**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе № 3

по дисциплине «**Администрирование систем управления базами данных**»

Вариант 45995

Выполнили: Ефименко К.А.

Факультет: ПИиКТ

Группа: P33202

Преподаватель: Николаев В.В.

# Задание

**1. Резервное копирование**

* Настроить резервное копирование с основного узла на резервный следующим образом: Периодические полные копии + непрерывное архивирование.  
  Включить для СУБД режим архивирования WAL; настроить копирование WAL (scp) на резервный узел; настроить полное резервное копирование (pg\_basebackup) по расписанию (cron) раз в неделю. Созданные полные копии должны сразу копироваться (scp) на резервный хост. Срок хранения копий на основной системе - 1 неделя, на резервной - 4 недели. По истечении срока хранения, старые архивы и неактуальные WAL должны автоматически уничтожаться.
* Подсчитать, каков будет объем резервных копий спустя месяц работы системы, исходя из следующих условий:
  + Средний объем новых данных в БД за сутки: 150МБ.
  + Средний объем измененных данных за сутки: 700МБ.
* 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:

* Симулировать сбой:
  + удалить с диска директорию конфигурационных файлов СУБД со всем содержимым.
* Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.
* Выполнить восстановление данных из резервной копии, учитывая следующее условие:
  + исходное расположение дополнительных табличных пространств недоступно - разместить в другой директории и скорректировать конфигурацию.
* 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.

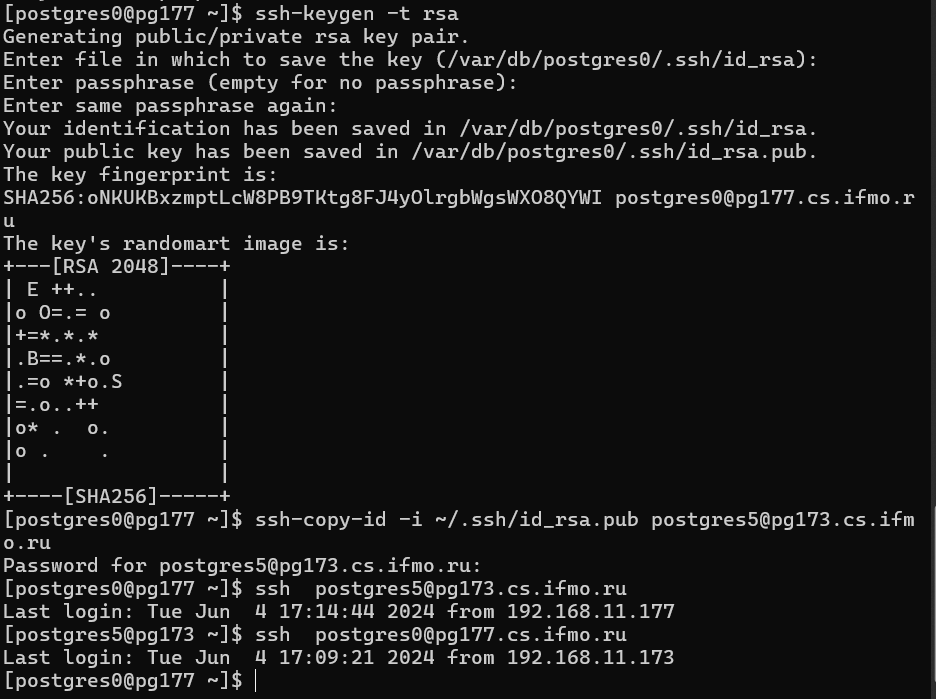
Ход работы:

* В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.
* Зафиксировать время и симулировать ошибку:
  + Удалить каждую вторую строку в любой таблице (DELETE)
* Продемонстрировать результат.
* Выполнить восстановление данных указанным способом.
* Продемонстрировать и проанализировать результат.

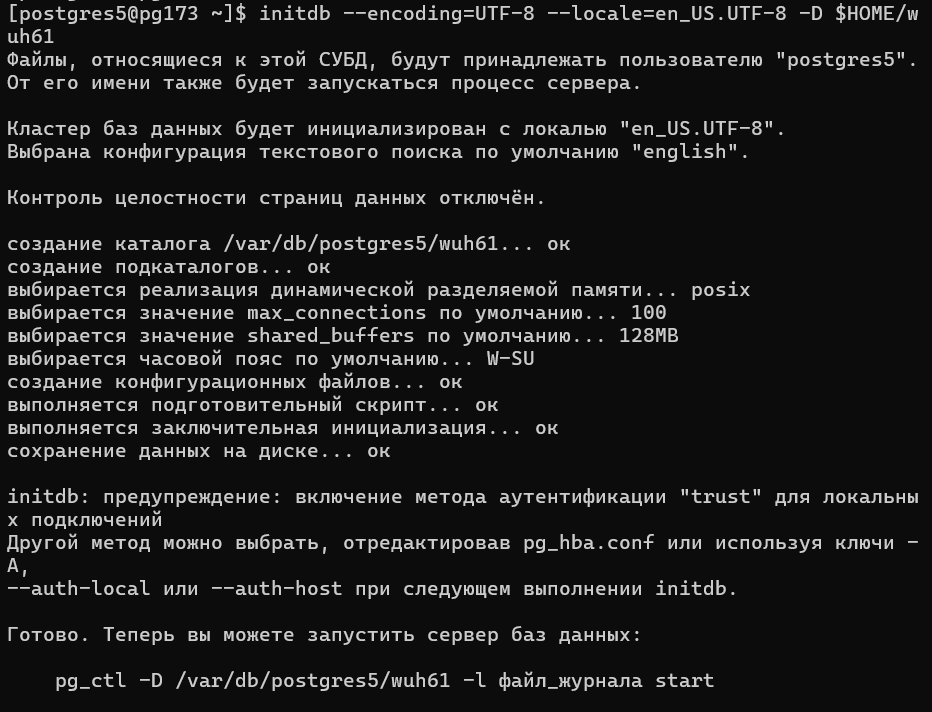
# Выполнение

**1. Резервное копирование**

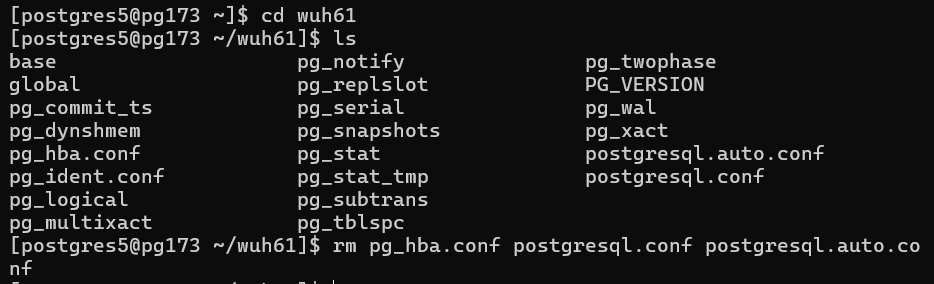
Создадим аутентификационный ключ SSH для подключения между основным узлом и резервным без ввода пароля

