

Федеральное государственное автономное образовательное учреждение
высшего образования

Национальный Исследовательский Университет ИТМО



Лабораторная работа №4

Дисциплина: "Распределенные системы хранения данных"

Вариант: 706

Выполнили:

Калабухов Максим Александрович

Коновалов Арсений Антонович

Группа: Р33131

Преподаватель:

Афанасьев Дмитрий Борисович

Город Санкт-Петербург

2024 год

Текст задания	3
Требования к выполнению работы	3
Этап 1. Конфигурация	3
Этап 2. Симуляция и обработка сбоя	3
2.1 Подготовка:	3
2.2 Сбой:	4
2.3 Обработка:	4
Этап 3. Восстановление	4
Выполнение	4
Этап 1	4
Этап 2	8
2.1	8
2.2	9
2.3	9
Этап 3	10
Вывод	11

Текст задания

Цель работы - ознакомиться с методами и средствами построения отказоустойчивых решений на базе СУБД Postgres; получить практические навыки восстановления работы системы после отказа.

Работа рассчитана на двух человек и выполняется в три этапа: настройка, симуляция и обработка сбоя, восстановление.

Требования к выполнению работы

- В качестве хостов использовать одинаковые виртуальные машины.
- В первую очередь необходимо обеспечить сетевую связность между ВМ.
- Для подключения к СУБД (например, через psql), использовать отдельную виртуальную или физическую машину.
- Демонстрировать наполнение базы и доступ на запись на примере **не менее, чем двух** таблиц, столбцов, строк, транзакций и клиентских сессий.

Этап 1. Конфигурация

Развернуть postgres на двух узлах в режиме горячего резерва (Master + Hot Standby). Не использовать дополнительные пакеты. Продемонстрировать доступ в режиме чтение/запись на основном сервере, в режиме чтение на резервном сервере, а также актуальность данных на нём.

Этап 2. Симуляция и обработка сбоя

2.1 Подготовка:

- Установить несколько клиентских подключений к СУБД.
- Продемонстрировать состояние данных и работу клиентов в режиме чтение/запись.

2.2 Сбой:

Симулировать неожиданное отключение основного узла - выполнить Power Off виртуальной машины.

2.3 Обработка:

- Найти и продемонстрировать в логах релевантные сообщения об ошибках.
- Выполнить переключение (failover) на резервный сервер.
- Продемонстрировать состояние данных и работу клиентов в режиме чтение/запись.

Этап 3. Восстановление

- Восстановить работу основного узла - откатить действие, выполненное с виртуальной машиной на этапе 2.2.
- Актуализировать состояние базы на основном узле - накатить все изменения данных, выполненные на этапе 2.3.
- Восстановить исправную работу узлов в исходной конфигурации (в соответствии с этапом 1).
- Продемонстрировать состояние данных и работу клиентов в режиме чтение/запись.

Выполнение

Этап 1

docker-compose.yml:

```
version: '3'

services:
  master:
    container_name: master
    restart: unless-stopped
    image: postgres:latest
    ports:
      - "9001:5432"
    environment:
      - PGDATA=/var/lib/postgresql/data
      - PGENCODING=UTF8
      - PGLOCALE=en_US.UTF8
      - PGUSERNAME=postgres
      - POSTGRES_PASSWORD=postgres
    volumes:
      - $HOME/master/pg_data:/var/lib/postgresql/data
    networks:
      - slons

  slave:
    container_name: slave
    restart: unless-stopped
    image: postgres:latest
    ports:
      - "9002:5432"
    depends_on:
      - master
    environment:
      - PGDATA=/var/lib/postgresql/data
      - PGENCODING=UTF8
      - PGLOCALE=en_US.UTF8
      - PGUSERNAME=postgres
      - POSTGRES_PASSWORD=postgres
    volumes:
      - $HOME/slave/pg_data:/var/lib/postgresql/data
    networks:
```

```
- slons
```

```
networks:  
  slons:  
    driver: bridge
```

