

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4
по дисциплине
«РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ»

Выполнил: Билбрам Денис Андреевич
Группа: Р3319

Преподаватель: Шибаев Семён Сергеевич

Санкт-Петербург
2025

1 Выполнение

1.1 Этап 1: Настройка

1.1.1 Настройка основного узла (VM1) 192.168.1.11

Цель этого шага — подготовить основной узел для потоковой репликации, настроив параметры PostgreSQL и разрешив подключения от резервного узла.

- **Редактирование конфигурационного файла `postgresql.conf`:**

```
listen_addresses = '*'
wal_level = replica
```

`listen_addresses = '*'` позволяет принимать подключения со всех сетевых интерфейсов. `wal_level = replica` включает уровень журналирования, необходимый для репликации.

- **Настройка доступа в `pg_hba.conf`:** Отредактирован файл `/etc/postgresql/14/m`

- **Создание пользователя для репликации:** Создан пользователь `replicator` с правами на репликацию:

```
sudo -u postgres psql
CREATE ROLE replicator WITH REPLICATION PASSWORD 'password' LOGIN;
```

1.1.2 Настройка резервного узла (VM2) 192.168.1.10

Цель — настроить VM2 как резервный узел, синхронизированный с VM1 через потоковую репликацию.

- **Создание базовой копии с VM1:** Использована утилита `pg_basebackup` для копирования данных с основного узла:

```
sudo -u postgres pg_basebackup -h 192.168.1.11 -D /var/lib/postgresql
-U replicator -P -R --wal-method=stream
```

Опция `-R` автоматически создает файл `standby.signal` и добавляет строку `primary_conninfo` в `postgresql.auto.conf` для подключения к VM1.

- **Запуск PostgreSQL:** После создания копии служба PostgreSQL запущена:

```
sudo systemctl start postgresql
```

1.1.3 Демонстрация репликации

Для проверки репликации созданы таблицы, добавлены данные и проверена их синхронизация.

- **Создание таблиц и данных на VM1:** На основном узле созданы две таблицы и добавлены данные:

```
sudo -u postgres psql
CREATE TABLE test1 (id serial PRIMARY KEY, data text);
INSERT INTO test1 (data) VALUES ('hello');
CREATE TABLE test2 (id serial PRIMARY KEY, name varchar(50));
INSERT INTO test2 (name) VALUES ('Alice'), ('Bob');
INSERT INTO test1 (data) VALUES ('transaction');
```

- **Проверка данных на VM2:** На резервном узле проверено наличие данных:

```
sudo -u postgres psql
SELECT * FROM test1;
SELECT * FROM test2;
```

Вывод для test1:

```
   id |      data
-----+-----
    1 | hello
    2 | transaction
(2 rows)
```

Для test2:

```
   id | name
-----+-----
    1 | Alice
    2 | Bob
(2 rows)
```

Попытка записи на VM2:

```
INSERT INTO test1 (data) VALUES ('world');
```

ERROR: cannot execute INSERT in a read-only transaction.

- **Проверка статуса репликации:** - На VM1:

```
sudo -u postgres psql -c "SELECT * FROM pg_stat_replication;"
```

Вывод показывает подключение VM2:

```
   pid | username | application_name | client_addr | state | sync_
-----+-----+-----+-----+-----+-----
  1234 | replicator | walreceiver      | 192.168.1.10 | streaming |
(1 row)
```

- На VM2:

```
sudo -u postgres psql -c "SELECT pg_is_in_recovery();" 
```

Вывод: t (true), указывающий на режим восстановления.

1.2 Этап 2: Симуляция и обработка сбоя

1.2.1 Симуляция сбоя

Цель — симулировать недоступность основного узла.

- **Отключение сети на VM1:** Выполнена команда для отключения сетевого интерфейса:

```
sudo ip link set enp0s3 down
```

Проверено отсутствие сетевой связности:

```
ping 192.168.1.10
```

ping: connect: Network is unreachable. Клиентские подключения завершились ошибкой:

```
psql: error: could not connect to server: Connection refused
```

1.2.2 Обработка сбоя

Цель — переключить резервный узел в режим основного и продолжить операции.

