

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

Лабораторная работа №3  
по дисциплине  
«Распределённые системы хранения данных»

Вариант - 38

Выполнил: студент группы Р33302  
Сабитов Д.Т.

Преподаватель:  
Шешуков Д. М.

Санкт-Петербург  
2023 г.

## **Задание:**

Лабораторная работа включает настройку резервного копирования данных с основного узла на резервный, а также несколько сценариев восстановления. Узел из предыдущей лабораторной работы используется в качестве основного; новый узел используется в качестве резервного. В сценариях восстановления необходимо использовать копию данных, полученную на первом этапе данной лабораторной работы.

## **Этапы работы:**

### **1. Резервное копирование**

1.1 Настроить резервное копирование с основного узла на резервный следующим образом:

Периодические обособленные (standalone) полные копии.

Полное резервное копирование (pg\_basebackup) по расписанию (cron) два раза в сутки.

Необходимые файлы WAL должны быть в составе полной копии, отдельно их не архивировать.

Срок хранения копий на основной системе - 1 неделя, на резервной - 1 месяц. По истечении срока хранения, старые архивы должны автоматически уничтожаться.

1.2 Подсчитать, каков будет объем резервных копий спустя месяц работы системы, исходя из следующих условий:

Средний объем измененных данных за сутки: ~350 МБ.

1.3 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:

3.1 Симулировать сбой:

удалить с диска директорию любого табличного пространства со всем содержимым.

3.2 Проверить работу СУБД, доступность данных, перезапустить СУБД, проанализировать результаты.

3.3 Выполнить восстановление данных из резервной копии, учитывая следующее условие: Исходное расположение дополнительных табличных пространств недоступно - разместить в другой директории и скорректировать конфигурацию.

3.4 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:

4.1 В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.

4.2 Зафиксировать время и симулировать ошибку:

В любой таблице с внешними ключами изменить внешние ключи случайным образом (INSERT, UPDATE)

4.3 Продемонстрировать результат.

4.4 Выполнить восстановление данных указанным способом.

4.5 Продемонстрировать и проанализировать результат.

## Выполнение:

### Создадим суперпользователя для тестов:

```
CREATE ROLE admin SUPERUSER CREATEDB CREATEROLE LOGIN PASSWORD 'admin';
```

### 1. Резервное копирование

Создание РК будет производиться следующим скриптом, который в свою очередь будет запускаться каждый день в 11 часов дня и вечера по расписанию:

Основной узел:

```
#!/usr/local/bin/bash

BACKUPS_DIR=$HOME/backups/
CURRENT_DATE=$(date "+%Y-%m-%d-%H:%M:%S")
BACKUP_DIR=$BACKUPS_DIR/BACKUP_${CURRENT_DATE}
pg_basebackup -h 127.0.0.1 -p 9056 -U postgres1 -D $BACKUP_DIR

scp -r $BACKUP_DIR postgres1@pg130:~/backups/

find $BACKUPS_DIR -name "*" -type d -mtime +7 -mindepth 1 -maxdepth 1 -exec rm -rf {} \;
ssh postgres1@pg130 "bash ~/remove_script.sh"
```

Резервный узел:

```
#!/usr/local/bin/bash

BACKUPS_DIR=~/.backups/
find $BACKUPS_DIR -name "*" -type d -mtime +31 -mindepth 1 -maxdepth 1 -exec rm -rf {} \;
```

На основном узле создадим стон-файл через команду (crontab -e), в котором опишем правило для запуска нашего скрипта два раза в сутки (в 11:00 и 23:00). Проверим список запланированных задач.

```
[postgres1@pg101 ~]$ crontab -l
0 0 11/12 * * ? /var/db/postgres1/313318/script.sh >> log_backup.log
```

### Проверим работу скрипта

```
[postgres1@pg101 ~/313318]$ bash ~/313318/script.sh
pg_ident.conf          100% 1636      6.4MB/s  00:00
0000000100000000000000 100% 16MB 147.8MB/s 00:00
0000                   100% 8192      24.9MB/s  00:00
postgres1.conf         100% 28KB  54.0MB/s  00:00
pg_hba.conf            100% 124    798.4KB/s  00:00
2680                   100% 8192      31.1MB/s  00:00
2619                   100% 152KB 112.7MB/s  00:00
2607_vm                100% 8192      31.8MB/s  00:00
2613                   100% 0        0.0KB/s    00:00
2664                   100% 16KB   39.0MB/s  00:00
2619_fsm               100% 24KB   48.8MB/s  00:00
2830                   100% 0        0.0KB/s    00:00
4166                   100% 8192      26.6MB/s  00:00
2836_vm                100% 8192      24.8MB/s  00:00
1249_fsm               100% 24KB   47.0MB/s  00:00
2617_fsm               100% 24KB   54.4MB/s  00:00
```

