

Тема 2. Как запустить PostgreSQL в Docker

В этом видео мы запустим СУБД с помощью Docker и подключимся к ней через интерфейс PyCharm. Также рассмотрим конфигурационный файл для запуска СУБД и основные атрибуты настроек.

На Docker Hub есть [официальный образ СУБД с качественной документацией](#).

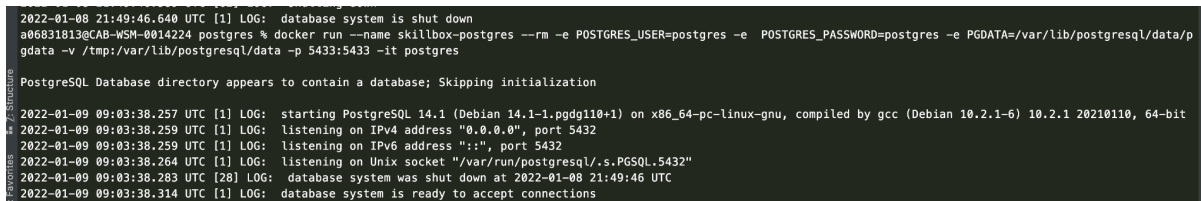
Рассмотрим два варианта развёртывания PostgreSQL:

- через Docker,
- через Docker Compose.

Docker

Чтобы запустить PostgreSQL через Docker, в терминале необходимо ввести команду:

```
docker run --name skillbox-postgres --rm -e POSTGRES_USER=postgres -e POSTGRES_PASSWORD=postgres -e PGDATA=/var/lib/postgresql/data/pgdata -v /tmp:/var/lib/postgresql/data -p 5432:5432 -it postgres
```



```
2022-01-08 21:49:46.640 UTC [1] LOG: database system is shut down
a06831813@CAB-WSM-0014224 postgres % docker run --name skillbox-postgres --rm -e POSTGRES_USER=postgres -e POSTGRES_PASSWORD=postgres -e PGDATA=/var/lib/postgresql/data/pgdata -v /tmp:/var/lib/postgresql/data -p 5432:5432 -it postgres
PostgreSQL Database directory appears to contain a database; Skipping initialization
2022-01-09 09:03:38.257 UTC [1] LOG: starting PostgreSQL 14.1 (Debian 14.1-1.pgdg110+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 10.2.1-6) 10.2.1 20210110, 64-bit
2022-01-09 09:03:38.259 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
2022-01-09 09:03:38.259 UTC [1] LOG: listening on IPv6 address ":::", port 5432
2022-01-09 09:03:38.264 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2022-01-09 09:03:38.283 UTC [28] LOG: database system was shut down at 2022-01-08 21:49:46 UTC
2022-01-09 09:03:38.314 UTC [1] LOG: database system is ready to accept connections
```

Скриншот: [Docker](#) / Skillbox

Мы использовали три параметра виртуального окружения с конфигами СУБД:

- имя пользователя;
- пароль;
- путь, в котором будут данные.

Также нужно указать ключ `-v` (volume), чтобы управлять физическим размещением создаваемых баз данных. Для этого необходимо подмонтировать в контейнер каталог, который будет хранить данные. В примере использована директория `/tmp` — временное хранилище. Если вы перезагрузите компьютер, все данные будут удалены. В базе хранят данные, которые нельзя терять.

