Университет ИТМО

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники

# Лабораторная работа №3 Распределенные системы хранения данных

Вариант: 37

Выполнил: Патутин В.М

Группа: P33101

Преподаватель: Шешуков Д.М.

Санкт-Петербург

2022 год

## 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.

**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 холодные полные копии.

Полная копия (rsync) по расписанию (cron) раз в сутки. СУБД на время

копирования должна отключаться. На резервном узле хранить 14 копий,

после успешного создания пятнадцатой копии, самую старую автоматически

уничтожать.

1.2 Подсчитать, каков будет объем резервных копий спустя месяц работы

системы,

исходя из следующих условий:

Средний объем измененных данных за сутки: ~700 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 любой таблицы 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.

Имя резервного узла — pg133.

Имя пользователя — postgres02.

## Выполнение:

Прежде чем делать лабораторную, были добавлены RSA ключи на узлы, чтобы скрипты работали.

### Резервное копирование

#### Настройка резервного копирования

Для начала был написан скрипт для того, чтобы сделать резервную копию и отправить ее на резервный узел (create\_backup.sh - для основного узла).

**create\_backup.sh - для основного узла:**

#!/bin/bash

backups\_dir="backups/"

pg\_ctl -D $HOME/u02/dtt19 -l logfile stop

backup\_name="db-$(date +"%m-%d-%Y-%H-%M-%S").tar"

tar -cf $backup\_name u02/dtt19 u05/dcj13

/opt/csw/bin/rsync --rsync-path=/opt/csw/bin/rsync --archive $backup\_name postgres2@pg133:$backups\_dir

rm -rf $backup\_name

number\_backups=$(ssh postgres2@pg133 ls $backups\_dir | wc -w)

if [ $number\_backups -gt $(echo "14") ]

then

ssh postgres2@pg133 rm -rf $backups\_dir$(ssh postgres2@pg133 ls -tr $backups\_dir | head -1)

fi

pg\_ctl -D $HOME/u02/dtt19 -l logfile start

Далее, по варианту, нужно, чтобы копии были раз в день. Для этого надо добавить задачу в crontab.

export VISUAL=vim #чтобы редактировать vim’ом

crontab -e

Добавляем туда такую строку:

0 0 \* \* \* bash /var/postgres/postgres4/[create\_backup.sh](https://vk.com/away.php?to=http%3A%2F%2Fcreate_backup.sh&cc_key=) > /dev/null 2>&1

0 0 \* \* \* - означает выполнение задачи каждый день в 00:00.

#### Объем резервных данных через месяц

В бд добавляются по 700мб в день.

14 / 2 \* (2 \* 700 + (29 - 1) \* 700) = 147000мб

### Потеря основного узла

При потере основного узла нам надо “поднять” БД на резервном узле. Для этого был написан скрипт (extract\_backup.sh - для резервного узла). Его надо вызвать с параметром - названием бекапа из директории backups. Например:

./extract\_backup.sh db-05-17-2022-18-19-00.tar

**extract\_backup.sh - для резервного узла:**

#!/bin/bash

backups\_dir="backups/"

