УНИВЕРСИТЕТ ИТМО

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
“Национальный исследовательский университет ИТМО”

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине
“Распределенные системы хранения данных”

Вариант: 696.

выполнил:
Студент группы P33131
Коновалов Арсений Антонович
Преподаватель
Афанасьев Дмитрий Борисович

г. Санкт-Петербург
2024 г.

1 Оглавление

(Ссылки кликабельны)

2 Задание

1 - Цель работы - на выделенном узле создать и сконфигурировать новый кластер БД Postgres, саму БД, табличные пространства и новую роль, а также произвести наполнение базы в соответствии с заданием.
Отчёт по работе должен содержать все команды по настройке, скрипты, а также измененные строки конфигурационных файлов. Способ подключения к узлу из сети Интернет через helios: ssh -J sXXXXXX@helios.cs.ifmo.ru:2222 postgresY@pgZZZ Способ подключения к узлу из сети факультета: ssh postgresY@pgZZZ Номер выделенного узла pgZZZ, а также логин и пароль для подключения Вам выдаст преподаватель.

Этап 1. Инициализация кластера БД
Директория кластера: $HOME/qkl81
Кодировка: ANSI1251
Локаль: английская
Параметры инициализации задать через переменные окружения

Этап 2. Конфигурация и запуск сервера БД
Способ подключения: сокет TCP/IP, принимать подключения к любому IP-адресу узла
Номер порта: 9696
Остальные способы подключений запретить.
Способ аутентификации клиентов: по паролю MD5
Настроить следующие параметры сервера БД:
max_connections
shared_buffers
temp_buffers
work_mem
checkpoint_timeout
effective_cache_size
fsync
commit_delay
Параметры должны быть подобраны в соответствии со сценарием OLTP:
500 транзакций в секунду размером 4КБ; обеспечить высокую доступность (High Availability) данных.
Директория WAL файлов: $ PGDATA/pg_wal
Формат лог-файлов: .log
Уровень сообщений лога: INFO
Дополнительно логировать: завершение сессий и продолжительность выполнения команд

Этап 3. Дополнительные табличные пространства и наполнение базы
Создать новые табличные пространства для различных таблиц: $HOME/kdu94, $HOME/vdk81, $HOME/ygl69
На основе template0 создать новую базу: lazybluelake
Создать новую роль, предоставить необходимые права, разрешить подключение к базе.
От имени новой роли (не администратора) произвести наполнение ВСЕХ созданных баз тестовыми наборами данных. ВСЕ табличные пространства должны использоваться по назначению.
Вывести список всех табличных пространств кластера и содержащиеся в них объекты.

Данные, выданные преподавателем: pg116:postgres0:*******

3 Выполнение

тап 0:
Подключение:
ssh -p 2222 s335094@se.ifmo.ru
ssh postgres0@pg116

Этап 1: Конфигурация

src/init.sql

```
[postgres0@pg116 ~]$ PGDATA=$HOME/qkl81
[postgres0@pg116 ~]$ PGLOCALE=eu_EU.ANSI1251
[postgres0@pg116 ~]$ PGENCODE=WIN1251
[postgres0@pg116 ~]$ PHUSERNAME=postgres0
[postgres0@pg116 ~]$ PGUSERNAME=postgres0
[postgres0@pg116 ~]$ PGHOST=pg116
[postgres0@pg116 ~]$ export PGDATA PGLOCALE PGUSERNAME PGHOST PGENCODE
[postgres0@pg116 ~]$ pg_ctl -D $HOME/qkl81 -l logfile start
```