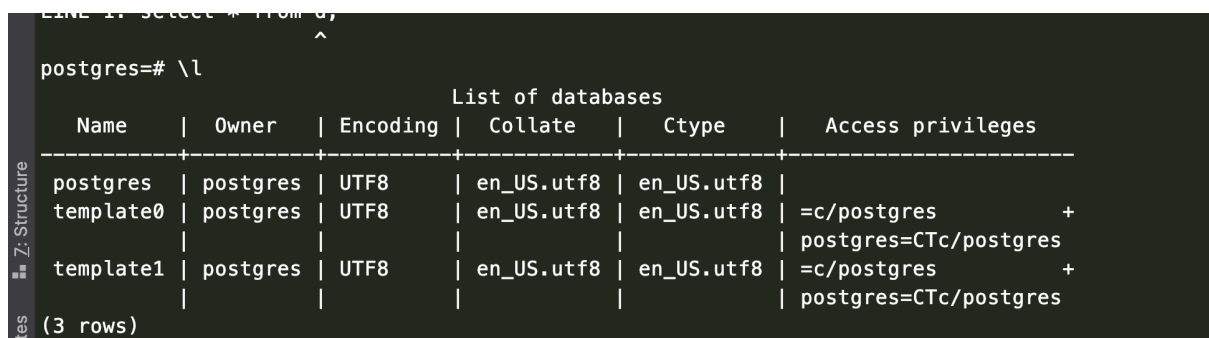
Контейнер запущен и готов для входных соединений по порту 5432. Чтобы зайти в терминал для работы с СУБД, нужно запустить ещё одну команду внутри PostgreSQL-контейнера:

```
docker exec -it skillbox-postgres /bin/sh

psql -u postgres
```

С помощью команды `psql` можно подключиться к СУБД под юзером `postgres`. Команда `psql` — терминальный клиент для работы с PostgreSQL. Она позволяет интерактивно вводить запросы, передавать их в PostgreSQL и видеть результаты. Кроме того, `psql` предоставляет ряд внутренних метакоманд:

- `\c` — подключение к базе данных, в которой после команды вводится имя базы;
- `\l` — список доступных БД;
- `\dt` — список таблиц;
- прочее.



```
LINE 1: select * from o,
^
postgres=# \l
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
postgres	postgres	UTF8	en_US.utf8	en_US.utf8	
template0	postgres	UTF8	en_US.utf8	en_US.utf8	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	en_US.utf8	en_US.utf8	=c/postgres + postgres=CTc/postgres

(3 rows)

Скриншот: [Docker](#) / Skillbox

Описание метакоманд можно найти [в документации](#).

Docker Compose

Рассмотрим запуск СУБД через Docker Compose.

```
version: '3.2'

services:

  postgres:

    image: postgres

    restart: always

    environment:

      - POSTGRES_USER=postgres

      - POSTGRES_PASSWORD=postgres

    ports:

      - '5432:5432'

    volumes:

      - ./db:/var/lib/postgresql/data
```

В папке /db расположен файл postgresql.conf, в котором задаются параметры. Они считываются при инициализации БД.

Теперь нужно подключиться к СУБД через PyCharm. Для начала важно создать тестовую БД и таблицу.

```
postgres=# create database test_db;

CREATE DATABASE

postgres=# \c test_db

You are now connected to database "test_db" as user "postgres".

