

Лабораторная работа №1: Основы работы с  
Docker и PostgreSQL

Ляхова Елизавета Олеговна  
группа РИМ-150950

Цель работы: Освоить фундаментальные концепции и базовые операции Docker: создание образов, запуск контейнеров, управление ими, работа с сетями и томами. На практике закрепить навыки, запустив изолированную базу данных PostgreSQL и подключившись к ней извне.

Задачи:

1. Установить и проверить работу Docker.
2. Изучить базовые команды Docker.
3. Запустить контейнер с PostgreSQL в изолированном режиме.
4. Запустить контейнер с pgAdmin и подключить его к контейнеру с БД через сеть Docker.
5. Подключиться к БД из pgAdmin, создать схему и выполнить запросы.
6. Обеспечить сохранность данных БД с помощью томов Docker.

## Ход выполнения:

### Часть 0: Установка и проверка Docker

```
Windows PowerShell      +  ×
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\lyaho> docker --version
Docker version 28.5.1, build e180ab8
PS C:\Users\lyaho> docker compose --version
Usage: docker compose [OPTIONS] COMMAND

Define and run multi-container applications with Docker

Options:
  --all-resources           Include all resources, even those not
                           used by services
  --ansi string             Control when to print ANSI control
                           characters ("never"|"always"|"auto")
                           (default "auto")
  --compatibility           Run compose in backward compatibility mode
  --dry-run                 Execute command in dry run mode
  --env-file stringArray    Specify an alternate environment file
  -f, --file stringArray    Compose configuration files
  --parallel int            Control max parallelism, -1 for
                           unlimited (default -1)
  --profile stringArray     Specify a profile to enable
  --progress string          Set type of progress output (auto,
                           tty, plain, json, quiet)
  --project-directory string Specify an alternate working directory
                           (default: the path of the, first
                           specified, Compose file)
  -p, --project-name string Project name

Management Commands:
  bridge      Convert compose files into another model

Commands:
  attach      Attach local standard input, output, and error streams to a service's running container
```

### 1) Проверка установки Docker

```
Windows PowerShell      +  ×
unpause   Unpause services
up        Create and start containers
version   Show the Docker Compose version information
volumes   List volumes
wait      Block until containers of all (or specified) services stop.
watch     Watch build context for service and rebuild/refresh containers when files are updated

Run 'docker compose COMMAND --help' for more information on a command.
PS C:\Users\lyaho> docker run hello-world

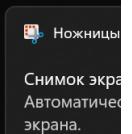
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
PS C:\Users\lyaho> |
```



