

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Распределенные системы хранения данных»

### **Лабораторная работа №3**

*Вариант 47*

Студент

*Кузнецов М. А.*

*P33131*

Преподаватель

*Шешуков Д. М.*

Санкт-Петербург, 2023 г.

# Описание задания

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.

## Требования к отчёту

Отчет должен быть самостоятельным документом (без ссылок на внешние ресурсы), содержать всю последовательность команд, содержимое скриптов по каждому пункту задания. Для демонстрации результатов приводить команду вместе с выводом (самой наглядной частью вывода, при необходимости).

### 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 новых данных в БД за сутки: ~650 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 конфигурационных файлов СУБД 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 директории PGDATA 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:  
Восстановление с использованием архивных WAL файлов.(СУБД должна работать в режиме архивирования WAL, потребуется задать параметры восстановления).  
Ход работы:  
4.1 В каждую таблицу базы добавить 2-3 новые строки, зафиксировать результат.  
4.2 Зафиксировать время и симулировать ошибку:  
Удалить любые две таблицы (DROP TABLE)  
4.3 Продемонстрировать результат.  
4.4 Выполнить восстановление данных указанным способом.  
4.5 Продемонстрировать и проанализировать результат.

## Этап 1. Резервное копирование

*Создаем пользователя с replicant*

```
create role replicant with replication login password '123';
```

*В файле postgresql.conf:*

```
wal_level = replica
```

*Подход -- создадим bash скрипт, в котором пропишем следующий код:*

```
#!/bin/bash

backup_dir="/var/db/postgres2/u06/dir6/backups"

current_datetime=$(date +%Y%m%d_%H%M')

backup_name="backup_${current_datetime}"

# Проверяем, существует ли директория backups
if [ ! -d "$backup_dir" ]; then

    mkdir -p "$backup_dir"

fi

# Создаем новую папку с именем вида "backup_20230522_1200"
new_backup_dir="$backup_dir/$backup_name"

mkdir "$new_backup_dir"

echo "Created backup directory: $new_backup_dir"

# Выполняем команду pg_basebackup и записываем резервную копию в новую папку
pg_basebackup --progress -p 9054 -U replicant --format=t --wal-method=fetch -D
"$new_backup_dir"

# Копируем
scp -r "$new_backup_dir" postgres2@pg132:backups/
```

Добавим в файл pg\_hba.conf также REPLICATION:

```
# TYPE DATABASE USER
host all,replication|
```

Для автоматического удаления:

```
0 12 * * * find /var/db/postgres2/u06/dir6/backups -type d -name "backup_*" -
mtime +7 -exec rm -rf {} \;/
```

*Также сделаем ssh ключ, чтобы не требовался пароль для отправки файлов на резервный узел.*

```
ssh-keygen -t rsa
```

```
ssh-copy-id -i ~/.ssh/id_rsa.pub postgres2@132
```

*Подсчет размера копий*

Бекап весит:

```
[postgres2@pg132 ~/backups]$ du -sh backup_20230522_1900/
291M    backup_20230522_1900/
[postgres2@pg132 ~/backups]$
```

$a1 = 291 \text{ Mb}$

$d = 650 \text{ Mb}$

$n = 30 \text{ days}$

$S_{30} = (2 \cdot 291 + 650 \cdot 29) / 2 \cdot 30 = 582966 \text{ Mb} = 569 \text{ GB}$

Инкрементальный подход будет эффективнее.

## Этап 2. Потеря основного узла

*Воссоздадим файловую структуру кластера для восстановления*

```
mkdir ~/u06/dir6
chmod 700 ~/u06/dir6
cd u06/dir6
```

```
tar xvf ~/backups/backup_*/base.tar
```

```
[postgres2@pg132 ~/u06/dir6]$ tar xvf ~/backups/backup_20230522_1637/base.tar
x backup_label
x tablespace_map
x postgres2@pg132/
x postgres2@pg132/16387.tar
x postgres2@pg132/base.tar
x base/
x base/1/
x base/1/2678
x base/1/2696
x base/1/2702
x base/1/2605
x base/1/2673_fsm
x base/1/3380
x base/1/2674_fsm
x base/1/2608_fsm
```

