УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Распределенные системы хранения данных»

**Лабораторная работа №3**

*Вариант 58*

Студент

*Ибадуллаев А. Э. Р33131*

Преподаватель

*Шешуков Д. М.*

Санкт-Петербург, 2023 г.

Описание задания

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.  
  
Требования к отчёту  
Отчет должен быть самостоятельным документом (без ссылок на внешние ресурсы), содержать всю последовательность команд, содержимое скриптов по каждому пункту задания. Для демонстрации результатов приводить команду вместе с выводом (самой наглядной частью вывода, при необходимости).  
  
1. Резервное копирование  
1.1 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).

1.2 Подсчитать, каков будет объем резервных копий спустя месяц работы системы,  
исходя из следующих условий:  
Средний объем новых данных в БД за сутки: ~100 МБ.  
1.3 Проанализировать результаты.  
  
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.  
  
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:  
3.1 Симулировать сбой:  
удалить с диска директорию конфигурационных файлов СУБД со всем содержимым.  
3.2 Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.  
3.3 Выполнить восстановление данных из резервной копии, учитывая следующее условие:  
Исходное расположение директории PGDATA недоступно - разместить в другой директории и скорректировать конфигурацию.  
3.4 Запустить СУБД, проверить работу и доступность данных, проанализировать результаты.  
  
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:  
4.1 В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.  
4.2 Зафиксировать время и симулировать ошибку:  
В любой таблице с внешними ключами изменить внешние ключи случайным образом (INSERT, UPDATE)  
4.3 Продемонстрировать результат.  
4.4 Выполнить восстановление данных указанным способом.  
4.5 Продемонстрировать и проанализировать результат.

Настройка WAL - архивации

*Создаем пользователя с привилегией replicant*

create role replicant with replication encrypted password ‘test’

login;

*postgesql.conf*



Изображение выглядит как текст

Автоматически созданное описание

*Создание первоначальной копии*

pg\_basebackup –progress -p 9022 -U replicant –format=t –wal-method=fetch -D cluster\_bak

*Формат – tar файлы, включаем необходимые журналы транзакций в нашу резервную копию, включаем отчет о прогрессе.*

*Прописываем свою директорию для сохранения*



*Переносим копию на резервный узел*

rsync –rsync-path=/opt/csw/bin/rsync –archive –verbose cluster\_bak

Изображение выглядит как текст

Автоматически созданное описание

*Подсчет размера копий*

*Через месяц изменения данных объем будет:*

30 \* 100 = 3 Гб

Изначальный бекап весит:



*В итоге: 3.2 Гб*

*Сравним результаты с размером бекапов в течение месяца:*

( 2\*187 + 200 \* (15-1) ) \* 15 / 2 = 23.8 Гб

*Очевидно, инкрементальные бекапы намного эффективнее.*

Потеря основного узла

*Воссоздадим файловую структуру кластера для восстановления*

mkdir ~/u03/gwt12

chmod 700 ~/ u03/gwt12

cd u03/gwt12

tar xvf ~/cluster\_bak/base.tar

*Файлы табличного пространства*

mkdir ~/u03/gtr127

cd ~/u03/gtr127

tar xvf ~/cluster\_bak/16384.tar

*Очищаем директорию для wal-файлов*

rm -rf ~/u03/gwt12/pg\_wal/\*

ln -s /var/postgres/postgres3/u03/gtr ~/u03/gwt12/pg\_tblspc/16384

*Указываем команду для загрузки wal-файлов*

Изображение выглядит как текст

Автоматически созданное описание

*Файл – флаг о восстановлении*

touch recovery.signal

*Запускаем кластер*

postgres -D ~/u03/gwt12

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание



*Проверяем работоспособность*

Изображение выглядит как текст

Автоматически созданное описание

Анализ выполнения

*Восстановление завершилось успешно. Но для корректного запуска необходимо подкорректировать ссылку*

Повреждение файлов БД

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

*Отчищаем директорию для wal-файлов*

rm -rf ~/u03/gwt12/pg\_wal/\*

*Подключаемся к базе, которая была в пространстве*

psql -p 9022 -d postgres

--

*Из-за того, что исходное пространство недоступно, распакуем копию в новое местоположение.*

mkdir gtr27\_new

cd gtr27\_new/

tar xvf ../cluster\_bak/16384.tar

*Запускаем поврежденный кластер в режиме восстановления wal-файлов и указываем команду для восстановления*

Изображение выглядит как текст

Автоматически созданное описание

*Меняем символическую ссылку*

rm ~/u03/gwt12/pg\_tblspc/16386

ln -s ~/gtr27\_new ~/u03/gwt12/pg\_tblspc/16386

*Запускаем в режиме восстановления*

touch ~/u03/gwt12/recovery.signal

postgres -D ~/u03/gwt12/

*Проверяем работоспособность*

Изображение выглядит как текст

Автоматически созданное описание

Анализ

*Восстановление завершилось успешно. Но для корректного запуска необходимо подкорректировать ссылку*

Логическое повреждение данных

*Попробуем создать некорректные изменения*

Изображение выглядит как стол

Автоматически созданное описание

*Восстановление*

Изображение выглядит как текст

Автоматически созданное описание**



*Очищаем директорию для wal-файлов*

rm -rf ~/u03/gwt12/pg\_wal/\*

*Перезапустим базу с восстановлением из wal-файла*

touch ~/u03/gwt12/recovery.signal

*Проверяем*

*Изображение выглядит как текст

Автоматически созданное описание*

*Восстановление прошло успешно*

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 я изучил способы непрерывного бекапа кластера PostgreSQL, на практике настроил и применил его при различных сбоях.*