ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

Факультет Программной Инженерии и Компьютерной Техники

Дисциплина:
«*Распределённые системы хранения данных*»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Выполнили:
Студенты гр. P33151
Соловьев Артемий Александрович
Понамарев Степан Андреевич

Проверил:
Перцев Тимофей Сергеевич

Санкт-Петербург
2024г.

Задание

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 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 новых данных в БД за сутки: 550МБ.
  - Средний объем измененных данных за сутки: 750МБ.
- 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 конфигурационных файлов СУБД 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.

Для подключения:

1) Для подключения к helios:
   ssh s334645@se.ifmo.ru
2) Для подключения к основному узлу:
   ssh postgres1@pg156
3) Для подключения к резервному узлу:
   ssh postgres0@pg191

Этап 1. Резервное копирование

Настройка резервного копирования

Создаем пользователя с привилегией REPLICATION

```
CREATE USER backup_user WITH REPLICATION;
```

Создание первоначальной копии

```
pg_dump -h localhost -d firstdb -p 9143 -U backup_user -O -Fc >
"$HOME/backup/firstdb_$(date +\%Y\%m\%d_\%H\%M).dump"
```

Перенос копии на резервный узел

```
rsync -avzP $HOME/backup/* postgres0@pg191:~/backup/
```

```
firstdb_20240510_0052.dump
          3.350 100%    2,53MB/s    0:00:00 (xfr#1, to-chk=2/3)
firstdb_20240510_0103.dump
              0 100%    0,00kB/s    0:00:00 (xfr#2, to-chk=1/3)
firstdb/

sent 229 bytes  received 92 bytes  642,00 bytes/sec
total size is 3.350  speedup is 10,44
```

Подсчет размера копий

Размер начального бэкапа: 7,4 Мб

```
[postgres0@pg191 ~]$ du -sh backup/
7,4M    backup/
```

При полном резервном копировании общий объем всех копий будет равен:

$$\frac{2 \cdot 7{,}4 + (30 - 1) \cdot 550}{2} \cdot 30 = 239{,}554 \text{ Гб}$$

При инкрементном копировании через месяц объем всех копий будет равен:

$$7{,}4 + 750 \cdot 30 = 22{,}517 \text{ Гб}$$

Очевидно, инкрементальные бекапы намного эффективнее.

Этап 2. Потеря основного узла

Восстанавливаем базу данных из копии

```
createdb -h localhost -p 9143 seconddb
pg_restore -h localhost -p 9143 -d seconddb -x -c $HOME/backup/firstdb.dump
psql -h localhost -p 9143 -d seconddb
```

```
[postgres0@pg191 ~/lab3]$ pg_restore -h localhost -p 9143 -d seconddb -x -c $HOME/backup/firstdb_20240509_2018.dump
pg_restore: при обработке оглавления:
pg_restore: из записи оглавления 3527; 2606 16393 CONSTRAINT orders orders_pkey postgres1
pg_restore: ошибка: could not execute query: ОШИБКА:  отношение "public.orders" не существует
Выполнялась команда: ALTER TABLE ONLY public.orders DROP CONSTRAINT orders_pkey;
pg_restore: из записи оглавления 3525; 2604 16389 DEFAULT orders id postgres1
pg_restore: ошибка: could not execute query: ОШИБКА:  отношение "public.orders" не существует
Выполнялась команда: ALTER TABLE public.orders ALTER COLUMN id DROP DEFAULT;
pg_restore: из записи оглавления 209; 1259 16385 SEQUENCE orders_id_seq postgres1
pg_restore: ошибка: could not execute query: ОШИБКА:  последовательность "orders_id_seq" не существует
Выполнялась команда: DROP SEQUENCE public.orders_id_seq;
pg_restore: из записи оглавления 210; 1259 16386 TABLE orders postgres1
pg_restore: ошибка: could not execute query: ОШИБКА:  таблица "orders" не существует
Выполнялась команда: DROP TABLE public.orders;
pg_restore: ошибка: could not execute query: ОШИБКА:  роль "postgres1" не существует
Выполнялась команда: ALTER TABLE public.orders OWNER TO postgres1;

pg_restore: из записи оглавления 209; 1259 16385 SEQUENCE orders_id_seq postgres1
pg_restore: ошибка: could not execute query: ОШИБКА:  роль "postgres1" не существует
Выполнялась команда: ALTER TABLE public.orders_id_seq OWNER TO postgres1;

pg_restore: предупреждение: при восстановлении проигнорировано ошибок: 6
```

Проверяем, что все восстановилось

```
seconddb=# SELECT * FROM orders;
 id | text | number
----+------+--------
  1 | 1    |    123
  2 | 2    |    123
  3 | 3    |    123
  4 | 4    |    123
  5 | 5    |    123
(5 строк)
```

```
seconddb=# \dt
           Список отношений
 Схема  |  Имя   |   Тип   | Владелец
--------+--------+---------+-----------
 public | orders | таблица | postgres0
(1 строка)
```

Анализ выполнения

Восстановление завершилось успешно.