backup\_name=$1

rm -rf u02

rm -rf u05

cp $backups\_dir$backup\_name ~/

tar -xf $backup\_name

rm -rf $backup\_name

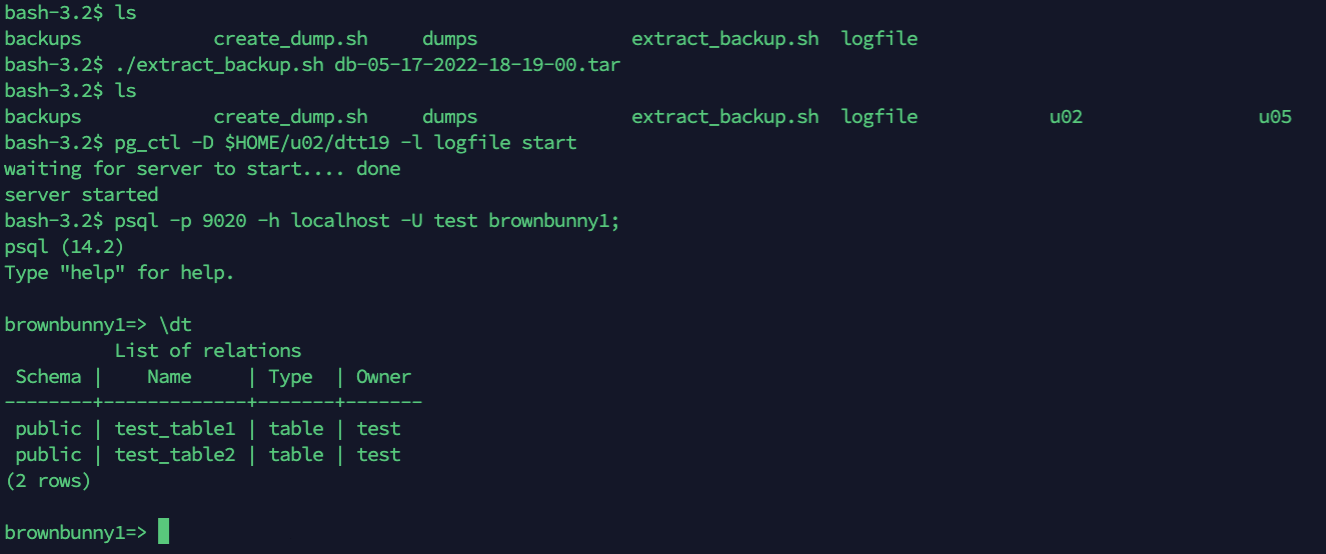
rm -f u02/dtt19/pg\_tblspc/16386

ln -s $(pwd)/u05/dcj13 ~/u02/dtt19/pg\_tblspc/16386

chmod 750 u02/dtt19

chmod 750 u05/dcj13

Пример работы:



Как можно увидеть, копия БД успешно развернулась.

### Повреждение файлов БД

При повреждении файлов БД нам надо получить backup с резервного узла, и развернуть его. Для этого был написан скрипт (extract\_backup.sh - для основного узла). Его надо вызвать с параметром - названием бекапа из директории backups резервного узла. Например:

./extract\_backup.sh db-05-17-2022-18-19-00.tar

**extract\_backup.sh - для основного узла:**

#!/bin/bash

backups\_dir="backups/"

backup\_name=$1

rm -rf u02

rm -rf u05

rsync --rsync-path=/opt/csw/bin/rsync --archive postgres2@pg133:$backups\_dir$backup\_name ~/

