MySql Master-Slave

- Настройка реплики
- Добавление нового slave-сервера
- Назначение нового master-сервера

В этой статье будет описана простая настройка репликации master-slave в MySQL. Преимуществами такой репликации является простота настройки и поддержки. Недостаток – в случаи отказа master-сервера выбор нового master-сервера и переключение кассового сервера на него не автоматизированы.

Настройка реплики

Для начала необходимо установить MySQL сервера, которые будут входить в реплику. На каждой машине выполняем (эти и дальнейшие действия выполняются от имени суперпользователя root):

```
curl http://update.artix.su/static/cs/install_scripts/stable/cs_env/install_db.sh -o /tmp/install_db.sh
chmod +x /tmp/install_db.sh
/tmp/install_db.sh mysql
```

На сервере, который будет мастером, в файл /etc/mysql/my.cnf нужно добавить:

```
[mysqld]
server-id = 1 # Какое-то число, какое - придумайте сами, главное чтобы в реплике не было двух серверов с одним id
log-bin = mysql-bin
log-bin-index = mysql-bin.index
log-error = mysql-bin.err
relay-log = relay-bin
relay-log-info-file = relay-bin.info
relay-log-index = relay-bin.index
# По истечению этого времени логи будут удалены и реплицировать эти изменения не получится, настраивайте под себя. Это свойство опционально, но если его не указать то логи сильно разрастутся
expire_logs_days=7
```

На slave-серверах в файл /etc/mysql/my.cnf нужно добавить:

```
[mysqld]
server-id = 2 # Для каждого сервера id должен быть уникальным
relay-log = relay-bin
relay-log-info-file = relay-log.info
relay-log-index = relay-log.index
```

После изменения настроек необходимо перезапустить MySQL сервер командой:

```
service mysql restart
```

Теперь на master-сервере необходимо создать пользователя для репликации:

```
use mysql;
CREATE USER 'internal_user'@'%' IDENTIFIED WITH mysql_native_password BY 'password';
GRANT REPLICATION SLAVE ON *.* TO 'internal_user'@'%';
flush privileges;
```

Для последующей настройки slave-серверов необходимо имя файла, в который пишутся логи на master-сервере. Чтобы файл логов создался, необходимо запустить кассовый сервер и дождаться окончания миграции базы данных (настройка подключения кассового сервера и сервера БД). После миграции на master -сервере выполняем команду:

```
SHOW MASTER STATUS;
```

Для добавления в реплику slave-сервера, необходимо указать для них подключение к master-серверу. Здесь понадобится результат предыдущего запроса к ma ster-серверу. Выполнить на всех slave-серверах команду:

```
CHANGE MASTER TO MASTER_HOST='<ip_master_сервера>', MASTER_USER='internal_user', MASTER_PASSWORD='password', MASTER_LOG_FILE = 'mysql-bin.
000001', MASTER_LOG_POS = 0;
START SLAVE;
```

На этом настройка репликации закончена. Далее будет описано как добавить в реплику новый slave-сервер и назначить один из slave-серверов master-сервером.

Добавление нового slave-сервера

Если необходимо добавить новый slave-сервер, то просто выполните для него те же действия, что и для настройки других slave-серверов.

Внимание! Если на master-сервере было указано свойство expire_logs_days и части логов уже нет, то надо заблокировать master-сервер для записи (на время добавления нового сервера рекомендуется приостановить все сервисы, работающие с репликой):

```
FLUSH TABLES WITH READ LOCK;
SET GLOBAL read_only = ON;
```

Снять дамп всех баз данных:

```
mysqldump -u root -p --lock-all-tables [dbname] > dbdump.sql
```

И восстановить их на новом slave-сервере:

```
mysql -u root -p [dbname] < dbdump.sql
```

При настройке подключения к master-серверу указать тот MASTER_LOG_POS, который будет при выполнении команды SHOW MASTER STATUS на master-сервере.

Не забудьте разблокировать для записи master-сервер:

```
SET GLOBAL read_only = OFF;
```

Назначение нового master-сервера

В случаи отказа master-сервера необходимо остановить все сервисы, осуществляющие работу с этим сервером.

Сейчас на всех slave-серверах необходимо остановить получение данных с master-сервера. Для этого выполните команды:

```
STOP SLAVE IO_THREAD;
```

Посмотрите потоки, которые выполняются на salve-сервере:

```
SHOW PRECESSLIST;
```

После того как от SQL_THREAD появится сообщение о том, что slave-сервер выполнил все команды из relay-лога в своей базе, можно останавливать slave-сервер командой:

```
STOP SLAVE;
```

Далее необходимо из всех slave-серверов выбрать тот, который имеет последние реплецированные данные. Для этого на каждом сервере смотрим текущий статус и выбираем тот сервер, у которого Read_Master_Log_Pos имеет наибольшее значение:

```
SHOW SLAVE STATUS;
```

Затем полностью останавливаем новый master – потоки чтения бинарного лога и выполнения SQL-команд:

```
STOP SLAVE;
RESET MASTER;
```

В завершении необходимо переключить остальные slave-сервера на новый master-сервер. На каждом slave-сервере выполнить:

```
STOP SLAVE;
CHANGE MASTER TO MASTER_HOST="<ip_нового_master_сервера>";
START SLAVE;
```

Стоит учитывать что разница данных между slave-серверами и новым master-сервером может быть, так как часть логов была утеряна (об этом будут свидетельствовать разные значения Read_Master_Log_Pos на slave-серверах). Если необходимо чтобы реплика после восстановления была полностью синхронизирована, то придется перед назначением нового master-сервера сделать дамп данных этого сервера и восстановить его на slave -серверах (подробнее этот процесс описан в разделе "Добавление нового slave-сервера").