МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

Лабораторная работа №3

по дисциплине

«Распределённые системы хранения данных»

Вариант - **333586**

Выполнил: студент группы P33212

Бондарь Б.М

Преподаватель: Шешуков Д.М.

Санкт-Петербург

2024 г.

**Задание:**

Лабораторная работа включает настройку резервного копирования данных с основного узла на резервный, а также несколько сценариев восстановления. Узел из предыдущей лабораторной работы используется в качестве основного; новый узел используется в качестве резервного. В сценариях восстановления необходимо использовать копию данных, полученную на первом этапе данной лабораторной работы.

**Этапы работы:**

Этап 1. Резервное копирование

Настроить резервное копирование с основного узла на резервный следующим образом:

Периодические обособленные (standalone) полные копии.

Полное резервное копирование (pg\_basebackup) по расписанию (cron) два раза в сутки. Необходимые файлы WAL должны быть в составе полной копии, отдельно их не архивировать. Срок хранения копий на основной системе - 1 неделя, на резервной - 1 месяц. 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 измененных данных за сутки: 350МБ.

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.

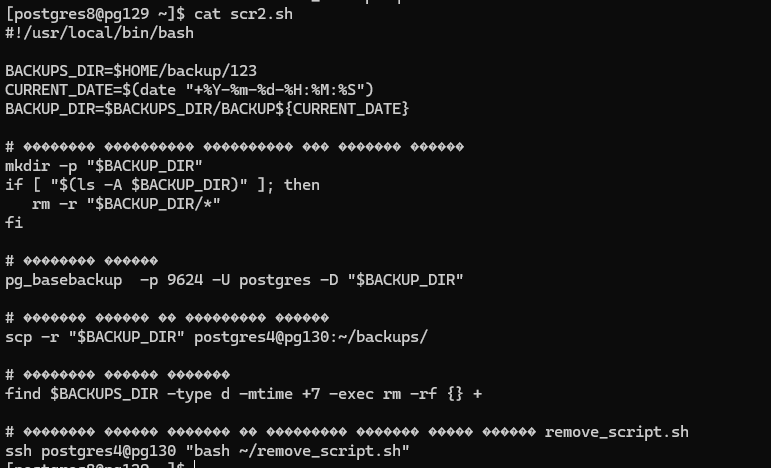
**Продемонстрировать и проанализировать результат.**

**Выполнение:**

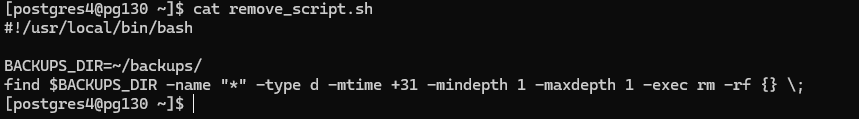
* **Резервное копирование**

Создание РК будет производиться следующим скриптом, который в свою очередь будет запускаться каждый день в 11 часов дня и вечера по расписанию:

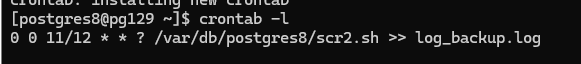
Основной узел:



Резервный узел:



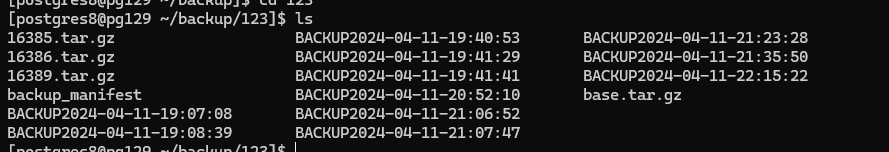
На основном узле создадим cron-файл через команду (crontab -e), в котором опишем правило для запуска нашего скрипта два раза в сутки (в 11:00 и 23:00). Проверим список запланированных задач.