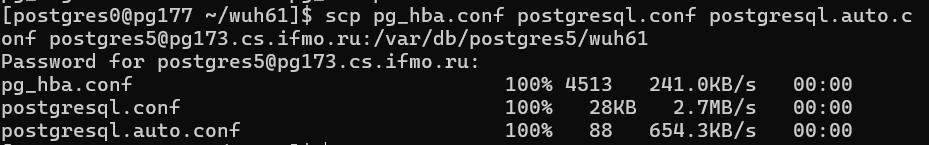


Создадим кластер БД postgres на резервном узле

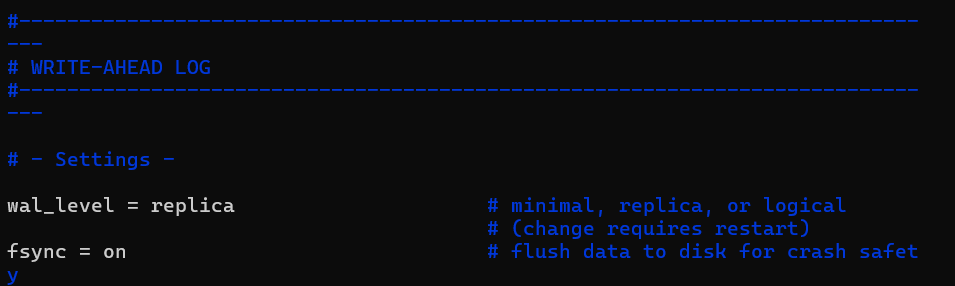


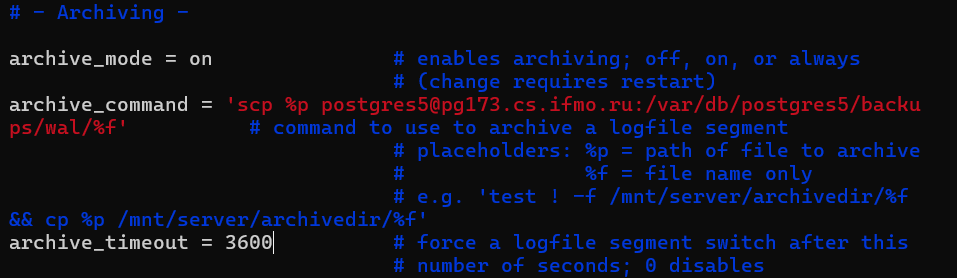
Конфиги postgresql.conf и pg\_hba.conf на резервном узле удалим и перенесем их с основного.



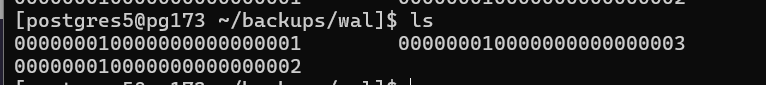


Вернемся на основной узел. Включим архивацию WAL-файлов и зададим директорию архивации – wal\_archive. Для этого изменим конфиг postgresql.conf. wal\_level должен быть не ниже replica, включен archive\_mode. Я задал команду копирования scp для каждой архивации wal. Также установил archive\_timeout 1 час.



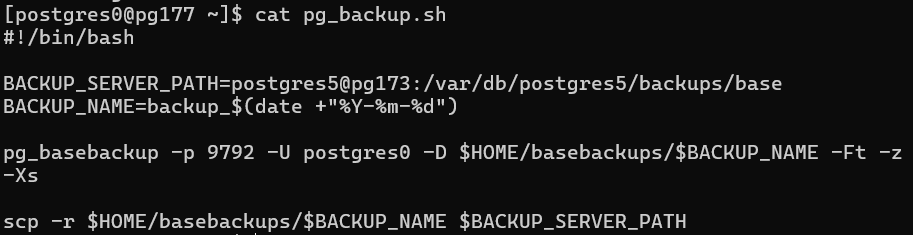


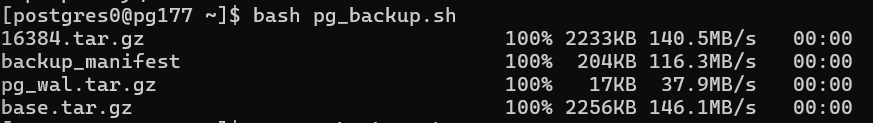
Чтобы проверить, что все работает, я задал archive\_timeout=60, то есть одна минута, потом зашел в psql и делал insert запросы в таблицу. Далее проверил, что лежит в папке backups/wal на резервном сервере

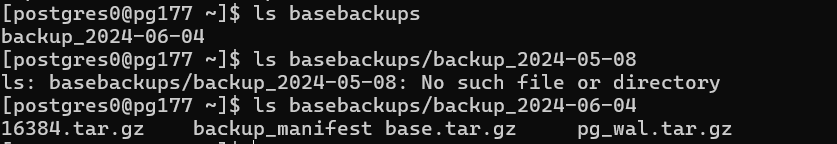


Далее напишем скрипт для резервного копирования и переноса резервной копии на резервный сервер. Запустим его и проверим создался ли бекап локально и на резервном сервере.

На основном сервере бекап создался.



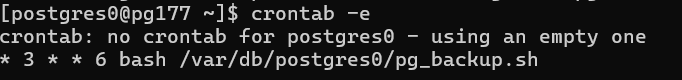


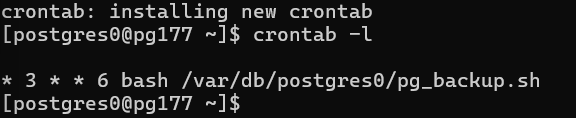


На резервном тоже.



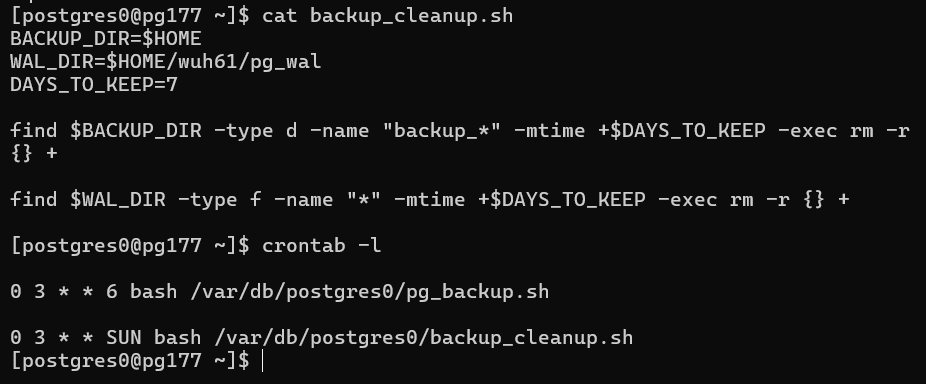
Далее с помощью crontab сделаем регулярную задачу – раз в неделю исполнять скрипт для создания резервных копий.

