**Национальный Исследовательский Университет**

**Информационных Технологий, Механики и Оптики**

Факультет программной инженерии и компьютерной техники

Лабораторная работа 2

по дисциплине

«Распределенные системы хранения данных»

Вариант - 11

Выполнил:

Студент группы P33101,

Патутин В.М.

Преподаватель:

Николаев В.В.

Санкт-Петербург

2021г.

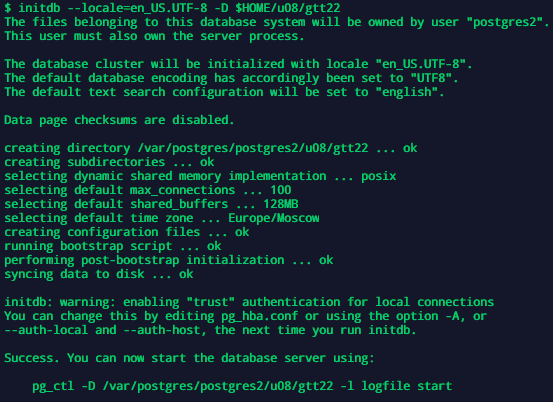
**Задание ЛР:**

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

Персональный пароль для работы с узлом выдается преподавателем.  
Обратите внимание, что домашняя директория пользователя  
/var/postgres/$LOGNAME  
  
Этапы выполнения работы:  
Инициализация кластера БД  
    • Имя узла — pg103.  
    • Имя пользователя — postgres2.  
    • Директория кластера БД — $HOME/u08/gtt22.  
    • Кодировка, локаль — UTF8, английская  
    • Перечисленные параметры задать через аргументы команды.  
Конфигурация и запуск сервера БД  
    • Способ подключения к БД — TCP/IP socket, номер порта 9011.  
    • 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 16 ГБ, хранение на жёстком диске (HDD);  
    • Директория WAL файлов — поддиректория в PGDATA.  
    • Формат лог-файлов — log.  
    • Уровень сообщений лога — NOTICE.  
    • Дополнительно логировать — завершение сессий и продолжительность выполнения команд.  
Дополнительные табличные пространства и наполнение  
    • На основе шаблона template1 пересоздать базу postgres в новом табличном пространстве:  
        ◦ $HOME/u07/gtr77.  
    • На основе template1 создать новую базу — whitecat.  
    • 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.  
      
Пароль для подключения к узлу: ldFwVgdm  
Пожалуйста, не выкладывайте пароль в общий доступ.  
Не забывайте останавливать свой экземляр БД, когда прекращаете с ним работать.

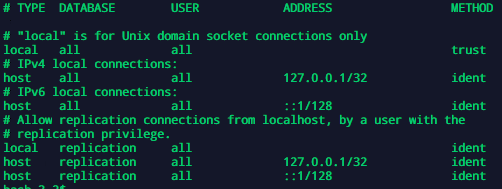
**Выполнение:**

**1)** Инициализация кластера БД:



2) Конфигурация и запуск сервера БД:

* Изменим pg\_hba.conf:



* Изменим postgresql.conf:

port - устанавливаем порт сервера:



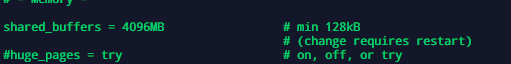
max\_connections - устанавливаем максимальное число одновременных подключений к серверу БД. В целом изначальное значение в 100 меня устраивало и я не нашел никаких точных рекомендаций, поэтому увеличил до 128, чтобы в будущем было удобно делить:



shared\_buffers - Задаёт объём памяти, который будет использовать сервер баз данных для буферов в разделяемой памяти.

Рекомендация - если вы используете выделенный сервер с объёмом ОЗУ 1 ГБ и более, разумным начальным значением shared\_buffers будет 25% от объёма памяти. Выделять для shared\_buffers более 40% ОЗУ вряд ли будет полезно.

Формула - ОЗУ\*25% = 0.25 \* 16384МБ = 4096МБ



При установлении такого значений была получена следующая ошибка:



Великий гугл сказал, что «Это может быть связано с тем, что shared\_buffers задано слишком большое значение», поэтому значение было изменено на 3072МБ. Проблема была решена.

temp\_buffers - задаёт максимальное число временных буферов для каждого сеанса. Сложно решить какое значение необходимо, потому что нет информации для чего будет использоваться кластер.



work\_mem - задаёт объём памяти, который будет использоваться для внутренних операций сортировки и хеш-таблиц, прежде чем будут задействованы временные файлы на диске.

Рекомендация - предел для work\_mem можно вычислить, разделив объём доступной памяти (физическая память минус объём под совместно используемые страницы shared\_buffers) на максимальное число одновременно используемых активных соединений.

Формула - (ОЗУ- shared\_buffers)/ max\_connections = 96МБ



checkpoint\_timeout - Максимальное время между автоматическими контрольными точками в WAL (в секундах). Дефолтное значение в 5 минут идеально сбалансировано.



effective\_cache\_size - Определяет представление планировщика об эффективном размере дискового кеша, доступном для одного запроса.

Рекомендация - В качестве начального значения можете использовать 25-50% доступной памяти. Точно должен быть >= shared\_buffers.

Формула – 0.5 \* 16Гб = 8 ГБ



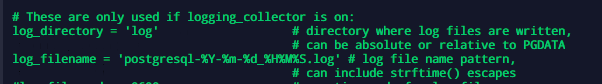
fsync - Если этот параметр установлен, сервер &project; старается добиться, чтобы изменения были записаны на диск физически, выполняя системные вызовы fsync() или другими подобными методами. Это даёт гарантию, что кластер баз данных сможет вернуться в согласованное состояние после сбоя оборудования или операционной системы. Отключение дает выигрыш в скорости, однако может привести к потери данных.



commit\_delay - добавляет паузу (в микросекундах) перед собственно выполнением сохранения WAL. Эта задержка может увеличить быстродействие при фиксировании множества транзакций, позволяя зафиксировать большее число транзакций за одну операции сохранения WAL, если система нагружена достаточно сильно и за заданное время успевают зафиксироваться другие транзакции. Эта задержка окажется бесполезной, если никакие другие транзакции не будут зафиксированы за это время, поэтому она добавляется, только в если момент запроса сохранения WAL активны как минимум commit\_siblings других транзакций.   


log\_directory - при включённом logging\_collector, определяет каталог, в котором создаются журнальные файлы.

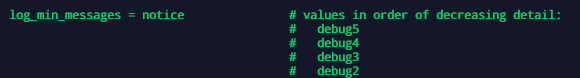
log\_filename - При включённом logging\_collector задаёт имена журнальных файлов.



logging\_collector - параметр включает сборщик сообщений.



log\_min\_messages - управляет минимальным уровнем сообщений, записываемых в журнал сервера.

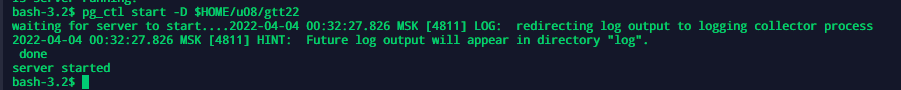


log\_disconnections – Включает протоколирование завершения сеанса.

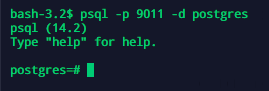
log\_duration - Записывает продолжительность каждой завершённой команды.



Запускаем сервер:

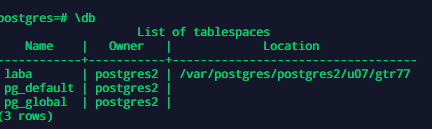


Подключаемся:



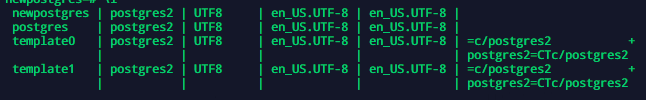
3)Создаем новое табличное пространство:



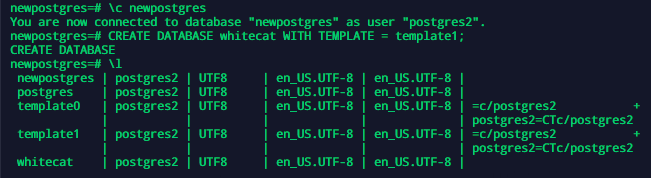


Пересоздаем базу postgres на основе шаблона template1 в новом табличном пространстве:





Создаем новую базу с именем whitecat:

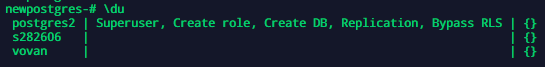


Создаем новую роль:



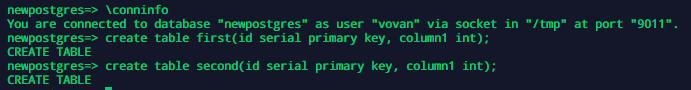




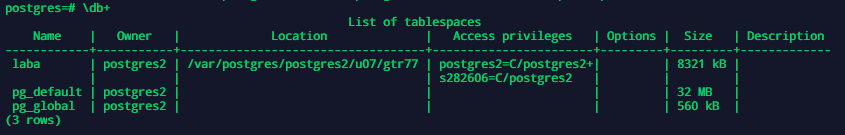


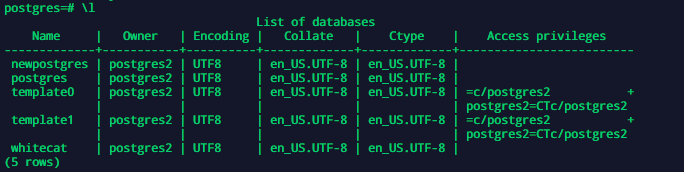


Наполнение базы данных:



4) Вывод таблиц:





**Выводы:**

Таким образом мною была проделана следующая работа: На выделенном узле создан и сконфигурирован новый кластер БД, сама БД, табличные пространства и новая роль в соответствии с заданием. Произведено наполнение базы.