tar -xf $backup\_name

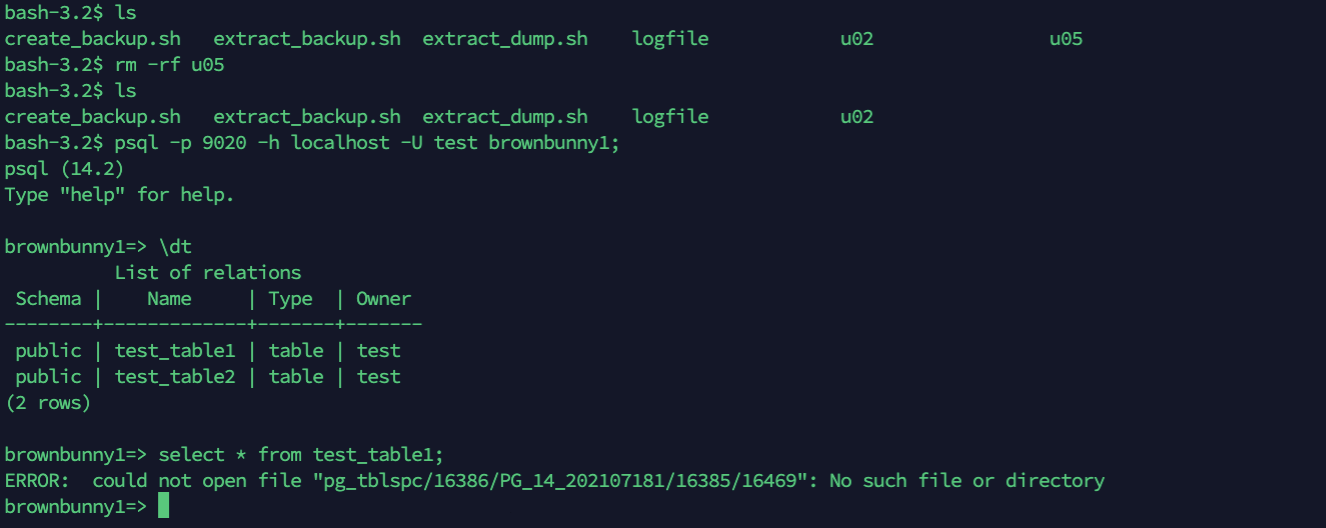
rm -rf $backup\_name

chmod 750 u02/dtt19

chmod 750 u05/dcj13

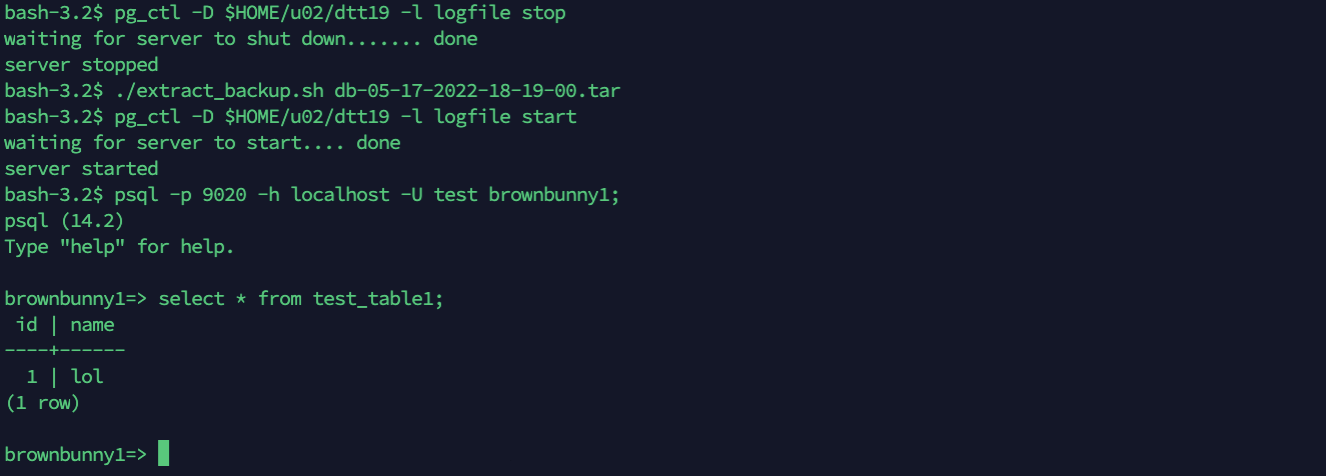
Пример работы:

Удалим директорию с табличным пространством на основном узле.



Как видим, БД работет, но мы не можем посмотреть данные таблицы.

Теперь надо восстановить данные из бекапа.



Как можно увидеть, работоспособность БД восстановлена.

### Логическое повреждение данных

Для восстановления БД после логического повреждения данных, был написан скрипт, который восстанавливает данные по dump.

При логическом повреждении файлов БД нам надо сделать сначала dump на резервном узле. Для этого был написан скрипт (create\_dump.sh - для резервного узла). Потом нужно получить dump с резервного узла, и развернуть его. Для этого был написан скрипт (extract\_dump.sh - для основного узла). Его надо вызвать с параметром - названием дампа из директории dumps резервного узла. Например:

./extract\_dump.sh db-05-16-2022-22-00-19.dump

**create\_dump.sh - для резервного узла:**

#!/bin/bash

dumps\_dir="dumps/"

dump\_name="db-$(date +"%m-%d-%Y-%H-%M-%S").dump"

pg\_dump -h localhost -p 9020 -d brownbunny1 -U test -Fc > $dumps\_dir$dump\_name