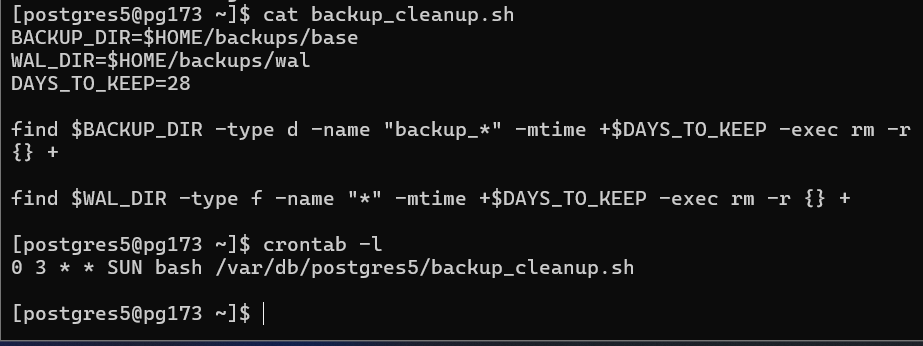




Также напишем скрипт для автоматического удаления старых резервных копий на основном и резервном серверах, регулярную задачу заведем также с помощью cron, но со сдвигом на +день.

Сначала напишем скрипт на основном сервере, также добавим задачу в crontab



Далее добавим скрипт и задачу на резервном сервере. Так как в cron нет поддержки синтаксиса для задачи на каждые 4 недели, то пусть задача выполняется каждую неделю, в условии поиска старых файлов есть ограничение на 28 дней. 

Теперь подсчитаем объем резервных копий на резервном сервере спустя месяц работы сервера. Так как каждые 4 недели все устаревшие резервные копии удаляются, то на резервном сервере максимально может быть 4 резервные копии, пусть первая резервная копия была сделана через 9 дней после старта сервера. Также мы используем сжатие gzip, сейчас резервная копия весит 4,4Мб



B то время как вес директрии wuh61 + вес директорий с табличными пространствами составляет 10Мб, пусть эффективность сжатия 60%. Объем новых данных в БД за день 150Мб, этот вес надо каждый день добавлять к весу кластера. Возьмем стартовый вес резервной копии в 4,4 Мб, тогда через 9 дней вторая резервная копия будет весить 4,4+(150\*9)\*0,4= 544,4 Мб и так далее. Тогда к концу месяца на сервере будет 4 резервных копии общим весом в   
 1. Первая резервная копия (через 7 дней):

Вес новых данных за 7 дней:

150 МБ/день × 7 дней = 1050 МБ

Сжатый вес новых данных:

1050 МБ × 0.4 = 420 МБ

Общий вес первой резервной копии:

4.4 МБ + 420 МБ = 424.4 МБ

2. Вторая резервная копия (через 14 дней):

Вес новых данных за 14 дней:

150 МБ/день×14 дней=2100 МБ

Сжатый вес новых данных:

2100 МБ×0.4=840 МБ

Общий вес второй резервной копии:

4.4 МБ+840 МБ=844.4 МБ

3. Третья резервная копия (через 21 день):

Вес новых данных за 21 день:

150 МБ/день×21 день=3150 МБ

Сжатый вес новых данных:

3150 МБ×0.4=1260 МБ

Общий вес третьей резервной копии:

4.4 МБ+1260 МБ=1264.4 МБ

4. Четвертая резервная копия (через 28 дней):

Вес новых данных за 28 дней:

150 МБ/день×28 дней=4200 МБ

Сжатый вес новых данных:

4200 МБ×0.4=1680 МБ

Общий вес четвертой резервной копии:

4.4 МБ+1680 МБ=1684.4 МБ

Общий вес 4-х резервных копий:

424.4 МБ+844.4 МБ+1264.4 МБ+1684.4 МБ=4217.6 МБ

Перевод в ГБ:

4217.6 МБ÷1024 ≈ 4.12 ГБ

Теперь подсчитаем объем wal файлов на резервном сервере через месяц после работы. Данные очищаются раз в неделю, то есть одновременно на сервере будут только данные за 5 недель (в нашем случае за 31 день). Измененных данных за день 700Мб, по сути все эти 700Мб будут хранится в виде wal файлов, размер wal файла 16Мб, то есть если записывать целымы блоками, то к концу месяца на резервном сервере в виде wal файлов будет хранится

Итого на резервном сервере будет хранится 25.8 гибибайта данных в виде резервных копий, если на резервном сервере не разворачивался кластер из бекапа.

Теперь посчитаем объем резервных копий на основном сервере через месяц, но с учетом хранения данных только 1 неделю. Тогда в пике будет 2 резервные копии, то есть

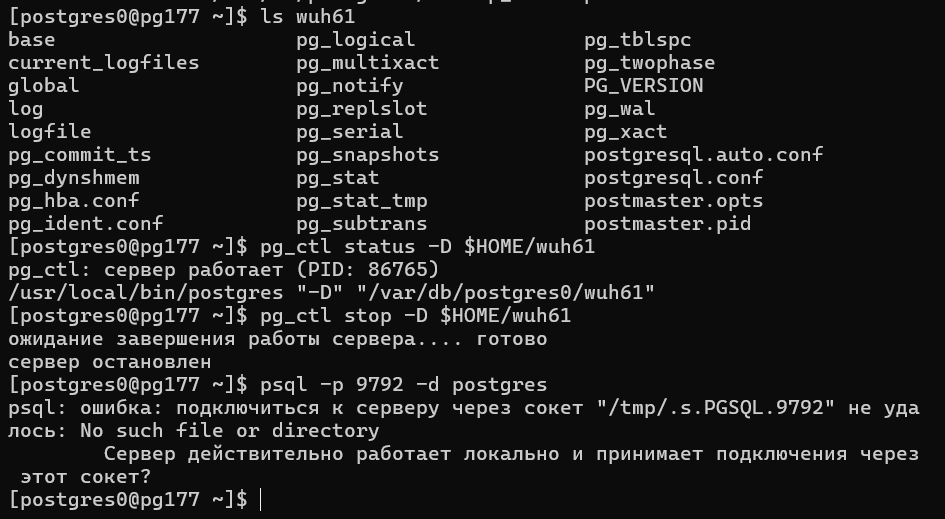
Объем WAL файлов составит

Итого к концу месяца скопится 12.7 гибибайта в виде резервных копий и WAL файлов.

