МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Лабораторная работа №3

по дисциплине

«Распределенные системы хранения данных»

Вариант 98523

Выполнил:
Векшин Арсений Иванович P3316
Преподаватель:
Николаев Владимир Вячеславович

Санкт-Петербург

~ 2025 ~

Задание

Данные для подключения

Основной узел

Виртуальная машина: pg107
Пользователь: postgres2
Пароль: /fE/sM38

Резервный узел

Виртуальная машина: pg113
Пользователь: postgres3
Пароль: nhW9fEGX

Лабораторная работа №3

Введите вариант: 98523

Внимание! У разных вариантов разный текст задания!

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.

Требования к отчёту

Отчет должен быть самостоятельным документом (без ссылок на внешние ресурсы), содержать всю последовательность команд и исходный код скриптов по каждому пункту задания. Для демонстрации результатов приводить команду вместе с выводом (самой наглядной частью вывода, при необходимости).

Этап 1. Резервное копирование

- Настроить резервное копирование с основного узла на резервный следующим образом:
  Периодические полные копии с помощью SQL Dump.
  По расписанию (cron) раз в сутки, методом SQL Dump с сжатием. Созданные архивы должны сразу перемещаться на резервный хост, они не должны храниться на основной системе. Срок хранения архивов на резервной системе - 4 недели. По истечении срока хранения, старые архивы 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: `600МБ`.
  - Средний объем измененных данных за сутки: `700МБ`.
- 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 любой таблицы со всем содержимым.
- Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.
- Выполнить восстановление данных из резервной копии, учитывая следующее условие:
  - исходное расположение директории PGDATA недоступно - разместить данные 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:

- Восстановление с использованием архивных WAL файлов. (СУБД должна работать в режиме архивирования WAL, потребуется задать параметры восстановления).

Ход работы:

- В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.
- Зафиксировать время и симулировать ошибку:
  - удалить каждую вторую строку в любой таблице (DELETE)
- Продемонстрировать результат.
- Выполнить восстановление данных указанным способом.
- Продемонстрировать и проанализировать результат.

Этап 1. Резервное копирование

Задание

Настроить резервное копирование с основного узла на резервный следующим образом:

- Периодические полные копии с помощью SQL Dump.
- По расписанию (cron) раз в сутки, методом SQL Dump с сжатием. Созданные архивы должны сразу перемещаться на резервный хост, они не должны храниться на основной системе. Срок хранения архивов на резервной системе - 4 недели. По истечении срока хранения, старые архивы 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: 600МБ.
- Средний объем измененных данных за сутки: 700МБ.

Проанализировать результаты.

Настройка резервного копирования с основного узла на резервный

1. Создадим суперпользователя для создания SQL dump и добавим его в конфигурационные файлы кластера.

```
psql -p 9296 -d postgres -c "create role backup_user with login password
'strongpass' superuser"
echo "localhost:9296:*:backup_user:strongpass" >> ~/.pgpass
chmod 600 ~/.pgpass
echo "local all backup_user scram-sha-256" >> $HOME/cbz5/pg_hba.conf
```

2. Напишем скрипт для резервного копирования, который будет запускаться при cron по четкому расписанию:

```
#var/db/postgres2/dump_script.sh
# Новая переменная окружения директории с дампами на основном узле
export LOCAL_DUMP_DIR=$HOME/dumps/postgres

# Новая переменная окружения директории с дампами на резервном узле
export REMOTE_DUMP_DIR=/var/db/dumps/postgres

# Создаём директорию для хранения дампов на основном узле
mkdir -p "$LOCAL_DUMP_DIR"

# Создаём директорию для хранения дампов на резервном узле
ssh postgres3@pg113 "mkdir -p $REMOTE_DUMP_DIR"

# Выполнение логического резервного копирования для всех баз данных на сервере с помощью
pg_dumpall
pg_dumpall -p 9296 -U postgres2 --exclude-database=template1 | gzip >
"$LOCAL_DUMP_DIR/dump_$(date +%Y-%m-%d).sql.gz"

# Проверка, что резервное копирование прошло успешно
if [ $? -eq 0 ]; then
        echo "Резервное копирование успешно завершено на основном узле: $LOCAL_DUMP_DIR"

        # Копирование только что созданных дампов на резервный узел
        scp -r "$LOCAL_DUMP_DIR"/* postgres3@pg113:"$REMOTE_DUMP_DIR/"

        if [ $? -eq 0 ]; then
        echo "Резервная копия успешно перенесена на резервный узел: $REMOTE_DUMP_DIR"

        # Удаление дампов после копирования на резервный узел
        find "$LOCAL_DUMP_DIR"/* -delete

        # Удаление дампов с резервного узла по истечении 4 недель
        ssh postgres3@pg113 "find $REMOTE_DUMP_DIR -type f -name '*.gz' -mtime +28 -exec
rm -f {} \;"

        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
```

