ИТМО ВТ

Лекция 3

Лектор:
Николаев В.В.

1. Установка и запуск PostgreSQL

ИТМО ВТ

Установка PostgreSQL

2 варианта:

- Графический установщик.
- Использование утилит:
  - использование подготовленных бинарных файлов
    - разные сборки для разных ОС;
  - компиляция исходных кодов (make)
    - нет сборки для используемой ОС;
    - нужна самая последняя версия\версия с внесенными изменениями;

ИТМО ВТ

Установка PostgreSQL

Базовые компоненты PostgreSQL

- сервер (postgresql);
- клиент (postgresql-client);
- contrib, docs и другие компоненты.

ИТМО ВТ

Процесс установки

1. Установка (сборка) базовых компонентов (пакетов):

sudo apt install postgresql-XX postgresql-client-XX ...

2. Создание системного пользователя (postgres):

adduser postgres

mkdir [PGDATA]

chown postgres [PGDATA]

3. Задание переменных окружения: PGDATA, …

ИТМО ВТ

Процесс установки

4. [/pg_path]/bin/initdb [-D PGDATA_path]

5. Запуск экземпляра кластера БД:

i. [/pg_path]/bin/postgres [-D PGDATA_path]

ii. [/pg_path]/bin/pg_ctl [-D PGDATA_path] -l logfile start

iii. Сервис: sudo service postgresql start

ИТМО ВТ

Инициализация кластера - initdb

- Создает структуру директорий PGDATA.
- Создает стандартные БД, которые можно использовать для создания своих баз.
- Задает локаль и кодировку, которая будет использоваться кластером БД.

initdb [-D PGDATA_path]

pg_ctl initdb

ИТМО ВТ

Инициализация

Базы данных, доступные после установки:

- пользовательская БД (**postgres**) - принадлежит администратору БД - пользователю postgres;
- создается **template1;**
- создается клон template1 — **template0;**

Создается суперпользователь — по умолчанию совпадает с именем пользователя, который запустил initdb;

- можно задать имя через -U: initdb -U имя

ИТМО ВТ

template1, template0

- Изначально template1 и template0 — идентичны.
- **template0** — служит в виде резервной копии.
- **template1** — используется в качестве шаблона при создании других БД.
- Можно создать свою БД, которая будет использоваться в качестве шаблона.
- Если внести измненения в template1 — они будут применены и для баз, созданных на его основе.
- К template0 — нельзя подключиться (она резервная).

ИТМО ВТ

Запуск сервера базы данных

- Запуск сервера баз данных:

  [/pg_path]/bin/postgres [-D PGDATA_path]

- Можно заранее установить переменную PGDATA:

  PGDATA = …

  export PGDATA

  postgres

- PGDATA - должна быть проинициализирована перед запуском PostgreSQL

ИТМО ВТ

Утилита pg_ctl

- Утилита pg_ctl — для упрощения взаимодействия с postgres.
- Можно запускать (по умолчанию в фоне), останавливать, перезапускать кластер:

  pg_ctl start -l logfile

  pg_ctl status [-D PGDATA_path]

- Можно использовать для инициализации кластера:

  pg_ctl [-D PGDATA_path] initdb

2. Подключение к БД PostgreSQL

ИТМО ВТ

Клиенты

- psql:
  - интерактивный терминал PostgreSQL;
  - стандартный клиент для работы с PostgreSQL;
- pgAdmin:
  - графический клиент;
  - доступен для Windows, Unix, macOS;
- Клиенты, использующие libpq.

