

НИЯУ МИФИ. Лабораторная работа №2-3

Базовые задачи администрирования СУБД

Журбенко Василий, Б21-525

2024

Цель работы

Изучение базовых задач администрирования СУБД PostgreSQL. Изучение изоляции контейнер в СУБД.

На защиту

Остановите экземпляр СУБД и выясните, сколько занимает на диске весь кластер БД. Затем определите, во сколько раз его можно сжать алгоритмом zstd с максимальной степенью компрессии.

```
root@ubuntu-jammy:/home/vagrant# sudo systemctl stop postgresql

root@ubuntu-jammy:/var/lib/postgresql/14# du -sh main/
84M      main/

root@ubuntu-jammy:/var/lib/postgresql/14# sudo tar --zstd -cf
postgresql_cluster.tar.zst -C /var/lib/postgresql/14/ main/
root@ubuntu-jammy:/var/lib/postgresql/14# ls -lh postgresql_cluster.tar.zst
-rw-r--r-- 1 root root 14M Jan 11 00:47 postgresql_cluster.tar.zst
```

Результаты:

- 84 mb до сжатия
 - 14 mb после сжатия
 - сжатие в 6 раз
-

Ход работы

1. Выяснить, в каком месте файловой системы расположен установленный в предыдущих работах кластер баз данных PostgreSQL:

```
postgres@ubuntu-jammy:/vagrant/2-1/scripts$ psql -c 'SHOW config_file;'
config_file
-----
/etc/postgresql/14/main/postgresql.conf
(1 row)
```

2. Выяснить, какие файлы хранятся в директории кластера базы данных:

```
postgres@ubuntu-jammy:/vagrant/2-1/scripts$ cd /etc/postgresql/14/main
postgres@ubuntu-jammy:/etc/postgresql/14/main$ ls
conf.d  environment  pg_ctl.conf  pg_hba.conf  pg_ident.conf  postgresql.conf  start.conf
# В директории кластера хранятся конфигурационные и служебные файлы.
```

3. Выяснить, какой командной строкой запущен экземпляр PostgreSQL:

```
postgres@ubuntu-jammy:/etc/postgresql/14/main$ systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sun 2025-01-05 12:54:32 UTC; 1 day 4h ago
   Process: 5659 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 5659 (code=exited, status=0/SUCCESS)
      CPU: 1ms
postgres@ubuntu-jammy:/etc/postgresql/14/main$ ps aux | grep postgres
...
postgres    5641  0.0  2.5 218756 25044 ?        Ss   00:03   0:03
/usr/lib/postgresql/14/bin/postgres -D /var/lib/postgresql/14/main -c
config_file=/etc/postgresql/14/main/postgresql.conf
# Сервер PostgreSQL запущен с указанными параметрами и директорией кластера.
```

4. Выполнить штатное завершение работы сервера PostgreSQL:

```

root@ubuntu-jammy:/home/vagrant# systemctl stop postgresql
root@ubuntu-jammy:/home/vagrant# systemctl status postgresql
○ postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Mon 2025-01-06 17:07:58 UTC; 7s ago
# Сервер PostgreSQL был успешно остановлен.

```

5. Вновь запустить экземпляр PostgreSQL вручную:

```

postgres@ubuntu-jammy:~$ /usr/lib/postgresql/14/bin/postgres -D
/var/lib/postgresql/14/main -c
config_file=/etc/postgresql/14/main/postgresql.conf
# Сервер запущен вручную от имени пользователя postgres.

```

6. Подключиться к экземпляру и проверить его работоспособность:

```

postgres@ubuntu-jammy:/vagrant/2-1/scripts$ psql
postgres=# SELECT version();

                                version
-----
PostgreSQL 14.15 (Ubuntu 14.15-0ubuntu0.22.04.1) on x86_64-pc-linux-gnu,
compiled by gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0, 64-bit
(1 row)
# Экземпляр PostgreSQL успешно функционирует.

```

7. Создать новую базу данных в кластере. Кто ее владелец? Какие объекты в ней содержатся?

```

postgres=# CREATE DATABASE test_db OWNER postgres;
CREATE DATABASE
postgres=# \l

                                List of databases
  Name  | Owner  | Encoding | Collate | Ctype  | Access privileges
-----+-----+-----+-----+-----+-----
jewelry_store | postgres | UTF8      | C.UTF-8 | C.UTF-8 | =Tc/postgres
+
postgres=# CTc/postgres+

```

```

postgres | postgres | UTF8 | C.UTF-8 | C.UTF-8 | admin=CTc/postgres
template0 | postgres | UTF8 | C.UTF-8 | C.UTF-8 | =c/postgres
+
postgres=CTc/postgres
template1 | postgres | UTF8 | C.UTF-8 | C.UTF-8 | =c/postgres
+
postgres=CTc/postgres
test_db | postgres | UTF8 | C.UTF-8 | C.UTF-8 |
(5 rows)
# Создана новая база test_db с владельцем postgres.

test_db=# \dt
Did not find any relations.
test_db=# \dv
Did not find any relations.
test_db=# \df
List of functions
Schema | Name | Result data type | Argument data types | Type
-----+-----+-----+-----+-----
(0 rows)

```

8. Подключиться к новой базе данных и создать в ней несколько пробных объектов:

```

postgres=# \c test_db
You are now connected to database "test_db" as user "postgres".
test_db=# CREATE TABLE test_table (
    id serial PRIMARY KEY,
    name text,
    created_at timestamp DEFAULT current_timestamp
);
CREATE TABLE
test_db=# CREATE VIEW test_view AS SELECT * FROM test_table;
CREATE VIEW
test_db=# CREATE FUNCTION test_function() RETURNS text AS $$
BEGIN
    RETURN 'Hello from test_function';
END;
$$ LANGUAGE plpgsql;
CREATE FUNCTION
# Созданы таблица, вид и функция в базе test_db.

```

9-10. Убедиться, что из новой базы данных нет доступа к исходной и обратно:

```
test_db=# SELECT * FROM jewelry_store.public.departments;
ERROR:  cross-database references are not implemented:
"jewelry_store.public.departments"
LINE 1: SELECT * FROM jewelry_store.public.departments;
test_db=# SELECT * FROM test_db.public.test_table;1~
 id | name | created_at
----+-----+-----
(0 rows)
```

```
jewelry_store=# SELECT * FROM test_db.public.test_table;
ERROR:  cross-database references are not implemented:
"test_db.public.test_table"
LINE 1: SELECT * FROM test_db.public.test_table;
```

Заключение

В работе были исследованы транзакции в PostgreSQL, выявлено расположение кластера базы данных, проведено создание новой базы данных, объектов в ней и проверка доступов между базами.

► Полная стенография