

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
Высшего образования  
*Факультет Программной Инженерии и Компьютерной Техники*

**Лабораторная работа 3 по РСХД**

Вариант 51486

Группа: Р3316

Выполнили:

Сиразетдинов, Шпинаева

Проверил:

Николаев В.В.

г. Санкт-Петербург

2025

# **Задание**

## **Цель работы**

Цель работы - настроить процедуру периодического резервного копирования базы данных, сконфигурированной в ходе выполнения лабораторной работы №2, а также разработать и отладить сценарии восстановления в случае сбоев.

Узел из предыдущей лабораторной работы используется в качестве основного. Новый узел используется в качестве резервного. Учётные данные для подключения к новому узлу выдаёт преподаватель. В сценариях восстановления необходимо использовать копию данных, полученную на первом этапе данной лабораторной работы.

## **Этап 1. Резервное копирование**

- Настроить резервное копирование с основного узла на резервный следующим образом:

Периодические полные копии + непрерывное архивирование. Включить для СУБД режим архивирования WAL; настроить копирование WAL () на резервный узел; настроить полное резервное копирование (pg\_basebackup) по расписанию (cron) раз в неделю. Созданные полные копии должны сразу копироваться (scp) на резервный хост. Срок хранения копий на основной системе - 1 неделя, на резервной - 4 недели. По истечении срока хранения, старые архивы и неактуальные WAL должны автоматически уничтожаться.

- Подсчитать, каков будет объем резервных копий спустя месяц работы системы, исходя из следующих условий:

Средний объем новых данных в БД за сутки: 350МБ. Средний объем измененных данных за сутки: 800МБ.

- Проанализировать результаты.

## **Этап 2. Потеря основного узла**

Этот сценарий подразумевает полную недоступность основного узла. Необходимо восстановить работу СУБД на РЕЗЕРВНОМ узле, продемонстрировать успешный запуск СУБД и доступность данных.

### **Этап 3. Повреждение файлов БД**

Этот сценарий подразумевает потерю данных (например, в результате сбоя диска или файловой системы) при сохранении доступности основного узла. Необходимо выполнить полное восстановление данных из резервной копии и перезапустить СУБД на ОСНОВНОМ узле.

Ход работы:

- Симулировать сбой:
  - удалить с диска директорию WAL со всем содержимым.
- Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.
- Выполнить восстановление данных из резервной копии, учитывая следующее условие:
  - исходное расположение дополнительных табличных пространств недоступно - разместить в другой директории и скорректировать конфигурацию.
- Запустить СУБД, проверить работу и доступность данных, проанализировать результаты.

### **Этап 4. Логическое повреждение данных**

Этот сценарий подразумевает частичную потерю данных (в результате нежелательной или ошибочной операции) при сохранении доступности основного узла. Необходимо выполнить восстановление данных на ОСНОВНОМ узле следующим способом:

- Генерация файла на резервном узле с помощью pg\_dump и последующее применение файла на основном узле.

Ход работы:

- В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.
- Зафиксировать время и симулировать ошибку:
  - в любой таблице с внешними ключами подменить значения ключей на случайные (INSERT, UPDATE)
- Продемонстрировать результат.
- Выполнить восстановление данных указанным способом.
- Продемонстрировать и проанализировать результат.

# Выполнение

## Этап 1. Резервное копирование

На основном узле зададим параметры для архивации WAL файлов и их копирования на резервный узел:

```
wal_level = replica
archive_mode = on
archive_command = 'scp %p postgres1@pg114:/var/db/postgres1/
wal_archive/%f'
```

Подключимся как суперпользователь

```
psql -h localhost -p 9455 -U postgres0 -d postgres
```

Создадим роль replicator с правами на репликацию:

```
CREATE ROLE replicator WITH REPLICATION LOGIN PASSWORD
'pass123';
```