Резервная копия создавалась:

```
[postgres1@pg130 ~/backups]$ ls
BACKUP_2023-06-05-15:56:34      BACKUP_2023-06-05-16:01:54      BACKUP_2023-06-05-16:07:27
BACKUP_2023-06-05-16:00:12      BACKUP_2023-06-05-16:05:25
[postgres1@pg130 ~/backups]$
```

### Расчет объема:

Размер одного бэкапа (изначально):

```
7,2М      backups/BACKUP_2023-06-05-16:18:12
```

Одновременно на основном узле могут храниться копии, время жизни которых меньше недели. В таком случае перед самой «чисткой» будут копии за 7 дней, которые делались 2 раза в день. Также нужно учитывать, что создаются новые wal файлы по 16 МБ каждый.

С помощью арифметической прогрессии вычислим объем на основном узле:

$$a_1 = 7.2 \text{ МБ} \quad d = 350 \text{ МБ} \quad n = 7 * 2 = 14$$

$$S = \frac{(16+7.2+350)}{2} * 14 = 2612 \text{ МБ} = \mathbf{2.6 \text{ ГБ}}$$

Для резервного узла:

$$n = 30 * 2 = 60$$

$$S = \frac{(16+7.2+350)}{2} * 60 = 11196 \text{ МБ} = \mathbf{11.2 \text{ ГБ}}$$

## 2. Потеря основного узла

Для демонстрации создания резервной копии будет создана база данных lazybox (из предыдущей работы)

```
postgres=# create database lazybox with template = template1;
CREATE DATABASE
postgres=#
```

Выполним следующий запрос:

```
CREATE TABLE TBL (
    num serial PRIMARY KEY,
    field varchar
);
insert into TBL (field) values ('1');
insert into TBL (field) values ('2');
insert into TBL (field) values ('3');
```

В итоге будут созданы следующие таблицы:

```
lazybox=# \d
```

| Список отношений |             |                    |           |
|------------------|-------------|--------------------|-----------|
| Схема            | Имя         | Тип                | Владелец  |
| public           | tbl         | таблица            | postgres1 |
| public           | tbl_num_seq | последовательность | postgres1 |

(2 строки)

После создания резервной копии на запасном узле появится директория “BACKUP\_2023\_05\_28\_23\_25\_58” со следующим содержимым:

```
[postgres1@pg130 ~/backups]$ ls BACKUP_2023-06-05-17:33:43
backup_label      pg_dynshmem      pg_notify        pg_stat_tmp      pg_wal
backup_manifest   pg_hba.conf      pg_replslot     pg_subtrans      pg_xact
base              pg_ident.conf    pg_serial        pg_tblspc        postgresql.auto.conf
global            pg_logical        pg_snapshots     pg_twophase      postgresql.conf
pg_commit_ts      pg_multixact     pg_stat          PG_VERSION
```

Чтобы восстановить работу СУБД на резервном узле для начала нужно выполнить следующий скрипт:

```
#!/usr/local/bin/bash

PGDATA=$HOME/u01/dir2
export PGDATA
mkdir -p $PGDATA
chmod 700 $PGDATA

WALBUFFS=$HOME/u07/dir7
export WALBUFFS
mkdir -p $WALBUFFS

rm -rf $PGDATA/*
cp -r ~/backups/MAIN_BACKUP/ $PGDATA
touch $PGDATA/recovery.signal
```

предварительно выделив наиболее актуальную копию (MAIN\_BACKUP)

После запуска скрипта добавились 2 директории для \$PGDATA и WAL’ов, куда их потом перенесёт кто-то при восстановлении.

Также содержимое основной копии подменило содержимое \$PGDATA. Помимо этого, сами скопированные WAL’ы также поместились в \$PGDATA в стандартную поддиректорию pg\_wal

```
[postgres1@pg130 ~]$ ls
313318      backups      remove_script.sh  u01      u07
[postgres1@pg130 ~]$

[postgres1@pg130 ~/u01/dir2]$ ls
backup_label      pg_dynshmem      pg_notify        pg_stat_tmp      pg_wal
backup_manifest   pg_hba.conf      pg_replslot     pg_subtrans      pg_xact
base              pg_ident.conf    pg_serial        pg_tblspc        postgresql.auto.conf
global            pg_logical        pg_snapshots     pg_twophase      postgresql.conf
pg_commit_ts      pg_multixact     pg_stat          PG_VERSION      recovery.signal
[postgres1@pg130 ~/u01/dir2]$
```

Дополнительно к копированию в «рабочем» пространстве заранее был создан файл recovery.signal, который является флагом на восстановление работы СУБД из РК.

