# Федеральное государственное автономное образовательное

учреждение высшего образования

Университет ИТМО

# Дисциплина: Распределённые системы хранения данных **Лабораторная работа 3**

Вариант 36589, pg111, postgres4, o05CgTYI

**Выполнил:**

# Митичев Иван Дмитриевич

**Группа:** P3316

**Преподаватель:**

# Николаев Владимир Вячеславович

2025 г.

Санкт-Петербург

# Ссылка на гитхаб со всеми скриптами/конфигами

[Karabas890/RSHDLab3](https://github.com/Karabas890/RSHDLab3)

# 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:

Первоначальная полная копия + непрерывное архивирование.  
Включить для СУБД режим архивирования WAL; настроить копирование WAL (scp) на резервный узел; создать первоначальную резервную копию (pg\_basebackup), скопировать на резервный узел (rsync).

* Подсчитать, каков будет объем резервных копий спустя месяц работы системы, исходя из следующих условий:
  + Средний объем новых данных в БД за сутки: 100МБ.
  + Средний объем измененных данных за сутки: 550МБ.
* 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 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:

* 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.

# Подключение

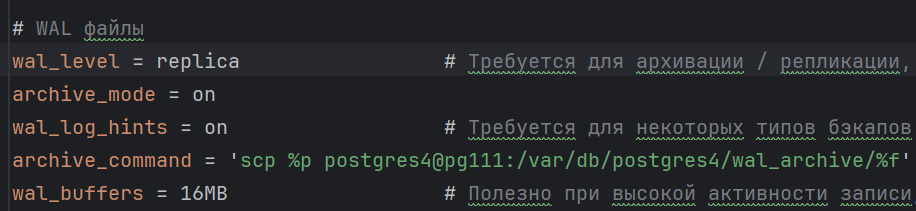
Способ подключения к узлу из сети Интернет через helios:  
ssh -J s368527@helios.cs.ifmo.ru:2222 postgres2@pg121  
Способ подключения к узлу из сети факультета:

ssh -p 2222 [s368527@se.ifmo.ru](mailto:s368527@se.ifmo.ru)  
ssh postgres2@pg121

# Выполнение

## Этап 1. Резервное копирование

Включение архивирования WAL:



Напишем скрпит для создания копии основного узла на резервном сервере:

**#!/bin/bash**set -euo pipefail  
  
# Пути  
BACKUP\_DIR="/var/db/postgres2/backups"  
REMOTE\_SERVER="postgres4@pg111"  
REMOTE\_BACKUP\_DIR="/var/db/postgres4/backups"  
  
# Создаём каталог локально  
mkdir -p "$BACKUP\_DIR"  
ssh "$REMOTE\_SERVER" "mkdir -p '$REMOTE\_BACKUP\_DIR'"  
  
# Уникальное имя  
BACKUP\_NAME="backup\_$(date +%F\_%H-%M-%S)"  
  
echo "Making base backup..."  
pg\_basebackup -D "$BACKUP\_DIR/$BACKUP\_NAME" \  
 -Ft -z -P -X stream \  
 -U postgres2 -h 127.0.0.1 -p 9426  
  
echo "Copying base backup on reserve server..."  
rsync -avz "$BACKUP\_DIR/$BACKUP\_NAME/" "$REMOTE\_SERVER:$REMOTE\_BACKUP\_DIR/"  
  
echo "✅ End of reserve copy (WAL archiving with archive\_command)."

**Расчеты:**

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

* + Средний объем новых данных в БД за сутки: 100МБ.
  + Средний объем измененных данных за сутки: 550МБ.

Изначальный размер данных – 100МБ. Каждый день добавляется еще 100МБ новых данных + 550МБ измененных = 650МБ. По формуле арифметической прогрессии вычислим объем резервных копий через 30 дней:

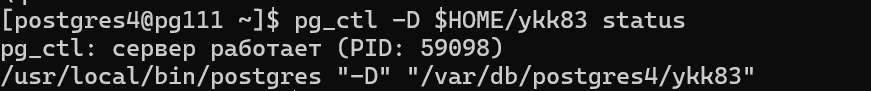
## Этап 2. Потеря основного узла

Восстановим файловую структуру кластера по бэкапам:

# 1. Остановим PostgreSQL (если вдруг запущен)  
pg\_ctl -D /var/db/postgres4/ykk83 stop  
  
# 2. Очистим старый кластер  
rm -rf /var/db/postgres4/ykk83/  
rm -rf /var/db/postgres4/twv39/  
mkdir -p /var/db/postgres4/ykk83  
chmod 750 /var/db/postgres4/ykk83  
  
# 3. Развернем базовый кластер  
tar -xzf /var/db/postgres4/backups/base.tar.gz -C /var/db/postgres4/ykk83/  
  
# 4. Развернем WAL  
mkdir -p /var/db/postgres4/ykk83/pg\_wal  
tar -xzf /var/db/postgres4/backups/pg\_wal.tar.gz -C /var/db/postgres4/ykk83/pg\_wal/  
  
# 5. Развернем tablespace (если был)  
mkdir -p /var/db/postgres4/twv39/  
tar -xzf /var/db/postgres4/backups/16388.tar.gz -C /var/db/postgres4/twv39/  
ln -s /var/db/postgres4/twv39 /var/db/postgres4/ykk83/pg\_tblspc/16388  
  
# 6. Настроим восстановление  
echo "restore\_command = 'cp /var/db/postgres4/wal\_archive/%f %p'" >> /var/db/postgres4/ykk83/postgresql.conf  
  
# 7. Создадим recovery.signal  
touch /var/db/postgres4/ykk83/recovery.signal  
  
# 8. Запустим PostgreSQL  
pg\_ctl -D /var/db/postgres4/ykk83 start  
# 8. Проверим работу  
psql -h localhost -p 9426 -U new\_role -d postgres

Проверим что все запустилось:

pg\_ctl -D $HOME/ykk83 status



Таблицы тоже на месте:

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Содержимое, созданное искусственным интеллектом, может быть неверным.

## Этап 3. Повреждение файлов БД

# Остановим сервер (чтобы имитировать некорректное завершение работы)  
pg\_ctl -D $PGDATA stop -m immediate  
# Удалим директорию WAL (qfg95 отдельный каталог)  
rm -rf $HOME/qfg95  
# Пробуем перезапустить сервер  
pg\_ctl -D $PGDATA start

Перезапускаем сервер:

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Содержимое, созданное искусственным интеллектом, может быть неверным.

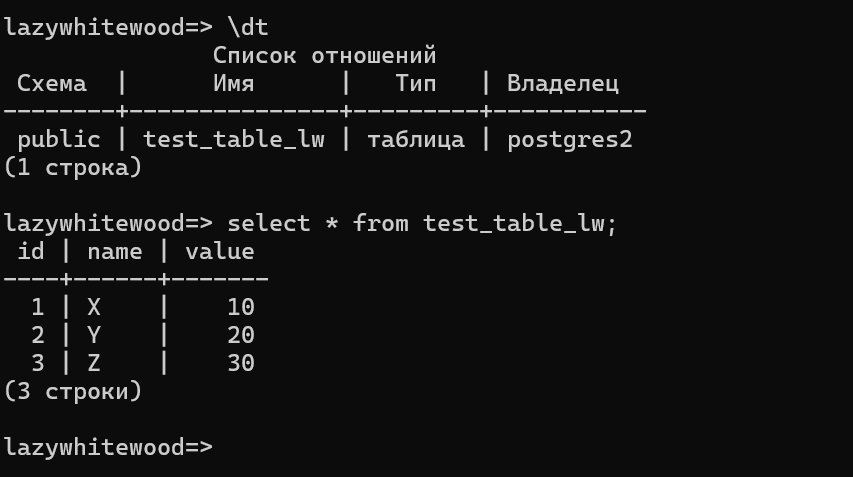
# Проверим корректность работы  
pg\_ctl -D $PGDATA status  
tail -n 30 $PGDATA/logfile  
# Остановим сервер (если вдруг он висит)  
pg\_ctl -D $PGDATA stop -m immediate  
  
