

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ

Лабораторная работа №2

по дисциплине

«Распределенные системы хранения данных»

Вариант 65296

Выполнил:

Векшин Арсений Иванович Р3316

Преподаватель:

Николаев Владимир Вячеславович

Санкт-Петербург

~ 2025 ~

Задание	3
Данные для подключения	3
Текст задания	3
Этап 1. Инициализация кластера БД.	5
Задание	5
Выполнение	5
Этап 2. Инициализация кластера БД.	6
Задание	6
Выполнение	6
postgres.conf	6
pg_hba.conf	7
Логи	8
Этап 3. Дополнительные табличные пространства и наполнение базы.	9
Задание	9
Выполнение	9
Скрипт (последний пункт)	10
Этап 4. Допзадание 1	11
Задание	11
Выполнение	11
postgresql.conf	11
pg_hba.conf	11
Код программы	12
Вывод	12

Задание

Данные для подключения

Виртуальная машина: pg107

Пользователь: postgres2

Пароль: /fE/sM38

Текст задания

Лабораторная работа №2

Введите вариант:

Внимание! У разных вариантов разный текст задания!

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/cbz5**
- Кодировка: KOI8-R
- Локаль: английская
- Параметры инициализации задать через аргументы команды

Этап 2. Конфигурация и запуск сервера БД

- Способы подключения: 1) Unix-domain socket в режиме peer; 2) socket TCP/IP, только localhost
- Номер порта: 9296
- Способ аутентификации TCP/IP клиентов: по паролю SHA-256
- Остальные способы подключений запретить.
- 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:

200 одновременных пользователей, 4 сессий на каждого; каждая сессия иницирует до 5 транзакций на запись размером 32КБ; обеспечить максимальную производительность.

- Директория WAL файлов: \$PGDATA/pg_wal
- Формат лог-файлов: .log
- Уровень сообщений лога: WARNING
- Дополнительно логировать: попытки подключения и завершение сессий

Этап 3. Дополнительные табличные пространства и наполнение базы

- На основе шаблона template0 пересоздать базу postgres в новом табличном пространстве: \$HOME/dpk35
- На основе template1 создать новую базу: goodblackrole
- 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.

Этап 1. Инициализация кластера БД.

Задание

- Директория кластера: `$HOME/cbz5`
- Кодировка: KOI8-R
- Локаль: английская
- Параметры инициализации задать через аргументы команды

Выполнение

```
mkdir cbz5
```

```
CLUSTER_DIR="$HOME/cbz5"
```

```
ENCODING="KOI8-R"
```

```
LOCALE="en_US.US-ASCII"
```

```
initdb -D "$CLUSTER_DIR" \  
  --encoding="$ENCODING" \  
  --locale="$LOCALE" \  
  --username=postgres2
```

Логи

```
[postgres2@pg107 ~]$ mkdir cbz5
CLUSTER_DIR="$HOME/cbz5"
ENCODING="KOI8-R"
[postgres2@pg107 ~]$ CLUSTER_DIR="$HOME/cbz5"
[postgres2@pg107 ~]$ ENCODING="KOI8-R"
[postgres2@pg107 ~]$ LOCALE="en_US.US-ASCII"
[postgres2@pg107 ~]$
[postgres2@pg107 ~]$ initdb -D "$CLUSTER_DIR" \
>   --encoding="$ENCODING" \
>   --locale="$LOCALE" \
>   --username=postgres2
Файлы, относящиеся к этой СУБД, будут принадлежать пользователю "postgres2".
От его имени также будет запускаться процесс сервера.

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

Контроль целостности страниц данных отключён.

исправление прав для существующего каталога /var/db/postgres2/cbz5... 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... Europe/Moscow
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
initdb: подсказка: Другой метод можно выбрать, отредактировав pg_hba.conf или ещё раз запустив initdb с ключом -A, --auth-local или --auth-host.

Готово. Теперь вы можете запустить сервер баз данных:

pg_ctl -D /var/db/postgres2/cbz5 -l файл_журнала start
```

Этап 2. Инициализация кластера БД.