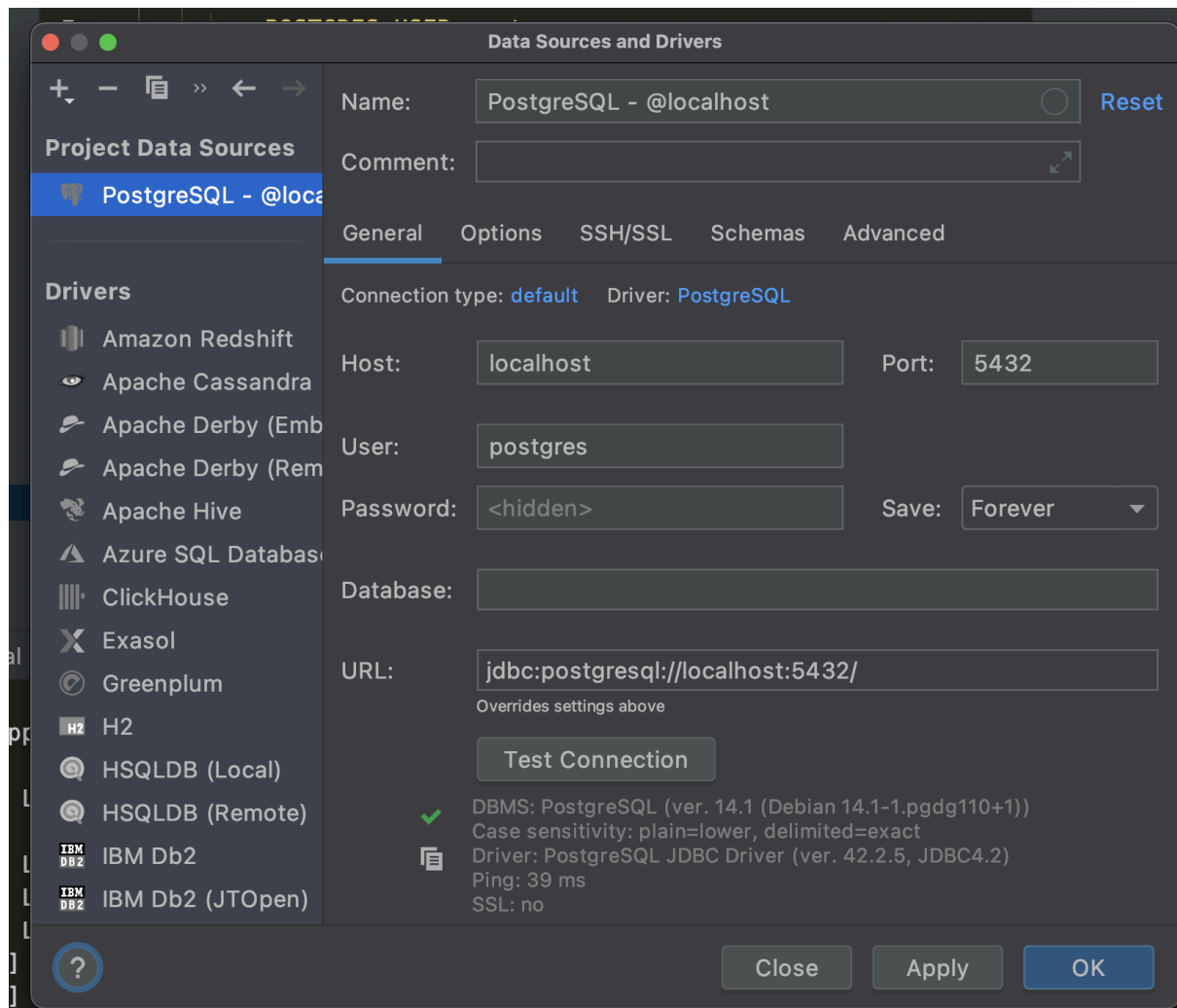
test_db=# create table test_table (id integer primary key, name varchar(255)
null);

CREATE TABLE

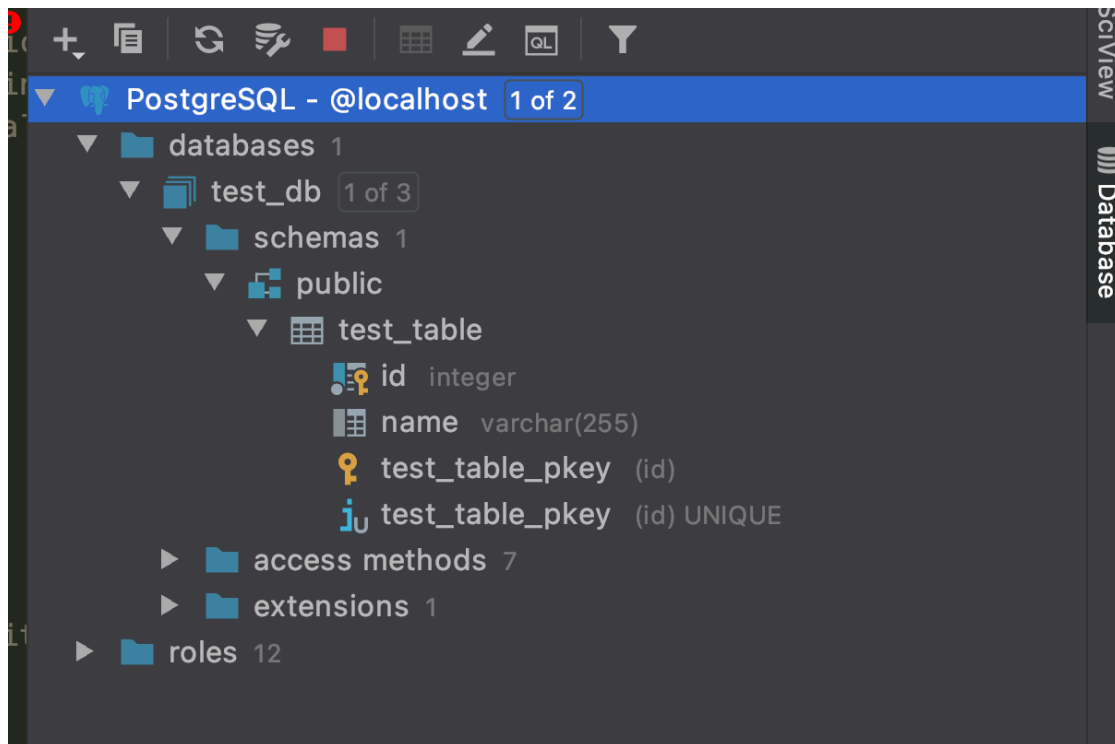
test_db=# insert into test_table (id,name) values (1,'test_name');