Проверим работу скрипта

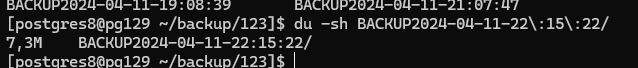


Резервная копия создалась:



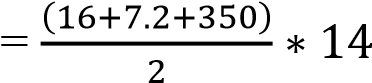
**Расчет объема:**

Размер одного бэкапа (изначально):

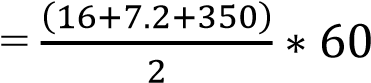


Одновременно на основном узле могут храниться копии, время жизни которых меньше недели. В таком случае перед самой «чисткой» будут копии за 7 дней, которые делались 2 раза в день. Также нужно учитывать, что создаются новые wal файлы по 16 МБ каждый.

С помощью арифметической прогрессии вычислим объем на основном узле: 𝑎1 = 7.2 МБ 𝑑 = 350 МБ 𝑛 = 7 \*2 = 14

𝑆 = 2612 МБ = **2.6 ГБ**

Для резервного узла: 𝑛 = 30 \* 2 = 60

𝑆 = 11196 МБ = **11.2 ГБ**

* **Потеря основного узла**

Для демонстрации создания резервной копии будет использована база данных postgres

-- Создаем таблицу

CREATE TABLE users (

user\_id SERIAL PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

created\_on TIMESTAMP WITH TIME ZONE DEFAULT CURRENT\_TIMESTAMP

)

-- Наполняем таблицу данными

INSERT INTO users (username, email) VALUES

('johndoe', 'johndoe@example.com'),

('janedoe', 'janedoe@example.com'),

('mikeb', 'mikeb@example.com');

-- Создаем таблицу

CREATE TABLE products (

product\_id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

price DECIMAL(10, 2) NOT NULL,

in\_stock BOOLEAN DEFAULT TRUE

)

-- Наполняем таблицу данными

INSERT INTO products (name, price, in\_stock) VALUES

('Laptop', 999.99, TRUE),

('Smartphone', 499.49, FALSE),

('Tablet', 250.00, TRUE);

-- Создаем таблицу

CREATE TABLE orders (

order\_id SERIAL PRIMARY KEY,

user\_id INT REFERENCES users(user\_id),

product\_id INT REFERENCES products(product\_id),

quantity INT NOT NULL,

order\_date TIMESTAMP WITH TIME ZONE DEFAULT CURRENT\_TIMESTAMP

) T

-- Наполняем таблицу данными

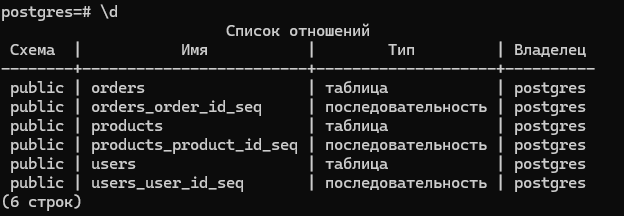
INSERT INTO orders (user\_id, product\_id, quantity) VALUES

(1, 1, 1),

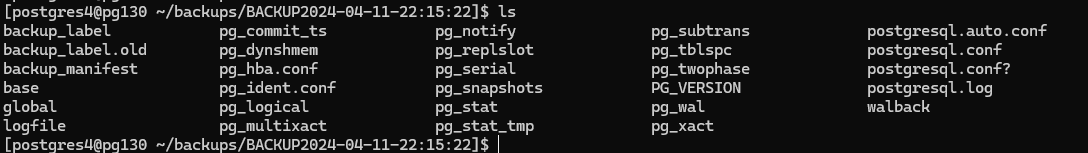
(2, 2, 2),

(3, 3, 1);

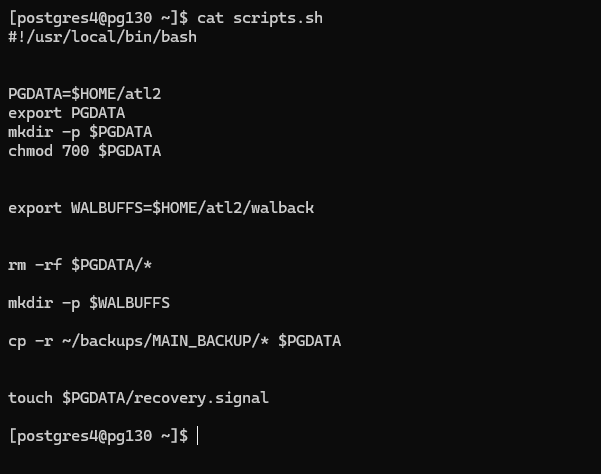
В итоге будут созданы следующие таблицы:



После создания резервной копии на запасном узле появится директория со следующим содержимым:



Чтобы восстановить работу СУБД на резервном узле для начала нужно выполнить следующий скрипт:



предварительно выделив наиболее актуальную копию (MAIN\_BACKUP)

После запуска скрипта добавились 2 директории для $PGDATA и WAL’ов, куда их потом перенесёт кто-то при восстановлении.

Также содержимое основной копии подменило содержимое $PGDATA. Помимо этого, сами скопированные WAL’ы также поместились в $PGDATA в стандартную поддиректорию pg\_wal

Дополнительно к копированию в «рабочем» пространстве заранее был создан файл recovery.signal, который является флагом на восстановление работы СУБД из РК.

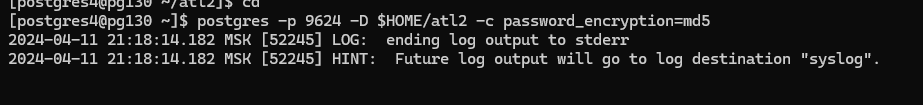
После этого осталось указать директорию для WAL’ов в конфигурационном файле для того, чтобы они туда скопировались:

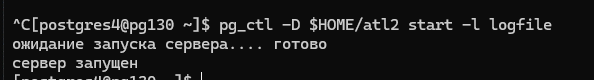
restore\_command = 'cp $PGDATA/pg\_wal/<> $WALBUFFS/' указать название файла

После подстановки в postgresql.conf будет доступна запись

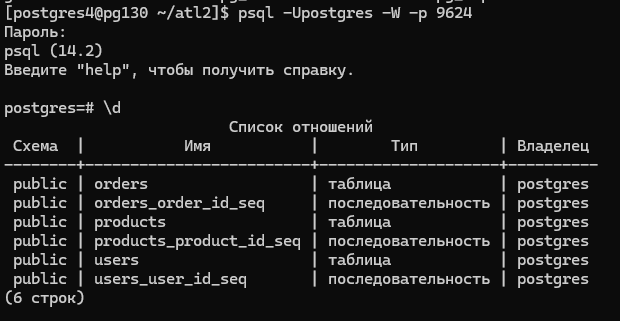


Далее после запуска postgres данные и WAL’ы «восстановятся», а файл recovery.signal исчезнет





Теперь можно запустить СУБД на резервном узле и проверить данные:

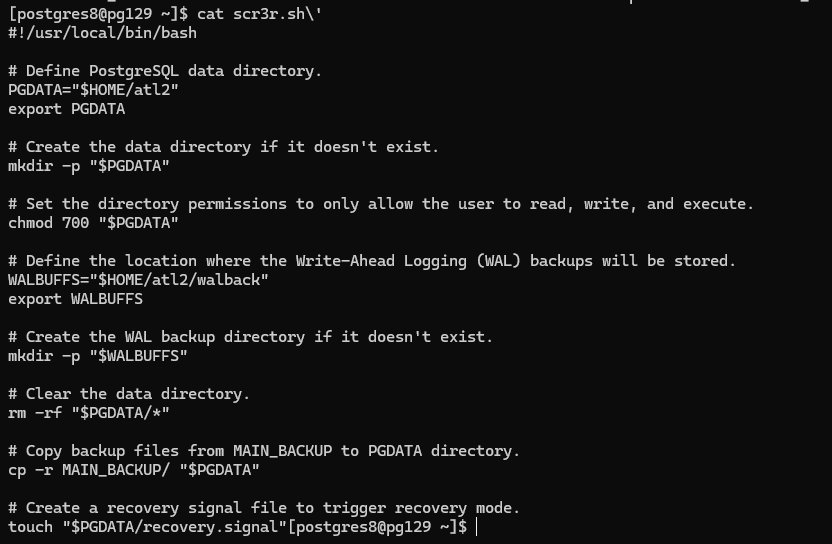


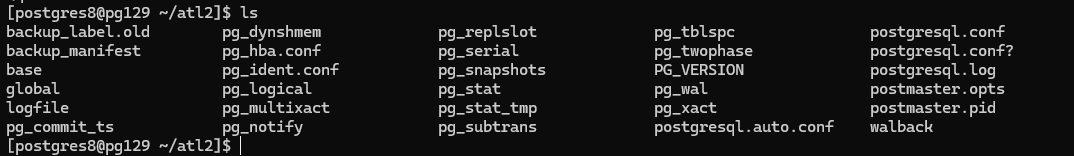
* **Повреждение файлов БД**

При запуске копии на основном узле при сбое алгоритм действий примерно такой же за исключение того, что в PGDATA и другие метапеременные прописываются новые значения. Т. е. нужно только перенести копию с резервного узла на основной, а потом сделать всё так же, как и в п. 2 уже там. Удалим папку atl2 на основной машине.

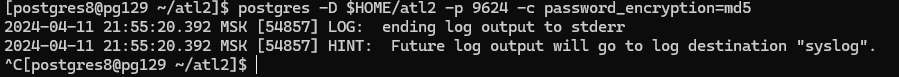
Переносим копию на основной узел и немного меняем скрипт для

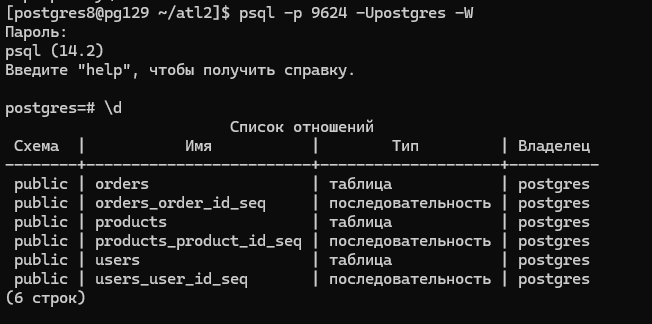






Далее перезапускаем сервер:

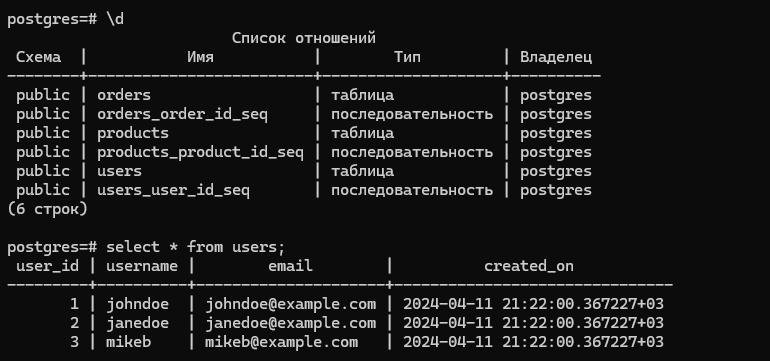




Работает!

* **Логическое повреждение данных**

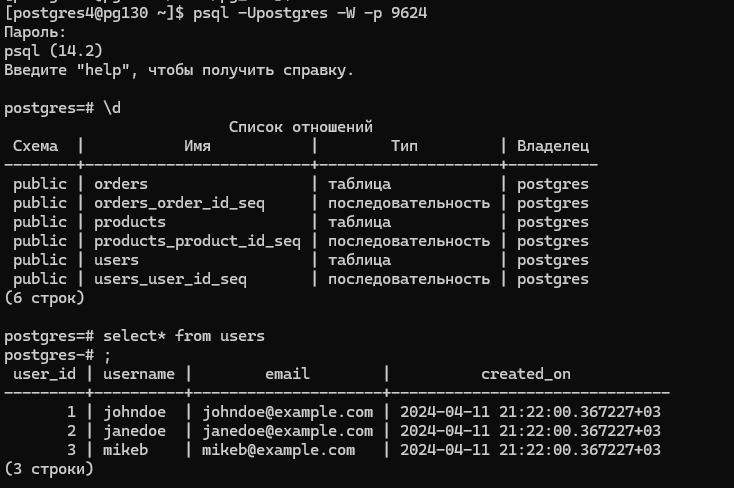
Сначала БД будет наполнена новыми данными, с которыми потом будут производиться неаккуратные действия.