Задание

- Способы подключения: 1) Unix-domain сокет в режиме peer; 2) сокет TCP/IP, только localhost
- Номер порта: 9296
- Способ аутентификации TCP/IP клиентов: по паролю SHA-256
- Остальные способы подключений запретить.
- 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: 200 одновременных пользователей, 4 сессий на каждого; каждая сессия иницирует до 5 транзакций на запись размером 32КБ; обеспечить максимальную производительность.
- Директория WAL файлов: \$PGDATA/pg_wal
- Формат лог-файлов: .log
- Уровень сообщений лога: WARNING
- Дополнительно логировать: попытки подключения и завершение сессий

Выполнение

postgres.conf

```
# CONNECTIONS AND AUTHENTICATION
# - Connection Settings -
listen_addresses = 'localhost'      # подключения только с localhost
port = 9296                          # Только локальные TCP/IP
# подключения
max_connections = 800                # 200 пользователей * 4 сессии
unix_socket_directories = '/tmp'    # Разрешён Unix-socket (по
# умолчанию)

# - Authentication -
password_encryption = scram-sha-256 # алгоритм кодировки пароля

# RESOURCE USAGE (except WAL)
# - Memory -
```

```

shared_buffers = 128MB      # Буфер shared memory для кеша страниц БД
temp_buffers = 8MB          # Буфер для временных объектов, создаётся
при подключении
work_mem = 16MB             # Используется для сортировок, хэшей и т.п.

# WRITE-AHEAD LOG
# - Settings -
fsync = on                  # Запись на диск до конца транзакции
commit_delay = 0            # Без задержек перед коммитом (максимальная
производительность)

# - Checkpoints -
checkpoint_timeout = 10min   # Как часто выполнять checkpoint (под
баланс запись/производительность)

# QUERY TUNING
# - Planner Cost Constants -
effective_cache_size = 4GB   # Зарезервированно ОС для postgres (для
планировщика запросов)

# REPORTING AND LOGGING
# - Where to Log -
log_destination = 'stderr'   # Поток ошибок -> логи
logging_collector = on       # Включает сбор логов
log_directory = 'log'        # Каталог для логов
log_filename = 'postgresl-%Y-%m-%d_%H%M%S.log' # Имя лог-файла (патерн)

# - What to Log -
log_connections = on         # Логировать подключения
log_disconnections = on     # Логировать отключения
log_statement = 'none'       # Не логировать запросы

# - When to Log -
log_min_messages = warning   # Логировать ошибки и все более критичное

```

pg_hba.conf

```

# Изменения в pg_hba.conf
# TYPE DATABASE      USER      ADDRESS      METHOD
# "local" is for Unix domain socket connections only
local    all         all                peer
# Разрешить TCP/IP подключения только с localhost с аутентификацией
SCRAM-SHA-256

```

host	all	all	127.0.0.1/32	scram-sha-256
host	all	all	:::1/128	scram-sha-256

Логи

```
[postgres2@pg107 ~/cbz5]$ pg_ctl -D "$PGDATA" start
ожидание запуска сервера....2025-04-20 22:55:41.867 MSK [25317] LOG:  redirecting log output to logging collector process
2025-04-20 22:55:41.867 MSK [25317] HINT:  Future log output will appear in directory "log".
готово
сервер запущен
[postgres2@pg107 ~/cbz5]$ ls
base                pg_hba.conf         pg_replslot         pg_subtrans         pg_xact
global              pg_ident.conf       pg_serial           pg_tblspc           postgresql.auto.conf
log                 pg_logical          pg_snapshots        pg_twophase          postgresql.conf
pg_commit_ts        pg_multixact        pg_stat             PG_VERSION          postmaster.opts
pg_dynshmem         pg_notify           pg_stat_tmp         pg_wal              postmaster.pid
[postgres2@pg107 ~/cbz5]$ cat log/postgresql-2025-04-20_225541.log
2025-04-20 22:55:41.867 MSK [25317] LOG:  ending log output to stderr
2025-04-20 22:55:41.867 MSK [25317] HINT:  Future log output will go to log destination "syslog".
```


Этап 3. Дополнительные табличные пространства и наполнение базы.

