

Министерство науки и высшего образования Российской Федерации  
ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»  
Институт радиоэлектроники и информационных технологий – РТФ

# **Основы работы с Docker и PostgreSQL**

## **Лабораторное задание №1**

Группа:

РИМ-150950

Студент:

Владимиров Н.А.

Преподаватель:

Кузьмин Д. И.

Екатеринбург  
2025

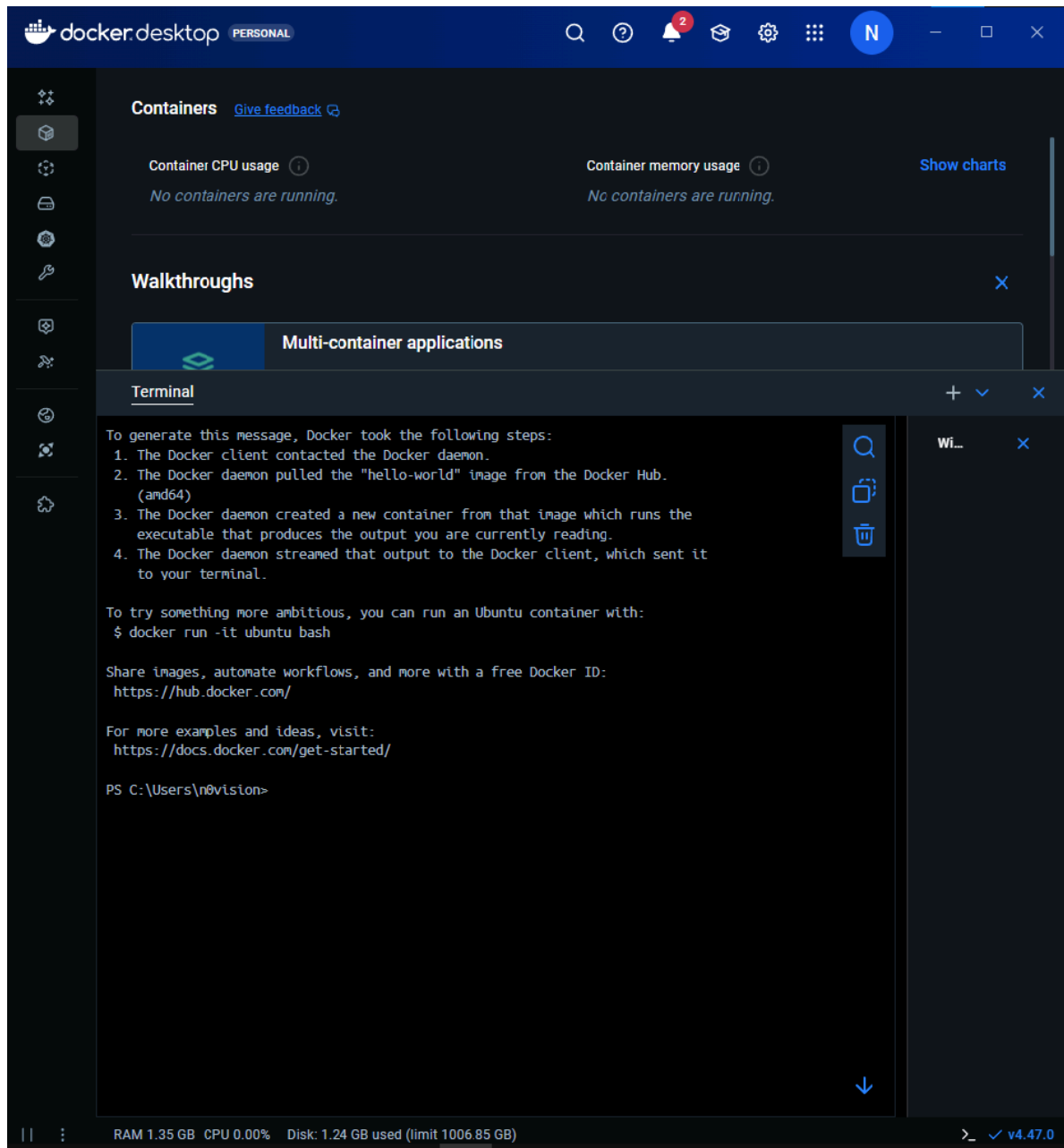
**Цель работы:** Освоить фундаментальные концепции и базовые операции Docker: создание образов, запуск контейнеров, управление ими, работа с сетями и томами. На практике закрепить навыки, запустив изолированную базу данных PostgreSQL и подключившись к ней извне.