После этого осталось указать директорию для WAL'ов в конфигурационном файле для того, чтобы они туда скопировались:

```
restore_command = 'cp $PGDATA/pg_wal/<> $WALBUFFS/'
```

 указать название файла

После подстановки в postgresql.conf будет доступна запись

```
restore_command = 'cp $PGDATA/pg_wal/00000000100000000000000014 $WALBUFFS'
```

Далее после запуска postgres данные и WAL'ы «восстановятся», а файл recovery.signal исчезнет

```
[postgres1@pg130 ~/u01/dir2]$ cd
[postgres1@pg130 ~]$ postgres -h 127.0.0.1 -p 9056 -c password_encryption=md5
2023-06-13 16:10:51.047 MSK [25367] СООБЩЕНИЕ: завершение вывода в stderr
2023-06-13 16:10:51.047 MSK [25367] ПОДСКАЗКА: В дальнейшем протокол будет выводиться в "syslog".

^C[postgres1@pg130 ~]$ ls -la
total 37
drwx----- 7 postgres postgres 9 5 июня 17:36 .
drwxr-xr-x 22 root wheel 25 9 июня 04:03 ..
-rw----- 1 postgres postgres 3495 5 июня 18:57 .bash_history
drwx----- 2 postgres postgres 3 29 мая 07:03 .ssh
drwxr-xr-x 2 postgres postgres 2 5 июня 14:52 313318
drwxr-xr-x 8 postgres postgres 8 5 июня 18:40 backups
-rw-r--r-- 1 postgres postgres 140 5 июня 17:02 remove_script.sh
drwxr-xr-x 3 postgres postgres 3 5 июня 17:36 u01
drwxr-xr-x 3 postgres postgres 3 5 июня 17:36 u07
[postgres1@pg130 ~]$ ls -l $PGDATA
total 85
-rw----- 1 postgres postgres 227 13 июня 16:09 backup_label.old
-rw----- 1 postgres postgres 181742 13 июня 16:09 backup_manifest
drwx----- 6 postgres postgres 6 13 июня 16:09 base
drwx----- 2 postgres postgres 59 13 июня 16:09 global
drwx----- 2 postgres postgres 2 13 июня 16:09 pg_commit_ts
drwx----- 2 postgres postgres 2 13 июня 16:09 pg_dynshmem
-rw----- 1 postgres postgres 124 13 июня 16:09 pg_hba.conf
-rw----- 1 postgres postgres 1636 13 июня 16:09 pg_ident.conf
drwx----- 4 postgres postgres 5 13 июня 16:11 pg_logical
drwx----- 4 postgres postgres 4 13 июня 16:09 pg_multixact
drwx----- 2 postgres postgres 2 13 июня 16:09 pg_notify
drwx----- 2 postgres postgres 2 13 июня 16:09 pg_replslot
drwx----- 2 postgres postgres 2 13 июня 16:09 pg_serial
drwx----- 2 postgres postgres 2 13 июня 16:09 pg_snapshots
drwx----- 2 postgres postgres 3 13 июня 16:11 pg_stat
drwx----- 2 postgres postgres 2 13 июня 16:11 pg_stat_tmp
drwx----- 2 postgres postgres 3 13 июня 16:10 pg_subtrans
drwx----- 2 postgres postgres 2 13 июня 16:09 pg_tblspc
drwx----- 2 postgres postgres 2 13 июня 16:09 pg_twophase
-rw----- 1 postgres postgres 3 13 июня 16:09 PG_VERSION
drwx----- 3 postgres postgres 6 13 июня 16:10 pg_wal
drwx----- 2 postgres postgres 3 13 июня 16:09 pg_xact
-rw----- 1 postgres postgres 88 13 июня 16:09 postgresql.auto.conf
-rw----- 1 postgres postgres 28904 13 июня 16:10 postgresql.conf
-rw----- 1 postgres postgres 84 13 июня 16:10 postmaster.opts
[postgres1@pg130 ~]$
```

Теперь можно запустить СУБД на резервном узле и проверить данные:

```
[postgres1@pg130 ~]$ psql -h 127.0.0.1 -p 9056 -d lazybox -U postgres1
Пароль пользователя postgres1:
psql (14.2)
Введите "help", чтобы получить справку.

lazybox=# \d

```

| Список отношений |             |                    |           |
|------------------|-------------|--------------------|-----------|
| Схема            | Имя         | Тип                | Владелец  |
| public           | tbl         | таблица            | postgres1 |
| public           | tbl_num_seq | последовательность | postgres1 |

```
(2 строки)

lazybox=#
```

### 3. Повреждение файлов БД

При запуске копии на основном узле при сбое алгоритм действий примерно такой же за исключением того, что в PGDATA и другие метaperеменные прописываются новые значения. Т. е. нужно только перенести копию с резервного узла на основной, а потом сделать всё так же, как и в п. 2 уже там. Удаляемая таблица находится здесь

Удаляемая таблица:

```
lazybox=# select 'tbl' ::regclass::oid;
 oid
-----
 16396
(1 строка)
```

```
[postgres1@pg101 ~/u01/dir2/base/16393]$ rm -rf 16396
```

```
[postgres1@pg101 ~]$ psql -h localhost -p 9056 -U postgres1 lazybox
Пароль пользователя postgres1:
psql (14.2)
Введите "help", чтобы получить справку.

lazybox=# \d

```

| Список отношений |             |                    |           |
|------------------|-------------|--------------------|-----------|
| Схема            | Имя         | Тип                | Владелец  |
| public           | tbl         | таблица            | postgres1 |
| public           | tbl_num_seq | последовательность | postgres1 |

```
(2 строки)

lazybox=# select * from tbl;
ОШИБКА: не удалось открыть файл "base/16393/16396": No such file or directory
lazybox=#
```

Переносим копию на основной узел и немного меняем скрипт для

```
[postgres1@pg130 ~]$ scp -r backups/BACKUP_2023-06-05-17:33:43/ postgres1@pg101:~/
```

```
#!/usr/local/bin/bash
```

```
PGDATA=$HOME/u01/dir2
```

```
export PGDATA
```

```
mkdir -p $PGDATA
```

```
chmod 700 $PGDATA
```

```
WALBUFFS=$HOME/u07/dir7
```

```
export WALBUFFS
```

```
mkdir -p $WALBUFFS
```

```
rm -rf $PGDATA/*
```

```
cp -r MAIN_BACKUP/ $PGDATA
```

```
touch $PGDATA/recovery.signal
```

```
[postgres1@pg101 ~]$ echo $PGDATA
```

```
/var/db/postgres1/u01/dir2
```

```
[postgres1@pg101 ~]$ ls $PGDATA
```

|                  |               |              |             |                      |
|------------------|---------------|--------------|-------------|----------------------|
| backup_label.old | pg_dynshmem   | pg_notify    | pg_stat_tmp | pg_wal               |
| backup_manifest  | pg_hba.conf   | pg_replslot  | pg_subtrans | pg_xact              |
| base             | pg_ident.conf | pg_serial    | pg_tblspc   | postgresql.auto.conf |
| global           | pg_logical    | pg_snapshots | pg_twophase | postgresql.conf      |
| pg_commit_ts     | pg_multixact  | pg_stat      | PG_VERSION  | recovery.signal      |

Далее перезапускаем сервер:

```
[postgres1@pg101 ~]$ postgres -h 127.0.0.1 -p 9056 -c password_encryption=md5
```

```
[postgres1@pg101 ~]$ psql -h 127.0.0.1 -p 9056 -d lazybox -U postgres1
```

Пароль пользователя postgres1:

```
psql (14.2)
```

Введите "help", чтобы получить справку.

```
lazybox=# select * from tbl;
```

```
 num | field
```

```
-----+-----
```

```
  1  |  1
```

```
  2  |  2
```

```
  3  |  3
```

```
(3 строки)
```

Работает!

#### 4. Логическое повреждение данных

Сначала БД будет наполнена новыми данными, с которыми потом будут производиться неаккуратные действия.



```

lazybox=# \d

```

| Схема  | Имя            | Тип                | Владелец  |
|--------|----------------|--------------------|-----------|
| public | newone         | таблица            | postgres1 |
| public | newone_num_seq | последовательность | postgres1 |
| public | tbl            | таблица            | postgres1 |
| public | tbl_num_seq    | последовательность | postgres1 |

(4 строки)

```

lazybox=# select * from newone;
 num | field
-----+-----
   1 | Danil
   2 | Tim
(2 строки)

```

После этого сделаем РК и отправим её на резервный узел

```

[postgres1@pg101 ~/313318]$ bash ~/313318/script.sh >> log.log

```

```

[postgres1@pg130 ~]$ cd backups
[postgres1@pg130 ~/backups]$ ls
BACKUP_2023-06-05-16:18:12  BACKUP_2023-06-05-17:05:16  BACKUP_2023-06-05-17:33:43  BACKUP_2023-06-05-17:05:39
BACKUP_2023-06-05-16:38:59  BACKUP_2023-06-05-17:07:46  BACKUP_2023-06-05-18:40:09
[postgres1@pg130 ~/backups]$

```

Далее после запуска на резервном узле появятся следующие данные и та самая пока не «испорченная» БД (данные) на которой будет сделан дамп

```

[postgres1@pg130 ~]$ psql -h 127.0.0.1 -p 9056 -d lazybox -U postgres1
Пароль пользователя postgres1:
psql (14.2)
Введите "help", чтобы получить справку.

```

```

lazybox=# \d

```

| Схема  | Имя            | Тип                | Владелец  |
|--------|----------------|--------------------|-----------|
| public | newone         | таблица            | postgres1 |
| public | newone_num_seq | последовательность | postgres1 |
| public | tbl            | таблица            | postgres1 |
| public | tbl_num_seq    | последовательность | postgres1 |

(4 строки)

```

lazybox=# select * from newone;
 num | field
-----+-----
   1 | Danil
   2 | Tim
(2 строки)

```

Далее делается сам дамп БД lazybox, которая будет испорчена на основном узле

```
[postgres1@pg130 ~]$ pg_dump -h 127.0.0.1 -p 9056 -U postgres1 -f ~/dump_file lazybox
Пароль:
[postgres1@pg130 ~]$ ls
313318          dump_file          u01
backups         remove_script.sh   u07
[postgres1@pg130 ~]$
```

Удаляем таблицу:

```
[postgres1@pg101 ~]$ psql -h 127.0.0.1 -p 9056 -d lazybox -U postgres1
Пароль пользователя postgres1:
psql (14.2)
Введите "help", чтобы получить справку.

lazybox=# \d
          Список отношений
 Схема |      Имя      | Тип      | Владелец
-----+-----+-----+-----
 public | newone         | таблица  | postgres1
 public | newone_num_seq | последовательность | postgres1
 public | tbl            | таблица  | postgres1
 public | tbl_num_seq    | последовательность | postgres1
(4 строки)

lazybox=# drop table tbl;
DROP TABLE
lazybox=# \d
          Список отношений
 Схема |      Имя      | Тип      | Владелец
-----+-----+-----+-----
 public | newone         | таблица  | postgres1
 public | newone_num_seq | последовательность | postgres1
(2 строки)
```

Теперь нужно скопировать дамп с резервного узла и применить его на основном:

```
[postgres1@pg130 ~]$ scp dump_file postgres1@pg101:~/
Password for postgres1@pg101.cs.ifmo.ru:
dump_file                               100% 3061    6.2MB/s   00:00
```

Далее восстанавливаем данные из файла в текстовом формате

```
[postgres1@pg101 ~]$ psql -h 127.0.0.1 -p 9056 -d lazybox < ~/dump_file
Пароль пользователя postgres1:
SET
SET
SET
SET
SET
SET
  set_config
-----
(1 строка)

SET
SET
SET
SET
SET
SET
: "newone"
ALTER TABLE
: "newone_num_seq"
ALTER TABLE
ALTER SEQUENCE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
ALTER TABLE
ALTER TABLE
: "newone_pkey"
ПОДРОБНОСТИ:  "(num)=(1)"
КОНТЕКСТ: COPY newone, 1
COPY 3
setval
```

Видно, что таблица опять появилась с теми же данными

```
[postgres1@pg101 ~]$ psql -h 127.0.0.1 -p 9056 -d lazybox -U postgres1
Пароль пользователя postgres1:
psql (14.2)
Введите "help", чтобы получить справку.
```

```
lazybox=# \d
```

| Список отношений |                |                    |           |
|------------------|----------------|--------------------|-----------|
| Схема            | Имя            | Тип                | Владелец  |
| public           | newone         | таблица            | postgres1 |
| public           | newone_num_seq | последовательность | postgres1 |
| public           | tbl            | таблица            | postgres1 |
| public           | tbl_num_seq    | последовательность | postgres1 |

(4 строки)

```
lazybox=# select * from tbl;
```

| num | field |
|-----|-------|
| 1   | 1     |
| 2   | 2     |
| 3   | 3     |

(3 строки)

```
lazybox=#
```

## Выводы

В результате выполнения работы были изучены различные способы создания резервных копий. Помимо этого, было сделано несколько резервных копий с помощью физического и логического резервного копирования, после чего было осуществлено восстановление данных и работы СУБД, используя эти резервные копии