Университет ИТМО

Факультет ФПИ и КТ P33131

**Отчет по лабораторной работе №2**

**«Распределенные системы хранения данных»**

Вариант 816

Студент:

Чжоу Хунсян

Гр.P33131

Преподаватель:

Афанасьев Дмитрий Борисович

[Цель работы 2](#_Toc169017272)

[Этап 1. Инициализация кластера БД 3](#_Toc169017273)

[Этап 2. Конфигурация и запуск сервера БД 3](#_Toc169017274)

[Этап 3. Дополнительные табличные пространства и наполнение базы 5](#_Toc169017275)

[Вывод 8](#_Toc169017276)

# 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 s336184@helios.cs.ifmo.ru:2222 postgres2@pg117

Способ подключения к узлу из сети факультета:

ssh postgres2@pg117

Номер выделенного узла pg117, а также логин и пароль для подключения Вам выдаст преподаватель.

pg117:postgres2:FrMa3dca

# Этап 1. Инициализация кластера БД

Директория кластера: $HOME/ewe49  
Кодировка: KOI8-R  
Локаль: русская  
Параметры инициализации задать через переменные окружения

ssh -J s336184@helios.cs.ifmo.ru:2222 postgres2@pg117

[postgres2@pg117 ~]$ export PGDATA=$HOME/ewe49

[postgres2@pg117 ~]$ export PGWAL=$HOME/svq55

[postgres2@pg117 ~]$ export LANG=ru\_RU.KOI8-R

[postgres2@pg117 ~]$ echo $PGDATA

/var/db/postgres2/ewe49

[postgres2@pg117 ~]$ echo $PGWAL

/var/db/postgres2/svq55

[postgres2@pg117 ~]$ echo $LANG

ru\_RU.KOI8-R

[postgres2@pg117 ~]$ initdb -D $PGDATA --locale=$LANG --waldir=$PGWAL

Файлы, относящиеся к этой СУБД, будут принадлежать пользователю "postgres2".

От его имени также будет запускаться процесс сервера.

Кластер баз данных будет инициализирован с локалью "ru\_RU.KOI8-R".

Кодировка БД по умолчанию, выбранная в соответствии с настройками: "KOI8R".

Выбрана конфигурация текстового поиска по умолчанию "russian".

Контроль целостности страниц данных отключён.

создание каталога /var/db/postgres2/ewe49... ок

создание каталога /var/db/postgres2/svq55... 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/postgres2/ewe49 -l файл\_журнала start

# Этап 2. Конфигурация и запуск сервера БД

* Способы подключения:
  1. Unix-domain сокет в режиме peer;
  2. сокет TCP/IP, принимать подключения к любому IP-адресу узла
* Номер порта: 9806
* Способ аутентификации TCP/IP клиентов: по имени пользователя
  1. Остальные способы подключений запретить.
  2. 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 с аппаратной конфигурацией:  
       оперативная память 12ГБ, хранение на жёстком диске (HDD).
* Директория WAL файлов: $HOME/svq55
* Формат лог-файлов: .csv
* Уровень сообщений лога: ERROR
* Дополнительно логировать: завершение сессий и продолжительность выполнения команд

$PGDATA/postgresql.conf

# Connection Settings

listen\_addresses = '\*' # Прослушивает все IP-адреса, позволяя подключаться к

серверу базы данных с любого IP-адреса.

port = 9806 # Номер порта, используемый сервером базы данных,

установлен на 9806.

# Resource Consumption

max\_connections = 100 # Максимальное количество подключений установлено на 100.

shared\_buffers = 3GB # Размер общего буфера, обычно 25% от общего объема

памяти, здесь установлен на 3 ГБ.

temp\_buffers = 128MB # Размер временного буфера на сеанс установлен на 128 МБ.

work\_mem = 64MB # Размер памяти, используемый каждой операцией

сортировки, установлен на 64 МБ.

effective\_cache\_size = 9GB # Размер кэша операционной системы оценивается при работе

базы данных, обычно он составляет 75% от общего объема

памяти, здесь он установлен на 9 ГБ.

# Checkpoints

checkpoint\_timeout = 10min # Интервал контрольной точки установлен на 15 минут.

# Write-Ahead Logging (WAL)

fsync = on # Перед записью на диск убедитесь, что данные

синхронизированы, чтобы обеспечить согласованность

данных.

commit\_delay = 0 # Время задержки записи на диск после фиксации

транзакции. Установите значение 0, чтобы указать

отсутствие задержки.

wal\_level = replica # Уровень детализации записей WAL, установленный на

реплику для поддержки репликации.