ИТМО ВТ

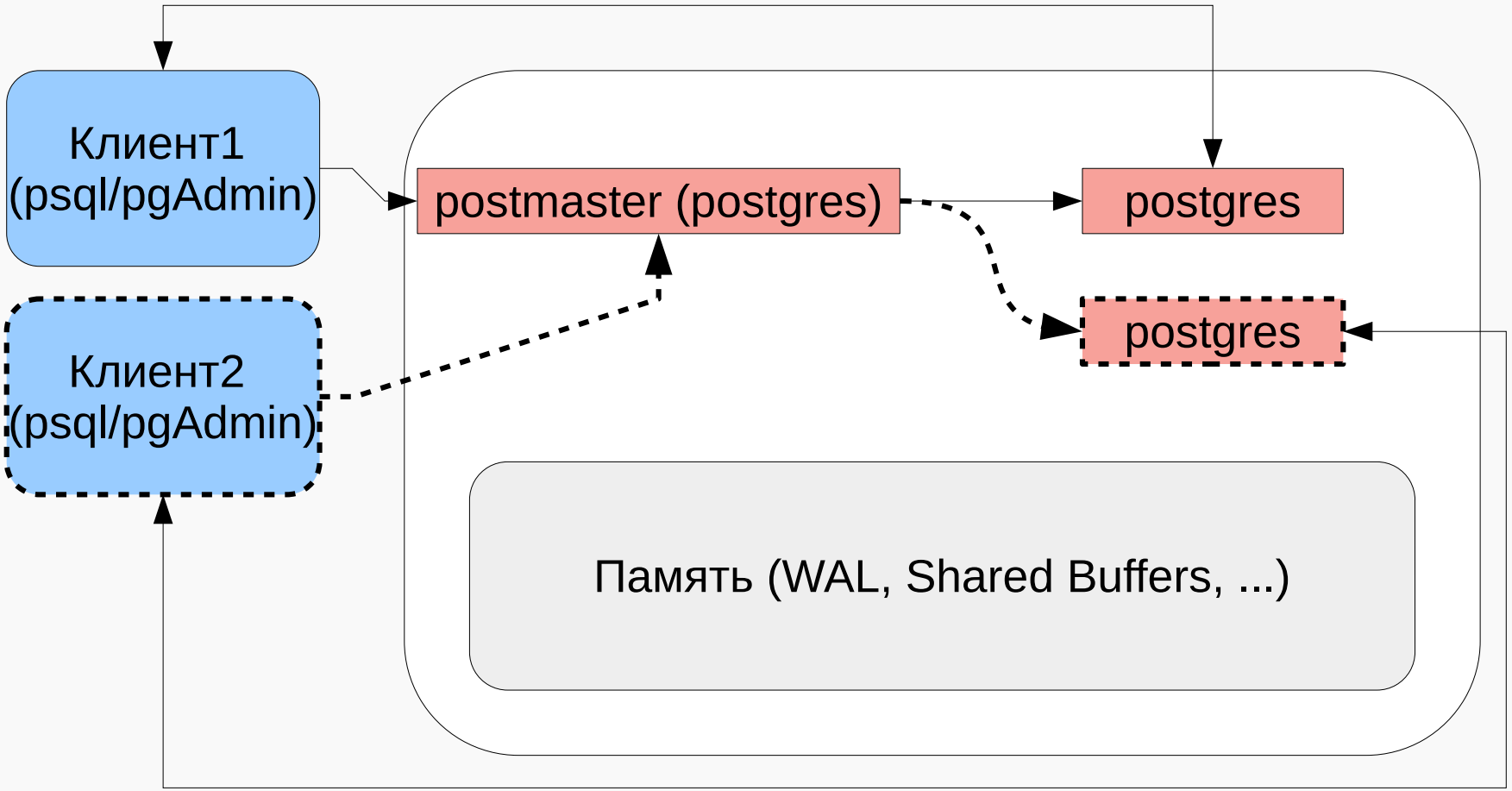
Подключение роли к БД

Для подключения к БД роль должна быть:

- LOGIN;
- содержать привилегию CONNECT на нужную БД;
- разрешение в pg_hba.conf;

