

Отчет о выполнении задания №2

Настроить репликацию СУБД master-slave. Воспроизвести ситуацию повреждения данных и восстановления их по состоянию до момента повреждения.

1) Настройка репликации на master sql сервере и на slave sql сервере.

Для этого в конфиг серверов добавлены id серверов, включен binlog, relaylog и задан список баз данных, подлежащих репликации. В mysql на master добавляем пользователя slave_user, ему даём права.

На мастере делаем дамп базы

На слейве создаем базу

create database sprintb;

Заливаем в неё дамп с мастер базы

mysql -uroot -p sprintb < sprintb.sql

На мастере:

SHOW MASTER STATUS;

```
aggy@gb:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 1638
Server version: 10.1.44-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000009 | 1035335 | sprintb,drop_test |
+-----+-----+-----+-----+
1 row in set (0.01 sec)

MariaDB [(none)]> █
```

смотрим Position

На слейве:

CHANGE MASTER TO MASTER_HOST='185.251.89.25', MASTER_USER='slave_user',

MASTER_PASSWORD='Пароль', MASTER_LOG_FILE = 'mysql-bin.000001',

MASTER_LOG_POS = Position которая была на мастере;

start slave;

Смотрим статус слейва

SHOW SLAVE STATUS\G

```

aggy@299652-mike199567:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 145
Server version: 5.7.29-0ubuntu0.18.04.1-log (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> SHOW SLAVE STATUS\G;
***** 1. row *****
Slave_IO_State: Waiting for master to send event
Master_Host: 185.251.89.25
Master_User: slave_user
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000009
Read_Master_Log_Pos: 1035335
Relay_Log_File: mysql-relay-bin.000118
Relay_Log_Pos: 4768
Relay_Master_Log_File: mysql-bin.000009
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table: phpmyadmin.%
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 1035335
Relay_Log_Space: 9585

```

```

Relay_Log_Space: 9585
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_UUID:
Master_Info_File: /var/lib/mysql/master.info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
Master_Retry_Count: 86400
Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
1 row in set (0.00 sec)
ERROR:
No query specified

```

Должны быть

Slave_IO_Running: Yes

Slave_SQL_Running: Yes

```

root@gb: /etc/mysql/mariadb.conf.d
#slow_query_log_file = /var/log/mysql/mariadb-slow.log
#long_query_time = 10
#log_slow_rate_limit = 1000
#log_slow_verbosity = query_plan
#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 = 1
log_bin = /var/log/mysql/mysql-bin.log
expire_logs_days = 10
max_binlog_size = 1M
binlog_do_db = sprintb
binlog_do_db = drop_test
#binlog_do_db = include_database_name
#binlog_ignore_db = exclude_database_name
#
:

```

```
aggy@299652-mike199567: /etc/mysql/mysql.conf.d
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#slow_query_log            = 1
#slow_query_log_file       = /var/log/mysql/mysql-slow.log
#long_query_time            = 2
#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                  = 2
relay_log                  = /var/log/mysql/mysql-relay-bin.log
log_bin                    = /var/log/mysql/mysql-bin.log
relay_log_index            = /var/log/mysql/mysql-relay-bin.index
expire_logs_days           = 10
max_binlog_size            = 1M
binlog_do_db               = sprintb, drop_test
#binlog_do_db              = include_database_name
#binlog_ignore_db          = include_database_name
# ignore pnpmyadmin internals from master
replicate_wild_ignore_table = phpmyadmin.%
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
#
# * Security Features
:
```

2) На slave сервере налажен периодический backup базы данных с помощью mysqldump и crontab .

Для организации периодических backup на slave написан bash скрипт , вызывающий утилиту mysqldump с нужными параметрами и периодически вызывающейся системой cron. Компрессированные дампы содержат в имени файла дату сохранения.

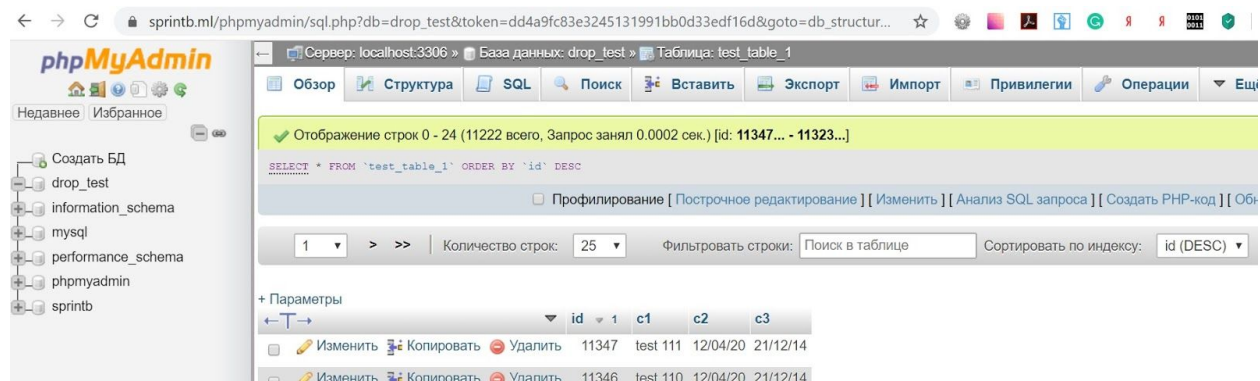
```
#!/bin/bash
#
# backup slave MySQL database
#
file="/backup/mysql/drop_test_$(date +%Y-%m-%d-%H%M').sql.gz"
echo ${file}
mysqldump -u root --apply-slave-statements --single-transaction --flush-logs --master-data=2 --databases drop_test | gzip > ${file}
# delete old backups (older than 180 minutes)
#find "/backup/mysql/drop_test*.sql.gz" -maxdepth 1 -type f -mmin +180 -exec rm -fv {} \;
(END)

# backup and replication tests
5 * * * * /backup/mysql/mysql_backup.sh >>/backup/mysql/mysql_backup.log 2>&1

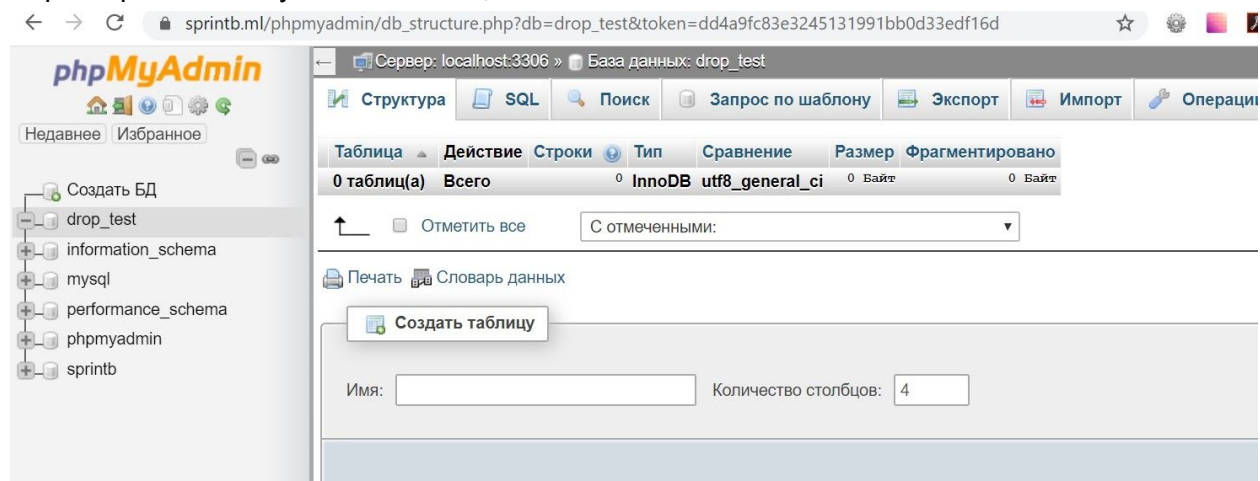
-rw-r--r-- 1 root root 52211 Apr 12 19:05 drop_test_2020-04-12-1905.sql.gz
-rw-r--r-- 1 root root 665 Apr 12 20:05 drop_test_2020-04-12-2005.sql.gz
-rw-r--r-- 1 root root 57055 Apr 12 21:05 drop_test_2020-04-12-2105.sql.gz
-rw-r--r-- 1 root root 57055 Apr 12 22:05 drop_test_2020-04-12-2205.sql.gz
-rw-r--r-- 1 root root 57055 Apr 12 23:05 drop_test_2020-04-12-2305.sql.gz
-rw-r--r-- 1 root root 57054 Apr 13 00:05 drop_test_2020-04-13-0005.sql.gz
-rw-r--r-- 1 root root 2555 Apr 13 00:05 mysql_backup.log
-rwxr-xr-x 1 root root 407 Apr 12 03:14 mysql_backup.sh*
```


3) Тест восстановления

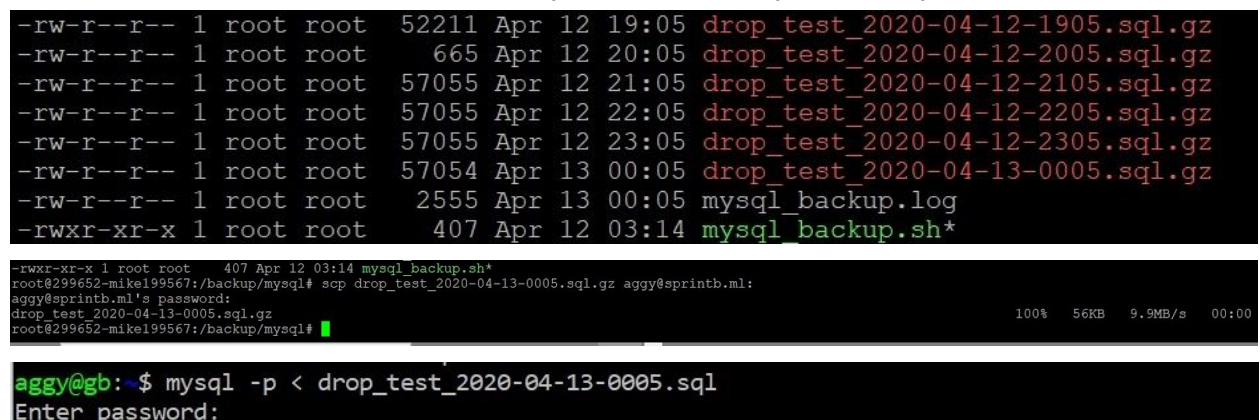
Через какое-то время после очередного дампа в тестовую базу было добавлено несколько записей с помощью bash скрипта.



Через несколько минут на мастере был произведен drop таблицы тестовой базы. Затем зафиксировали отсутствие таблицы на slave.



Соответствующий файл периодического дампа был скопирован на master сервер и его содержимое было внесено в master базу посредством утилиты mysql.



