Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №2**

по дисциплине

‘Распределённые системы хранения данных’

Вариант №35455

*Выполнил:*

Студент группы P33312

Соболев Иван Александрович

*Преподаватель:*

Осипов Святослав Владимирович



Санкт-Петербург, 2024

**Задание:**

Цель работы - на выделенном узле создать и сконфигурировать новый кластер БД Postgres, саму БД, табличные пространства и новую роль, а также произвести наполнение базы в соответствии с заданием. Отчёт по работе должен содержать все команды по настройке, скрипты, а также измененные строки конфигурационных файлов.

Способ подключения к узлу из сети Интернет через helios:  
ssh -J sXXXXXX@helios.cs.ifmo.ru:2222 postgresY@pgZZZ  
Способ подключения к узлу из сети факультета:  
ssh postgresY@pgZZZ  
Номер выделенного узла pgZZZ, а также логин и пароль для подключения Вам выдаст преподаватель.

Обратите внимание, что домашняя директория пользователя /var/postgres/$LOGNAME

Этап 1. Инициализация кластера БД

* Директория кластера: $HOME/ifg51
* Кодировка: ISO\_8859\_5
* Локаль: русская
* Параметры инициализации задать через аргументы команды

Этап 2. Конфигурация и запуск сервера БД

* Способ подключения: сокет TCP/IP, принимать подключения к любому IP-адресу узла
* Номер порта: 9455
* Остальные способы подключений запретить.
* Способ аутентификации клиентов: по паролю в открытом виде
* Настроить следующие параметры сервера БД:
  + max\_connections
  + shared\_buffers
  + temp\_buffers
  + work\_mem
  + checkpoint\_timeout
  + effective\_cache\_size
  + fsync
  + commit\_delay

Параметры должны быть подобраны в соответствии со сценарием OLAP:  
9 одновременных пользователей, пакетная запись/чтение данных по 64МБ.

* Директория WAL файлов: $HOME/krv1
* Формат лог-файлов: .csv
* Уровень сообщений лога: NOTICE
* Дополнительно логировать: контрольные точки и попытки подключения

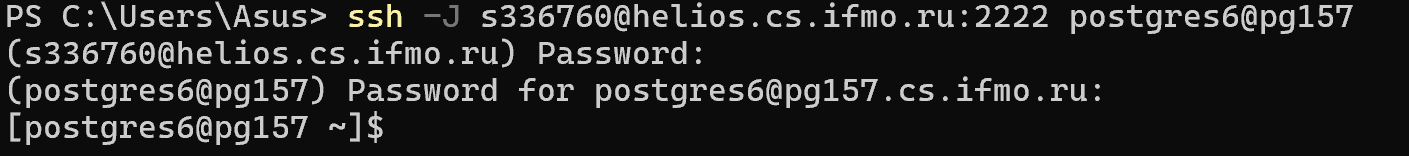
Этап 3. Дополнительные табличные пространства и наполнение базы

* Пересоздать шаблон template1 в новом табличном пространстве: $HOME/zzo32
* На основе template1 создать новую базу: wetbluelove
* Создать новую роль, предоставить необходимые права, разрешить подключение к базе.
* От имени новой роли (не администратора) произвести наполнение ВСЕХ созданных баз тестовыми наборами данных. ВСЕ табличные пространства должны использоваться по назначению.
* Вывести список всех табличных пространств кластера и содержащиеся в них объекты.

**Выполнение:**

Подключение к узлу:

ssh -J s336760@helios.cs.ifmo.ru:2222 postgres6@pg157



Инициализация кластера БД:

1. Создание директории кластера БД – mkdir -p $HOME/ifg51
2. Изменение системного пользователя - chown postgres6 $HOME/ifg51
3. Инициализация кластера - initdb --encoding=ISO\_8859\_5 --locale=ru\_RU.ISO8859-5 --username=postgres6 -D $HOME/ifg51

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Конфигурация и запуск сервера БД:

Передача конфигурационных файлов через scp:

На узле нет vim'a, а работать в vi некомфортно. Поэтому передает на гелиос конфигурационные файлы:

scp postgres6@pg157:ifg51/pg\_hba.conf .

scp postgres6@pg157:ifg51/postgresql.conf .

