

Homework_Lesson 18

Цель: научиться реализовывать отказоустойчивое решение для баз данных (на примере PostgreSQL либо MySQL)

1. Создать два сервера, установить на них PostgreSQL либо MySQL и подключить их к одной сети.

Развернуты 2 виртуалки Ubuntu:

```
● sttewie@sttewie:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.20.204 netmask 255.255.254.0 broadcast 192.168.21.255
    inet6 fe80::215:5dff:fe15:522 prefixlen 64 scopeid 0x20<link>

● sttewie@sttewie:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.20.202 netmask 255.255.254.0 broadcast 192.168.21.255
```

2. Настроить репликацию между серверами, чтобы изменения, вносимые на одном сервере, автоматически реплицировались на другой. (Зеркалирование)

3. Настроить отказоустойчивость, используя репликацию и механизм автоматического переключения между серверами. (Кластер*) Кластер по желанию и по возможности.

Отредактируем конфигурационный файл MySQL (my.cnf) на обоих серверах

Для машины 192.168.20.202

```
server-id=1                # Уникальный ID сервера
log-bin=mysql-bin          # Включение бинарных логов
binlog_format=ROW          # Формат логов для репликации
gtid_mode=ON               # Включение GTID для репликации
enforce_gtid_consistency=ON
log-slave-updates=ON       # Логи обновлений на слейв-узлах
```

Для машины 192.168.20.204

```
[my.cnf]
server-id=2
log-bin=mysql-bin
binlog_format=ROW
gtid_mode=ON
enforce_gtid_consistency=ON
log-slave-updates=ON
```

На каждом сервере создаем пользователя, который будет использоваться для репликации.

```
mysql> CREATE USER 'replica'@'%' IDENTIFIED BY '5550';
ANT REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'replica'@'%';
FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.04 sec)
```

На машине 192.168.20.204 выполним команду для настройки связи с мастер машиной 192.168.20.202:

```
mysql> CHANGE MASTER TO
-> MASTER_HOST='192.168.20.202',
-> MASTER_USER='replica',
-> MASTER_PASSWORD='5550',
-> MASTER_LOG_FILE='mysql-bin.000001',
-> MASTER_LOG_POS=154,
-> GET_MASTER_PUBLIC_KEY=1;
TART SLAVE;
Query OK, 0 rows affected, 9 warnings (0.21 sec)
```

На машине 192.168.20.202 выполним аналогичную команду для настройки связи с 192.168.20.204:

```
mysql> CHANGE MASTER TO
-> MASTER_HOST='192.168.20.204',
-> MASTER_USER='replica',
-> MASTER_PASSWORD='5550',
-> MASTER_LOG_FILE='mysql-bin.000001',
-> MASTER_LOG_POS=154,
-> GET_MASTER_PUBLIC_KEY=1;
TART SLAVE;
Query OK, 0 rows affected, 9 warnings (0.16 sec)
```

Добавляем тестовые данные на машине 192.168.20.202:

```
mysql> CREATE DATABASE test_db;
Query OK, 1 row affected (0.04 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test_db |
+-----+
5 rows in set (0.00 sec)

mysql> USE test_db;
PRIMARY KEY, value VARCHAR(50));
INSERT INTO replDatabase changed
mysql> CREATE TABLE replication_test (id INT PRIMARY KEY, value VARCHAR(50));
ication_test VALUES (1, 'Server 202');
Query OK, 0 rows affected (0.26 sec)

mysql> INSERT INTO replication_test VALUES (1, 'Server 202');
Query OK, 1 row affected (0.06 sec)
```

На машине 192.168.20.204 выполним тест репликации и проверим выполнение:

```
mysql> USE test_db;
2, 'Server 204');
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> INSERT INTO replication_test VALUES (2, 'Server 204');
Query OK, 1 row affected (0.02 sec)
```

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test_db |
+-----+
5 rows in set (0.00 sec)

mysql> USE test_db;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_test_db |
+-----+
| replication_test |
+-----+
1 row in set (0.01 sec)

mysql> SELECT * FROM test_db.replication_test;
+----+-----+
| id | value |
+----+-----+
| 1 | Server 202 |
| 2 | Server 204 |
+----+-----+
2 rows in set (0.00 sec)
```

4. Проверить работу отказоустойчивого решения, симулируя отказ одного из серверов и убедившись, что второй сервер продолжает работу и все данные сохранены с обоими видами репликации (Зеркалирование подразумевает, что реплика автоматически подключится к главному ноду в случае, когда она станет вновь доступна)

Сделаем симуляцию отказа одной из машин:

Останавливаем службу MySQL на 192.168.20.202

```
sttewie@sttewie:~$ sudo systemctl stop mysql
```

На машине 192.168.20.204 добавим новую запись:

```
mysql> USE test_db;
3, 'Server 204 after 202 failure');
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> INSERT INTO replication_test VALUES (3, 'Server 204 after 202 failure');
Query OK, 1 row affected (0.02 sec)
```

```
sttewie@sttewie:~$ sudo systemctl start mysql
```

Проверим синхронизацию данных:

```
mysql> SELECT * FROM test_db.replication_test;
+----+-----+
| id | value |
+----+-----+
| 1  | Server 202 |
| 2  | Server 204 |
| 3  | Server 204 after 202 failure |
+----+-----+
3 rows in set (0.00 sec)
```

Аналогично, на 192.168.20.204

Остановим 204 и на 202 добавим новую запись:

```
mysql> USE test_db;
4, 'Server 202 after 204 failure');
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> INSERT INTO replication_test VALUES (4, 'Server 202 after 204 failure');
Query OK, 1 row affected (0.02 sec)
```

Перезапускаем 204 и проверяем синхронизацию:

```
sttewie@sttewie:~$ sudo systemctl start mysql
sttewie@sttewie:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.40-0ubuntu0.24.10.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

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> SELECT * FROM test_db.replication_test;
+----+-----+
| id | value |
+----+-----+
| 1  | Server 202 |
| 2  | Server 204 |
| 3  | Server 204 after 202 failure |
| 4  | Server 202 after 204 failure |
+----+-----+
4 rows in set (0.00 sec)
```

Опционально:

- Настроить систему резервного копирования, чтобы регулярно создавать бэкапы данных и сохранять их на отдельном сервере (либо на отдельном диске, либо папке) через SSH.
- Документировать все шаги по настройке и проверке отказоустойчивого решения и подготовить отчет о выполненной работе.