*Файлы табличного пространства*

```
mkdir ~/u06/dir7
```

```
cd ~/u06/dir7
```

```
tar xvf ~/backups/backup_*/16387.tar
```

```
[postgres2@pg132 ~/u06/dir7]$ tar xvf ~/backups/backup_20230522_1637/16387.tar
x PG_14_202107181/
x PG_14_202107181/16388/
x PG_14_202107181/16388/2619
x PG_14_202107181/16388/2680
x PG_14_202107181/16388/2613
x PG_14_202107181/16388/2664
x PG_14_202107181/16388/13938
```

*Отчищаем директорию для wal-файлов*

```
rm -rf ~/u06/dir6/pg_wal/*
```

```
ln -s /var/db/postgres2/u06/dir7 ~/u06/dir6/pg_tblspc/16387
```

```
[postgres2@pg132 ~/u06/dir6/pg_tblspc]$ ln -s /var/db/postgres2/u06/dir7 /var/db/postgres2/u06/dir6/
pg_tblspc/16387
[postgres2@pg132 ~/u06/dir6/pg_tblspc]$ ls
16387
[postgres2@pg132 ~/u06/dir6/pg_tblspc]$
```

*Указываем команду для загрузки wal-файлов*

```
restore_command = 'cp /var/db/postgres2/wal_archive/%f %p' # command to use to restore an archived logfile segment
# placeholders: %p = path of file to restore
#               %f = file name only
```

*Файл – флаг о восстановлении*

```
touch recovery.signal
```

*Запускаем кластер*

```
2023-05-28 17:33:09.319 MSK [99367] СООБЩЕНИЕ: передача вывода в протокол процессу сбора протоколов
2023-05-28 17:33:09.319 MSK [99367] ПОДСКАЗКА: В дальнейшем протоколы будут выводиться в каталог "log".
```

## Проверяем работоспособность

```
postgres=# select * from users;
 id |          name          |          email          | phone_number
-----+-----+-----+-----
  1 | Иванов Иван Иванович  | ivanov@mail.ru         | +79161234567
  2 | Петров Петр Петрович  | petrov@gmail.com       | +79162345678
  3 | Сидорова Анна Сергеевна | sidorova@yahoo.com    | +79163456789
  4 | Кузнецова Ольга Игоревна | kuznetsova@hotmail.com | +79164567890
  5 | Васильев Алексей Николаевич | vasiliev@yandex.ru    | +79165678901
  6 | Макаров Сергей Дмитриевич | makarov@mail.ru       | +79166789012
  7 | Тимофеева Наталья Валерьевна | timofeeva@gmail.com   | +79167890123
  8 | Андреева Елена Игоревна | andreeva@yahoo.com     | +79168901234
  9 | Козлов Артем Игоревич  | kozlov@hotmail.com     | +79169012345
 10 | Смирнова Екатерина Александровна | smirnova@yandex.ru    | +79160123456
(10 строк)
```

## Анализ выполнения

*Восстановление завершилось успешно. Но для корректного запуска необходимо подкорректировать ссылку*

## Повреждение файлов БД

*Последняя актуальная копия находится на узле, скачивать еще раз нет смысла.*

*Отчищаем директорию для wal-файлов*

```
rm -rf ~/u06/dir6/pg_wal/*
```

*Подключаемся к базе, которая была в пространстве*

```
~/u06/dir6]$ pg_ctl start
```

```
ожидание запуска сервера...2023-05-28 17:40:20.143 MSK [443] СООБЩЕНИЕ: передача вывода в протокол
процессу сбора протоколов
2023-05-28 17:40:20.143 MSK [443] ПОДСКАЗКА: В дальнейшем протоколы будут выводиться в каталог "log
".
прекращение ожидания
pg_ctl: не удалось запустить сервер
Изучите протокол выполнения.
```

