НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Администрирование СУБД

Лабораторная работа № 3

Вариант 34834

Выполнил студент:

Подольский Вячеслав

Группа № Р3320

Преподаватель: Николаев Владимир Вячеславович

г. Санкт-Петербург

2024

Оглавление

[Задание 3](#_Toc177332637)

[Ход работы: 4](#_Toc177332638)

[Вывод: 6](#_Toc177332639)

# Задание

Этап 1. Резервное копирование

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

Периодические холодные полные копии.  
Полная копия (rsync) по расписанию (cron) раз в сутки. СУБД на время копирования должна останавливаться. На резервном узле хранить 14 копий, после успешного создания пятнадцатой копии, самую старую автоматически уничтожать.

* Подсчитать, каков будет объем резервных копий спустя месяц работы системы, исходя из следующих условий:
  + Средний объем новых данных в БД за сутки: 650МБ.
  + Средний объем измененных данных за сутки: 600МБ.
* 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:

* 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 любые две таблицы (DROP TABLE)
* Продемонстрировать результат.
* Выполнить восстановление данных указанным способом.
* Продемонстрировать и проанализировать результат.

## Ход работы:

Этап 1:

* 1. Настройка резервного копирования

Напишем bash-cкрипт, который будет останавливать кластер, отправлять полную копию на резервный узел, после удаленно запускать скрипт (для удаления).

Основной узел

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Резервный узел

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

На основном узле создадим cron-файл через команду (crontab -e) d котором опишем правило для запуска нашего скрипта каждую полночь (раз в сутки). После чего запустим.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Сделаем для основного узла подключение без пароля к резервному узлу:

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

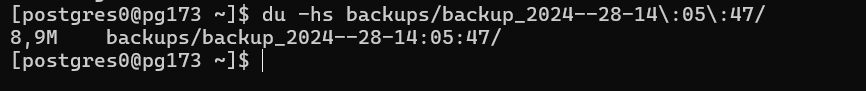
Автоматически созданное описание

Запустим вручную несколько раз скрипт и проверим логи на резервном узле:

Изображение выглядит как текст, снимок экрана, Шрифт, черно-белый

Автоматически созданное описание

1.2 Расчет объема данных



Общий объем данных, занимаемый бэкапами образует арифметическую прогрессию

Плюс к каждому бэкапу нужно прибавлять 600 МБ измененных данных

Тогда сумма будет выглядеть следующим образом

По условию мы сохраняем 14 последних резервных копий, поэтому из суммы нам нужно вычесть 16 первых членов:

1.3 Анализ результатов

По результатам объем резервных копий на резервном узле спустя месяц работы сервера бд составит 208 Гб. Полные холодные копии имеют большой размер и остановки сервера, что значительно хуже, например, горячих инкрементных копий.

Этап 2:

Восстановление

Воссоздадим файловую структуру кластера и табличного пространства на резервном узле из нашего бэкапа

cp -r ~/backups/backup\_2024--29-01:45:46/\* ~/

Создадим потерянные символические ссылки на табличные пространства и wal-файлы в связи с копированием данных на другой узел:

ln -s /var/db/postgres0/iat59 /var/db/postgres0/eaw92/pg\_tblspc/16393

ln -s /var/db/postgres0/agt19 /var/db/postgres0/eaw92/pg\_wal

Изображение выглядит как текст, снимок экрана, черный, Шрифт

Автоматически созданное описание

restore\_command = 'cp ~/agt19/%f %p'

touch ~/eaw92/recovery.signal

postgres -D eaw92 >~/logfile 2>&1 &

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

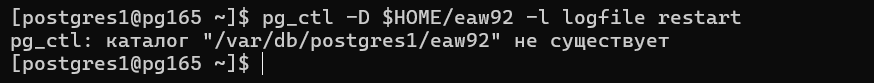
Этап 3:

3.1 Симуляция сбоя и проверка работы  
Удаляем директорию конфигурационных файлов:  
rm -rf eaw92/

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Перезапускаем сервер



3.2 Восстановление данных

Выполним восстановление с последней резервной копии:

rsync -avv ~/backups/backup\_2024--29-01:45:46/eaw92/\* postgres1@pg165:~/new\_eaw92

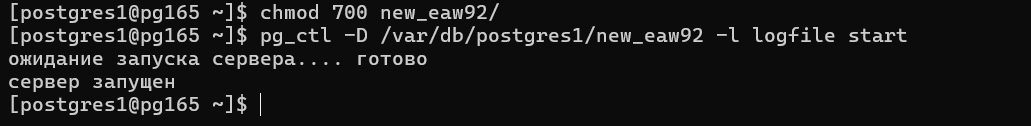
Так как директория eaw92 недоступна, тогда скопируем директорию даты в new\_eaw92.

Выдаем необходимые права для запуска сервера:

chmod 700 new\_eaw92/

Запускаем сервер:

pg\_ctl -D /var/db/postgres1/new\_eaw92 -l logfile start



Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Этап 4:

Сначала нужно включить архивирование. Изменим postgresql.conf



Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание



4.1 В каждую таблицу добавить 2-3 новые записи:

До изменений:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

После изменений:



Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, черный

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, меню

Автоматически созданное описание

4.2 Зафиксировать время и симулировать ошибку

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

2024-11-12 00:40:44.461631+03Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

4.3 Восстановление данных

Скопировал pg\_wal во временную папку, чтобы не потерять.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Сравним с тем, что находится на резервном узле:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

В общем, новые wal-файлы сохранились в обоих узлах.

Возьмем backup на резервном узле, чтобы восстановить файлы из него:



Далее из папки pg\_wal на основном узле удалили файлы и скопировали в него новый wal-файл



Установим время и команду восстановления в postgresql.conf







Создадим файл recovery.signal для старта сервера в режиме восстановления

Далее запускаем сервер

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Как видим, получилось восстановить состояние бд до возникновения повреждения.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Анализ

При восстановлении логических повреждений данных wal – архивация полезна, так как помимо восстановления к самому последнему состоянию позволяет возвратиться в какой-то определенный момент времени.

# Вывод

В ходе выполнения данной лабораторной работы я познакомился с прерывным бэкапом PostgreSQL кластера, посмотрел на практике как его настроить и применять при различных сбоях: полной потери основного узла, повреждении файлов бд или логического повреждения данных. Во всех случаях WAL бэкапы решили задачу.