Поднимаем контейнеры

```
docker-compose up
```

```
~/Users/arsenykonovalov/master# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS		NAMES
a3d7adf245e5	postgres:15.7	"docker-entrypoint.s..."	2 minutes ago
Up 12 seconds	0.0.0.0:9002->5432/tcp, :::9002->5432/tcp		slave
2fcd7331675d	postgres:15.7	"docker-entrypoint.s..."	2 minutes ago
Up 12 seconds	0.0.0.0:9001->5432/tcp, :::9001->5432/tcp		master

Подключаемся psql и создаем роль репликации

```
~/Users/arsenykonovalov/master# sh connect_to_psql.sh
```

```
Password for user postgres:
```

```
psql (14.11 (Ubuntu 14.11-0ubuntu0.22.04.1), server 15.7 (Debian  
15.7-1.pgdg120+1))
```

```
WARNING: psql major version 14, server major version 15.
```

```
Some psql features might not work.
```

```
Type "help" for help.
```

```
postgres=# create user replica with replication;
```

```
CREATE ROLE
```

```
~/Users/arsenykonovalov/master# sh connect_to_container.sh
```

```
root@2fcd7331675d:/# cd $PGDATA
```

```
# Allow replication connections from localhost, by a user with the  
# replication privilege.
```

local	replication	all		trust
host	replication	all	127.0.0.1/32	trust
host	replication	all	:::1/128	trust
host	replication	replica	all	trust

Изменяем postgresql.conf на мастере:

```
hot_standby=on  
wal_level = hot_standby  
wal_keep_size = 64
```

Делаем первоначальный бэкап мастера

```
root@2fcd7331675d:/# pg_basebackup -h master -U replica -D /backup/
```

Размещаем бэкап на резервном узле:

```
~/Users/arsenykonovalov/master/pg_data# rm -rf  
/Users/arsenykonovalov/pg_data/*  
~/Users/arsenykonovalov/master/pg_data# mv backup/*  
/Users/arsenykonovalov/pg_data/
```

Создаем файл standby.signal на резервном узле

```
root@a3d7adf245e5:/var/lib/postgresql/data# touch standby.signal
```

Конфигурируем подключение к мастеру на резервном узле

```
primary_conninfo='host=master user=replica'
```

```
~/Users/arsenykonovalov# docker-compose up  
Starting master ... done  
Creating slave ... done  
Attaching to master, slave  
master      |  
master      | PostgreSQL Database directory appears to contain a database;  
Skipping initialization  
master      |  
master      | 2024-05-27 02:59:28.594 UTC [1] LOG:  starting PostgreSQL  
15.7 (Debian 15.7-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc  
(Debian 12.2.0-14) 12.2.0, 64-bit  
master      | 2024-05-27 02:59:28.594 UTC [1] LOG:  listening on IPv4  
address "0.0.0.0", port 5432  
master      | 2024-05-27 02:59:28.594 UTC [1] LOG:  listening on IPv6  
address ":::", port 5432  
master      | 2024-05-27 02:59:28.600 UTC [1] LOG:  listening on Unix  
socket "/var/run/postgresql/.s.PGSQL.5432"  
master      | 2024-05-27 02:59:28.612 UTC [30] LOG:  database system was  
shut down at 2024-05-27 02:52:06 UTC  
master      | 2024-05-27 02:59:28.636 UTC [1] LOG:  database system is  
ready to accept connections  
slave       |  
slave       | PostgreSQL Database directory appears to contain a database;  
Skipping initialization  
slave       |
```