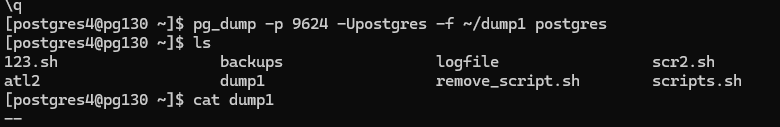
После этого сделаем РК и отправим её на резервный узел

Далее после запуска на резервном узле появятся следующие данные и та самая пока не

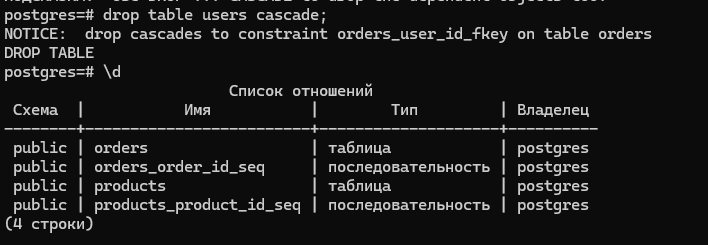
«испорченная» БД (данные) на которой будет сделан дамп



Далее делается сам дамп БД postgres, которая будет испорчена на основном узле



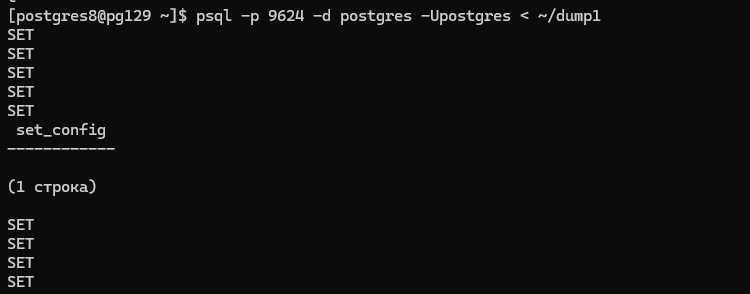
Удаляем таблицу:



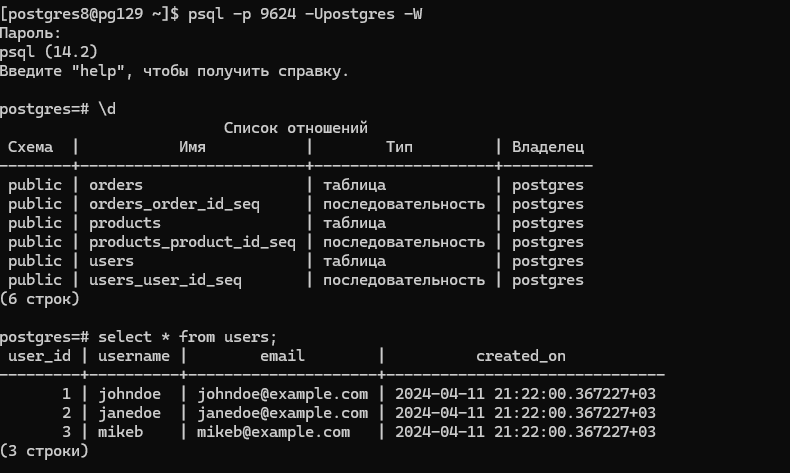
Теперь нужно скопировать дамп с резервного узла и применить его на основном:



Далее восстанавливаем данные из файла в текстовом формате



Видно, что таблица опять появилась с теми же данными



**Выводы**

В результате выполнения работы были изучены различные способы создания резервных копий. Помимо этого, было сделано несколько резервных копий с помощью физического и логического резервного копирования, после чего было осуществлено восстановление данных и работы СУБД, используя эти резервные копии