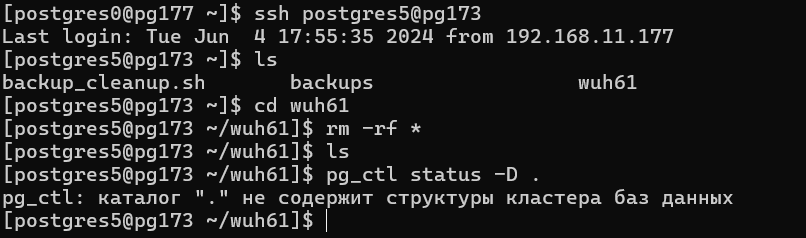
**2. Потеря основного узла**

Теперь, когда у нас есть копия WAL-файлов и basebackup на резервном узле, восстановим работу БД из этих копий.

Сначала сымитируем недоступность главного сервера, остановим его и попробуем подключится.

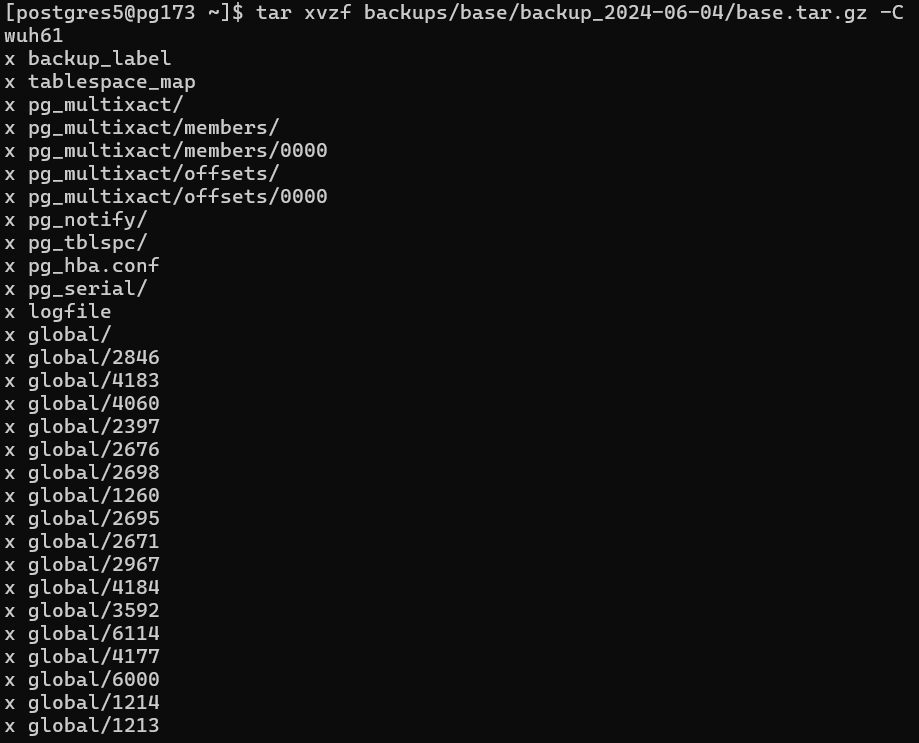


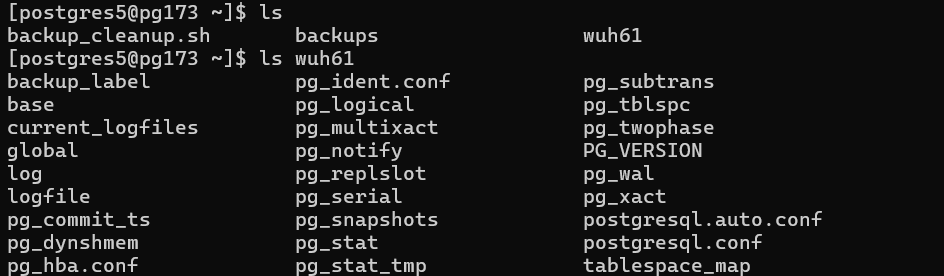
Теперь, когда сервер не доступен удалим все данные из папки wuh61, в которой лежат данные кластера.



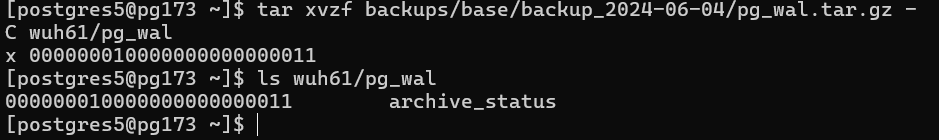
Теперь начнем восстановление файлов из pg\_basebackup.

Разархивируем базовые файлы из архива base последней резервной копии:

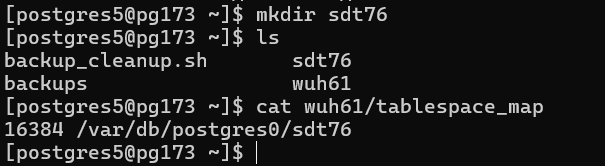


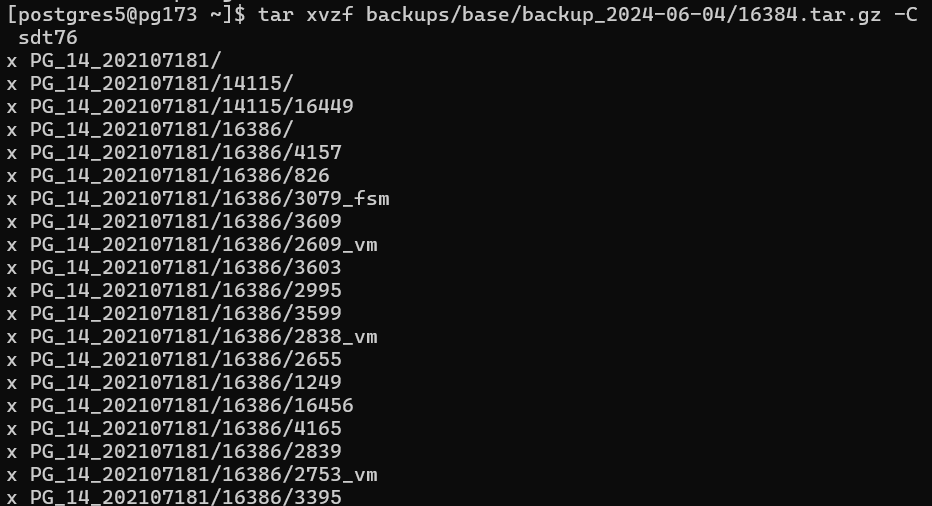


Далее разархивируем содержание архива pg\_wal в соответствующую директрию.