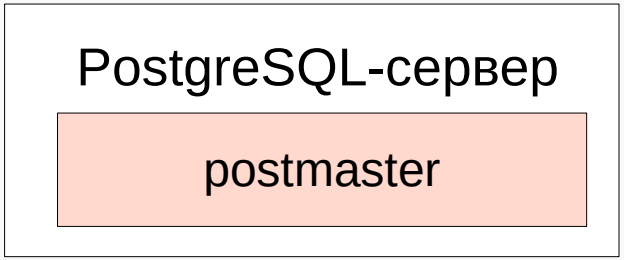
ИТМО ВТ

Подключение к PostgreSQL

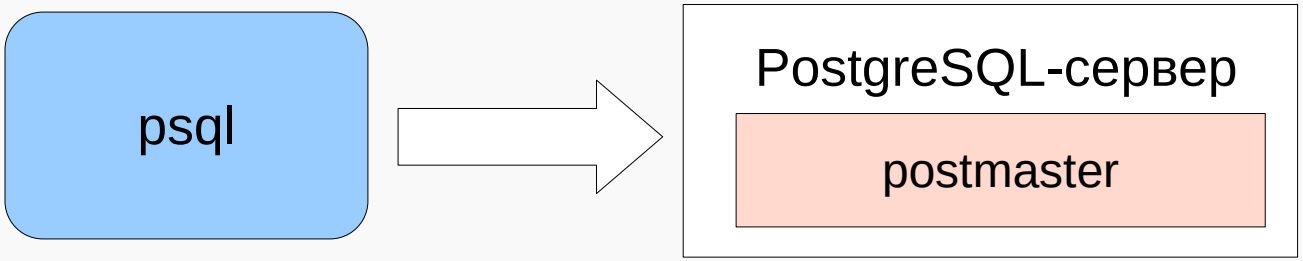


Подключение к серверу БД (1)

1) Запуск сервера PostgreSQL:
   postgres -p 2378 -D /home/myuser/pgd/data

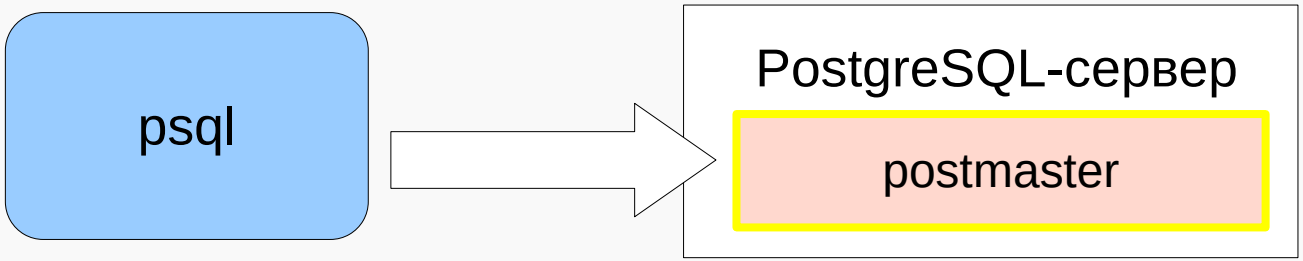


2) Подключаем клиент: psql -p 2347

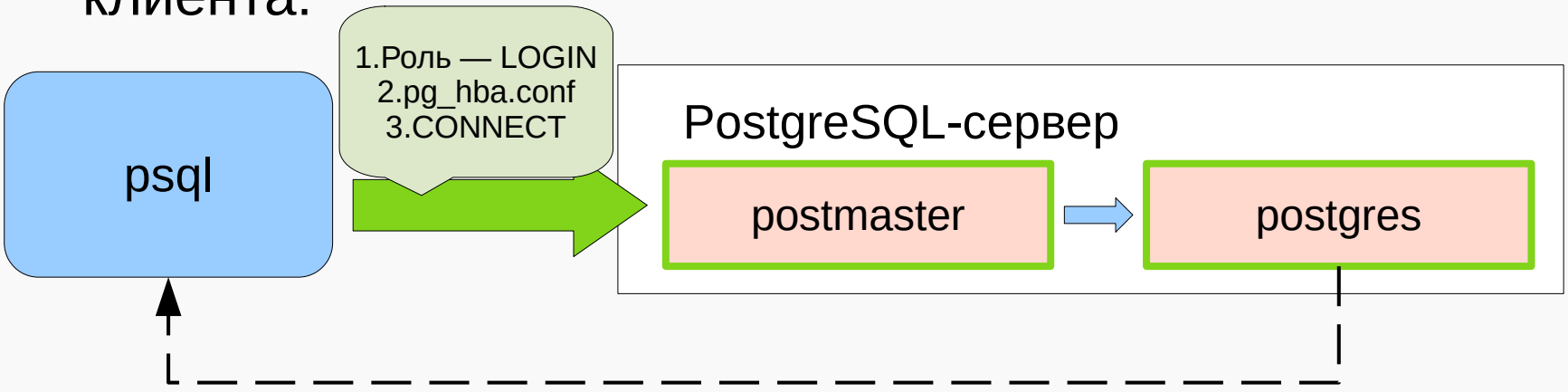


Подключение к серверу БД (2)