```

slave      | 2024-05-27 02:59:29.017 UTC [1] LOG:  starting PostgreSQL
15.7 (Debian 15.7-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc
(Debian 12.2.0-14) 12.2.0, 64-bit
slave      | 2024-05-27 02:59:29.017 UTC [1] LOG:  listening on IPv4
address "0.0.0.0", port 5432
slave      | 2024-05-27 02:59:29.017 UTC [1] LOG:  listening on IPv6
address "::", port 5432
slave      | 2024-05-27 02:59:29.025 UTC [1] LOG:  listening on Unix
socket "/var/run/postgresql/.s.PGSQL.5432"
slave      | 2024-05-27 02:59:29.033 UTC [29] LOG:  database system was
interrupted; last known up at 2024-05-27 02:50:51 UTC
slave      | 2024-05-27 02:59:29.118 UTC [29] LOG:  entering standby mode
slave      | 2024-05-27 02:59:29.119 UTC [29] LOG:  starting backup
recovery with redo LSN 0/6000028, checkpoint LSN 0/6000060, on timeline
ID 1
slave      | 2024-05-27 02:59:29.123 UTC [29] LOG:  redo starts at
0/6000028
slave      | 2024-05-27 02:59:29.125 UTC [29] LOG:  completed backup
recovery with redo LSN 0/6000028 and end LSN 0/6000100
slave      | 2024-05-27 02:59:29.125 UTC [29] LOG:  consistent recovery
state reached at 0/6000100
slave      | 2024-05-27 02:59:29.125 UTC [1] LOG:  database system is
ready to accept read-only connections
slave      | 2024-05-27 02:59:29.136 UTC [30] LOG:  started streaming WAL
from primary at 0/7000000 on timeline 1

```

Создаем таблицу на мастере:

```

~/Users/arsenykonovalov/master# sh connect_to_psql.sh
psql (14.11 (Ubuntu 14.11-0ubuntu0.22.04.1), server 15.7 (Debian
15.7-1.pgdg120+1))
WARNING: psql major version 14, server major version 15.
        Some psql features might not work.
Type "help" for help.

postgres=# create table test();
CREATE TABLE

```

Проверяем актуальность данных на слейве:

```

~/Users/arsenykonovalov/slave# sh connect_to_psql.sh
psql (14.11 (Ubuntu 14.11-0ubuntu0.22.04.1), server 15.7 (Debian
15.7-1.pgdg120+1))
WARNING: psql major version 14, server major version 15.
        Some psql features might not work.
Type "help" for help.

```



```
postgres=# \d
          List of relations
 Schema | Name   | Type  | Owner
-----+-----+-----+-----
 public | test   | table | postgres
(1 row)
```

Доступность только на чтение:

```
postgres=# create table slave_is_slave();
ERROR:  cannot execute CREATE TABLE in a read-only transaction
---
slave    | 2024-05-27 03:01:29.535 UTC [31] ERROR:  cannot execute
CREATE TABLE in a read-only transaction
slave    | 2024-05-27 03:01:29.535 UTC [31] STATEMENT:  create table
slave_is_slave();
```

Этап 2

2.1

Дополнительная демонстрация данных:

```
MASTER
postgres=# create table students(id serial primary key, name text);
CREATE TABLE
postgres=# insert into students values (1, 'Maxim');
INSERT 0 1
postgres=# insert into students values (2, 'Arseniy');
INSERT 0 1

SLAVE
postgres=# \d
          List of relations
 Schema | Name           | Type      | Owner
-----+-----+-----+-----
 public | students       | table     | postgres
 public | students_id_seq | sequence  | postgres
 public | test           | table     | postgres
(3 rows)

postgres=# select * from students;
```

```
id | name
---+-----
 1 | Maxim
 2 | Arseniy
(2 rows)
```

2.2

Убиваем контейнер мастера (имитация Power Off)

```
~/Users/arsenykonovalov# docker kill master
master
```

2.3

Логи слейва после отключения мастера:

```
master exited with code 137
slave      | 2024-05-27 03:11:24.301 UTC [33] FATAL: could not connect
to the primary server: could not translate host name "master" to
address: Name or service not known
slave      | 2024-05-27 03:11:24.301 UTC [29] LOG: waiting for WAL to
become available at 0/703AA48
slave      | 2024-05-27 03:11:29.265 UTC [34] FATAL: could not connect
to the primary server: could not translate host name "master" to
address: Name or service not known
slave      | 2024-05-27 03:11:29.265 UTC [29] LOG: waiting for WAL to
become available at 0/703AA48
slave      | 2024-05-27 03:11:34.269 UTC [35] FATAL: could not connect
to the primary server: could not translate host name "master" to
address: Name or service not known
slave      | 2024-05-27 03:11:34.270 UTC [29] LOG: waiting for WAL to
become available at 0/703AA48
slave      | 2024-05-27 03:11:39.269 UTC [36] FATAL: could not connect
to the primary server: could not translate host name "master" to
address: Name or service not known
slave      | 2024-05-27 03:11:39.270 UTC [29] LOG: waiting for WAL to
become available at 0/703AA48
```

```
~/Users/arsenykonovalov# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS        PORTS
NAMES
7a18b073350d   postgres:15.7 "docker-entrypoint.s..." 12 minutes ago Up 12
minutes       0.0.0.0:9002->5432/tcp, :::9002->5432/tcp slave
```

```
d1031451b725 postgres:15.7 "docker-entrypoint.s..." 39 minutes ago
Exited (137) 54 seconds ago
```

Осуществляем переключение на резервный сервер:

```
~/Users/arsenykonovalov# docker exec -it slave bashroot@7a18b073350d:/#
su - postgres
postgres@7a18b073350d:~$ /usr/lib/postgresql/15/bin/pg_ctl promote -D
/var/lib/postgresql/data
waiting for server to promote... done
server promoted

exit
```

Проверяем возможность писать и читать на слейве:

```
~/Users/arsenykonovalov# sh connect_to_psql.sh
WARNING: psql major version 14, server major version 15.
        Some psql features might not work.
Type "help" for help.

postgres=# insert into students values(3, 'Dima');
INSERT 0 1
postgres=# select * from students;
 id | name
----+-----
  1 | Maxim
  2 | Arseniy
  3 | Dima
(3 rows)
```

Этап 3

Восстанавливаем работу мастера:

```
pg_basebackup -P -X stream -c fast -h slave -U postgres -D $PGDATA
```

Создаем standby.signal на слейве:

```
root@a3d7adf245e5:/var/lib/postgresql/data# touch standby.signal
```

Копируем на мастер, возвращаем первоначальную конфигурацию мастера и поднимаем оба узла:

```
~/Users/arsenykonovalov/pg_data# rm -rf
/Users/arsenykonovalov/master/pg_data/*
~/Users/arsenykonovalov/pg_data# mv backup/*
/Users/arsenykonovalov/master/pg_data/
~/Users/arsenykonovalov# docker-compose up master
```

Проверяем read/write:

```
MASTER
postgres=# insert into students values (4, 'AlexG2004');
INSERT 0 1
SLAVE
postgres=# \d
```

List of relations			
Schema	Name	Type	Owner
public	students	table	postgres
public	students_id_seq	sequence	postgres
public	test	table	postgres

(3 rows)

```
postgres=# select * from students;
 id | name
----+-----
  1 | Maxim
  2 | Arseniy
  3 | Dima
  4 | AlexG2004
(4 rows)
```

Вывод

Научились работать с конфигурацией postgres на двух узлах в режиме горячего резерва (Master + Hot Standby) и осуществлять failover

< laba kaif >

```
-----
\
\      | \_ / |      / \ // \
      / 0 0 \_ /      // | \ \
      /      / \_ /      // | \ \
      @_^_@' / \_ //      | \ \ \
      //^_ / \_ //      | \ \ \
      ( // ) | \_ //      | \ \ \
      ( / / ) _|_ / ) //      | \ \ \
      ( // / ) ' / , _ _ / ( ; - .      | _ _ \ . --
      (( / / )) , - {      _      \ - . | . --
      (( // / )) ' / \      /      ~ - . _ . --
      (( /// )) \ . {      }      /      . -- ^ -
      (( / ))      . ---- ~ - \      \      . ^ -
      /// . ---- . >      \      _ - ~
      /// - . _ _ _ _ _ } ^ - _ _ _ _ ~
                                  / . ~
```