Результат:

```
либо выберите подходящее сочетание параметров локализации.
[[postgres0@pg116 ~]$ PGENCODE=ISO8859-1
[[postgres0@pg116 ~]$ initdb --locale=$PGLOCALE --encoding=$PGENCODE --username=$PGUSERNAME
Файлы, относящиеся к этой СУБД, будут принадлежать пользователю "postgres0".
От его имени также будет запускаться процесс сервера.

Кластер баз данных будет инициализирован с локалью "en_US.ISO8859-1".
Выбрана конфигурация текстового поиска по умолчанию "english".

Контроль целостности страниц данных отключён.

исправление прав для существующего каталога /var/db/postgres0/qkl81... ок
создание подкаталогов... ок
выбирается реализация динамической разделяемой памяти... posix
выбирается значение max_connections по умолчанию... 100
выбирается значение shared_buffers по умолчанию... 128MB
выбирается часовой пояс по умолчанию... W-SU
создание конфигурационных файлов... ок
выполняется подготовительный скрипт... ок
выполняется заключительная инициализация... ок
сохранение данных на диске... ок

initdb: предупреждение: включение метода аутентификации "trust" для локальных подключений
Другой метод можно выбрать, отредактировав pg_hba.conf или используя ключи -A,
--auth-local или --auth-host при следующем выполнении initdb.

Готово. Теперь вы можете запустить сервер баз данных:

    pg_ctl -D /var/db/postgres0/qkl81 -l файл_журнала start
```

Этап 2: Подключение

Файл postgresql.conf и его изменения:

src/postgresql.conf

```
#ports and network
listen_address = '*'
port = 9696
password_encryption = md5
#OLTP
max_connections = 500
shared_buffers = 1GB
temp_buffers = 1MB
work_mem = 1MB
max_prepared_transactions = 500
effective_cache_size = 1GB
fsync = on
commit_delay = 1000
checkpoint_timeout = 5min
archive_mode = on
#WAL
archive_command = 'cp␣%p␣PGDATA/pg\_wal/%f'
#log format
logging_collector = on
log_destination = 'stderr'
log_directory = 'log'
#log lvl
log_min_messages = info
```

```
#etc
log_min_duration_sample = 100
log_disconnections = on
```

Файл pg_hba.conf и его изменения:

src/pg_hba.conf

```
- local   all             all                                     trust
- host    all             all             127.0.0.1/32            trust
- host    all             all             ::1/128                 trust
- local   replication     all                                     trust
- host    replication     all             127.0.0.1/32            trust
- host    replication     all             ::1/128                 trust
+ #local   all             all                                     trust
+ #host    all             all             127.0.0.1/32            trust
+ #host    all             all             ::1/128                 trust
+ #local   replication     all                                     trust
+ #host    replication     all             127.0.0.1/32            trust
+ #host    replication     all             ::1/128                 trust

host    all             all             localhost       md5
host    all             all             192.168.0.0/16  md5
local   all             all                             reject
host    all             all             all             reject
```

Этап 3: Работа с БД
Создать новые табличные пространства для различных таблиц: $HOME/kdu94, $HOME/vdk81, $HOME/ygl69

На основе template0 создать новую базу: lazybluelake

src/lazybluelake_create.bash

```
mkdir -p $HOME/kdu94
mkdir -p $HOME/vdk81
mkdir -p $HOME/ygl69

CREATE TABLESPACE kdu94 LOCATION '/var/db/postgres0/kdu94';
CREATE TABLESPACE vdk81 LOCATION '/var/db/postgres0/vdk81';
CREATE TABLESPACE ygl69 LOCATION '/var/db/postgres0/ygl69';

CREATE DATABASE lazybluelake1 TEMPLATE template0;
CREATE DATABASE lazybluelake2 TEMPLATE template0;
CREATE DATABASE lazybluelake3 TEMPLATE template0;

alter DATABASE lazybluelake1 SET TABLESPACE kdu94;
alter DATABASE lazybluelake2 SET TABLESPACE vdk81;
alter DATABASE lazybluelake3 SET TABLESPACE ygl69;
CREATE ROLE USERS;
CREATE ROLE s335094 WITH LOGIN PASSWORD '228';

#inserting data

psql -d lazybluelake1  -p 9696 -h pg116 -U s335094 -W -f ./fill.sql
psql -d lazybluelake2  -p 9696 -h pg116 -U s335094 -W -f ./fill.sql
psql -d lazybluelake3  -p 9696 -h pg116 -U s335094 -W -f ./fill.sql
```