Установим данному скрипту права на выполнение

```
$ chmod +x dump_script.sh
```

Через утилиту cron установим выполнение скрипта каждый день в 22:07

```
$ crontab -e
7 22 * * * /var/db/postgres2/dump_script.sh
```

```
crontab -e
```

```
[postgres2@pg107 ~]$ crontab -l
35 23 * * * /var/db/postgres2/dump_script.sh
```

Подсчет объема системных копий спустя месяц работы системы

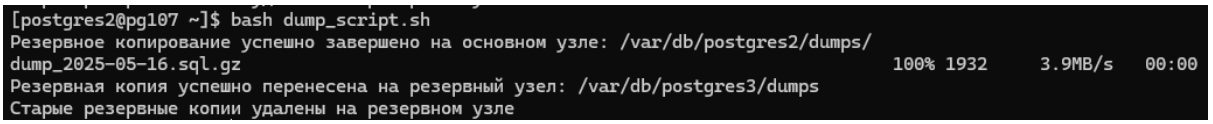
Резервная копия создается раз в день, дольше 4 недель на резервном узле они не хранятся. Следовательно, максимальный срок хранения = 28 дней.

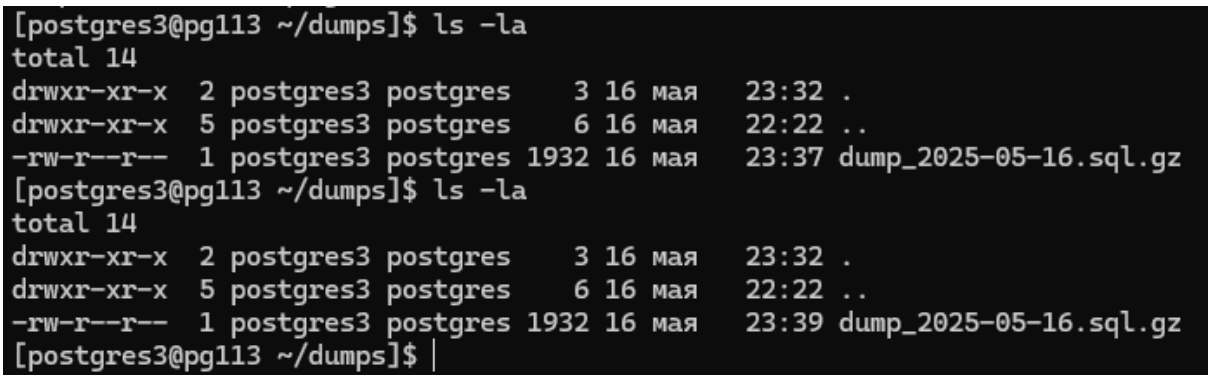
Допустим, за первый день работы посчитаем измененные данные также, как новые. Тогда размеры SQL дампа за первые n дней:
1300, 1300 + 600, 1300 + 600 · 2, … , 1300 + 600 · 28, 1300 + 600 · 29.

Вычтем два первых дня, так как они уже успеют удалиться, итого получим:
1300 · 28 + 600 · (2 + 3 + ··· + 28 + 29) = 1300 · 28 + 600 · 434 = 36400 + 260400 ≈ 296.8 Гб несжатых данных.

Если применять обычное gzip сжатие, то объем уменьшится еще в примерно 2 раза. Таким образом, объем резервных копий пропорционален росту БД, а не количеству изменений. Изменения строк не дублируются, а просто затирают старые.

Логи





Этап 2. Потеря основного узла

Задание

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

Выполнение

Представим, что основной узел поврежден и теперь перейдём на резервный узел и попробуем восстановить БД.

