiers reçus
-->relay-log-index=slave1	tous les fichiers reçus y sont répertoriés
-->master-user=replic2	login de l'esclave chez le maitre
-->master-password=jslpdl	mot de passe de l'esclave chez le maitre

Il faut configurer de la même façon sur le deuxième serveur.

Il faut à nouveau redémarrer MySQL:

```
> /etc/init.d/mysql restart
```

Afin de vérifier les états des maîtres et des esclaves (sous MySQL):

```
> show master ;
> show slave ;
```

Si les status sont propres et identiques, on peut lancer la réplication avec :

```
> start slave ;
```

Si les fichiers ont commencé à évoluer, il vaut mieux faire une mise à zéro de ces fichiers afin de ne pas corrompre notre base avec des informations obsolètes:

```
> reset master ;
> reset slave ;
```

On revérifie alors les états de chacun.

On ne peut se connecter sur le même compte sur les 2 sites en même temps. Le besoin réel n'est pas présent car l'administrateur ne se connecte jamais sur les 2 sites en même temps. Pour l'exercice, il faut alors créer un autre utilisateur ayant tous les droits.

Dans MySQL :

```
> use bruz
> insert into membres (prenom,login,password,type) values ('david', 'admindavid2',
md5('jslpdl'),'2') ;
```

Dès à présent, on peut se connecter sur les 2 sites en même temps avec les droits de l'administrateur. N'importe quelles des opérations seront synchronisées sur l'autre serveur de façon quasiment instantanée.

Si on le souhaite, il est possible de regarder les logs de mysql en tapant :

```
> mysqlbinlog /var/log/mysql/
```

Analyse des résultats obtenus

Objectif atteint : les 2 serveurs sont opérationnels et synchronisés

Bilan de l'activité : les 2 serveurs sont opérationnels et synchronisés