3) экземпляр проверяет, может ли клиент получить доступ:



4) Если проверка прошла успешно, создается серверный процесс для обработки соединения данного клиента:



ИТМО ВТ

pg_hba.conf

- В pg_hba.conf задаются способы подключения для различных пользователей к различным базам данных.
- Создается при работе initdb.
- По умолчанию располагается в PGDATA.
- Можно поменять через hba_file параметр:
  - при запуске postgres;
  - в postgresql.conf;

ИТМО ВТ

Формат pg_hba.conf

- Каждая запись в файле определяет вид подключения для разных категорий пользователей.

- Порядок расположения записей влияет на права:
  - чтение происходит последовательно;
  - pg_hba.conf читается во время запуска;
  - Если файл изменен:
    - pg_reload_conf()
    - pg_ctl reload

ИТМО ВТ

Пример pg_hba.conf

```
# PostgreSQL Client Authentication Configuration File
...
# local      DATABASE  USER  METHOD         [OPTIONS]
# host       DATABASE  USER  ADDRESS        METHOD  [OPTIONS]
…
# "local" is for Unix domain socket connections only
  local      all       all   peer
# IPv4 local connections:
  host       all       all   127.0.0.1/32    md5
...
```

ИТМО ВТ

Виды подключений

- Локальный UNIX-сокет (**local**):
  - psql (без имени хоста), используется libpq:
    - подключается пользователь системы к БД с тем же именем, а также есть пользователь БД с тем же именем.
  - клиент и сервер на одной машине;
  - параметр unix_socket_directories
- TCP/IP соединение (**host**):
  - указывается хост, порт:

    psql -h *host* -p *port database*

    psql -h *host* -p *port* -U *username database*
  - Предполагается, что задан параметр PGDATA.

ИТМО ВТ

Методы подключений

- **trust —** предоставить доступ всем из данной категории (**опасно!**);
- **по паролю** (scram-sha-256, md5, password). Смена пароля:
  - изменение роли;
  - \password
- **Ident —** похоже на peer, но для tcp/ip (host).
- **peer —** сравнивается пользователь ОС с пользователем БД:
  - можно задать правила отображения пользователей в **pg_ident.conf** и параметр **map** в **pg_hba.conf**;
  - только local;
- ...

3. Создание БД в PostgreSQL

ИТМО ВТ

Создание Базы Данных

- *CREATE DATABASE*:

  CREATE DATABASE newDb1 [WITH опции];

- *createdb* — утилита в директории [/postgresql]/bin:

  createdb -h host -p port -U user [опции] newDb2;

- Чтобы создать базу пользователь должен быть:
  - *суперпользователем*;
  - обладать ролью *CREATEDB*;

ИТМО ВТ

Создание базы из указанного шаблона (1)

- По умолчанию при создании новой базы данных клонируется **template1**.

- Можно использовать в качестве шаблона другую базу:

  CREATE DATABASE newDb3 TEMPLATE newDB1;

  createdb -T newDB2 newDB4

ИТМО ВТ

Создание базы из указанного шаблона (2)

- Чтобы база могла быть шаблоном нужно, чтобы к ней не было подключений **других пользователей**.

- Системный каталог **pg_database**:

  datistemplate — БД создана, чтобы быть шаблоном;
  - может быть использована (как шаблон) *владельцем*, *суперпользователями* и пользователями с *CREATEDB*;

  datallowconn — запрещаются новые подключения.

