

Homework_Lesson18_Report

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

Задание 1:

1. Создать два сервера, установить на них PostgreSQL либо MySQL и подключить их к одной сети.
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)
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 главной ноде в случае, когда она станет вновь доступна)

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.

В прошлой домашней работе задании мы установили на сервере 192.168.1.210 и создали на нем базу cours. Установим на второй сервер 192.168.1.213 mysql.

```
av11@ubuntu24test:~$ sudo apt install mysql-server
```

На машине 192.168.1.210 Мастер добавляем параметры в конфиг sudo nano /etc/mysql/my.cnf

```
[mysqld]
server-id = 1
log_bin = /var/log/mysql/mysql-bin.log
binlog_do_db = cours
```

Перезапускаем mysql: sudo systemctl restart mysql

В конфиге sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf меняем параметр что бы mysql слушал порт в сети: bind-address = 127.0.0.1 на bind-address = 0.0.0.0

Создаем пользователя replic указываем что подключиться он может с 192.168.1.213 и устанавливаем пароль:

```
CREATE USER 'rep'@'192.168.1.213' IDENTIFIED BY 'my_password';
```

Устанавливаем права REPLICATION SLAVE ко всем базам пользователю и хосту 'replic'@'192.168.1.213':

```
GRANT REPLICATION SLAVE ON *.* TO 'rep'@'192.168.1.213';
```

Перезапустить изменения, связанные с пользователями и их правами:
FLUSH PRIVILEGES;

Получим координат бинарного лога для этого выполним команду SHOW MASTER STATUS; на мастере. Нам нужны значения File: mysql-bin.000001 и Position: 887.

```
mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000004 |      987 | cours        |                   |                   |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Переключаемся на машину 192.168.1.213 Слейв.

В конфиге добавляем параметры sudo nano /etc/mysql/my.cnf

```
[mysqld]
server-id = 2
log_bin = /var/log/mysql/mysql-bin.log
```

Перезапускаем mysql: sudo systemctl restart mysql

Заходим в mysql
Настраиваем репликацию:

```
CHANGE MASTER TO
MASTER_HOST='192.168.1.210',
MASTER_USER='rep',
MASTER_PASSWORD='my_password',
MASTER_LOG_FILE='mysql-bin.000004',    -- Укажите значение File
MASTER_LOG_POS=987;                   -- Укажите значение Position STATUS
```

Запустите репликацию:
START SLAVE;

Проверка репликации
SHOW SLAVE STATUS\G;

```

mysql> SHOW SLAVE STATUS\G;
***** 1. row *****
      Slave_IO_State: Waiting for source to send event
        Master_Host: 192.168.1.210
        Master_User: rep
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: mysql-bin.000004
        Read_Master_Log_Pos: 987
        Relay_Log_File: ubuntu24test-relay-bin.000003
        Relay_Log_Pos: 863
        Relay_Master_Log_File: mysql-bin.000004
        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:
          Last_Errno: 0
          Last_Error:
          Skip_Counter: 0
        Exec_Master_Log_Pos: 987
        Relay_Log_Space: 1249
        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: 0473f2d0-b155-11ef-9e40-00155d462e10
        Master_Info_File: mysql.slave_master_info
        SQL_Delay: 0
        SQL_Remaining_Delay: NULL
        Slave_SQL_Running_State: Replica 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:

```

Если база была создана до настройки репликации, она не была записана в бинарный лог, поэтому слейв её не получил.

На сервере-мастере (192.168.1.210), выполните команду для создания дампа базы:
`mysqldump -u root -p --databases cours --master-data=2 > cours_dump.sql`

Передайте файл дампа на слейв (например, с помощью `scp`):
`scp cours_dump.sql avl1@192.168.1.213:/home/avl1`

Останавливает репликацию.
`STOP SLAVE;`

Восстанавливаем базу.

```
mysql -u root -p < /home/avl1/cours_dump.sql
```

Приступим к проверке.

На мастере проверим содержание таблицы в базе cours.

```
mysql> SELECT * FROM `test_table`;  
+-----+  
| id | name |  
+-----+  
| 1 | Replication Test |  
+-----+  
1 row in set (0.00 sec)
```

Добавим ее папу строк в таблицу.

```
mysql> USE cours;  
Database changed  
mysql> INSERT INTO test_table VALUES (2, 'Replication Test2');  
Query OK, 1 row affected (0.01 sec)  
  
mysql> INSERT INTO test_table VALUES (3, 'Replication Test3');  
Query OK, 1 row affected (0.01 sec)
```

Проверим что данные добавлены на Мастере 192.168.1.210

```
avl@ubuntu-s24:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 25  
Server version: 8.0.40-0ubuntu0.24.04.1 (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_table`;  
ERROR 1046 (3D000): No database selected  
mysql> USE cours;  
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> SELECT * FROM `test_table`;  
+-----+  
| id | name |  
+-----+  
| 1 | Replication Test |  
| 2 | Replication Test2 |  
| 3 | Replication Test3 |  
+-----+  
3 rows in set (0.00 sec)  
  
mysql> █
```

Проверим данные на слейве 192.168.1.213.

```
av1@ubuntu24test:~$ sudo mysql
[sudo] password for av1:
Sorry, try again.
[sudo] password for av1:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 8.0.40-0ubuntu0.24.04.1 (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_table`;
ERROR 1046 (3D000): No database selected
mysql> USE cours;
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> SELECT * FROM `test_table`;
+----+-----+
| id | name          |
+----+-----+
|  1 | Replication Test |
|  2 | Replication Test2 |
|  3 | Replication Test3 |
+----+-----+
3 rows in set (0.00 sec)
```

Напишем скрипт для бекапа и запустим его. Наш бекап по rsync будет копироваться на другую машину. При необходимости можем добавить в конце что бы бекап удалялся с первой машины.

```
av1@ubuntu-s24:~$ skripts/backup_mysql
mysqldump: [Warning] Using a password on the command line interface can be insecure.
sending incremental file list
cours_backup_2024-12-04_13-26-11.sql

sent 1,726 bytes  received 35 bytes  704.40 bytes/sec
total size is 4,574  speedup is 2.60
av1@ubuntu-s24:~$ ls -l backup/
total 8
-rw-rw-r-- 1 av1 av1 4574 Dec  4 13:26 cours_backup_2024-12-04_13-26-11.sql
av1@ubuntu-s24:~$
```

Проверим на второй машине есть ли файл

```
av1@ubuntu24test:~$ ls -l backup/
total 8
-rw-rw-r-- 1 av1 av1 4574 Dec  4 13:26 cours_backup_2024-12-04_13-26-11.sql
av1@ubuntu24test:~$
```