

# УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Распределенные системы хранения данных»

## **Лабораторная работа №2**

*Вариант 122*

Студент

*Иванов Е.Д.*

*Р33111*

Преподаватель

*Николаев В. В.*

Санкт-Петербург, 2023 г.

## Задание:

На выделенном узле создать и сконфигурировать новый кластер БД, саму БД, табличные пространства и новую роль в соответствии с заданием. Произвести наполнение базы.

Отчёт должен содержать все команды по настройке, а также измененные строки конфигурационных файлов.

Подключение к узлу через helios:

1) `ssh s666666@se.ifmo.ru -p 2222`

2) `ssh пользователь@узел`

Персональный пароль для работы с узлом выдается преподавателем.

Обратите внимание, что домашняя директория пользователя

`/var/db/postgres1`

Этапы выполнения работы:

Инициализация кластера БД

- Имя узла — `pg160`.
- Имя пользователя — `postgres1`.
- Директория кластера БД — `$HOME/u08/dir8`.
- Кодировка, локаль — `ANSI1251`, русская
- Перечисленные параметры задать через переменные окружения.

Конфигурация и запуск сервера БД

- Способ подключения к БД — `TCP/IP socket`, номер порта `9120`.
- 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 OLAP: 5 пользователей, пакетная запись/чтение данных в среднем по 20 МБ;

- Директория WAL файлов — `$HOME/u08/dir9`.
- Формат лог-файлов — `log`.
- Уровень сообщений лога — `WARNING`.
- Дополнительно логировать — контрольные точки.

Дополнительные табличные пространства и наполнение

- Создать новое табличное пространство для индексов:
  - `$HOME/u10/dir1`.
- На основе `template0` создать новую базу — `whitebunny10`.
- От имени новой роли (не администратора) произвести наполнение

существующих баз тестовыми наборами данных. Предоставить права по необходимости. Табличные пространства должны использоваться по назначению.

- Вывести список всех табличных пространств кластера и содержащиеся в них объекты.

## Выполнение:

Создал кластер базы данных:(сначала задаём переменные окружения)

- `export PGDATA=$HOME/u08/dir8`
- `export PGHOST=pg160`
- `export PGUSERNAME=postgres1`
- `export PGENCODING=WIN1251`
- `export PGLOCALE=ru_RU.KOI8-R`
- Создание базы  
`initdb --locale=$PGLOCALE --encoding=$PGENCODING --username=$PGUSERNAME --pgdata=$PGDATA`

Далее необходимо изменить на нужные параметры файла `pg_hba.conf`:

- для подключения по паролю(md5) по tcp добавляем информацию о данном доступе, при этом остальные методы доступа запрещаем:

```
# TYPE  DATABASE        USER            ADDRESS          METHOD
# MY CONNECT tcp/ip with port 9120
host    all            all             all              md5
# "local" is for Unix domain socket connections only
local   all            all              reject
# IPv4 local connections:
host    all            all             127.0.0.1/32     reject
# IPv6 local connections:
host    all            all             ::1/128          reject
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication  all             reject
host    replication  all             127.0.0.1/32     reject
host    replication  all             ::1/128          reject
```

(перед этим я создал пользователя со всеми привилегиями ADMIN для удобства последующей работы)

Изменения в файле `postgresql.conf`:

- ставим порт подключения на 9120, разрешаем доступ с любых адресов, ставим максимальное количество подключённых пользователей равным пяти:

```
# - Connection Settings -
```

```
listen_addresses = '*'          # what IP address(es) to listen on;  
                                # comma-separated list of addresses;  
                                # defaults to 'localhost'; use '*' for all  
                                # (change requires restart)  
port = 9120                     # (change requires restart)  
max_connections = 5             # (change requires restart)
```

- меняю метод шифрования на md5:

```
password_encryption = md5      # scram-sha-256 or md5
```

- Установка размеров буферов shared\_buffers, temp\_buffers и work\_mem:

```
# - Memory -
```

```
shared_buffers = 2GB           # min 128kB  
                                # (change requires restart)  
#huge_pages = try              # on, off, or try  
                                # (change requires restart)  
#huge_page_size = 0           # zero for system default  
                                # (change requires restart)  
temp_buffers = 128MB           # min 800kB  
#max_prepared_transactions = 0 # zero disables the feature  
                                # (change requires restart)  
# Caution: it is not advisable to set max_prepared_transactions nonzero unless  
# you actively intend to use prepared transactions.  
work_mem = 64MB                # min 64kB
```

Из документации PostgreSQL следует, что shared\_buffers следует держать от 25% до 40% от всего выделенного ОЗУ. Размер временных буферов следует сделать достаточно большим(128МБ), так как система соответствует OLAP с пакетной передачей в среднем по 20 МБ. Так как операции большие, и, скорее всего, используют множество сортировок и хэш таблиц, то значение work\_mem я выставил в 64МБ(объём памяти для внутренних операций сортировок и хэш-таблиц).

- effective\_cache\_size = 4GB. Оставил по умолчанию(должен быть не меньше чем shared\_buffers).
- fsync = on. Оставил данный параметр включенным, чтобы запись на диск происходила.
- commit\_delay = 0(мс). Сохранение на WAL начинается сразу после выполнения операции.
- checkpoint\_timeout = 10min. Поставил значение на 10 минут, так как предполагаю, что операции не частые и их немного. Не сильно отошел от стандартных 5 минут.
- Директория WAL файлов — \$HOME/u08/dir9:  
archive\_mode = on  
archive\_command = 'cp %p \$HOME/u08/dir9'

- Настройка логов(дополнительно логировать чекпоинты):

*log\_destination = 'stderr'*

*logging\_collector = on*

*log\_min\_messages = warning*

*log\_checkpoints = on*

## Работа с БД:

- Запуск сервера *pg\_ctl -D \$HOME/u08/dir8 -l logfile start*
- создаем БД — *createdb -p 9120 -T template0 whitebunny10*
- *mkdir -p \$HOME/u10/dir1*
- *psql -p 9120 -d whitebunny10*
- Создание пространства индексов *create tablespace indexspace location '/var/db/postgres1/u10/dir1';*
- *create role s311715 login password '1234';*
- *create role ADMIN login password '1234'*(Добавляем эту роль до изменений параметров подключения к БД);
- Далее я создал табличное пространство и добавил тестовую таблицу football с индексом:  
*mkdir -p \$HOME/u11/tables*  
*create tablespace tablespace1 location '/var/db/postgres1/u11/tables';*  
*create table football (id bigserial primary key, name text );*  
*create index on football (name) tablespace indexspace;*
- Выдача прав пользователю s311715:  
*grant select, insert on football to s311715;*  
*grant usage, select on sequence football\_id\_seq to s311715;*
- Вход от пользователя s311715:  
*psql -h pg160 -p 9120 -d whitebunny10 -U s311715* (далее приглашение на ввод пароля, "1234")  
Наполнение таблицы тестовыми данными:  
*insert into football values (DEFAULT, 'zenit');*  
*insert into football values (DEFAULT, 'barsa');*

**Вывести список всех табличных пространств кластера и содержащиеся в них объекты.**

```
whitebunny10=> select * from pg_tablespace
whitebunny10-> ;
  oid |   spcname   | spcowner | spcacl | spcoptions
-----+-----+-----+-----+-----
 1663 | pg_default  |      10  |        |
 1664 | pg_global   |      10  |        |
16386 | indexspace  |      10  |        |
16389 | tablespace1 |      10  |        |
(4 строки)
```

```
whitebunny10=> select t.spcname, STRING_AGG(c.relname, E',\n') FROM pg_class c JOIN pg_tablespace t ON c.reltablespace = t
 spcname | string_agg
-----+-----
pg_global | pg_toast_1262,
          | pg_toast_1262_index,
          | pg_toast_2964,
          | pg_toast_2964_index,
          | pg_toast_1213,
          | pg_toast_1213_index,
          | pg_toast_1260,
          | pg_toast_1260_index,
          | pg_toast_2396,
          | pg_toast_2396_index,
          | pg_toast_6000,
          | pg_toast_6000_index,
          | pg_toast_3592,
          | pg_toast_3592_index,
          | pg_toast_6100,
          | pg_toast_6100_index,
          | pg_database_datname_index,
          | pg_database_oid_index,
          | pg_db_role_setting_databaseid_rol_index,+
          | pg_tablespace_oid_index,
          | pg_tablespace_spcname_index,
          | pg_authid_rolname_index,
          | pg_authid_oid_index,
          | pg_auth_members_role_member_index,
          | pg_auth_members_member_role_index,
          | pg_shdepend_depender_index,
          | pg_shdepend_reference_index,
          | pg_shdescription_o_c_index,
          | pg_replication_origin_roiident_index,
          | pg_replication_origin_roname_index,
          | pg_shseclabel_object_index,
          | pg_subscription_oid_index,
          | pg_subscription_subname_index,
          | pg_authid,
          | pg_subscription,
          | pg_database,
          | pg_db_role_setting,
          | pg_tablespace,
          | pg_auth_members,
          | pg_shdepend,
          | pg_shdescription,
          | pg_replication_origin,
          | pg_shseclabel
indexspace | football_name_idx
(2 строки)
```

**Выводы:**