**Задачи:**

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

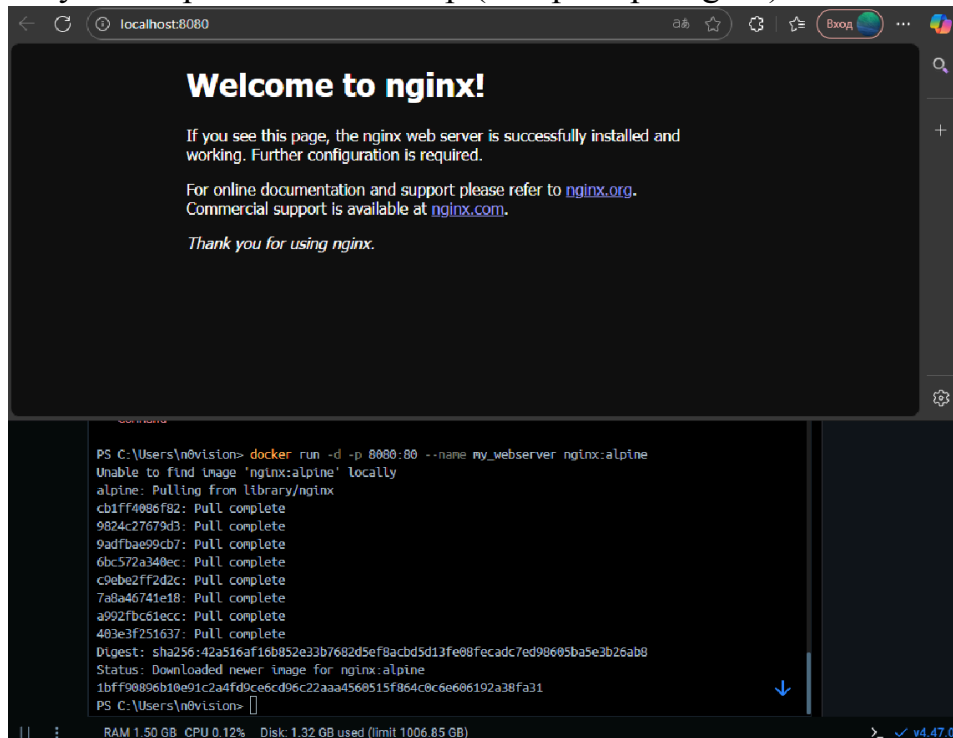
**Ход работы:**

## Часть 0: Установка и проверка Docker



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

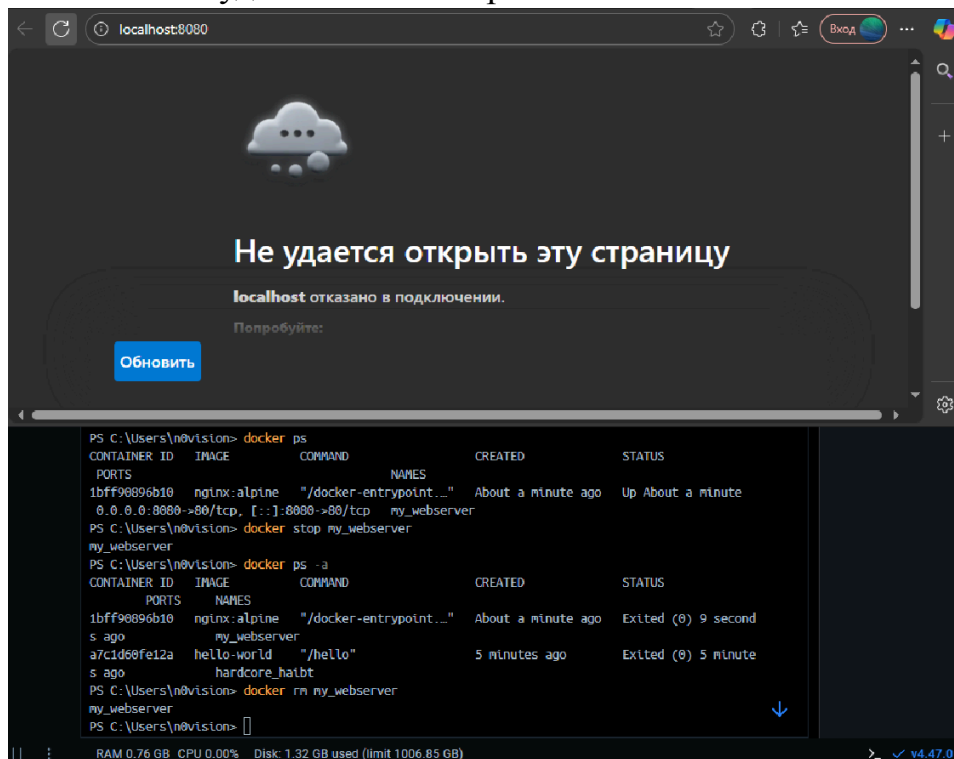
### Запустим простой контейнер (на примере Nginx):