Обратно отправляем на узел:

scp pg\_hba.conf [postgres6@pg157:ifg51/pg\_hba.conf](mailto:postgres6@pg157:ifg51/pg_hba.conf)

scp postgresql.conf [postgres6@pg157:ifg51/postgresql.conf](mailto:postgres6@pg157:ifg51/postgresql.conf)

**pg\_hba.conf**

Разрешаем подключение по паролю через host, остальные способы подключения запрещаем:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**postgresql.conf**

Меняем следующие параметры:

порт и порты слушания:

Изображение выглядит как текст, снимок экрана, Шрифт, белый

Автоматически созданное описание

ставим максимальное количество подключённых пользователей равным девяти:



Установка размеров буферов shared\_buffers, temp\_buffers и work\_mem:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

shared\_buffers: параметр, который задает количество памяти, которое PostgreSQL будет использовать для кэширования данных из таблиц и индексов в оперативной памяти.

Из документации PostgreSQL следует, что shared\_buffers следует держать от 25% до 40% от всего выделенного ОЗУ. Так как в варианте не сказано сколько ОЗУ выделено на кластер, то будем использовать наиболее частое значение 8Гб, тогда на shared\_buffers выделим 2Гб.

temp\_buffers: параметр устанавливает максимальный размер оперативной памяти, которую сервер может использовать для хранения временных файлов сессии, создаваемых в процессе выполнения операций сортировки и объединения данных.

Размер временных буферов следует сделать достаточно большим(256МБ), так как система соответствует OLAP с пакетной передачей в среднем по 64 МБ.

work\_mem: параметр отвечает за количество памяти, выделяемой для выполнения операций сортировки, хэширования, агрегирования и других операций обработки данных для каждой сессии.

Так как операции большие, и, скорее всего, используют множество сортировок и хэш-таблиц, то значение work\_mem я выставил в 128МБ (объём памяти для внутренних операций сортировок и хэш-таблиц).

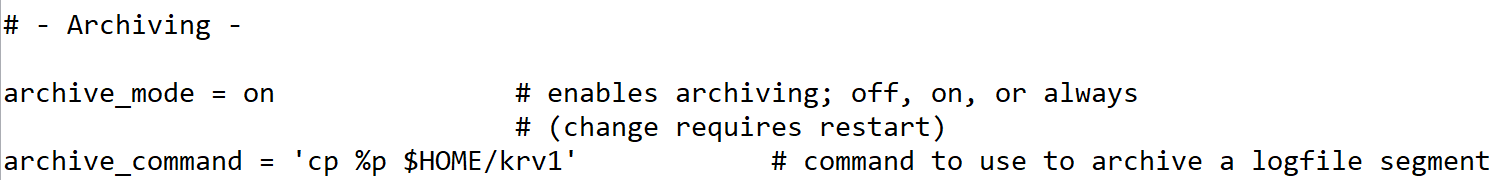
- effective\_cache\_size = 4GB. Оставил по умолчанию (должен быть не меньше чем shared\_buffers).

- fsync = on. Оставил данный параметр включенным, чтобы запись на диск происходила. fsync: Параметр определяет, включена ли синхронизация записи на диск в PostgreSQL. Флаг fsync имеет смысл отключать на read-only копиях бд, в других случаях нужно включать для повышения отказоустойчивости независимо от конфигурации системы.

- commit\_delay = 0(мс). Сохранение на WAL начинается сразу после выполнения операции. Изменение задержки перед сохранением WAL имеет смысл только в том случае, если есть возможность протестировать его влияние на общую производительность.

- checkpoint\_timeout = 10min. Параметр checkpoint\_timeout в PostgreSQL определяет интервал времени в секундах между запусками процесса контрольной точки (checkpoint). Поставил значение на 10 минут, так как предполагаю, что операции не частые и их немного. Не сильно отошел от стандартных 5 минут.

Включаем архивирование и указываем директорию, в которую будут копироваться WAL-файлы:



Формат лог-файлов: .csv

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описание

Уровень сообщений лога: NOTICE



Дополнительно логировать: контрольные точки и попытки подключения

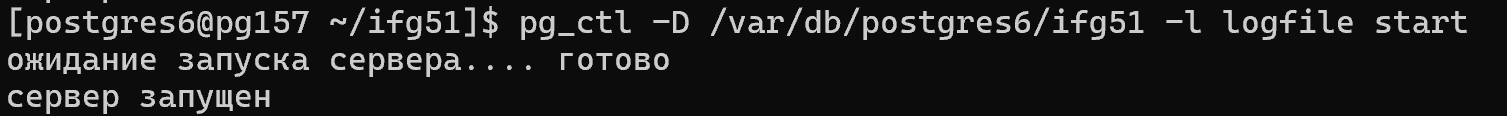
Изображение выглядит как текст, снимок экрана, Шрифт, чек

Автоматически созданное описание

### **Этап 3. Дополнительные табличные пространства и наполнение базы**

Запускаем сервер

pg\_ctl -D /var/db/postgres6/ifg51 -l logfile start



Для запуска БД необходимо сначала установить `trust` in `pg\_hba.conf` и обновить дефолтный пароль для postgres6 (Я установил 123). Затем можем подключаться.

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Пересоздать шаблон template1 в новом табличном пространстве: $HOME/zzo32

mkdir -p $HOME/zzo32

psql -h localhost -p 9455 -U postgres6 postgres

Создаем табличное пространство - CREATE TABLESPACE zzo32 LOCATION '/var/db/postgres6/zzo32';

Перезаписываем туда template 1:

UPDATE pg\_database SET datistemplate = false WHERE datname = 'template1';

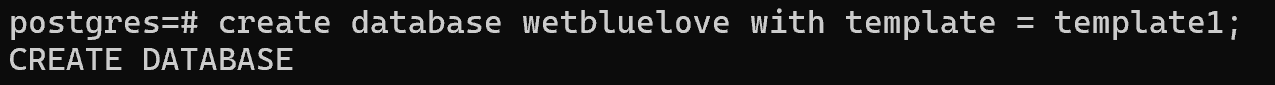
DROP DATABASE template1;

CREATE DATABASE template1 TEMPLATE template0 TABLESPACE zzo32;

UPDATE pg\_database SET datistemplate = true WHERE datname = 'template1';

На основе template1 создать новую базу: wetbluelove

create database wetbluelove with template = template1;



Переподключаемся к базе wetbluelove:

psql -h localhost -p 9455 -U postgres6 wetbluelove

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Создать новую роль, предоставить необходимые права, разрешить подключение к базе.

create role ivan login password 'ivan';

Создадим таблицы в бд и заполним их:

psql -h localhost -p 9455 -U ivan wetbluelove -f create.sql

psql -h localhost -p 9455 -U ivan wetbluelove -f insert.sql

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Вывести список всех табличных пространств кластера и содержащиеся в них объекты.

select \* from pg\_tablespace;

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

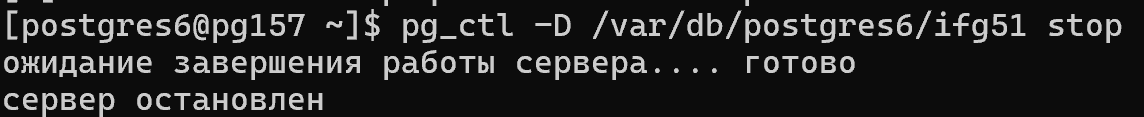
SELECT c.relname, t.spcname FROM pg\_class c JOIN pg\_tablespace t ON c.reltablespace = t.oid;

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Завершаем работу на узле, чтобы не потреблять ресурсы зря.

pg\_ctl -D /var/db/postgres6/ifg51 stop



**Выводы**

Во время выполнения лабораторной работы я научился создавать и конфигурировать кластер БД PostgreSQL. Я познакомился с созданием и работой табличных пространств и ролей.