Задание

- На основе шаблона template0 пересоздать базу postgres в новом табличном пространстве: `$HOME/dpk35`
- На основе template1 создать новую базу: `goodblackrole`
- 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.

Выполнение

Создание баз

```
CREATE TABLESPACE dpk35 LOCATION '/var/db/postgres2/dpk35';
```

```
CREATE DATABASE dpk35 TEMPLATE template0 TABLESPACE dpk35;
```

```
CREATE DATABASE goodblackrole TEMPLATE template1;
```

```
CREATE ROLE gooduser WITH LOGIN PASSWORD 'password';
```

```
GRANT ALL PRIVILEGES ON DATABASE dpk35 TO gooduser;
```

```
GRANT ALL PRIVILEGES ON DATABASE goodblackrole TO gooduser;
```

```
GRANT USAGE ON SCHEMA public TO gooduser;
```

```
GRANT CREATE ON SCHEMA public TO gooduser;
```

```
ALTER SCHEMA public OWNER TO gooduser;
```

```
ALTER ROLE gooduser SET default_tablespace = goodblackrole;
```

Список баз данных								
Имя	Владелец	Кодировка	Провайдер локали	LC_COLLATE	LC_CTYPE	локаль ICU	Правила ICU	Права доступа
dpk35db	postgres2	KOI8R	libc	en_US.UTF-8	en_US.UTF-8			=Tc/postgres2 + postgres2=CTc/postgres2+
goodblackrole	postgres2	KOI8R	libc	en_US.UTF-8	en_US.UTF-8			gooduser=CTc/postgres2 + =Tc/postgres2 + postgres2=CTc/postgres2+
postgres	postgres2	KOI8R	libc	en_US.UTF-8	en_US.UTF-8			gooduser=CTc/postgres2
template0	postgres2	KOI8R	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres2 + postgres2=CTc/postgres2
template1	postgres2	KOI8R	libc	en_US.UTF-8	en_US.UTF-8			=c/postgres2 + postgres2=CTc/postgres2

```
CREATE TABLE test_table2 (  
    id SERIAL PRIMARY KEY,  
    name TEXT  
);  
  
INSERT INTO test_table2 (name) VALUES ('Alice1'), ('Bob1'),  
('Charlie1');
```

Скрипт (последний пункт)

```
WITH spaced_tables AS (  
    SELECT COALESCE(t.spcname, 'pg_default') AS  
        spcname,  
        c.relname,  
        ROW_NUMBER() OVER (PARTITION BY COALESCE(t.spcname, 'pg_default')  
ORDER BY c.relname) AS rn  
        FROM pg_tablespace t  
        FULL JOIN pg_class c ON c.reltablespace = t.oid  
        WHERE spcname NOT IN ('pg_default', 'pg_global', 'information_schema')  
        ORDER BY spcname, c.relname  
    )  
SELECT  
    CASE WHEN rn = 1 THEN spcname ELSE NULL END AS spcname, relname  
FROM spaced_tables;
```

Этап 4. Допзадание 1

Задание

Подключиться к БД через helios по TCP/IP

Выполнение

postgresql.conf

```
listen_addresses = '*'
```

pg_hba.conf

```
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# Разрешить TCP/IP подключения только с localhost с аутентификацией
SCRAM-SHA-256
host all gooduser all scram-sha-256
host all all 127.0.0.1/32 scram-sha-256
host all all ::1/128 scram-sha-256
```

Изменения применяются после перезапуска БД:

```
[s367133@helios ~]$ psql -h pg107 -p 9296 gooduser -d postgres
Пароль пользователя gooduser:
psql (16.4)
Введите "help", чтобы получить справку.

postgres=> ^C
```

Код программы

<https://github.com/ArsenyVekshin/ITMO/tree/master/RSHD/lab2>

Вывод