Напишем скрипт для создания копии:

```
#!/bin/bash
# Директория для хранения резервных копий на основном узле
LOCAL_BACKUP_DIR="/var/db/postgres0/backups/$(date +%Y-%m-%d_%H-%M-%S)"
# Директория для хранения резервных копий на резервном узле
REMOTE_BACKUP_DIR="/var/db/postgres1/backups/$(date +%Y-%m-%d_%H-%M-%S)"
# Создание директории на основном узле
mkdir -p $LOCAL_BACKUP_DIR
# Создание директории на резервном узле
ssh pg1 "mkdir -p $REMOTE_BACKUP_DIR"

# Выполнение резервного копирования на основном узле
pg_basebackup -D $LOCAL_BACKUP_DIR -Ft -z -Xs -P -U replicator -h
pg109 -p 9455
# Проверка успешности выполнения
if [ $? -eq 0 ]; then
    echo "Резервное копирование успешно завершено на основном узле:
$LOCAL_BACKUP_DIR"
    # Копирование резервной копии на резервный узел
    scp -r $LOCAL_BACKUP_DIR/* pg1:$REMOTE_BACKUP_DIR/
    # Проверка успешности копирования
    if [ $? -eq 0 ]; then
        echo "Резервная копия успешно перенесена на резервный узел:"
```

```
$REMOTE_BACKUP_DIR"
```

```
# Удаление старых резервных копий на основном узле (старше 1 недели)
```

```
find /var/db/postgres0/backups -type d -mtime +7 -exec rm -rf {} \;
```

```
echo "Старые резервные копии удалены на основном узле"
```

```
# Удаление старых резервных копий на резервном узле (старше 4 недель)
```

```
ssh pg1 "find /var/db/postgres1/backups -type d -mtime +28 -exec rm -rf {} \;"
```

```
echo "Старые резервные копии удалены на резервном узле"
```

```
else
```

```
echo "Ошибка при переносе резервной копии на резервный узел"
```

```
exit 1
```

```
fi
```

```
else
```

```
echo "Ошибка при выполнении резервного копирования на основном узле"
```

```
exit 1
```

```
fi
```

Добавим права на выполнение

```
chmod +x /var/db/postgres0/script1.sh
```

Настроим выполнение скрипта с помощью кроны раз в неделю

```
crontab -e
```

```
0 3 * * 1 /var/db/postgres0/script1.sh
```

```
[crontab: installing new crontab
```

## Подсчет объемов и анализ результатов

### Этап 2. Потеря основного узла

Для восстановления воссоздадим файловую структуру кластера и табличного пространства на резервном узле из нашего бэкапа:

```
mkdir -p /var/db/postgres1/u08/djs10
```

```
cp -r ~/backups/2025-04-06_20-13-07/* /var/db/postgres1/u08/djs10
```

```
chmod 750 /var/db/postgres1/u08/djs10
```

```
# Распакуем резервную копию:
```

```
cd ~/u08/djs10
```

```
tar -xzf base.tar.gz
```

```
tar -xzf pg_wal.tar.gz
```

```
tar -xzf 16387.tar.gz
```

```

tar -xzf 16388.tar.gz
# Укажем в postgresql.conf команду для загрузки wal-файлов:
echo "restore_command = 'cp /var/db/postgres1/u08/djs10/%f %p'"
>> /var/db/postgres1/u08/djs10/postgresql.conf
# Создадим в директории кластера файл, сигнализирующий о
восстановлении:
touch ~/u08/djs10/recovery.signal
# Запустим резервный кластер:
pg_ctl -D ~/u08/djs10 start
# Проверим статус:
pg_ctl -D ~/u08/djs10 status

```

```

[[postgres1@pg114 ~/u08/djs10]$ ls
16387.tar.gz  16388.tar.gz  backup_manifest base.tar.gz  pg_wal.tar.gz
[[postgres1@pg114 ~/u08/djs10]$ tar -xzf base.tar.gz
tar -xzf pg_wal.tar.gz
tar -xzf 16387.tar.gz
tar -xzf 16388.tar.gz
# Укажем в postgresql.conf команду для загрузки wal-файлов:
echo "restore_command = 'cp /var/db/postgres1/u08/djs10/%f %p'" >> /var/db/postgres1/u08/djs10/postgresql.conf
# Создадим в директории кластера файл, сигнализирующий о восстановлении:
touch ~/u08/djs10/recovery.signal
# Запустим резервный кластер:
pg_ctl -D ~/u08/djs10 start
# Проверим статус:
pg_ctl -D ~/u08/djs10 status
ожидание запуска сервера...2025-04-06 20:20:26.699 MSK [68331] СООБЩЕНИЕ: передача вывода в протокол
процессу сбора протоколов
2025-04-06 20:20:26.699 MSK [68331] ПОДСКАЗКА: В дальнейшем протоколы будут выводиться в каталог "log".
... готово
сервер запущен
pg_ctl: сервер работает (PID: 68331)
/usr/local/bin/postgres "-D" "/var/db/postgres1/u08/djs10"
[[postgres1@pg114 ~/u08/djs10]$

```

Проверим таблицы в восстановленной бд:

```

psql -h pg109 -p 9455 -U new_user -d leftbrownmom
\d

```

```

[[postgres1@pg114 ~/u08/djs10]$ psql -p 9455 -U new_user -d leftbrownmom -h pg114
Пароль пользователя new_user:
psql (16.4)
Введите "help", чтобы получить справку.

leftbrownmom=> \d