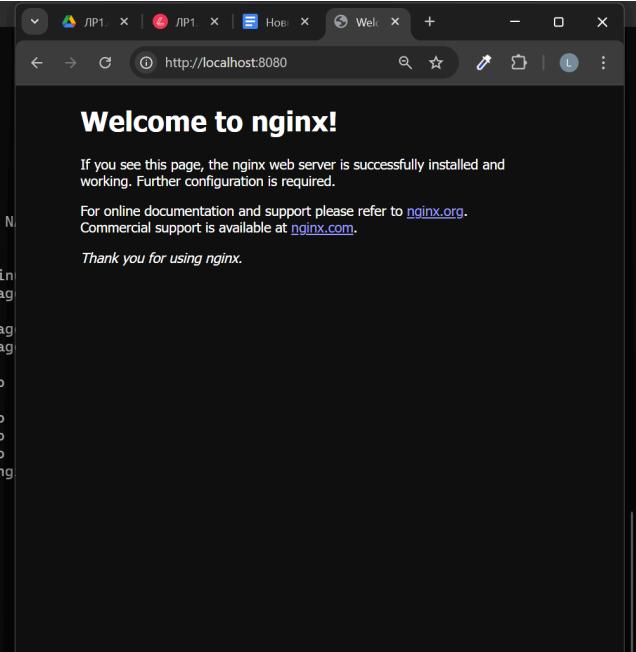
## 2) Запуск hello world

### Часть 1: Базовые команды Docker. Работа с образами и контейнерами

```
PS C:\Users\lyaho> docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
<none>            <none>  6c5db3b991ba  22 hours ago   727MB
<none>            <none>  1bf13fc47662  5 days ago    1.61GB
node               latest   92c60ed197ed  10 days ago   1.64GB
postgres           latest   1ffc019dae94  10 days ago   643MB
dpage/pgadmin4     latest   5d9624a93634  12 days ago   787MB
alpine             latest   4b7ce07002c6  2 weeks ago   12.8MB
nginx              alpine   61e01287e546  2 weeks ago   79.7MB
ubuntu             latest   66460d557b25  3 weeks ago   117MB
<none>            <none>  b5b154cfefbf  4 weeks ago   643MB
hello-world        latest   6dc565aa6309  2 months ago  20.3kB
ubuntu             20.04   8feb4d8ca535  6 months ago  109MB
openjdk            latest   9b448de897d2  2 years ago   727MB
PS C:\Users\lyaho> docker ps
CONTAINER ID        IMAGE      COMMAND       CREATED        STATUS        PORTS     NAMES
PS C:\Users\lyaho> docker ps -a
CONTAINER ID        IMAGE      COMMAND       CREATED        STATUS        PORTS     NAMES
ad390a030421        hello-world "hello"      About a minute ago  Exited (0)  About a minute ago
bdb54f4a5995        ubuntu     "/bin/bash"   23 hours ago   Exited (255)  2 minutes ago
upta
ea8d9c226e31        openjdk    "jshell"     23 hours ago   Exited (255)  2 minutes ago
fa1c797cfb8f        openjdk    "jshell"     23 hours ago   Exited (0)  23 hours ago
ver
870fa3dec981        ubuntu:20.04 "echo 'hello docker'"  4 days ago    Exited (0)  4 days ago
t
0af88ffe5223        b5b154cfefbf  "docker-entrypoint.s..."  4 days ago    Exited (255)  23 hours ago
7c09b8e66ea4        b5b154cfefbf  "docker-entrypoint.s..."  5 days ago    Exited (255)  23 hours ago
e2613a4727eb        b5b154cfefbf  "docker-entrypoint.s..."  5 days ago    Exited (255)  23 hours ago
PS C:\Users\lyaho> docker run -d -p 8080:80 --name my_webserver nginx:alpine
3dd6d5aa3cde833580f1e86c5b5eea7a01cf5dbfa7fdb843356dc7fe605ed108
PS C:\Users\lyaho>
```

## 3) Просмотр информации

```
PS C:\Users\lyaho> docker ps
CONTAINER ID        IMAGE      COMMAND       CREATED        STATUS        PORTS     NAMES
PS C:\Users\lyaho> docker ps -a
CONTAINER ID        IMAGE      COMMAND       CREATED        STATUS        PORTS     NAMES
ad390a030421        hello-world "hello"      About a min
bdb54f4a5995        ubuntu     "/bin/bash"   23 hours ago   Exited (0)
upta
ea8d9c226e31        openjdk    "jshell"     23 hours ago   Exited (0)
fa1c797cfb8f        openjdk    "jshell"     23 hours ago   Exited (0)
ver
870fa3dec981        ubuntu:20.04 "echo 'hello docker'"  4 days ago    Exited (0)
t
0af88ffe5223        b5b154cfefbf  "docker-entrypoint.s..."  4 days ago    Exited (0)
7c09b8e66ea4        b5b154cfefbf  "docker-entrypoint.s..."  5 days ago    Exited (0)
e2613a4727eb        b5b154cfefbf  "docker-entrypoint.s..."  5 days ago    Exited (0)
PS C:\Users\lyaho> docker run -d -p 8080:80 --name my_webserver ng
3dd6d5aa3cde833580f1e86c5b5eea7a01cf5dbfa7fdb843356dc7fe605ed108
PS C:\Users\lyaho>
```