- **Переключение на VM2:** Выполнено повышение VM2 до основного узла:

```
sudo -u postgres /usr/lib/postgresql/14/bin/pg_ctl  
promote -D /var/lib/postgresql/14/main
```

Вывод:

```
pg_ctl: server promoted
```

- **Подключение клиентов к VM2:** Подключение к новому основному узлу:

```
psql -h 192.168.1.10 -U postgres -d postgres
```

- **Проверка логов:** Просмотрены логи на VM2:

```
tail /var/log/postgresql/postgresql-14-main.log
```

Сообщения:

```
FATAL: could not connect to the primary server: connection to server
port 5432 failed: No route to host
Is the server running on that host and accepting TCP/IP connections?
LOG: received promote request
FATAL: terminating walreceiver process due to administrator command
LOG: redo done at 0/302C110 system usage: CPU: user: 0.00 s, system:
LOG: last completed transaction was at log time 2025-06-08 22:36:13.
LOG: selected new timeline ID: 2
LOG: archive recovery complete
LOG: database system is ready to accept connections
```

1.3 Этап 3: Восстановление

1.3.1 Восстановление сети на VM1

Цель — вернуть сетевую связность основного узла.

- **Включение сетевого интерфейса:** Выполнена команда:

```
sudo ip link set enp0s3 up
```

Проверена связность:

```
ping 192.168.1.10
```

результат: успешный отклик.

1.3.2 Восстановление VM1

Цель — синхронизировать VM1 с VM2.

- **Остановка PostgreSQL:**

```
sudo systemctl stop postgresql
```

- **Синхронизация с помощью pg_rewind:** Выполнена команда для синхронизации данных:

```
sudo -u postgres pg_rewind -D /var/lib/postgresql/14/main
—source-server='host=192.168.1.11 port=5432
user=replicator password=password'
```

Вывод:

```
pg_rewind: connected to server
pg_rewind: source and target cluster are on the same timeline
pg_rewind: rewinding from last common checkpoint at 0/30006B8
pg_rewind: done
```

- **Запуск PostgreSQL:**

```
sudo systemctl start postgresql
```

1.3.3 Восстановление исходной конфигурации

Цель — вернуть VM1 в роль основного узла, а VM2 — в роль резервного.

- **Остановка PostgreSQL на VM2:**

```
sudo systemctl stop postgresql
```

- **Повышение VM1 до основного узла:**

```
sudo -u postgres pg_ctl promote -D /var/lib/postgresql/14/main
```

Вывод:

```
pg_ctl: server promoted
```

- **Настройка VM2 как резервного узла:** - Удаление содержимого каталога данных:

```
sudo -u postgres rm -rf /var/lib/postgresql/14/main/*
```

- Создание новой базовой копии:

```
sudo -u postgres pg_basebackup -h 192.168.1.11 -D /var/lib/postgresql/14/main
16384/16384 kB (100%), 1/1 tablespace
```

- Запуск PostgreSQL:

```
sudo systemctl start postgresql
```

1.3.4 Демонстрация восстановленной конфигурации

- **Операции на VM1:** Выполнены операции записи:

```
sudo -u postgres psql
INSERT INTO test1 (data) VALUES ('back to original');
INSERT INTO test2 (name) VALUES ('Charlie');
```

- **Проверка данных на VM2:**

```
sudo -u postgres psql
SELECT * FROM test1;
SELECT * FROM test2;
```

Вывод:

```
id | data
---+-----
 1 | hello
 2 | transaction
 3 | back to original
(3 rows)
```

```
id | name
----+-----
 1 | Alice
 2 | Bob
 3 | Charlie
(3 rows)
```

Попытка записи:

```
INSERT INTO test1 (data) VALUES ('should fail');
```

ERROR: cannot execute INSERT in a read-only transaction.

2 Заключение

Лабораторная работа позволила освоить настройку отказоустойчивого решения на базе PostgreSQL, включая конфигурацию репликации, обработку сбоев и восстановление системы. Полученные навыки включают настройку параметров PostgreSQL, создание базовых копий, симуляцию сбоев и синхронизацию данных. Эти знания применимы для обеспечения высокой доступности баз данных в реальных системах.