Этап 3. Повреждение фалов базы данных.

Последняя актуальная копия находится на узле, скачивать еще раз нет смысла.

Очищаем директорию для wal-файлов

```
rm -rf ~/u03/gwt12/pg_wal/*
```

Подключаемся к базе, которая была в пространстве

```
psql -p 9022 -d postgres
```

```
firstdb=# select * from orders;
 id | text | number
----+------+--------
  1 | 1    |    123
  2 | 2    |    123
  3 | 3    |    123
  4 | 4    |    123
  5 | 5    |    123
(5 строк)

firstdb=# insert into orders values (6,6,123);
ПАНИКА:  не удалось создать файл "pg_wal/xlogtemp.78346": ??? ?????? ????? ??? ????????
сервер неожиданно закрыл соединение
        Скорее всего сервер прекратил работу из-за сбоя
        до или в процессе выполнения запроса.
Подключение к серверу потеряно. Попытка восстановления неудачна.
!?>
```

Видим, что сервер аварийно завершился и не смог восстановиться, при повторной попытке подключиться возникает ошибка:

```
[postgres1@pg156 ~/pg_database1]$ psql -h localhost -p 9143 -d firstdb
psql: ошибка: подключиться к серверу "localhost" (::1), порту 9143 не удалось: Connection refused
        Сервер действительно работает по данному адресу и принимает TCP-соединения?
подключиться к серверу "localhost" (127.0.0.1), порту 9143 не удалось: Connection refused
        Сервер действительно работает по данному адресу и принимает TCP-соединения?
```

Из-за того, что старое пространство недоступно, сделаем новое в другом месте ☺

```
export BACKUPDIR=$HOME/backup_dir
rm -rf BACKUPDIR
mkdir BACKUPDIR
tar -xvf $HOME/backup/step3/base.tar -C BACKUPDIR
```

После распаковки бэкапа, нужно восстановить wal.

```
mkdir $PGDATA/pg_wal/
cp -r BACKUPDIR/pg_wal/* $PGDATA/pg_wal/
pg_ctl start
```

Проверяем доступность данных:

```
СТРОКА 1: selecet * from orders;
          ^
firstdb=# select * from orders;
 id | text | number
----+------+--------
  1 | 1    |    123
  2 | 2    |    123
  3 | 3    |    123
  4 | 4    |    123
  5 | 5    |    123
(5 строк)
```

Анализ

Восстановление было проведено успешно.

Этап 4. Логическое повреждение данных.

Делаем «нежелательные» изменения.

```
firstdb=# select * from orders;
 id | text | number
----+------+--------
  1 | 1    |    123
  2 | 2    |    123
  3 | 3    |    123
  4 | 4    |    123
  5 | 5    |    123
(5 строк)

firstdb=# insert into orders values (6, 6, 123);
 firstdb=# insert into orders values (7,6,123);
 INSERT 0 1
```

Восстановление

Переносим нужную резервную копию на основной узел

```
scp  $HOME/backup/firstdb_20240510_0052.dump lab2:~/backup/
```

Восстанавливаем базу данных из резервной копии

```
pg_restore -h localhost -p 9143 -d firstdb -x -c $HOME/backup/firstdb_20240510_0052.dump
psql -h localhost -p 9143 -d firstdb
```

Проверка данных.

```
[postgres1@pg156 ~/backup]$ pg_restore -h localhost -p 9143 -d firstdb -x -c $HOME/backup/firstdb_20240510_0052.dump
[postgres1@pg156 ~/backup]$ psql -h localhost -p 9143 -d firstdb
psql (14.2)
Введите "help", чтобы получить справку.

firstdb=# select * from orders;
 id | text | number
----+------+--------
  1 | 1    |    123
  2 | 2    |    123
  3 | 3    |    123
  4 | 4    |    123
  5 | 5    |    123
(5 строк)
```

Анализ

Wal-архивация невероятно полезна, так как позволяет не только вернуться к последнему состоянию, но и выбрать определенный момент времени

Вывод

Во время выполнения лабораторной работы мы изучили способы непрерывного бекапа кластера PostgreSQL, на практике настроили и применили его при различных сбоях базы данных.