The screenshot shows a web browser window at localhost:8080 displaying the "Welcome to nginx!" page. Below the browser, a terminal window shows the execution of the command `docker run -d -p 8080:80 --name my_webserver nginx:alpine`. The terminal output indicates that the image was pulled from the library and the container is running.

```
PS C:\Users\n0vision> docker run -d -p 8080:80 --name my_webserver nginx:alpine
Unable to find image 'nginx:alpine' locally
alpine: Pulling from library/nginx
cb1ff4086f82: Pull complete
9824c27679d3: Pull complete
9adfbac99cb7: Pull complete
6bc572a348ec: Pull complete
c9eb2ff2d2c: Pull complete
7a8a46741e18: Pull complete
a992fbc61ecc: Pull complete
483e3f251637: Pull complete
Digest: sha256:42a516af16b952e33b7682d5ef8acbd5d13fe08fecadc7ed98605ba5e3b26ab8
Status: Downloaded newer image for nginx:alpine
1bfff90896b10e91c2a4fd9ce6cd96c22aaa4568515f864c0c6e606192a38fa31
PS C:\Users\n0vision>
```

### Остановим и удалим контейнер:



The screenshot shows a web browser window at localhost:8080 displaying a 404 error page. Below the browser, a terminal window shows the execution of the command `docker ps` to list running containers, followed by `docker stop my_webserver` to stop the container, and `docker rm my_webserver` to remove it.

```
PS C:\Users\n0vision> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
1bfff90896b10 nginx:alpine "/docker-entrypoint..." About a minute ago Up About a minute
0.0.0.0:8080->80/tcp, [::]:8080->80/tcp my_webserver
PS C:\Users\n0vision> docker stop my_webserver
my_webserver
PS C:\Users\n0vision> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
1bfff90896b10 nginx:alpine "/docker-entrypoint..." About a minute ago Exited (0) 9 second
s ago my_webserver
a7c1d68fe12a hello-world "/hello" 5 minutes ago Exited (0) 5 minute
s ago hardcore_hatbt
PS C:\Users\n0vision> docker rm my_webserver
my_webserver
PS C:\Users\n0vision>
```

## Часть 2: Запуск PostgreSQL в контейнере

Запустим контейнер с PostgreSQL и подключимся к БД прямо из контейнера:

The screenshot shows the Docker Desktop interface. At the top, a table lists the containers:

Name	Container ID	Image	Port(s)	Actions
my_postgres_db	6fb1a96fb80d	postgres:11	5432:5432	[Restart] [Stop] [Logs] [Remove]

Below the table is a terminal window for the 'my\_postgres\_db' container. The terminal shows the following commands and output:

```
0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp my_postgres_db
PS C:\Users\n8vision> ^C
PS C:\Users\n8vision> docker exec -it my_postgres_db psql -U postgres -d test_db
psql (15.14 (Debian 15.14-1.pgdg13+1))
Type "help" for help.

test_db=# help
You are using psql, the command-line interface to PostgreSQL.
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit
test_db=#
```

## Часть 3: Подключение к БД через pgAdmin из второго контейнера

Создадим сеть Docker, подключим контейнер с PostgreSQL к сети и запустим pgAdmin в той же сети