Скрипты для взаимодействия:

src/fill.sql

```
create table lake (id serial primary key, name text, square int);
insert into lake (name, square) values ("Baikal", 5647);
```

```
insert into lake (name, square) values ("Ozero", 999);

WITH objects as (SELECT
    current_database() AS database_name,
      pg_namespace.nspname AS schema_name,
                pg_class.relname AS object_name,
                        pg_class.reltablespace as tbspc FROM
            pg_catalog.pg_class
                        INNER JOIN     pg_catalog.pg_namespace ON pg_class.
                            relnamespace = pg_namespace.oid
                        LEFT JOIN pg_tablespace ON pg_tablespace.oid =
                            pg_class.reltablespace WHERE
                        pg_class.relname NOT LIKE 'pg_%%' ORDER BY
                            database_name,
                             schema_name,object_name), tblspcs as (
                        SELECT * FROM pg_database JOIN pg_tablespace ON
                            pg_database.dattablespace = pg_tablespace.oid)
                        SELECT (CASE WHEN objects.tbspc = 0 THEN (SELECT
                            tblspcs.spcname FROM tblspcs WHERE datname =
                            objects.database_name)
                        ELSE (SELECT spcname FROM pg_tablespace WHERE oid =
                            objects.tbspc)
                        END), objects.object_name FROM objects;
```

Вывод:

src/out.sql

```
 spcname |                object_name
---------+-----------------------------------------------
 ygl69   | _pg_foreign_data_wrappers
 ygl69   | _pg_foreign_servers
 ygl69   | _pg_foreign_table_columns
 ygl69   | _pg_foreign_tables
 ygl69   | _pg_user_mappings
 ygl69   | administrable_role_authorizations
 ygl69   | applicable_roles
 ygl69   | attributes
 ygl69   | character_sets
 ygl69   | check_constraint_routine_usage
 ygl69   | check_constraints
 ygl69   | collation_character_set_applicability
 ygl69   | collations
 ygl69   | column_column_usage
 ygl69   | column_domain_usage
 ygl69   | column_options
 ygl69   | column_privileges
 ygl69   | column_udt_usage
 ygl69   | columns
 ygl69   | constraint_column_usage
 ygl69   | constraint_table_usage
 ygl69   | data_type_privileges
 ygl69   | domain_constraints
 ygl69   | domain_udt_usage
 ygl69   | domains
 ygl69   | element_types
 ygl69   | enabled_roles
 ygl69   | foreign_data_wrapper_options
 ygl69   | foreign_data_wrappers
 ygl69   | foreign_server_options
 ygl69   | foreign_servers
 ygl69   | foreign_table_options
 ygl69   | foreign_tables
```

```
 ygl69    | information_schema_catalog_name
 ygl69    | key_column_usage
 ygl69    | parameters
 ygl69    | referential_constraints
 ygl69    | role_column_grants
 ygl69    | role_routine_grants
 ygl69    | role_table_grants
 ygl69    | role_udt_grants
 ygl69    | role_usage_grants
 ygl69    | routine_column_usage
 ygl69    | routine_privileges
 ygl69    | routine_routine_usage
 ygl69    | routine_sequence_usage
 ygl69    | routine_table_usage
 ygl69    | routines
 ygl69    | schemata
 ygl69    | sequences
 ygl69    | sql_features
 ygl69    | sql_implementation_info
 ygl69    | sql_parts
 ygl69    | sql_sizing
 ygl69    | table_constraints
 ygl69    | table_privileges
 ygl69    | tables
 ygl69    | transforms
 ygl69    | triggered_update_columns
 ygl69    | triggers
 ygl69    | udt_privileges
 ygl69    | usage_privileges
 ygl69    | user_defined_types
 ygl69    | user_mapping_options
 ygl69    | user_mappings
 ygl69    | view_column_usage
 ygl69    | view_routine_usage
 ygl69    | view_table_usage
 ygl69    | views
 ygl69    | lake
 ygl69    | lake_id_seq
 ygl69    | lake_pkey
```

4 Вывод

По итогам выполнения работы я победил создание кластера и его настройку суммарно всего за 13 часов.