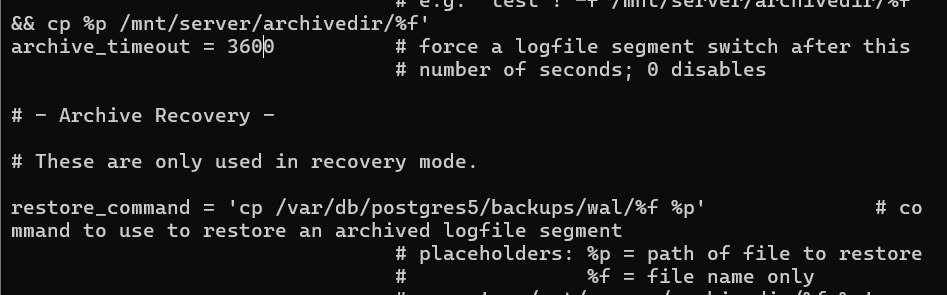


Далее создадим директории для табличных пространств, такие же как на основном сервере и поправим содержимое файла tablespace\_map, чтобы он содержал правильные пути. Далее разархивируем табличные пространства в свои папки

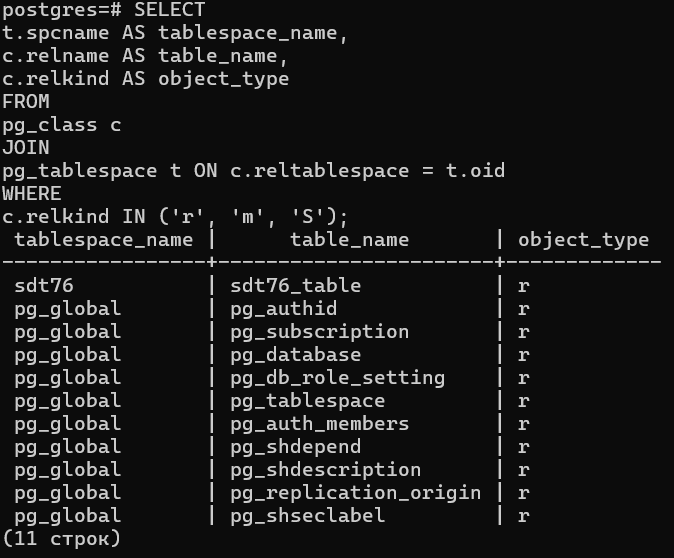




В конфиге postgresql.conf необходимо настроить параметры restore\_command и archive\_command в соответствии с новой структурой директорий на резервном узле:

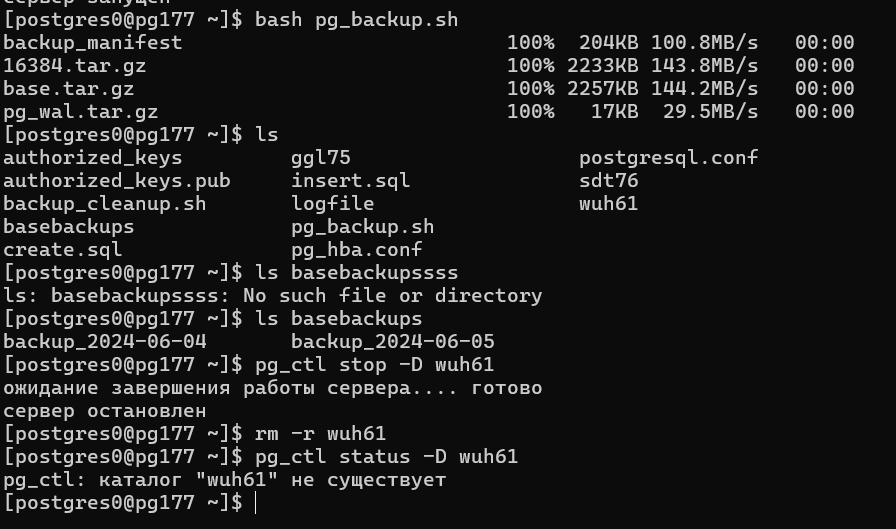


Теперь запустим сервер БД и попробуем подключиться

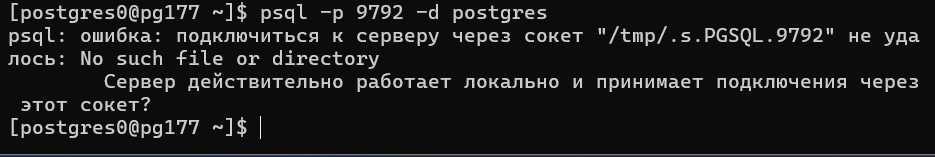


**3. Повреждение файлов БД**

Симулируем сбой – удалим с диска директорию конфигурационных файлов БД со всем содержимым:

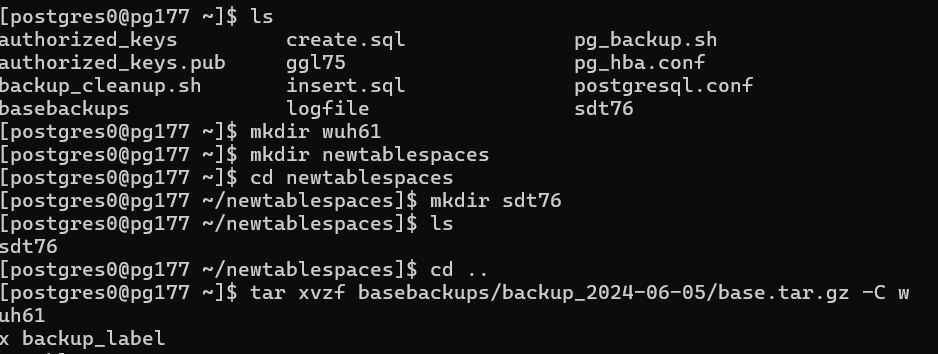


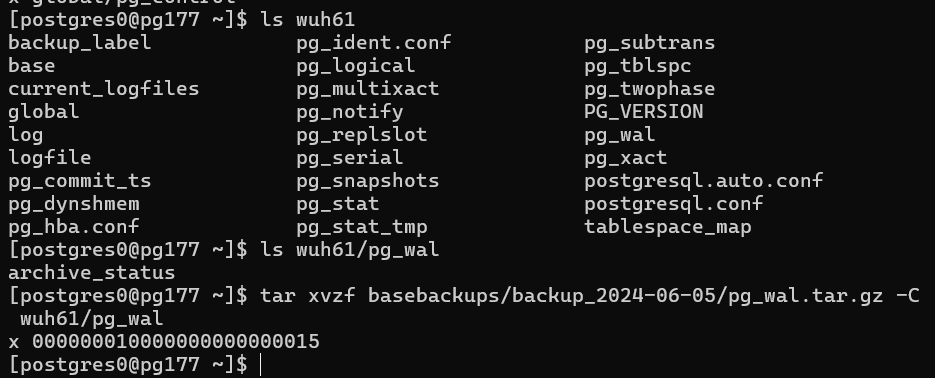
Попробуем узнать статус сервера и подключиться к нему:



Как мы видим, сервер “упал” и не принимает новых подключений.