4. Файловая структура и конфигурация PostgreSQL

ИТМО ВТ

Структура файлов PostgreSQL

- **PGDATA** – путь к директории данных кластера, установлена заранее.
  - **global** – содержит таблицы уровня кластера (pg_database).
  - **pg_wal** – директория с WAL-файлами.
  - **pg_xact** – директория с данными о коммитах транзакций (CLOG).
  - **base** – файлы данных базы данных кластера:
    - 1
    - 13201
    - 13202

Просмотр файлов данных (1)

sudo ls /PGDATA/base

В результате получим список oid, соответствующих базам:

- 1
- 13201
- 13202

Просмотр файлов данных (2)

Утилита oid2name — для просмотра имен (по oid или filenode):

```
oid2name

Oid        Database Name        Tablespace

----------------------------------------------------------

13202      template1            pg_default

13201      template0            pg_default

1          postgres             pg_default
```

ИТМО ВТ

Файлы данных

- Объектам БД (PostgreSQL) — соответствуют файлы данных:
  - если до 1Gb (по умолчанию): объекту соответствует 1 файл данных;
  - если > 1Gb: объекту соответствуют файлы-сегменты;
- Находятся в $PGDATA/base в директории соответствующей БД.

Базовые конфигурационные файлы

- Стандартный путь, по которому располагаются конфигурационные файлы:

  /etc/postgresql/версия/main/
- **postgresql.conf** — базовый файл для хранения настроек.
- **postgresql.auto.conf** — динамически изменяемые настройки (через ALTER SYSTEM).
- **pg_hba.conf** — конфигурация подключений к БД.
- **pg_ident.conf** — файл отображения имен.

ИТМО ВТ

postgresql.conf

- **postgresql.conf** — базовый файл для хранения настроек.
- Создается при работе initdb.
- По умолчанию располагается в PGDATA.
- Представляет из себя набор параметров (имя, значение):

  search_path = 'someSchema, public'

  shared_buffers = 128MB
- Определяют значения для всего кластера.

Изменение кофигурационных параметров

1. postgresql.conf/ALTER SYSTEM — на уровне кластера;
2. ALTER DATABASE — изменить 1. на уровне базы данных;
3. ALTER ROLE — переписать 1. и 2. на уровне пользователя.
4. SET — изменить для сессии:

   SET param TO value/DEFAULT;
5. Параметры можно задать в команде запуска сервера БД (postgres) — для переписи параметров postgresql.conf.

Значения применяются при запуске новой сессии

5. Табличные пространства

ИТМО ВТ

Табличные пространства

- **Табличные пространства** — позволяют задать пути (директорию в файловой системе), где будут храниться объекты БД (вне PGDATA):
  - позволяют управлять физическим расположением объектов БД.
- У табличных пространств есть имя.
- Просмотр размера табличного пространства:
  - Функция pg_tablespace_size('имя_тп')
- Системный каталог: pg_tablespace
- $PGDATA/pg_tblspc содержит символические ссылки на директории внешних табличных пространств.

Табличные пространства (1)

- Табличные пространства можно задать для разных объектов БД: таблиц, индексов, последовательностей, представлений.

```
CREATE TABLESPACE newTablespace LOCATION
'/diskA/someDir';

CREATE TABLE STUDENT (
   id  integer NOT NULL
) TABLESPACE newTablespace;
```

ИТМО ВТ

Стандартные табличные пространства

- **pg_default** — соответствует директории base в PGDATA;
  - по умолчанию здесь хранятся файлы данных, соответствующие объектам БД.

- **pg_global** — соответствует директории global в PGDATA;
  - pg_database, pg_authid, pg_tablespace, некотоые другие каталоги и индексы.

ИТМО ВТ

Табличные пространства (2)

- Табличное пространство можно задать для БД целиком:

  CREATE DATABASE TestDB

  TABLESPACE newTablespace;

- Табличное пространство может быть изменено:

  ALTER TABLE STUDENT

  SET TABLESPACE pg_default;

ИТМО ВТ

Зачем нужны табличные пространства?

- Можно контролировать что где хранится.
- Критически важные объекты можно размещать на более быстрых физических дисках.
- Если заканчивается место на жестком диске, можно использовать другой диск для размещения объектов.

ИТМО ВТ

Материалы

Документация PostgreSQL:
https://www.postgresql.org/docs/14/index.html

Лицензия PostgreSQL: