УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Распределенные системы хранения данных»

**Лабораторная работа №4**

*Вариант 23*

Студент

*Митрофанов Е. Ю.*

*Патутин В. М.*

*P33101*

Преподаватель

*Шешуков Д. М.*

Санкт-Петербург, 2022 г.

Описание задания

Требования к выполнению лабораторной работы:  
    Работа рассчитана на двух человек. Для выполнения и демонстрации лабораторной работы разрешено (и рекомендуется) использование своих компьютеров. В случае отсутствия возможности использования своего компьютера для выполнения лабораторной работы обратитесь к преподавателю для корректировки варианта и получения доступа к узлам.  
    В качестве хостов использовать одинаковые виртуальные машины.  
    В первую очередь настроить сеть виртуальных машин:  
        Если ВМ запускаются на одном хосте, рекомендуется использовать NAT сеть.  
        Если ВМ запускаются на различных хостах, рекомендуется использовать сетевые интерфейсы в режиме “Bridge”; для связи рекомендуется использовать проводное соединение.  
        Проверить сетевую связность между всеми узлами (ping, ssh).  
    Для подключения к СУБД (например, через psql), использовать отдельную виртуальную или физическую машину.  
    Перед тем как “сломать” узел на этапе 2, рекомендуется выполнить снапшот виртуальной машины.  
    Для демонстрации наполнения базы, а также доступа на запись (см. задание ниже) использовать не меньше двух таблиц, трёх столбцов, пяти строк, двух транзакций, двух клиентских сессий. Данные не обязаны быть осмысленными, но должны быть легко отличимы - повторяющиеся строки запрещены.  
  
Этап 1 Настройка:  
    Развернуть postgres на двух узлах в режиме трансляции логов. Не использовать дополнительные пакеты. Продемонстрировать доступ в режиме чтение/запись на основном сервере, а также что новые данные синхронизируются на резервный.  
  
Этап 2.1 Подготовка:  
    a. Установить несколько клиентских подключений к СУБД.  
    b. Продемонстрировать состояние данных и работу клиентов в режиме чтение/запись.  
  
Этап 2.2 Сбой:  
    Симулировать отказ основного узла - выполнить жесткое выключение виртуальной машины.

Этап 2.3 Отработка:  
    a. Найти продемонстрировать в логах релевантные сообщения об ошибках.  
    b. Выполнить фейловер на резервный сервер.  
    c. Продемонстрировать состояние данных и работу клиентов в режиме чтение/запись.  
  
Этап 3 Восстановление:  
    a. Восстановить работу основного узла - откатить действие, выполненное с виртуальной машиной на этапе 2.2.  
    b. Актуализировать состояние базы на основном узле - накатить все изменения данных, выполненные на этапе 2.3.  
    c. Восстановить работу узлов в исходной конфигурации (в соответствии с этапом 1).  
    d. Продемонстрировать состояние данных и работу клиентов в режиме чтение/запись.

# Этап 1

*Подключаем сеть виртуальных машин:*

Изображение выглядит как текст

Автоматически созданное описание

*Проверяем доступность хостов в сети:*

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

*Создадим кластер:*

Изображение выглядит как текст

Автоматически созданное описание

*Настроим параметры входа в pg\_hba.conf:*

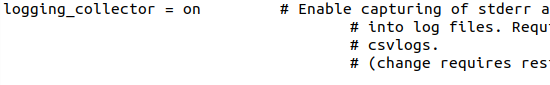
Изображение выглядит как текст

Автоматически созданное описание

*Настроим параметры входа в postgresql.conf:*

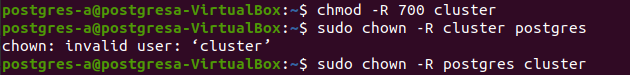
Изображение выглядит как текст

Автоматически созданное описание



*Дополнительные шаги настройки:*







*Запуск БД:*

Изображение выглядит как текст

Автоматически созданное описание

*Создадим пользователя с правом replication*:



Standby кластер

*Создадим полный первоначальный бекап:*



*Запуск БД:*

Изображение выглядит как текст

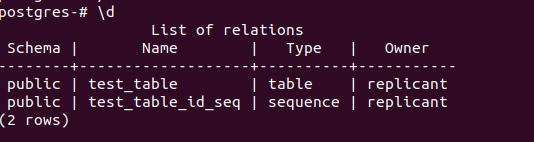
Автоматически созданное описание

Демонстрация

*Наполним данными основной кластер:*



*Убедимся в синхронизации на второй машине:*



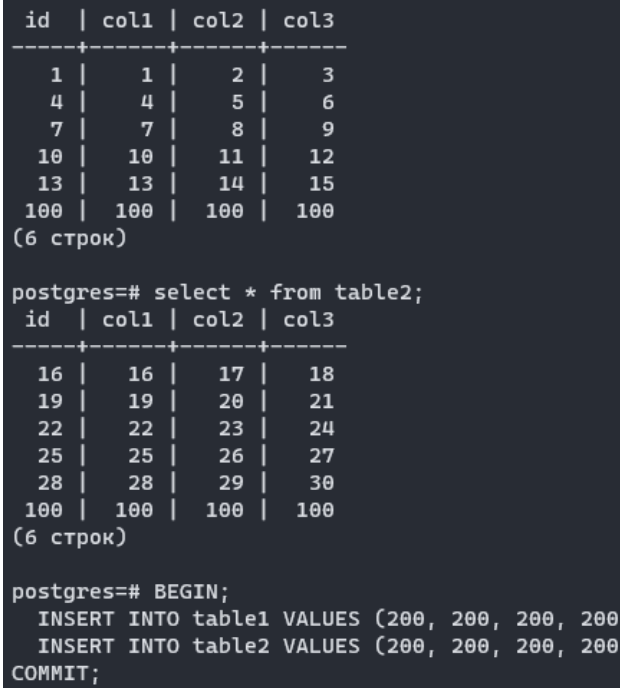
# Этап 2.1 Подготовка

*Подключаемся к основному кластеру с двух сессий и вносим изменения с первой:*

Изображение выглядит как текст

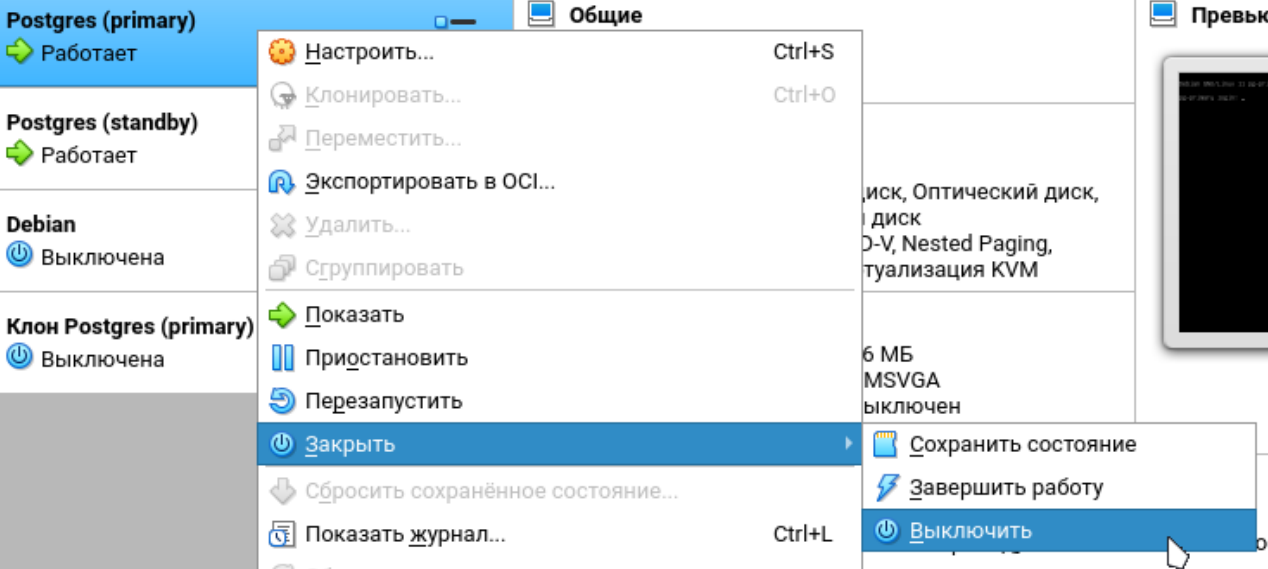
Автоматически созданное описание

*Проверяем изменения во второй и делаем транзакцию:*



# Этап 2.2 Сбой

*Выключаем работающую виртуальную машину:*



# Этап 2.3 Отработка

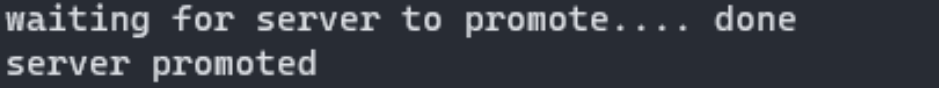
*В логах видим сообщение о потере соединения с главным кластером*

Изображение выглядит как текст

Автоматически созданное описание

*Переводим Standby кластер в ручной режим*

pg\_ctl -D cluster promote



*Подключаемся к резервному кластеру с двух сессий:*

Изображение выглядит как текст

Автоматически созданное описание

*Проверяем данные со второй сессии и так же вносим изменения:*

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Этап 3 Восстановление

*Вновь запускаем primary cluster.*

*Создадим новый basebackup на standby и перенесем на primary:*

pg\_basebackup -U replicant --format=t --wal-method=fetch -D cluster\_bak1

scp -r cluster\_bak1 pg-primary: Разворачиваем новый бекап на primary: tar -xvf cluster\_bak1/base.tar -C cluster

*Восстанавливает postgresql.conf на primary:*

archive\_mode = on

archive\_command = 'scp %p postgresa@postgresa-VirtualBox:~/wal\_archive/%f'

*Перезапускаем Standby в резервном режиме и проверяем коннект:*

*Изображение выглядит как текст

Автоматически созданное описание*

*Проверяем все данные на целостность на основной машине:*

Изображение выглядит как стол

Автоматически созданное описание

# Вывод

*Во время выполнения лабораторной работы мы настроили сеть виртуальных машин и настроили синхронизацию двух кластеров postgresql. Мы познакомились с механизмами отказоустойчивости, балансировки и репликации – симулировали сбой одного из кластеров и восстановили его после.*