INSERT 0 1
```

Заходим в tool — Database:



Скриншот: [Docker](#) / Skillbox



Скриншот: [Docker](#) / Skillbox

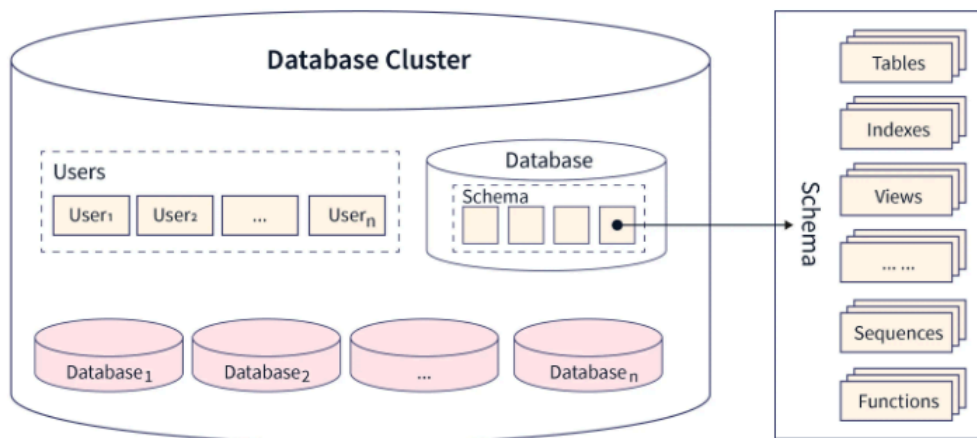


Схема: ScalerTopics / [Scaler.com](#)

Папка `pg_data` (в контейнере папка `db`) хранит множество папок и файлов. Помимо двух файлов, которые мы уже рассматривали, в папке `base` есть каталоги баз данных с идентификатором, файл `PG_VERSION` — версия СУБД, файл `postmaster.opts` с параметрами, при которых инициализировалась СУБД. Папка `global` содержит глобальные таблицы.

Подробнее обо всех файлах и директориях вы можете узнать [в документации](#).

```
postgres=# \dt pg_catalog.*
List of relations
 Schema | Name | Type | Owner
-----+-----+-----+-----
 pg_catalog | pg_aggregate | table | postgres
 pg_catalog | pg_am | table | postgres
 pg_catalog | pg_amop | table | postgres
 pg_catalog | pg_amproc | table | postgres
 pg_catalog | pg_attrdef | table | postgres
 pg_catalog | pg_attribute | table | postgres
 pg_catalog | pg_auth_members | table | postgres
 pg_catalog | pg_authid | table | postgres
 pg_catalog | pg_cast | table | postgres
 pg_catalog | pg_class | table | postgres
 pg_catalog | pg_collation | table | postgres
 pg_catalog | pg_constraint | table | postgres
 pg_catalog | pg_conversion | table | postgres
```

Скриншот: [Docker](#) / Skillbox

О системных каталогах СУБД можно прочитать [в документации](#).

В следующем видео мы с помощью библиотек настроим синхронное и асинхронное подключение к БД из веб-приложения, а также используем ORM.