```

Схема	Имя	Тип	Владелец
public	test_table1	таблица	new_user
public	test_table1_id_seq	последовательность	new_user
public	test_table2	таблица	new_user
public	test_table2_id_seq	последовательность	new_user

(4 строки)

## Этап 3. Повреждение файлов БД

Удалим директорию с WAL файлами и запустим сервер

```
ервано: последний момент работы: 2025-04-06 21:08:18 MSK",,,,,,,,"", "startup",,0
2025-04-06 21:08:27.763 MSK,,,76274,,67f2c31b.129f2,2,,2025-04-06 21:08:27 MSK,,0,ВАЖНО,XX000,"требуемый каталог WAL ""pg_wal""
не существует",,,,,,,,"", "startup",,0
2025-04-06 21:08:27.765 MSK,,,76270,,67f2c31b.129ee,6,,2025-04-06 21:08:27 MSK,,0,СООБЩЕНИЕ,00000,"стартовый процесс (PID 76274)
завершился с кодом выхода 1",,,,,,,,"", "postmaster",,0
2025-04-06 21:08:27.765 MSK,,,76270,,67f2c31b.129ee,7,,2025-04-06 21:08:27 MSK,,0,СООБЩЕНИЕ,00000,"прерывание запуска из-за ошиб
ки в стартовом процессе",,,,,,,,"", "postmaster",,0
2025-04-06 21:08:27.766 MSK,,,76270,,67f2c31b.129ee,8,,2025-04-06 21:08:27 MSK,,0,СООБЩЕНИЕ,00000,"система БД выключена",,,,,,,
,"", "postmaster",,0
```

Восстановим данные в новую директорию

```
mkdir -p ~/u08_new/djs10
chmod 750 ~/u08_new/djs10
cd ~/u08_new/djs10
# Распакуем резервную копию:
tar -xzf ~/backups/2025-04-06_19-29-44/base.tar.gz -C ~/u08_new/
djs10
mkdir -p ~/u08_new/djs10/pg_wal
tar -xzf ~/backups/2025-04-06_19-29-44/pg_wal.tar.gz -C ~/u08_new/
djs10/pg_wal
mkdir -p ~/cje38_new
tar -xzf ~/backups/2025-04-06_19-29-44/16387.tar.gz -C ~/cje38_new/
mkdir -p ~/qdx64_new
tar -xzf ~/backups/2025-04-06_19-29-44/16388.tar.gz -C ~/qdx64_new/
# Создадим в директории кластера файл, сигнализирующий о
восстановлении:
touch ~/u08_new/djs10/recovery.signal
```

Укажем в postgresql.conf команду для загрузки wal файлов:

```
echo "restore_command = 'cp /home/postgres0/u08_new/djs10/pg_wal/%f
%p'" > postgresql.conf
```

Запустим восстановленный кластер: pg\_ctl -D /u08\_new/djs10 start

```
[postgres0@pg109 ~/u08_new/djs10]$ pg_ctl -D /var/db/postgres0/u08_new/djs10 -l logfile start
ожидание запуска сервера.... готово
сервер запущен
```

Подключимся к бд и посмотрим таблицы:

```
psql -h pg109 -p 9455 -U new_user -d leftbrownmom
\d
```

```
[postgres0@pg109 ~/u08_new/djs10]$ pg_ctl -D /var/db/postgres0/u08_new/djs10 -l logfile start
ожидание запуска сервера.... готово
сервер запущен
```

## Этап 4. Логическое повреждение данных

Восстановим и запустим кластер

```
pg_resetwal -f ~/u08/dsj10  
pg_ctl start
```

Включим архивирование

```
archive_mode = on  
archive_command = 'scp %p pgl:/var/db/postgres1/wal_archive/%f'  
archive_timeout = 60  
restore_command = 'cp /var/db/postgres1/u08/djs10/pg_wal/%f %p'
```

Добавим строки в таблицы

```
insert into test_table1 (data) values ('Test data 666');
```

```
leftbrownmom=> insert into test_table1 (data) values ('Test data 666');  
INSERT 0 1  
leftbrownmom=> select * from test_table1;  
 id |      data  
----+-----  
  1 | Test data 1  
  2 | Test data 2  
  3 | Test data 1  
  4 | Test data 2  
  5 | Test data 1  
  6 | Test data 2  
  7 | Test data 1  
  8 | Test data 2  
  9 | Test data 1  
 10 | Test data 2  
 11 | Test data 666  
(11 строк)  
  
leftbrownmom=> █
```

## Вывод