# Logging

logging\_collector = on # Включите сборщик журналов для записи журналов в файл.

log\_destination = 'csvlog' # Формат вывода журнала — CSV.

log\_directory = 'log' # Каталог хранения файлов журналов.

log\_filename = 'postgresql-%Y-%m-%d\_%H%M%S.log' # Формат именования файлов журнала.

log\_truncate\_on\_rotation = on

log\_rotation\_age = 1d # Время ротации файла журнала установлено так, чтобы оно

менялось один раз в день.

log\_rotation\_size = 0 # Максимальный размер файла журнала до ротации.

log\_statement = 'none' # Тип записываемых операторов SQL. Этот параметр настроен

на отсутствие записи каких-либо операторов.

log\_duration = on # Запишите время выполнения каждой команды.

log\_connections = on # Регистрируйте каждое новое соединение.

log\_disconnections = on # Регистрируйте каждое отключение.

log\_min\_messages = error # Минимальный уровень ведения журнала, установленный для

регистрации сообщений об ошибках.

log\_error\_verbosity = default # Подробность журнала ошибок, установлена ​​по умолчанию.

$PGDATA/pg\_hba.conf

# TYPE DATABASE USER ADDRESS METHOD

# "local" is for Unix domain socket connections only

local all all peer

# IPv4 local connections:

host all all 127.0.0.1/32 md5

# IPv6 local connections:

host all all ::1/128 md5

# 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

Использует UNIX Socket с одноранговым узлом для подключения базы данных Postgres

[postgres2@pg117 ~]$ pg\_ctl -D /var/db/postgres2/ewe49 -l logfile start

ожидание запуска сервера.... готово

сервер запущен

[postgres2@pg117 ~/ewe49]$ psql -U postgres2 -p 9806 -d postgres

psql (14.2)

Введите "help", чтобы получить справку.

postgres=# ALTER USER postgres2 WITH PASSWORD '123456';

ALTER ROLE

# Этап 3. Дополнительные табличные пространства и наполнение базы

* Создать новые табличные пространства для партицированной таблицы: $HOME/gje71, $HOME/xca33
* На основе template1 создать новую базу: fatbluefish
* 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.

[postgres2@pg117 ~]$ mkdir $HOME/gje71

[postgres2@pg117 ~]$ mkdir $HOME/xca33

[postgres2@pg117 ~]$ ls

ewe49 gje71 logfile svq55 xca33

[postgres2@pg117 ~/gje71]$ psql -U postgres2 -p 9806 -d postgres

psql (14.2)

Введите "help", чтобы получить справку.

postgres=# CREATE TABLESPACE ts\_gje71 LOCATION '/var/db/postgres2/gje71';

CREATE TABLESPACE

postgres=# CREATE TABLESPACE ts\_xca33 LOCATION '/var/db/postgres2/xca33';

CREATE TABLESPACE

postgres=# \l

Список баз данных

Имя | Владелец | Кодировка | LC\_COLLATE | LC\_CTYPE | Права доступа

-----------+-----------+-----------+--------------+--------------+-------------------------

postgres | postgres2 | KOI8R | ru\_RU.KOI8-R | ru\_RU.KOI8-R |

template0 | postgres2 | KOI8R | ru\_RU.KOI8-R | ru\_RU.KOI8-R | =c/postgres2 +

| | | | | postgres2=CTc/postgres2

template1 | postgres2 | KOI8R | ru\_RU.KOI8-R | ru\_RU.KOI8-R | =c/postgres2 +

| | | | | postgres2=CTc/postgres2

(3 строки)

postgres=# CREATE DATABASE fatbluefish TEMPLATE template1;

CREATE DATABASE

postgres=# \l

Список баз данных

Имя | Владелец | Кодировка | LC\_COLLATE | LC\_CTYPE | Права доступа

-------------+-----------+-----------+--------------+--------------+-------------------------

fatbluefish | postgres2 | KOI8R | ru\_RU.KOI8-R | ru\_RU.KOI8-R |

postgres | postgres2 | KOI8R | ru\_RU.KOI8-R | ru\_RU.KOI8-R |

template0 | postgres2 | KOI8R | ru\_RU.KOI8-R | ru\_RU.KOI8-R | =c/postgres2 +

| | | | | postgres2=CTc/postgres2

template1 | postgres2 | KOI8R | ru\_RU.KOI8-R | ru\_RU.KOI8-R | =c/postgres2 +

| | | | | postgres2=CTc/postgres2

(4 строки)

postgres=# CREATE ROLE tolia WITH LOGIN PASSWORD '123456';

CREATE ROLE

postgres=# GRANT CONNECT ON DATABASE fatbluefish TO tolia;

GRANT

postgres=# GRANT ALL PRIVILEGES ON DATABASE fatbluefish TO tolia;

GRANT

postgres=# GRANT CREATE ON TABLESPACE ts\_gje71 TO tolia;

GRANT

postgres=# GRANT CREATE ON TABLESPACE ts\_xca33 TO tolia;

GRANT

postgres=# \q

[postgres2@pg117 ~]$ psql -U postgres2 -p 9806 -d fatbluefish

psql (16.4)

Введите "help", чтобы получить справку.

fatbluefish=>

fatbluefish=> CREATE TABLE sales (

id SERIAL,

sale\_date DATE,

amount DECIMAL,

PRIMARY KEY (id, sale\_date)

) PARTITION BY RANGE (sale\_date);

CREATE TABLE

fatbluefish=> CREATE TABLE sales\_2023 PARTITION OF sales

FOR VALUES FROM ('2023-01-01') TO ('2023-12-31')

TABLESPACE ts\_gje71;

CREATE TABLE

fatbluefish=> CREATE TABLE sales\_2024 PARTITION OF sales

FOR VALUES FROM ('2024-01-01') TO ('2024-12-31')

TABLESPACE ts\_xca33;

CREATE TABLE

fatbluefish=> INSERT INTO sales (sale\_date, amount) VALUES

('2023-02-15', 100.00),

('2023-06-20', 150.00),

('2024-03-10', 200.00);

INSERT 0 3

fatbluefish=> \q

[postgres2@pg117 ~/gje71]$ psql -U postgres2 -p 9806 -d fatbluefish

psql (14.2)

Введите "help", чтобы получить справку.

fatbluefish=# SELECT spcname AS "Name", pg\_tablespace\_location(oid) AS "Location" FROM pg\_tablespace;

Name | Location

------------+-------------------------

pg\_default |

pg\_global |

ts\_gje71 | /var/db/postgres2/gje71

ts\_xca33 | /var/db/postgres2/xca33

(4 строки)

fatbluefish=# SELECT

t.spcname AS "Tablespace",

c.relname AS "Object Name",

CASE c.relkind

WHEN 'r' THEN 'Table'

WHEN 'i' THEN 'Index'

WHEN 'S' THEN 'Sequence'

WHEN 'v' THEN 'View'

WHEN 'm' THEN 'Materialized View'

WHEN 'c' THEN 'Composite Type'

WHEN 't' THEN 'TOAST Table'

WHEN 'f' THEN 'Foreign Table'

END AS "Object Type"

FROM

pg\_class c

JOIN

pg\_tablespace t ON c.reltablespace = t.oid

ORDER BY

t.spcname, c.relname;

Tablespace | Object Name | Object Type

------------+-----------------------------------------+-------------

pg\_global | pg\_auth\_members | Table

pg\_global | pg\_auth\_members\_member\_role\_index | Index

pg\_global | pg\_auth\_members\_role\_member\_index | Index

pg\_global | pg\_authid | Table

pg\_global | pg\_authid\_oid\_index | Index

pg\_global | pg\_authid\_rolname\_index | Index

pg\_global | pg\_database | Table

pg\_global | pg\_database\_datname\_index | Index

pg\_global | pg\_database\_oid\_index | Index

pg\_global | pg\_db\_role\_setting | Table

pg\_global | pg\_db\_role\_setting\_databaseid\_rol\_index | Index

pg\_global | pg\_replication\_origin | Table

pg\_global | pg\_replication\_origin\_roiident\_index | Index

pg\_global | pg\_replication\_origin\_roname\_index | Index

pg\_global | pg\_shdepend | Table

pg\_global | pg\_shdepend\_depender\_index | Index

pg\_global | pg\_shdepend\_reference\_index | Index

pg\_global | pg\_shdescription | Table

pg\_global | pg\_shdescription\_o\_c\_index | Index

pg\_global | pg\_shseclabel | Table

pg\_global | pg\_shseclabel\_object\_index | Index

pg\_global | pg\_subscription | Table

pg\_global | pg\_subscription\_oid\_index | Index

pg\_global | pg\_subscription\_subname\_index | Index

pg\_global | pg\_tablespace | Table

pg\_global | pg\_tablespace\_oid\_index | Index

pg\_global | pg\_tablespace\_spcname\_index | Index

pg\_global | pg\_toast\_1213 | TOAST Table

pg\_global | pg\_toast\_1213\_index | Index

pg\_global | pg\_toast\_1260 | TOAST Table

pg\_global | pg\_toast\_1260\_index | Index

pg\_global | pg\_toast\_1262 | TOAST Table

pg\_global | pg\_toast\_1262\_index | Index

pg\_global | pg\_toast\_2396 | TOAST Table

pg\_global | pg\_toast\_2396\_index | Index

pg\_global | pg\_toast\_2964 | TOAST Table

pg\_global | pg\_toast\_2964\_index | Index

pg\_global | pg\_toast\_3592 | TOAST Table

pg\_global | pg\_toast\_3592\_index | Index

pg\_global | pg\_toast\_6000 | TOAST Table

pg\_global | pg\_toast\_6000\_index | Index

pg\_global | pg\_toast\_6100 | TOAST Table

pg\_global | pg\_toast\_6100\_index | Index

ts\_gje71 | pg\_toast\_16400 | TOAST Table

ts\_gje71 | pg\_toast\_16400\_index | Index

ts\_gje71 | sales\_2023 | Table

ts\_xca33 | pg\_toast\_16408 | TOAST Table

ts\_xca33 | pg\_toast\_16408\_index | Index

ts\_xca33 | sales\_2024 | Table

(49 строк)

# Вывод

В ходе выполнения работы научился создавать, инициализировать, настраивать и использовать базы данных с помощью команд.