Из-за того, что расположение директории PGDATA недоступно, распакуем копию в новое местоположение.

```
mkdir dir6_new
cd dir6_new/
scp -r postgres2@pg132:u06/dir6/ ~/u06/dir6_new/
```

```
[postgres2@pg110 ~/u06]$ scp -r postgres2@pg132:u06/dir6/* /var/db/postgres2/u06/dir6_new/
00000001000000000000000000000006      100% 16MB 107.1MB/s   00:00
00000001000000000000000000000015      100% 16MB 115.3MB/s   00:00
0000000100000000000000000000000D      100% 16MB 118.8MB/s   00:00
00000001000000000000000000000027      100% 16MB 114.9MB/s   00:00
00000001000000000000000000000012.00000028.backup      100% 341    9.6KB/s     00:00
00000001000000000000000000000022.00000028.backup      100% 341   828.8KB/s   00:00
00000001000000000000000000000020      100% 16MB  97.5MB/s   00:00
0000000100000000000000000000000C      100% 16MB 130.2MB/s   00:00
00000001000000000000000000000018      100% 16MB 143.0MB/s   00:00
00000001000000000000000000000012      100% 16MB 147.0MB/s   00:00
00000001000000000000000000000001      100% 16MB 131.0MB/s   00:00
```

Запускаем поврежденный кластер в режиме восстановления wal-файлов и указываем команду для восстановления

```
restore_command = 'cp /var/db/postgres2/u06/dir6_new/archives/%f %p'
```

Запускаем в режиме восстановления

```
touch recovery.signal
postgres -D ~/u06/dir6_new/
```

Проверяем работоспособность

```
[postgres2@pg110 ~]$ psql -p 9054 -d template1 -h pg110
psql (14.2)
Введите "help", чтобы получить справку.

template1=# select * from pg_tables;
      schemaname      |      tablename      | tableowner | tablespace | hasindexes | hasrules | ha
triggers | rowsecurity
-----+-----
pg_catalog | pg_statistic        | postgres2 |            | t          | f        | f
pg_catalog | pg_type             | postgres2 |            | t          | f        | f

template1=# select * from pg_tablespace;
 oid | spcname | spcowner | spcacl | spcoptions
-----+-----
 1663 | pg_default |      10 |      | 
 1664 | pg_global  |      10 |      | 
 16387 | newtablespace |      10 |      | 
(3 строки)

template1=# SELECT d.datname, t.spcname FROM pg_tablespace t JOIN pg_database d ON d.datdatabasespace =
t.oid;
 datname | spcname
-----+-----
 postgres | newtablespace
 template1 | pg_default
 template0 | pg_default
 newtestdb4 | pg_default
(4 строки)
```

## Анализ

*Восстановление завершилось успешно.*

## Логическое повреждение данных

*Попробуем создать некорректные изменения*

```
postgres=# select * from users;
```

id	name	email	phone_number
1	Иванов Иван Иванович	ivanov@mail.ru	+79161234567
2	Петров Петр Петрович	petrov@gmail.com	+79162345678
3	Сидорова Анна Сергеевна	sidorova@yahoo.com	+79163456789
4	Кузнецова Ольга Игоревна	kuznetsova@hotmail.com	+79164567890
5	Васильев Алексей Николаевич	vasiliev@yandex.ru	+79165678901
6	Макаров Сергей Дмитриевич	makarov@mail.ru	+79166789012
7	Тимофеева Наталья Валерьевна	timofeeva@gmail.com	+79167890123
8	Андреева Елена Игоревна	andreeva@yahoo.com	+79168901234
9	Козлов Артем Игоревич	kozlov@hotmail.com	+79169012345
10	Смирнова Екатерина Александровна	smirnova@yandex.ru	+79160123456

(10 строк)

```
postgres=# insert into users values(11, 'Кузнецов М. А.', 'maxmax@gmail.com', '+79111234567');
INSERT 0 1
postgres=# insert into users values(12, 'Kovalchuk М. А.', 'maxmax@ya.ru', '+79119876543');
INSERT 0 1
postgres=# select * from users;
```

id	name	email	phone_number
1	Иванов Иван Иванович	ivanov@mail.ru	+79161234567
2	Петров Петр Петрович	petrov@gmail.com	+79162345678
3	Сидорова Анна Сергеевна	sidorova@yahoo.com	+79163456789
4	Кузнецова Ольга Игоревна	kuznetsova@hotmail.com	+79164567890
5	Васильев Алексей Николаевич	vasiliev@yandex.ru	+79165678901
6	Макаров Сергей Дмитриевич	makarov@mail.ru	+79166789012
7	Тимофеева Наталья Валерьевна	timofeeva@gmail.com	+79167890123
8	Андреева Елена Игоревна	andreeva@yahoo.com	+79168901234
9	Козлов Артем Игоревич	kozlov@hotmail.com	+79169012345
10	Смирнова Екатерина Александровна	smirnova@yandex.ru	+79160123456
11	Кузнецов М. А.	maxmax@gmail.com	+79111234567
12	Kovalchuk М. А.	maxmax@ya.ru	+79119876543