С помощью утилиты mysqlbinlog на master сервере был произведен поиск необходимых временных интервалов для окончательного восстановления базы в дополнение к периодическому дампу (от начала последних изменений до команды drop).

```
aggy@gb:~$ ll /var/log/mysql/
total 7164
drwxr-s-- 2 mysql adm      4096 Apr 12 17:20 ./
drwxrwxr-x 13 root  syslog  4096 Apr 12 06:25 ../
-rw-r----- 1 mysql adm     5988 Apr 12 17:11 error.log
-rw-r----- 1 mysql adm    12165 Apr 12 16:41 error.log.1
-rw-rw---- 1 mysql adm   1026764 Apr 11 12:20 mysql-bin.000001
-rw-rw---- 1 mysql adm   1048736 Apr 11 18:21 mysql-bin.000002
-rw-rw---- 1 mysql adm   1048844 Apr 12 10:43 mysql-bin.000003
-rw-rw---- 1 mysql adm   1048702 Apr 12 11:51 mysql-bin.000004
-rw-rw---- 1 mysql adm   1049388 Apr 12 12:59 mysql-bin.000005
-rw-rw---- 1 mysql adm     76983 Apr 12 16:41 mysql-bin.000006
-rw-rw---- 1 mysql adm      346 Apr 12 16:45 mysql-bin.000007
-rw-rw---- 1 mysql adm   1073627 Apr 12 17:20 mysql-bin.000008
-rw-rw---- 1 mysql adm   898722 Apr 12 21:33 mysql-bin.000009
-rw-rw---- 1 mysql adm      288 Apr 12 17:20 mysql-bin.index
aggy@gb:~$ date
Sun Apr 12 21:36:02 UTC 2020
aggy@gb:~$ mysqlbinlog /var/log/mysql/mysql-bin.000009 >tmp.sql
```

```
aggy@gb: ~
#200412 21:25:53 server id 1  end_log_pos 448489      Query  thread_id=1440  exec_time=0      error_code=0
SET TIMESTAMP=1586726753/*!*/;
SET @@session.pseudo_thread_id=1440/*!*/;
/*!C utf8mb4 *//*!*/;
SET @@session.character_set_client=45,@@session.collation_connection=224,@@session.collation_server=45/*!*/;
DROP TABLE `test_table_1` /* generated by server */
/*!*/;
# at 448489
#200412 21:25:53 server id 1  end_log_pos 448527      GTID 0-1-14267 trans
/*!100001 SET @@session.gtid_seq_no=14267/*!*/;
BEGIN
/*!*/;
# at 448527
#200412 21:25:53 server id 1  end_log_pos 448690      Query  thread_id=1439  exec_time=0      error_code=0
SET TIMESTAMP=1586726753/*!*/;
/*!C utf8mb4 *//*!*/;
SET @@session.character_set_client=45,@@session.collation_connection=45,@@session.collation_server=45/*!*/;
DELETE FROM `phpmyadmin`.`pma__column_info` WHERE db_name = 'drop_test' AND table_name = 'test_table_1'
/*!*/;
# at 448690
#200412 21:25:53 server id 1  end_log_pos 448755      Query  thread_id=1439  exec_time=0      error_code=0
SET TIMESTAMP=1586726753/*!*/;
COMMIT
/*!*/;
# at 448755
#200412 21:25:53 server id 1  end_log_pos 448793      GTID 0-1-14268 trans
/*!100001 SET @@session.gtid_seq_no=14268/*!*/;
BEGIN
#
```

С помощью mysqlbinlog все транзакции с тестовой базой внутри этого временного интервала сохранены в sql файл, который потом был загружен на master сервер утилитой mysql.

```
aggy@gb:~$ mysqlbinlog /var/log/mysql/mysql-bin.000009 --start-datetime="2020-04-12 21:12:00" --stop-datetime="2020-04-12 21:25:00" >tmp.sql
```

```
aggy@gb:~$ less tmp.sql
aggy@gb:~$ mysql -p < tmp.sql
```

Для проверки правильности восстановления было сравнено количество записей в тестовой базе и содержимое последней записи до и после теста. Они оказались идентичными.

The screenshot shows the phpMyAdmin web interface. The left sidebar displays a database tree with 'drop_test' selected, containing 'Новая' and 'test_table_1'. The main panel shows the 'test_table_1' table structure and data. A green message bar at the top indicates 'Отображение строк 0 - 24 (11222 всего, Запрос занял 0.0002 сек.) [id: 11347... - 11323...]'. Below this, the SQL query 'SELECT * FROM `test_table_1` ORDER BY `id` DESC' is shown. The table navigation controls show 'Количество строк: 25' and 'Сортировать по индексу: id (DESC)'. The table data is displayed with columns 'id', 'c1', 'c2', and 'c3'. The visible rows are:

	id	c1	c2	c3
<input type="checkbox"/>	11347	test 111	12/04/20	21/12/14
<input type="checkbox"/>	11346	test 110	12/04/20	21/12/14
<input type="checkbox"/>	11345	test 109	12/04/20	21/12/14
<input type="checkbox"/>	11344	test 108	12/04/20	21/12/14