Для этого остановим кластер на основном узле

```
pg_ctl -D $HOME/cbz5/ stop
```

Скрипт для автоматического восстановления кластера на резервном узле:

```
# recovery_rezerv_stg2.sh
export PGDATA=$HOME/cbz5
export TBDATA=$HOME/rez5
export PGPORT=9296


mkdir -p $PGDATA
mkdir -p $TBDATA
initdb -D $PGDATA


pg_ctl -D $PGDATA start

# Создаём  табличное пространство
psql -U postgres2 -p $PGPORT -d postgres -c "CREATE TABLESPACE rez5 LOCATION
'/var/db/postgres2/rez5';"
# Разархивируем дамп, чтобы восстановить кластер по нему.
gunzip -c "$(ls -t dumps/postgres/dump_*.sql.gz | head -n1)" | psql -U
postgres2 -d postgres -p $PGPORT

# Проверка, что сервер отвечает и всё хорошо
pg_ctl -D $PGDATA status
```

Логи

```
[postgres2@pg107 ~]$ pg_ctl -D $HOME/cbz5/ stop
ожидание завершения работы сервера.... готово
сервер остановлен
```

```
[postgres3@pg113 ~]$ bash recovery_rezerv_stg2.sh
Файлы, относящиеся к этой СУБД, будут принадлежать пользователю "postgres3".
От его имени также будет запускаться процесс сервера.

Кластер баз данных будет инициализирован с локалью "ru_RU.UTF-8".
Кодировка БД по умолчанию, выбранная в соответствии с настройками: "UTF8".
Выбрана конфигурация текстового поиска по умолчанию "russian".

Контроль целостности страниц данных отключён.

исправление прав для существующего каталога /var/db/postgres3/cbz55... ок
создание подкаталогов... ок
выбирается реализация динамической разделяемой памяти... posix
выбирается значение max_connections по умолчанию... 100
выбирается значение shared_buffers по умолчанию... 128MB
выбирается часовой пояс по умолчанию... Europe/Moscow
создание конфигурационных файлов... ок
выполняется подготовительный скрипт... ок
выполняется заключительная инициализация... ок
сохранение данных на диске... ок

initdb: предупреждение: включение метода аутентификации "trust" для локальных подключений
initdb: подсказка: Другой метод можно выбрать, отредактировав pg_hba.conf или ещё раз запустив initdb с ключом -A, --auth-local или --auth-host.

Готово. Теперь вы можете запустить сервер баз данных:

    pg_ctl -D /var/db/postgres3/cbz55 -l файл_журнала start

ожидание запуска сервера....2025-05-16 23:53:47.546 MSK [94070] СООБЩЕНИЕ:  завершение вывода в stderr
2025-05-16 23:53:47.546 MSK [94070] ПОДСКАЗКА:  В дальнейшем протокол будет выводиться в "syslog".
 готово
сервер запущен
```

```
CREATE TABLESPACE
SET
SET
SET
CREATE ROLE
ALTER ROLE
CREATE ROLE
ALTER ROLE
CREATE ROLE
ALTER ROLE
:    "dpk35"
ALTER ROLE
:   "/var/db/postgres2/dpk35"
:   "/var/db/postgres2/meow"
SET
SET
SET
SET
SET
 set_config
------------

(1 строка)

SET
SET
SET
SET
:    "dpk35"
:    "dpk35"
\connect: подключиться к серверу через сокет "/tmp/.s.PGSQL.9296" не удалось: ВАЖНО:  база данных "dpk35" не существует
pg_ctl: сервер работает (PID: 94070)
/usr/local/bin/postgres "-D" "/var/db/postgres3/cbz55"
```

Этап 3. Повреждение файлов БД

Задание

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 любой таблицы со всем содержимым.
- Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.
- Выполнить восстановление данных из резервной копии, учитывая следующее условие:
  - исходное расположение директории PGDATA недоступно - разместить данные в другой директории и скорректировать конфигурацию.
- Запустить СУБД, проверить работу и доступность данных, проанализировать результаты.

Выполнение

Симулируем сбой (основной узел)

```
# simulate_table_file_loss.sh
export PGDATA=$HOME/cbz5
BASE_DIR="$PGDATA/base"

# Найти первый файл таблицы
FILE_TO_DELETE=$(find "$BASE_DIR" -type f | head -n 1)

# Проверка
if [[ -z "$FILE_TO_DELETE" ]]; then
    echo "Файл таблицы не найден!"
    exit 1
fi
rm -f "$FILE_TO_DELETE"

TABLE_FILE_NAME="$FILE_TO_DELETE"
echo "Удалён файл таблицы: $TABLE_FILE_NAME"
# Проверка состояния сервера
pg_ctl -D $PGDATA status
```

Этап 4. Логическое повреждение данных

Задание

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:

- Восстановление с использованием архивных WAL файлов. (СУБД должна работать в режиме архивирования WAL, потребуется задать параметры восстановления).