Начнем восстановление аналогично пункту 2. В условии указано, что директория доп. табличных пространств недоступна, поэтому нам также необходимо будет создать новую директорию, в которую мы разархивируем доп. табличное пространство из копии pg\_basebackup:

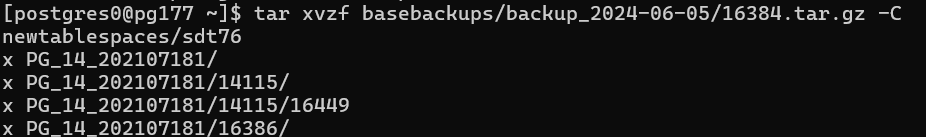




Файлы успешно разархивированы из копии. Редактируем tablespace\_map:

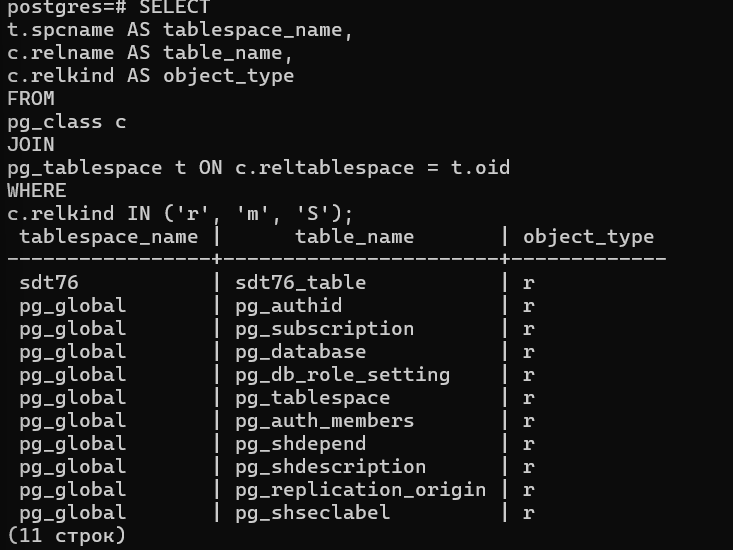
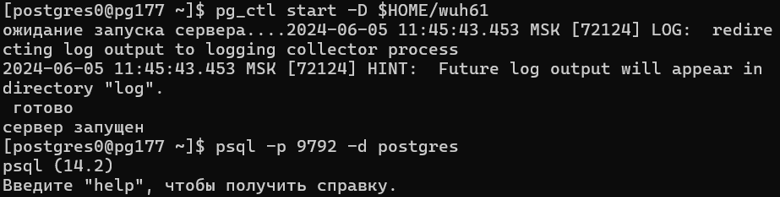


Далее разархивируем файлы табличных пространств в новую папку

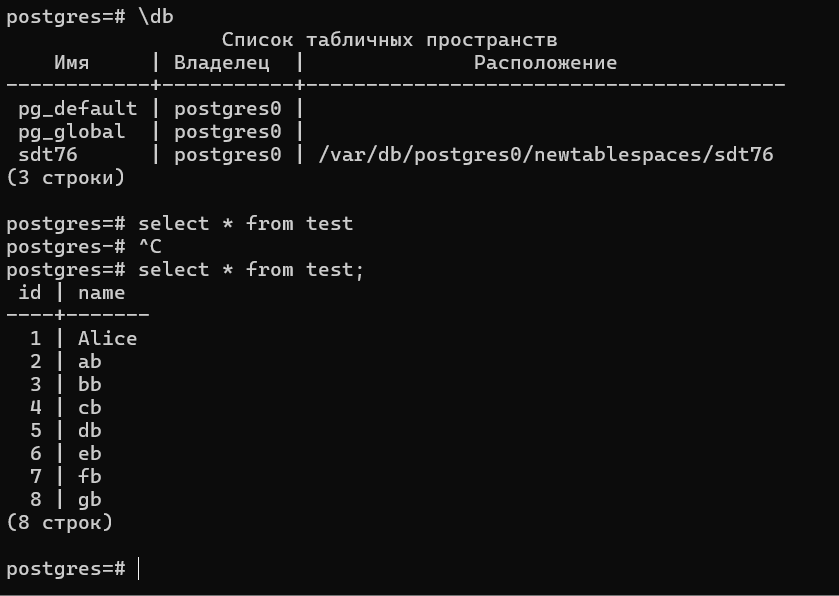


В этом случае нам не нужно изменять команды в postgresql.conf, т.к. структура директорий осталась прежней, и, соответственно, путь к директории wal\_archive не изменился.

Попробуем запустить сервер и подключится, после чего проверить наличие таблиц и пути к табличным пространствам, а также наличие в них данных.



Как мы видим, postgres восстановился из WAL-файлов, и тестовые данные доступны. Проверим новые пути к табличным пространствам