# Создадим новые каталоги для восстановления  
rm -rf $HOME/ykk83\_recovered $HOME/qfg95\_recovered $HOME/twv39\_recovered  
mkdir -p $HOME/ykk83\_recovered  
mkdir -p $HOME/qfg95\_recovered  
mkdir -p $HOME/twv39\_recovered  
chmod 700 $HOME/ykk83\_recovered $HOME/qfg95\_recovered $HOME/twv39\_recovered  
# Скопируем последнюю резервную копию с резервного узла  
# Скопировать base.tar.gz  
scp postgres4@pg111:/var/db/postgres4/backups/base.tar.gz $HOME/  
  
# Скопировать pg\_wal.tar.gz  
scp postgres4@pg111:/var/db/postgres4/backups/pg\_wal.tar.gz $HOME/  
  
# Скопировать tablespace (16388.tar.gz)  
scp postgres4@pg111:/var/db/postgres4/backups/16388.tar.gz $HOME/  
  
# Развернём базовый кластер  
tar -xzf $HOME/base.tar.gz -C $HOME/ykk83\_recovered  
  
# Развернём WAL  
mkdir -p $HOME/ykk83\_recovered/pg\_wal  
tar -xzf $HOME/pg\_wal.tar.gz -C $HOME/ykk83\_recovered/pg\_wal/  
  
# Развернём tablespace  
tar -xzf $HOME/16388.tar.gz -C $HOME/twv39\_recovered  
ln -s $HOME/twv39\_recovered $HOME/ykk83\_recovered/pg\_tblspc/16388  
  
# Настроим конфигурацию  
echo "restore\_command = 'cp /var/db/postgres2/wal\_archive/%f %p'" >> $HOME/ykk83\_recovered/postgresql.conf  
#echo "port = 9426" >> $HOME/ykk83\_recovered/postgresql.conf  
#echo "listen\_addresses = '\*'" >> $HOME/ykk83\_recovered/postgresql.conf  
  
#Создаём recovery.signal  
touch $HOME/ykk83\_recovered/recovery.signal  
  
#Запускаем сервер из новой директории  
pg\_ctl -D $HOME/ykk83\_recovered -l $HOME/ykk83\_recovered/logfile start  
  
#Проверяем доступность  
psql -p 9426 -U postgres2 -d postgres

Проверим целостность данных:

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

## Этап 4. Логическое повреждение данных

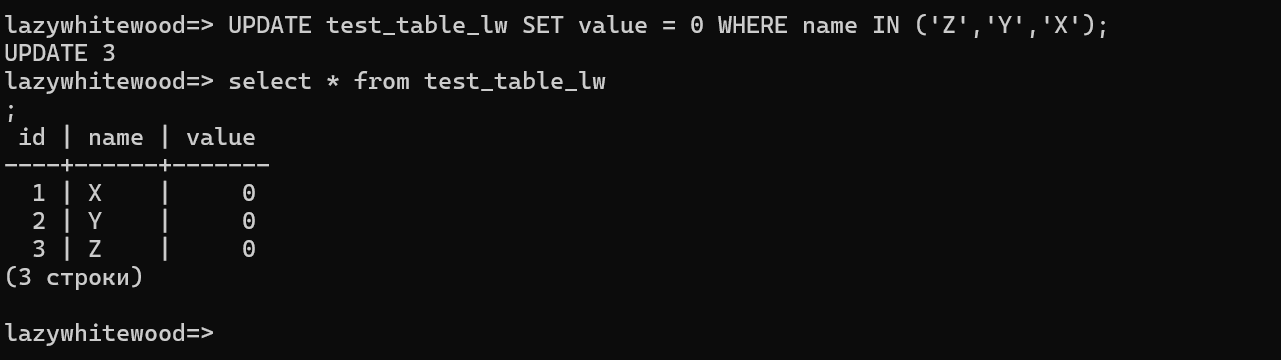
Исходные данные в таблицах:

# Добавляем новые данные:

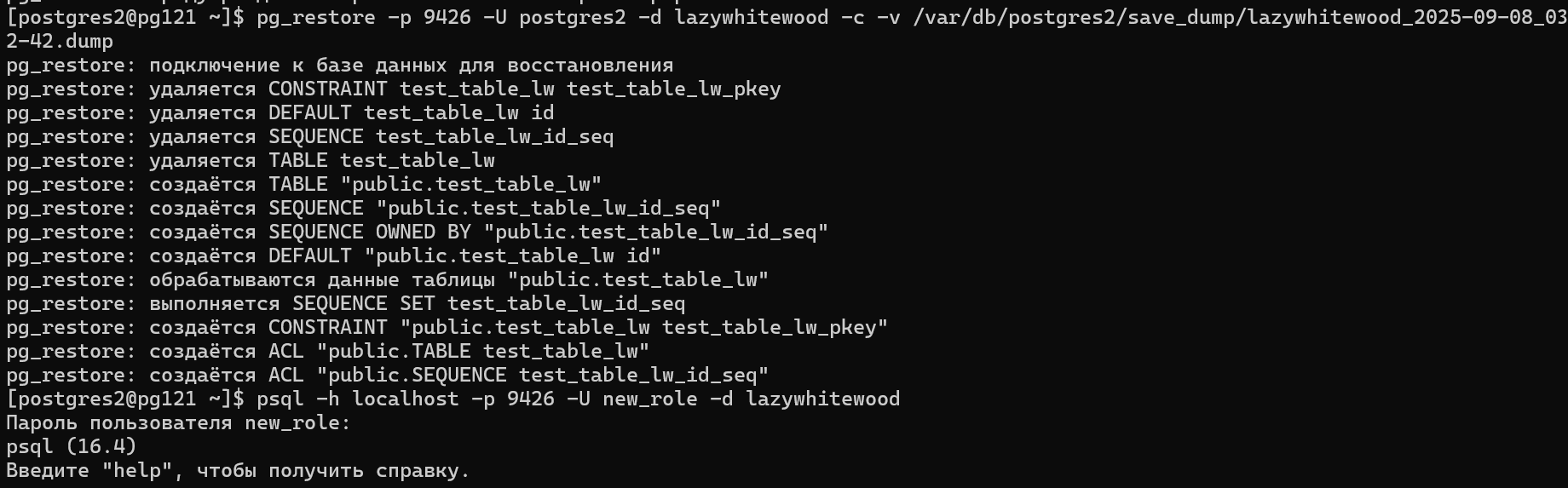
Изображение выглядит как текст, снимок экрана, Шрифт

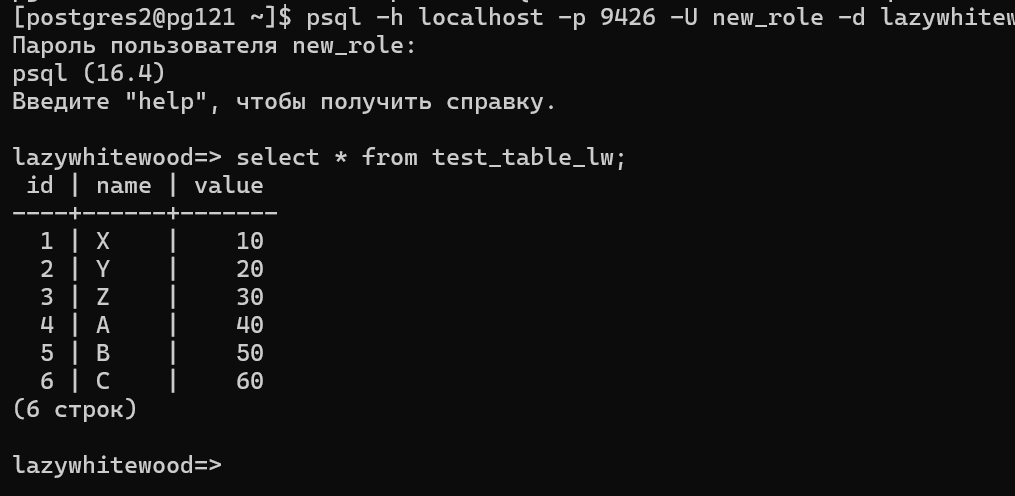
Содержимое, созданное искусственным интеллектом, может быть неверным.

Пример «логической ошибки» — перезаписываем строки



Восстанавливаем данные из дампа



Подключаемся к базе для проверки

# Вывод

В ходе лабораторной была выполнена настройка резервного копирования базы данных, сконфигурированной в ходе выполнения лабораторной работы №2, а также

разработаны сценарии восстановления в случае сбоев. Во всех случаях удалось успешно восстановить работоспособность системы.