Ход работы:

- В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.
- Зафиксировать время и симулировать ошибку:
  - удалить каждую вторую строку в любой таблице (DELETE)
- Продемонстрировать результат.
- Выполнить восстановление данных указанным способом.
- Продемонстрировать и проанализировать результат.
-

Выполнение

Внесем изменения в postgres.conf основного узла:

```
archive_mode = on
archive_command = 'scp %p pg113:/var/db/postgres3/wal_archive/%f'
wal_level = replica
archive_timeout = 60
```

Создадим копию кластера с помощью команды, так как восстановление при помощи wal-архивирования работает только с физической копией:

```
pg_basebackup -p 9296 -D $HOME/backup -Ft -Xs -P
```

Состояние test_table3 на момент бэкапа:

```
postgres=# select * from test_table3;
 id |   name
----+----------
  1 | Alice1
  2 | Bob1
  3 | Charlie1
(3 строки)
```

Добавим новые данные в таблицу и зафиксируем изменения

```
postgres=# select pg_switch_wal();
 pg_switch_wal
---------------
 0/30007A0
(1 строка)

postgres=# select now();
              now
-------------------------------
 2025-05-18 18:44:40.145363+03
(1 строка)

postgres=# select * from test_table3;
 id |   name
----+----------
  1 | Alice1
  2 | Bob1
  3 | Charlie1
  4 | Alexy
  5 | Maria
  6 | Ivan
  7 | Olga
  8 | Dmitry
  9 | Ekaterina
 10 | Sergey
 11 | Ann
 12 | Pavel
 13 | Natalya
(13 строк)
```

команды:

```
INSERT INTO test_table3 (name) VALUES ('Alexy'), ('Maria'), ('Ivan'),
('Olga'), ('Dmitry'), ('Ekaterina'), ('Sergey'), ('Ann'), ('Pavel'),
('Natalya');

select pg_switch_wal();
select now();

select * from test_table3;
```

Теперь удалим каждую вторую строку из таблицы test_table3 как симуляция ошибки

```
DO $$
DECLARE
      table_name TEXT := 'test_table3';
      column_name TEXT := 'id';
BEGIN
```

```
      EXECUTE format('
      DELETE FROM %I
      WHERE %I IN (
            SELECT %I
            FROM (
                  SELECT %I, ROW_NUMBER() OVER (ORDER BY %I) as rn
                  FROM %I
            ) t
            WHERE rn %% 2 = 0
      )',
      table_name, column_name, column_name, column_name, column_name,
table_name);
END $$;
```

Новое состояние таблицы:

```
postgres=# select * from test_table3;
 id |    name
----+-----------
  1 | Alice1
  3 | Charlie1
  5 | Maria
  7 | Olga
  9 | Ekaterina
 11 | Ann
 13 | Natalya
(7 строк)
```

Востановим:

```
pg_ctl -D $HOME/cbz5 stop
cp $HOME/cbz5/pg_wal/* $HOME/wal_dir/
scp wal_dir/* pg113:~/wal_archive/
rm-rf cbz5/
rm-rf rez5/
mkdir pg_data_recovery
mkdir rez5
chmod 750 pg_data_recovery/
tar -xvf $HOME/backup/base.tar -C $HOME/pg_data_recovery
tar -xvf $HOME/backup/pg_wal.tar -C $HOME/pg_data_recovery/pg_wal
TABLESPACE_ARCHIVE=$(ls $HOME/backup | grep -E '^[0-9]+\.tar$' | head -n
1)
tar -xvf $HOME/backup/$TABLESPACE_ARCHIVE -C $HOME/rez5
cp wal_dir/* pg_data_recovery/pg_wal/
echo "
restore_command = 'scp pg113:~/wal_archive/%f %p'
recovery_target_time = ''
recovery_target_action = promote " >>
$HOME/pg_data_recovery/postgres.conf
```

```
touch recovery.signal
pg_ctl -D $HOME/pg_data_recovery start
```

После восстановления таблица выглядит так:

Код программы

https://github.com/ArsenyVekshin/ITMO/tree/master/RSHD/lab3

Вывод