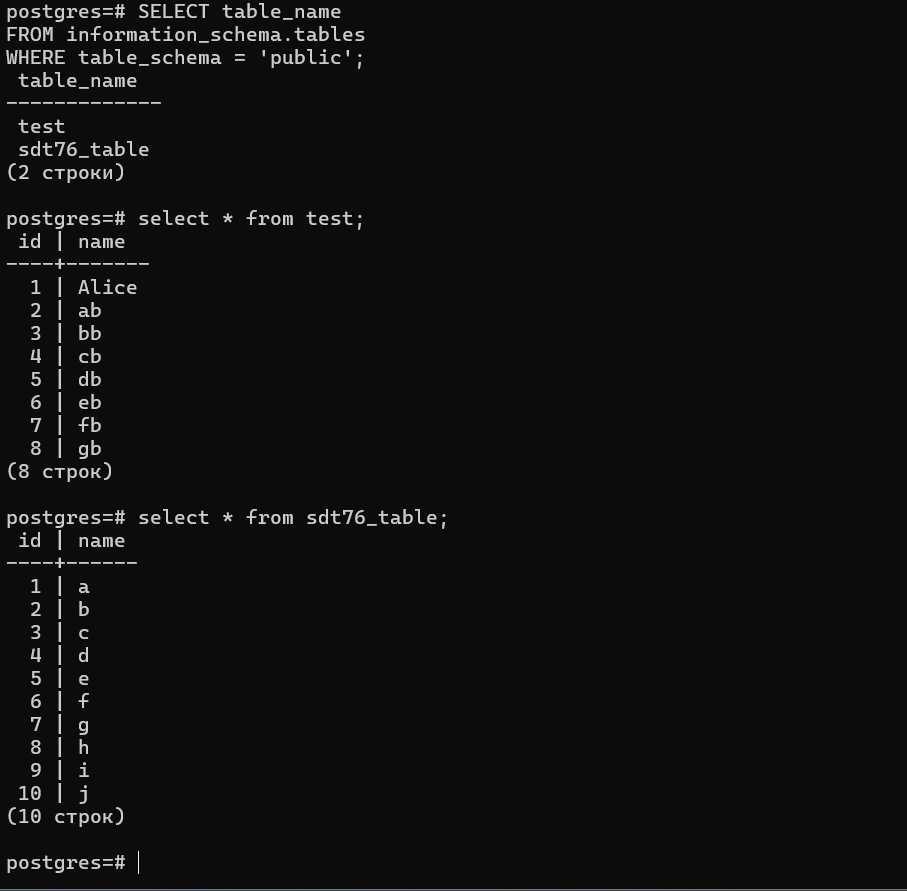


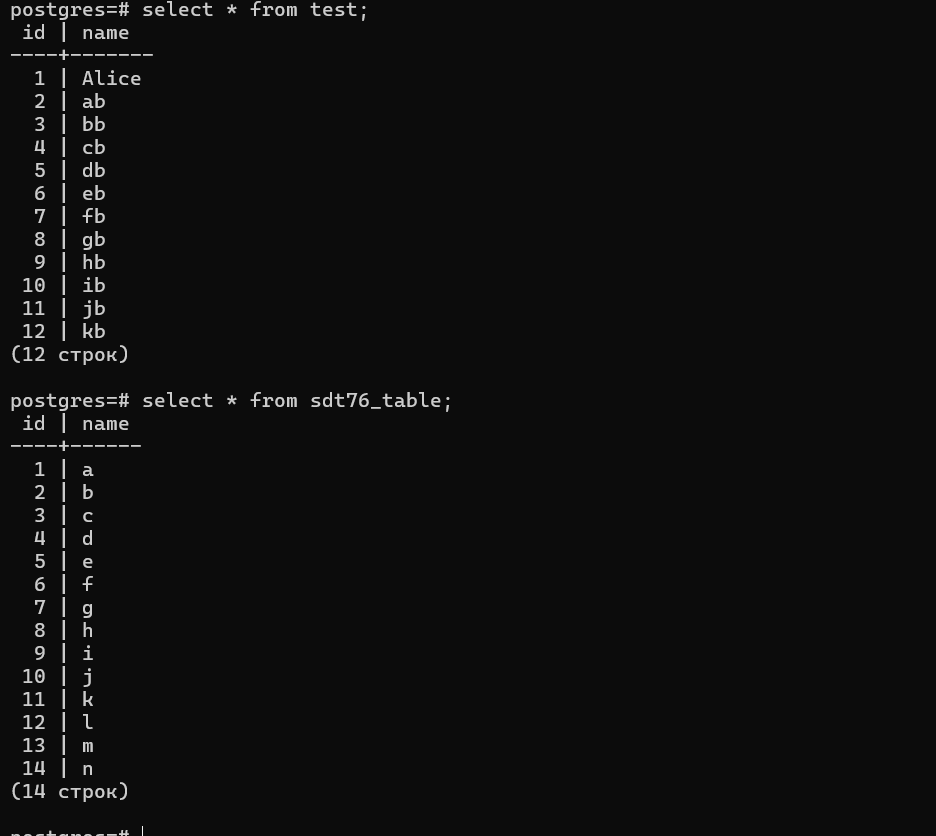
**4. Логическое повреждение данных**

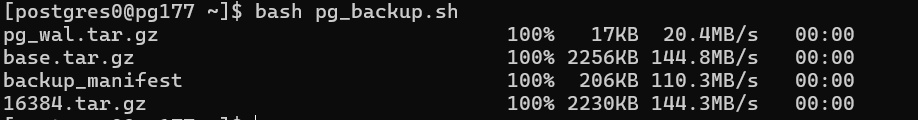
По варианту данные необходимо восстановить следующим образом:

* **Генерация файла на резервном узле с помощью pg\_dump и последующее применение файла на основном узле.**

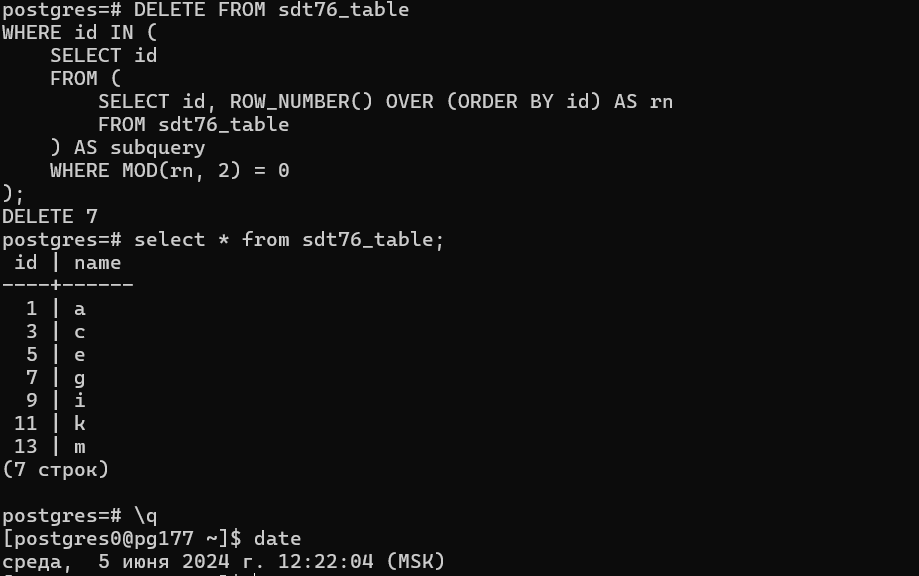
Для этого сначала в схеме public (дефолтная схема) создадим новые строки, после чего сделаем резервную копию и почистим директорию wal на резервном сервере, чтобы увидеть активность архивирования.