**extract\_dump.sh - для основного узла:**

#!/bin/bash

dumps\_dir="dumps/"

dump\_name=$1

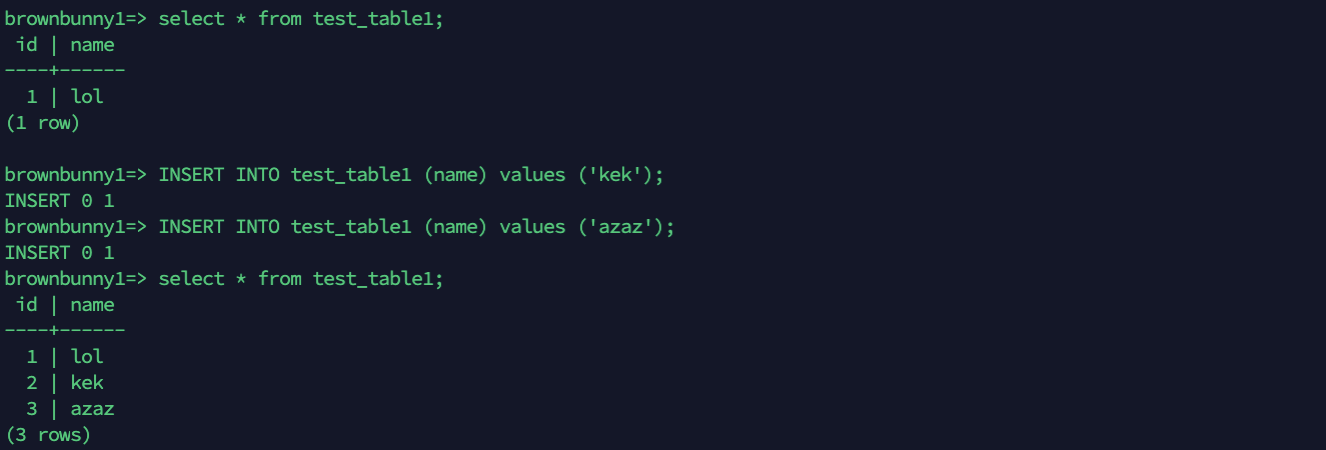
rsync --rsync-path=/opt/csw/bin/rsync --archive postgres2@pg133:$dumps\_dir$dump\_name ~/

pg\_restore -h localhost -p 9020 -d brownbunny1 -c $dump\_name

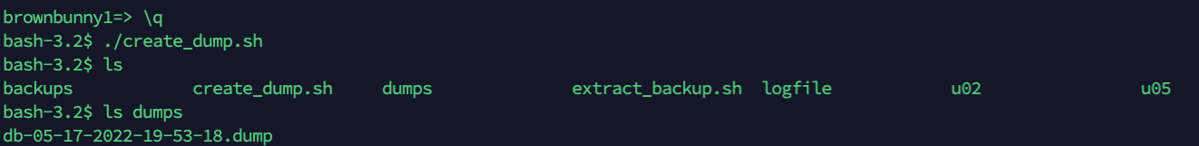
rm -rf $dump\_name

Пример работы:

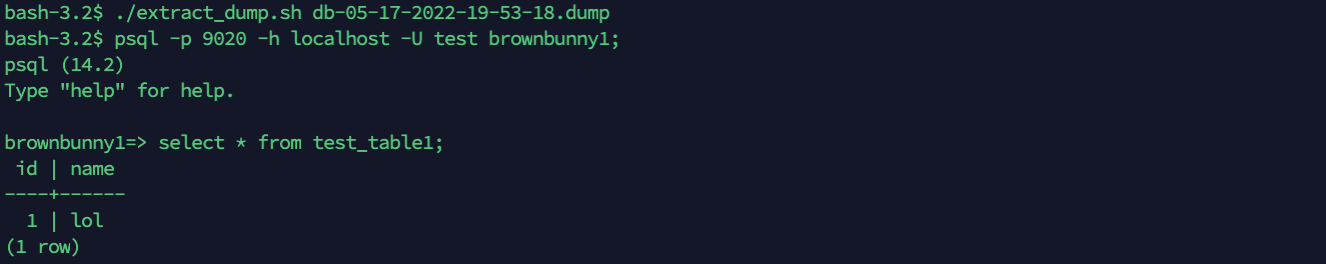
Произведем логическое повреждение данных на основном узле.



Сделаем дамп на резервном узле с нужными исходными нам данными.



Произведем выгрузку дампа на основном узле.



Как можно увидеть данные успешно восстановились.

## Вывод:

В процессе выполнения лабораторной работы были получены навыки создания периодических холодных бекапов БД на удаленный узел с помощью rsync.