



ОНЛАЙН-ОБРАЗОВАНИЕ

# MySQL репликация

Ставим **percona-server-57** по [инструкции](#). В целом она сводится к следующему:

```
[root@otuslinux ~] yum install  
http://www.percona.com/downloads/percona-release/redhat/0.1-6/percona-release-0.1-6.noarch.rp  
m -y  
[root@otuslinux ~] yum install Percona-Server-server-57 -y
```

Стенд для повторения всех команд можно взять [отсюда](#).

По умолчанию Persona хранит файлы в таком виде:

- Основной конфиг в `/etc/my.cnf`
- Так же инклюдится директория `/etc/my.cnf.d/` - куда мы и будем складывать **наши** конфиги.
- Дата файлы в `/var/lib/mysql`

Копируем конфиги из `/vagrant/conf.d` в `/etc/my.cnf.d/`

```
[root@otuslinux ~] cp /vagrant/conf/conf.d/* /etc/my.cnf.d/
```

После этого можно запустить службу:

```
[root@otuslinux ~] systemctl start mysql
```

При установке `Percona` автоматически генерирует пароль для пользователя `root` и кладет его в файл `/var/log/mysqld.log`:

```
[root@otuslinux ~] cat /var/log/mysqld.log | grep 'root@localhost:' | awk '{print $11}'
```

`*mulP>&68v/A`

Подключаемся к `mysql` и меняем пароль для доступа к полному функционалу:

```
[root@otuslinux ~] mysql -uroot -p'*mulP>&68v/A'
```

```
mysql > ALTER USER USER() IDENTIFIED BY 'YourStrongPassword';
```

Репликацию будем настраивать с использованием **GTID**. Что это такое и зачем это надо можно почитать [здесь](#).

Следует обратить внимание, что атрибут **server-id** на мастер-сервере должен **обязательно** отличаться от **server-id** слейв-сервера. Проверить какая переменная установлена в текущий момент можно следующим образом:

```
mysql> SELECT @@server_id;
```

```
+-----+
| @@server_id |
+-----+
|           1 |
+-----+
```

Убеждаемся что GTID включен:

```
mysql> SHOW VARIABLES LIKE 'gtid_mode';
```

Variable_name	Value
gtid_mode	ON

Создадим тестовую базу bet и загрузим в нее дамп и проверим:

```
mysql> CREATE DATABASE bet;
```

```
Query OK, 1 row affected (0.00 sec)
```

```
[root@otuslinux ~]# mysql -uroot -p -D bet < /vagrant/bet.dmp
```

```
mysql> USE bet;
```

```
mysql> SHOW TABLES;
```

Tables_in_bet
bookmaker
competition
events_on_demand
market
odds
outcome
v_same_event



Создадим пользователя для репликации и даем ему права на эту самую репликацию:

```
mysql> CREATE USER 'repl'@'%' IDENTIFIED BY '!OtusLinux2018';
```

```
mysql> SELECT user,host FROM mysql.user where user='repl';
```

user	host
repl	%

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'%' IDENTIFIED BY '!OtusLinux2018';
```

Дампим базу для последующего залива на слэйв и игнорируем таблицы по заданию:

```
[root@otuslinux ~] mysqldump --all-databases --triggers --routines --master-data  
--ignore-table=bet.events_on_demand --ignore-table=bet.v_same_event -uroot -p > master.sql
```

На этом настройка Master-а завершена. Файл дампа нужно залить на слейв.

Так же точно копируем конфиги из `/vagrant/conf.d` в `/etc/my.cnf.d/`  
`[root@otuslinux ~]# cp /vagrant/conf/conf.d/* /etc/my.cnf.d/`

Правим в `/etc/my.cnf.d/01-basics.cnf` директиву `server-id = 2`

```
mysql> SELECT @@server_id;
```

@@server_id
2

Раскомментируем в `/etc/my.cnf.d/05-binlog.cnf` строки:

```
#replicate-ignore-table=bet.events_on_demand
```

```
#replicate-ignore-table=bet.v_same_event
```

Таким образом указываем таблицы которые будут игнорироваться при репликации

Заливаем дампы мастера и убеждаемся что база есть и она без лишних таблиц:

```
mysql> SOURCE /mnt/master.sql
```

```
mysql> SHOW DATABASES LIKE 'bet';
```

```
+-----+  
| Database (bet) |  
+-----+  
| bet            |  
+-----+
```

```
mysql> USE bet;
```

```
mysql> SHOW TABLES;
```

```
+-----+  
| Tables_in_bet |  
+-----+  
| bookmaker     |  
| competition   |  
| market        |  
| odds          |  
| outcome       |  
+-----+
```

```
+-----+ # видим что таблиц v_same_event и events_on_demand нет
```

Ну и собственно подключаем и запускаем слейв:

```
mysql> CHANGE MASTER TO MASTER_HOST = "192.168.11.150", MASTER_PORT = 3306,  
MASTER_USER = "repl", MASTER_PASSWORD = "!OtusLinux2018", MASTER_AUTO_POSITION = 1;
```

```
mysql> START SLAVE;
```

```
mysql> SHOW SLAVE STATUS\G
```

```
***** 1. row *****
```

```
Slave_IO_State: Waiting for master to send event
```

```
Master_Host: 192.168.11.150
```

```
Master_User: repl
```

```
Master_Port: 3306
```

```
Connect_Retry: 60
```

```
Master_Log_File: mysql-bin.000001
```

```
Read_Master_Log_Pos: 313
```

```
Relay_Log_File: mysql2-relay-bin.000002
```

```
Relay_Log_Pos: 526
```

```
Relay_Master_Log_File: mysql-bin.000001
```

```
Slave_IO_Running: Yes
```

```
Slave_SQL_Running: Yes
```

Видно что репликация работает, gtid работает и игнорятся таблички по заданию:

Slave\_IO\_State: Waiting for master to send event

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

Replicate\_Ignore\_Table: bet.events\_on\_demand,bet.v\_same\_event

Retrieved\_Gtid\_Set: 2861c789-9f2a-11e8-a9fb-525400c9c704:1

Executed\_Gtid\_Set: 2861c789-9f2a-11e8-a9fb-525400c9c704:1

Проверим репликацию в действии. На мастере:

```
mysql> USE bet;
```

```
mysql> INSERT INTO bookmaker (id,bookmaker_name) VALUES(1,'1xbet');
```

```
mysql> SELECT * FROM bookmaker;
```

id	bookmaker_name
1	1xbet
4	betway
5	bwin
6	ladbrokes
3	unibet



На слейве:

```
mysql> SELECT * FROM bookmaker;
```

```
+----+-----+  
| id | bookmaker_name |  
+----+-----+  
| 1 | 1xbet          |  
| 4 | betway         |  
| 5 | bwin           |  
| 6 | ladbrokes      |  
| 3 | unibet         |  
+----+-----+
```

В binlog-ах на слейве также видно последнее изменение, туда же он пишет информацию о GTID:

```
SET @@SESSION.GTID_NEXT= '2861c789-9f2a-11e8-a9fb-525400c9c704:2'/*!*/;  
# at 418  
#180814 2:42:38 server id 1  end_log_pos 491 CRC32 0xd18180a2  Query  thread_id=26  exec_time=0  
error_code=0  
SET TIMESTAMP=1534196558/*!*/;  
BEGIN  
/*!*/;  
# at 491  
#180814 2:42:38 server id 1  end_log_pos 618 CRC32 0x609871c1  Query  thread_id=26  exec_time=0  
error_code=0  
use `bet`/*!*/;  
SET TIMESTAMP=1534196558/*!*/;  
INSERT INTO bookmaker (id,bookmaker_name) VALUES(1,'1xbet')  
/*!*/;  
# at 618  
#180814 2:42:38 server id 1  end_log_pos 649 CRC32 0x8d303d9d  Xid = 1251  
COMMIT/*!*/;
```