## 4) Запуск простого контейнера

```
Windows PowerShell      +  X
7c09b8e66ea4  b5b154cf5bf  "docker-entrypoint.s..."  5 days ago      Exited (255) 23 hours ago      5432/tcp  my_postgres_db
e2613a4727eb  b5b154cf5bf  "docker-entrypoint.s..."  5 days ago      Exited (255) 23 hours ago      5432/tcp  habr_pg
PS C:\Users\lyaho> docker run -d -p 8080:80 --name my_webserver nginx:alpine
3dd6d5aa3cde833580f1e86c5b5ea7a01cf5dbfa7fdb843356dc7fe605ed108
PS C:\Users\lyaho> docker ps
CONTAINER ID  IMAGE          COMMAND           CREATED        STATUS          PORTS
ES
3dd6d5aa3cde  nginx:alpine  "/docker-entrypoint..."  58 seconds ago  Up 57 seconds  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  my_webserver
PS C:\Users\lyaho> docker stop my_webserver
Error response from daemon: No such container: my_webserver
PS C:\Users\lyaho> docker ps -a
CONTAINER ID  IMAGE          COMMAND           CREATED        STATUS          PORTS
NAMES
3dd6d5aa3cde  nginx:alpine  "/docker-entrypoint..."  About a minute ago  Up About a minute  0.0.0.0:8080->80/tcp, [::]:8080->80/tcp  my_webserver
ad390a030421  hello-world   "/hello"          3 minutes ago   Exited (0) 3 minutes ago
pedantic_dirac
ddb54f4a5995  ubuntu         "/bin/bash"       23 hours ago    Exited (255) 4 minutes ago
quirky_brahmagupta
ea8d9c226e31  openjdk        "jshell"         23 hours ago    Exited (255) 4 minutes ago
myjava
fa1c797cfb8f  openjdk        "jshell"         23 hours ago    Exited (0) 23 hours ago
suspicious_hoover
870fa3dec981  ubuntu:20.04   "echo 'hello docker'"  4 days ago     Exited (0) 4 days ago
fervent_lamport
0af88ffe5223  b5b154cf5bf  "docker-entrypoint.s..."  4 days ago     Exited (255) 23 hours ago  5432/tcp
postgresdb
7c09b8e66ea4  b5b154cf5bf  "docker-entrypoint.s..."  5 days ago     Exited (255) 23 hours ago  5432/tcp
my_postgres_db
e2613a4727eb  b5b154cf5bf  "docker-entrypoint.s..."  5 days ago     Exited (255) 23 hours ago  5432/tcp
habr_pg
PS C:\Users\lyaho> docker rm my_webserver
Error response from daemon: cannot remove container "my_webserver": container is running: stop the container before removing or force
remove
PS C:\Users\lyaho>
```

## 5) Остановка и удаление контейнера

## Часть 2: Запуск PostgreSQL в контейнере

```
0000@MSI MINGW64 ~
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
0000@MSI MINGW64 ~
$ docker run -d \
> --name my_postgres_db \
> -e POSTGRES_USER=postgres \
> -e POSTGRES_PASSWORD=postgres \
> -e POSTGRES_DB=test_db \
> -p 5432:5432 \
> postgres:15
f0684d1a2124340e8cb71c52daa0e3f7566ec425f25ce14fc20e1075624a9117

0000@MSI MINGW64 ~
$ doker ps
bash: doker: command not found

0000@MSI MINGW64 ~
$ docker ps
psql (15.14 (Debian 15.14-1.pgdg13+1))
Type "help" for help.

test_db=# \dt
Did not find any relations.
test_db=# CREATE TABLE users (id SERIAL PRIMARY KEY, name VARCHAR(50));
CREATE TABLE
test_db=# INSERT INTO users (name) VALUES ('Alice'), ('Bob');
INSERT 0 2
test_db=# SELECT * FROM users;
 id | name
----+-----
  1 | Alice
  2 | Bob
(2 rows)

test_db=# |
```

- 6) Запуск контейнера с Postresql, а также проверка, что контецнер запущен (docker ps) и подключение БД прямо из контейнера

Часть 3: Подключение к БД через pgAdmin из второго контейнера

```
MINGW64:/c/Users/lyaho
0000@MSI MINGW64 ~
$ docker network connect my_network my_postgres_db

0000@MSI MINGW64 ~
$ docker run -d \
> --name my_pgadmin \
> -e PGADMIN_DEFAULT_EMAIL=admin@example.com \
> -e PGADMIN_DEFAULT_PASSWORD=admin \
> -p 8080:80 \
> --network my_network \
> dpage/pgadmin4
3c64cea6313543b2086ee9b093d6dff698034c841cbc0d6e7e18b0a0ccf587c7
docker: Error response from daemon: failed to set up container networking: drive
r failed programming external connectivity on endpoint my_pgadmin (c3d0c51f2101d
c35clf933af48c14f368cd1c67bff549df8c0e0a6b3a4e98a33): Bind for 0.0.0.0:8080 fail
ed: port is already allocated

Run 'docker run --help' for more information

0000@MSI MINGW64 ~
$ docker stop my_webserver
my_webserver

0000@MSI MINGW64 ~
$ docker run -d \
> --name my_pgadmin \
> -e PGADMIN_DEFAULT_EMAIL=admin@example.com
docker: 'docker run' requires at least 1 argument

Usage: docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
see 'docker run --help' for more information

0000@MSI MINGW64 ~
$ docker run -d \
> --name my_admin \
> -e PGADMIN_DEFAULT_EMAIL=admin@example.com \
> -e PGADMIN_DEFAULT_PASSWORD=admin \
> -p 8080:80 \
> --network my_network \
> dpage/pgadmin4
689d060f63acb684adb13cafd19fb43e9171ce9fa121334088897d1bde2b85f6

0000@MSI MINGW64 ~
$ |
```

