НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет Программной Инженерии и Компьютерной Техники

Администрирование систем управления базами данных

Лабораторная работа № 2

191 вариант

Выполнили студенты:

Мизевич Илья

Группа №: P33202

Преподаватель: Николаев Владимир Вячеславович

г. Санкт-Петербург

2023

**Этапы выполнения работы**

На выделенном узле создать и сконфигурировать новый кластер БД, саму БД,  
табличные пространства и новую роль в соответствии с заданием. Произвести  
наполнение базы.

1. Инициализация кластера БД  
        • Имя узла — pg191.  
        • Имя пользователя — postgres0.  
        • Директория кластера БД — $HOME/u23/znt06.  
        • Кодировка, локаль — UTF8, английская  
        • Перечисленные параметры задать через переменные окружения.
2. Конфигурация и запуск сервера БД  
        • Способ подключения к БД — TCP/IP socket, номер порта 9191.  
        • Остальные способы подключений запретить.  
        • Способ аутентификации клиентов — по имени пользователя.  
        • 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 с аппаратной конфигурацией: оперативная память 8 ГБ, хранение на жёстком диске (HDD);
3. • Директория WAL файлов — $HOME/u34/znt08.  
        • Формат лог-файлов — log.  
        • Уровень сообщений лога — INFO.  
        • Дополнительно логировать — контрольные точки.
4. Дополнительные табличные пространства и наполнение  
        • Пересоздать шаблон template1 в новом табличном пространстве:  
            ◦ $HOME/u34/znt08.  
        • На основе template0 создать новую базу — theovermind4.  
        • От имени новой роли (не администратора) произвести наполнение  
   существующих баз тестовыми наборами данных. Предоставить права по  
   необходимости. Табличные пространства должны использоваться по назначению.  
        • Вывести список всех табличных пространств кластера и содержащиеся  
   в них объекты.

**Ход работы**

1. **Инициализация кластера БД**

PGHOST=pg191

PGUSERNAME=postgres0

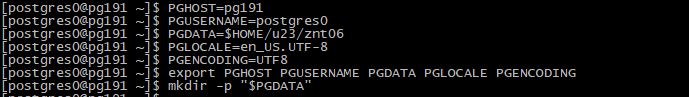
PGDATA=$HOME/u23/znt06

PGLOCALE=en\_US.UTF-8

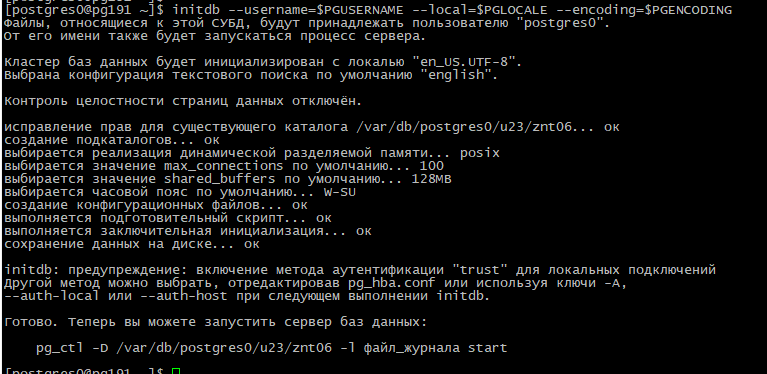
PGENCODING=UTF8

export PGHOST PGUSERNAME PGDATA PGLOCALE PGENCODING

mkdir -p "$PGDATA"



initdb --username=$PGUSERNAME --local=$PGLOCALE --encoding="$PGENCODING

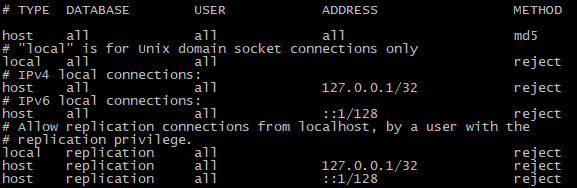
c

1. **Конфигурация и запуск сервера БД**

Установка нужных параметров происходит в файлах pg\_hba.conf и postgresql.conf.

Редактирование файлов осуществлялось с помощью редактора vi.