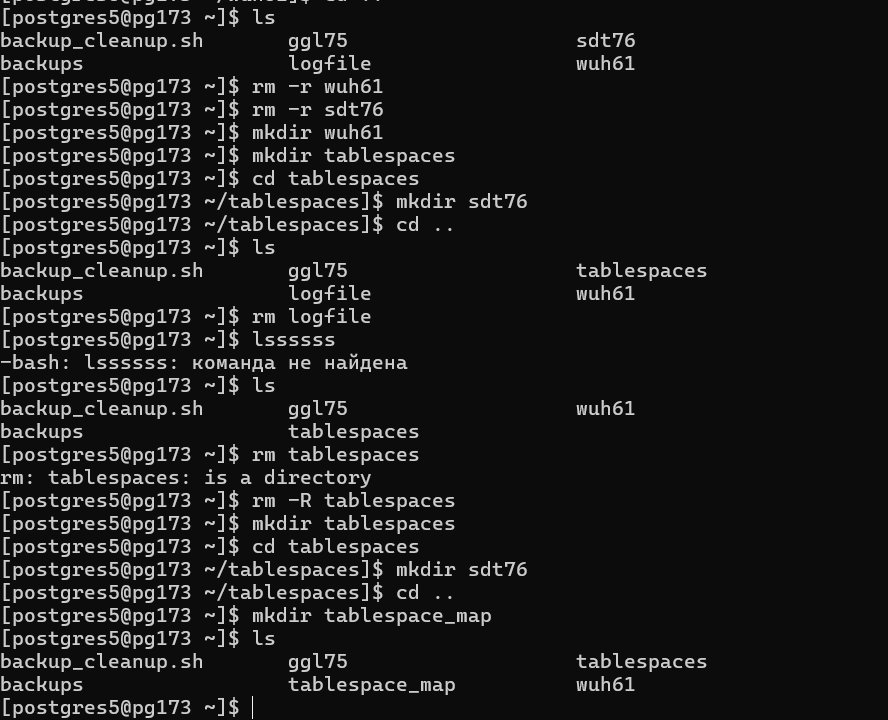


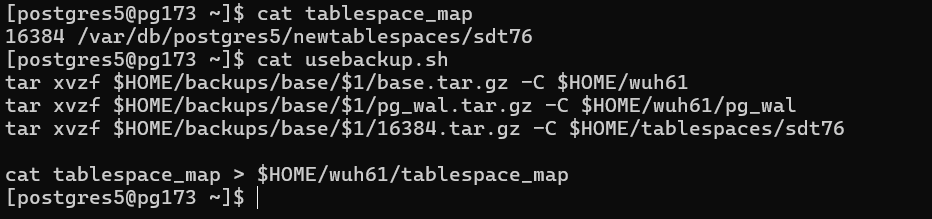
Далее зафиксируем время и затрем новые данные (симулируем случайное редактирование с помощью update)

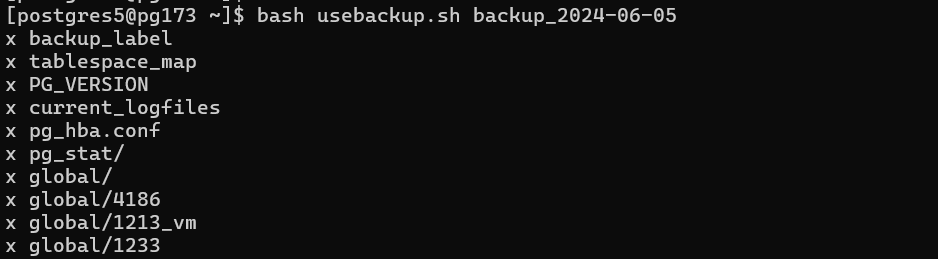


Далее на резервном узле восстановим кластер без дублирования новых wal, сделаем дамп и отправим на главный узел.

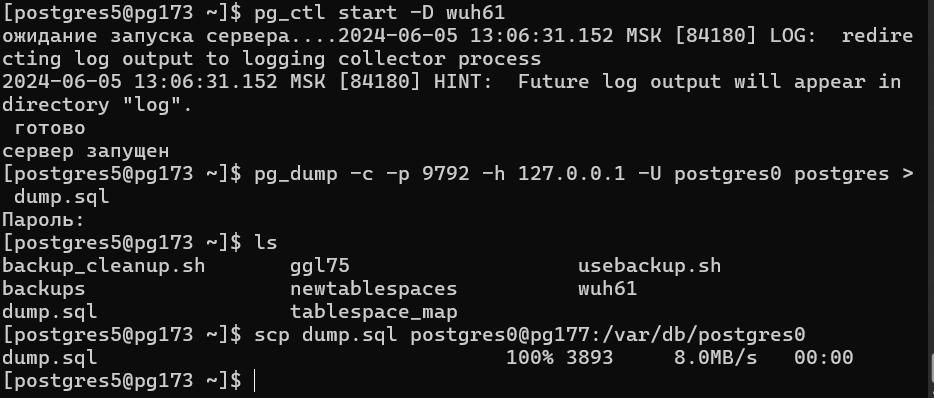
Для этого создадим папки под кластер и под табличные пространства, после чего прогоним скрипт.



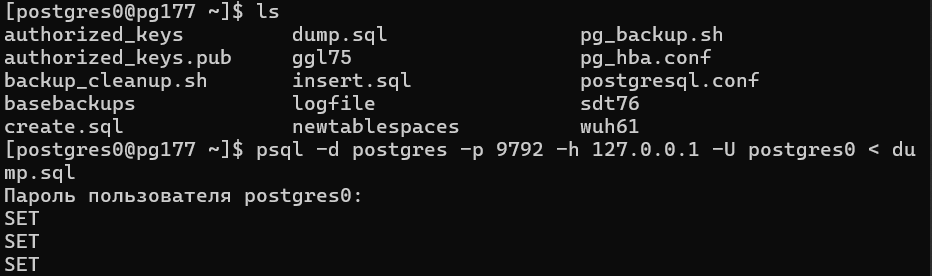




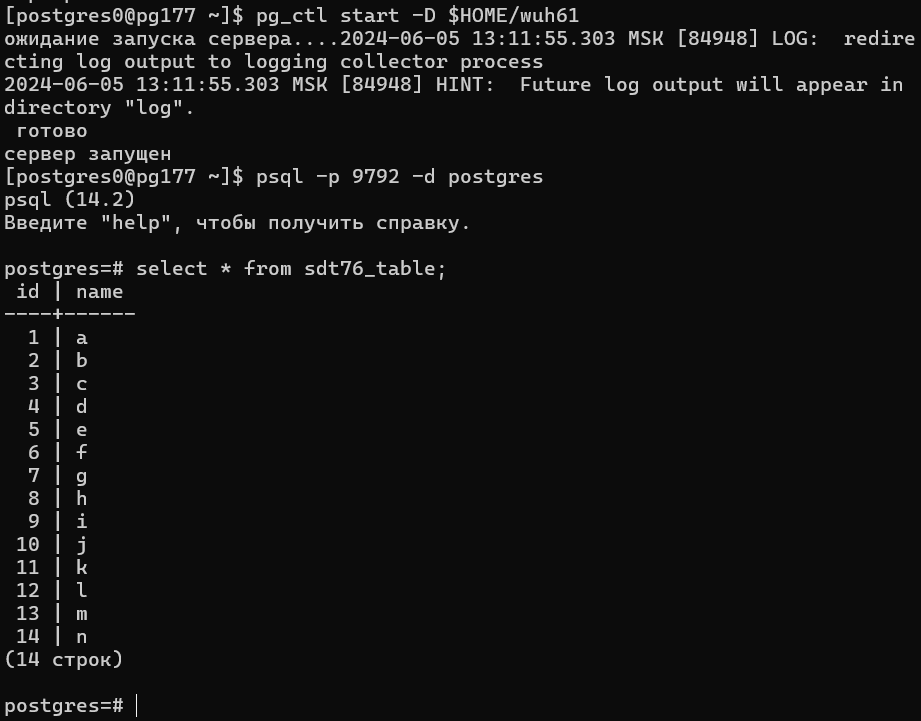
Далее запустим кластер и сделаем дамп, после чего отправим дамп на основной узел



Далее применим дамп



Подключимся и проверим данные.



Данные восстановились. Я обработал почти «идеальный» случай, когда дамп был сделан сразу после внесения новых данных, но до их порчи.