(12 строк)



```
postgres=# select * from payments
postgres=# ;
```

id	card_num	card_date	card_cvv
1	1234567812345678	10/24	123
2	2345678923456789	11/25	234
3	3456789034567890	12/26	345
4	4567890145678901	01/27	456
5	5678901256789012	02/28	567

(5 строк)

```
postgres=# insert into payments values(6, '1111222233334444', '03/29', '888');
INSERT 0 1
postgres=# insert into payments values(7, '4444333322221111', '04/30', '999');
INSERT 0 1
postgres=# select * from payments;
```

id	card_num	card_date	card_cvv
1	1234567812345678	10/24	123
2	2345678923456789	11/25	234
3	3456789034567890	12/26	345
4	4567890145678901	01/27	456
5	5678901256789012	02/28	567
6	1111222233334444	03/29	888
7	4444333322221111	04/30	999

(7 строк)

*Фиксируем время*

```
postgres=# select now();
```

now
2023-05-28 21:55:12.3683+03

(1 строка)

*Удаляем таблички выше*

```
postgres=# drop table if exists users, payments;
```

*Останавливаем сервер*

```
pg_ctl stop
```

*Произведем синхронизацию между основным узлом и резервной копией*

```
scp postgres2@pg132:/u06/dir6/ /var/db/postgres2/u06/dir6/
```

*Очищаем директорию для wal-файлов*

```
rm -rf ~/u06/dir6/pg_wal/*
```

*Скопируем последний файл*

```
[postgres2@pg110 ~/u06/dir6/archives]$ cp 00000001000000000000002B ~/u06/dir6/pg_wal/
```

*Восстановление (прописываем в postgresql.conf)*

```
recovery_target_time = '2023-05-28 21:55:12.3683+03'
```

*Перезапустим базу с восстановлением из wal-файла*

```
touch recovery.signal
```

```
pg_ctl start
```

Далее остается прописать команду `pg_wal_replay_resume()` - это функция, которая используется для возобновления воспроизведения журнала транзакций

*Проверяем*

```
postgres=# select * from users;
```

id	name	email	phone_number
1	Иванов Иван Иванович	ivanov@mail.ru	+79161234567
2	Петров Петр Петрович	petrov@gmail.com	+79162345678
3	Сидорова Анна Сергеевна	sidorova@yahoo.com	+79163456789
4	Кузнецова Ольга Игоревна	kuznetsova@hotmail.com	+79164567890
5	Васильев Алексей Николаевич	vasiliev@yandex.ru	+79165678901
6	Макаров Сергей Дмитриевич	makarov@mail.ru	+79166789012
7	Тимофеева Наталья Валерьевна	timofeeva@gmail.com	+79167890123
8	Андреева Елена Игоревна	andreeva@yahoo.com	+79168901234
9	Козлов Артем Игоревич	kozlov@hotmail.com	+79169012345
10	Смирнова Екатерина Александровна	smirnova@yandex.ru	+79160123456

```
(10 строк)

postgres=# select * from payments;
```

id	card_num	card_date	card_cvv
1	1234567812345678	10/24	123
2	2345678923456789	11/25	234
3	3456789034567890	12/26	345
4	4567890145678901	01/27	456
5	5678901256789012	02/28	567

```
(5 строк)

postgres=#
```

*Восстановление прошло успешно*

## Анализ

*Wal-архивация полезна, так как можно не только вернуться к последнему состоянию, но и также выбрать определенный момент времени*

## Вывод

*Во время выполнения лабораторной работы я:*

- 1. изучил способы непрерывного создания резервной копии кластера PostgreSQL.*
- 2. настроил и применил его при различных сбоях.*