7) Создание сети Docker, подключение контейнера к Postresq и запуск pgAdmin в той же сети

```
1 SELECT * FROM users;
2
```

	<b>id</b> [PK] integer	<b>name</b> character varying (50)
1	1	Alice
2	2	Bob

8) Выполнение запроса через query Tool в pgAdmin

Часть 4: Сохранение данных с помощью Томов (Volumes)

```
0000@MSI MINGW64 ~
$ docker volume create postgres_data
postgres_data

0000@MSI MINGW64 ~
$ docker volume ps
docker: unknown command: docker volume ps

usage: docker volume COMMAND

Run 'docker volume --help' for more information

0000@MSI MINGW64 ~
$ docker volume ls
DRIVER      VOLUME NAME
local      00eca7635c7f0eb51bb4699208234ee0090bac2db8b0
8db65ac33bb5f1a8463d
local      1a04fcf96dc5255b4b82aacbbed48f1303e64b46886a
5fd964b4234e7fb21e7a
local      02c7d371d75393d5b363140e98b4dd60fdfed90595fd
364744fbb6436b181c42
local      7d127ca7c04ec4f1bba302b0559dbd5e71e90ee4cd94
c73997f520aad91297ff
local      47f823cd0dee71fecd337d190bcb069e109a78cdbfa9
f8b6a5afdf00226ab191
local      774cf60fbe03c00b7156585665d2ecc1f8a08d1f6e4b
3b301930194606b9a7ea
local      817e30500300d524f8156ae3fd5a2752cd43f6f4f830
8ce3fe47ec7d2bfeea96
local      8060572a4a3410d07ec59796c97141e86ca875dfc540
84371ac57a88820df887
```

## 9) Создание тома для хранения данных БД

```

0000@MSI MINGW64 ~
$ docker run -d \
> --name my_postgres_db_persistent \
> -e POSTGRES_USER=postgres \
> -e POSTGRES_PASSWORD=postgres \
> -e POSTGRES_DB=test_db \
> -p 5432:5432 \
> -v postgres_data:/var/lib/postgresql/data \
> --network my_network \
> postgres:15
8e230d936f2aef82b56c8fdb66bc35dafb64bc4c694a88602ce21
72cff32d16

0000@MSI MINGW64 ~
$ docker run -d \
> --name my_pgadmin \
> -e PGADMIN_DEFAULT_EMAIL=admin@example.com \
> -e PGADMIN_DEFAULT_PASSWORD=admin \
> -p 8080:80 \
> --network my_network \
> dpage/pgadmin4
02d8c1ffa9b615a6af48a09995563027caaf420050e81e3ba9eaea
bdf5d1268e

0000@MSI MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE          COMMAND
CREATED        STATUS         PORTS
                           NAMES
02d8c1ffa9b6   dpage/pgadmin4   "/entrypoint.sh"
    About a minute ago   Up About a minute   0.0.0.0:80
80->80/tcp, [::]:8080->80/tcp      my_pgadmin

```

10) Запуск нового контейнера с PostgreSQL, подключив том

Ответы на вопросы:

1.Что такое Docker?

Docker — это платформа для контейнеризации, которая позволяет упаковывать приложение со всеми его зависимостями в изолированную среду (контейнер) и запускать его на любом компьютере, независимо от операционной системы.

2.Для чего нужны тома и сети docker?

Тома Docker используются для перманентного хранения данных за пределами контейнера, чтобы они не терялись при его удалении или перезапуске. Сети Docker используются для коммуникации между

контейнерами, позволяя им взаимодействовать друг с другом, а также с внешним миром и хост-системой.

### 3.Как подключится к контейнеру и выполнить в нём команды?

Чтобы подключиться к контейнеру и выполнить в нём команды, используйте команду docker exec с флагами -it для интерактивного режима и последующей командой, например, docker exec -it my-container bash. Также можно использовать команду docker attach, которая подключает терминал к основному процессу контейнера, но exec более гибкая и позволяет запускать новые команды.

### 4.Для чего нужен pgAdmin?

pgAdmin нужен для управления базами данных PostgreSQL через графический интерфейс, что позволяет выполнять различные задачи, такие как создание и изменение баз данных, написание и выполнение SQL-запросов, а также визуализация структуры данных. Он служит инструментом для администрирования и настройки СУБД PostgreSQL, предоставляя удобный интерфейс для работы с ней без необходимости использовать только командную строку.