Измененная конфигурация файла pg\_hba.conf:





Изменение конфигурации postgresql.conf осуществлялось в соответствии с заданием:

port – 9190, указываем TCP-порт, открываемый сервером. (На указанном в варианте порту база не запускалась).

max\_connections – 100, определяет максимальное количество одновременных подключений к серверу БД. Сервер выдавал ошибку нехватки ресурсов при изменении этого значения. Поэтому оставляем стандартное значение



shared\_buffers – 2048MB, задает объем памяти, который будет использовать сервер для буферов разделяемой памяти. В документации рекомендуется устанавливать начальное значение 25% от объема памяти, но т.к. мы создаем тестовый кластер на общем сервере, то выделять такое количество ресурсов нецелесообразно. 0,25 \* 8 = 2 гб



temp\_buffers – 16MB, задает максимальное число временных буферов для каждого сеанса пользователя.





work\_mem – 32MB, задает объем памяти, который будет использоваться для внутренних операций.

checkpoint\_timeout – 5min, определяет максимальное время в секундах между автоматическими контрольным точками в WAL. При увеличении времени может привести к увеличению времени восстановления после сбоя, а также повышает риск потери данных, так как чекпоинты производятся реже. Если время уменьшить то это создаст нагрузку на чтение/запись с диска , что может существенно тормозить систему т.к в качестве хранилища у нас используется HDD.



effective\_cache\_size – 4GB, определяет представление планировщика об эффективном размере дискового кеша, доступном для одного запроса. Это представление влияет на оценку стоимости использования индекса; чем выше это значение, тем больше вероятность, что будет применяться сканирование по индексу, чем ниже, тем более вероятно, что будет выбрано последовательное сканирование. Рекомендуется ставить 40%-50% от доступной ОЗУ.



fsync – on, данный параметр отвечает за сброс данных из кэша на диск при завершении транзакций. Так как нам важна доступность данных, то мы активируем этот параметр для большей надежности.

commit\_delay – 0, добавляет паузу в микросекундах перед выполнением сохранения WAL. При включенном fsync задержка не добавляется, поэтому оставим параметр равным 0.

Директория WAL файлов — поддиректория $PGDATA

Ничего не меняем, оставляем стандартный путь $PGDATA/pg\_wal

Формат лог-файлов — log:

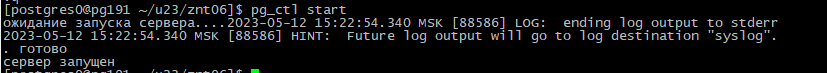
Уровень сообщений лога — info:



Дополнительно логировать — контрольные точки:



Запуск сервера:



1. **919Дополнительные табличные пространства и наполнение**

новое табличное пространство $HOME/u13/nrur3:





Удаляем и пересоздаем template1 в новом табличном пространстве:

https://sun9-60.userapi.com/impg/xAgsQCeGzXN4GSEhBUcSaBoVcHhCyQhx7hjsag/fOS0Q5-l2IY.jpg?size=756x57&quality=96&sign=65111d59332c9ebe4169c065637fc2cf&type=album

  
На основе template0 новая база — theovermind4:



Создадим пользователя и таблицу:

CREATE ROLE yranikus WITH LOGIN PASSWORD '1234';



G  
Выдаем права:





Заходим в бд от имени нового пользователя:



Вставляем данные:

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

**SELECT**

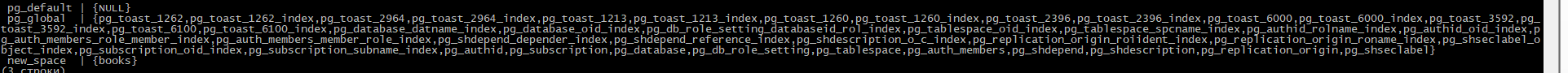
**t.spcname AS tablespace,**

**c.relname**

**FROM pg\_tablespace t**

**LEFT JOIN pg\_class c ON c.reltablespace = t.oid \g**

Объекты, содержащиеся в табличных пространствах:



**Вывод**

В ходе лабораторной работы мы научились инициализировать кластер используя initdb, настроили его с помощью конфигурационных файлов pg\_hba.conf и postgresql.conf, запустили используя pg\_ctl. Также создали пользователей, выдали им права и посмотрели как они работают.