The screenshot shows the Docker Desktop interface with two containers listed in a network:

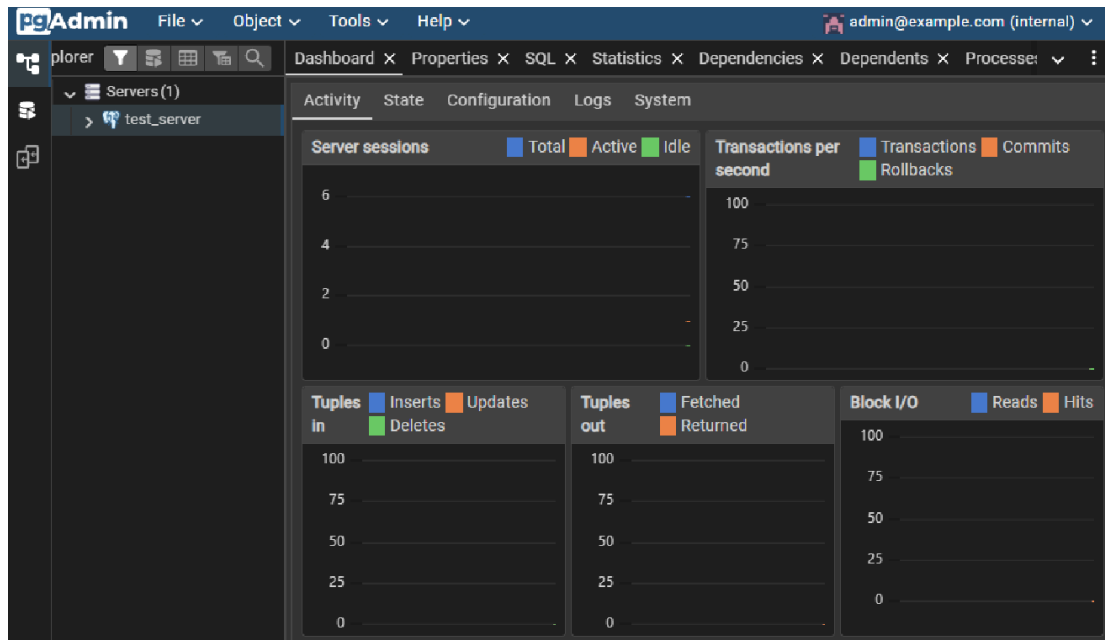
Name	Container ID	Image	Port(s)	Actions
my_postgres_db	6fb1a96fb80d	postgres:11	5432:5432	[Restart] [Stop] [Logs] [Remove]
my_pgadmin	5b0777b09f	dgryse/pgadmin:8.0		[Restart] [Stop] [Logs] [Remove]

Below the table is a terminal window for the 'my\_pgadmin' container. The terminal shows the following commands and output:

```
godot@4: /#
root@my_pgadmin:~# cat /etc/docker/daemon.json | jq '. "log-driver": "json-file", "log-opts": {"type": "json"}'
{
  "log-driver": "json-file",
  "log-opts": {
    "type": "json"
  }
}
root@my_pgadmin:~# docker exec -it my_postgres_db psql -U postgres -d test_db
psql (15.14 (Debian 15.14-1.pgdg13+1))
Type "help" for help.

test_db=# help
You are using psql, the command-line interface to PostgreSQL.
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit
test_db=#
```

## Настроим подключение в pgAdmin



Через Query Tool в pgAdmin выполним запрос:

The screenshot shows the pgAdmin Query Tool interface. The query editor contains the following SQL query:

```
SELECT * FROM user
```

The query has been executed, and the results are displayed in the 'Data Output' tab. The results show a single row with the column 'user name' and the value 'postgres'.

	user name
1	postgres

The interface includes a sidebar with a tree view of the database structure and a top menu bar with options like File, Object, Tools, and Help.

## Часть 4: Сохранение данных с помощью Томов (Volumes)

Остановим и удалим текущий контейнер с БД

```
PS C:\Users\n0vision> docker stop 6fb1a96fb80d
6fb1a96fb80d
PS C:\Users\n0vision> docker rm 6fb1a96fb80d
6fb1a96fb80d
PS C:\Users\n0vision>
```

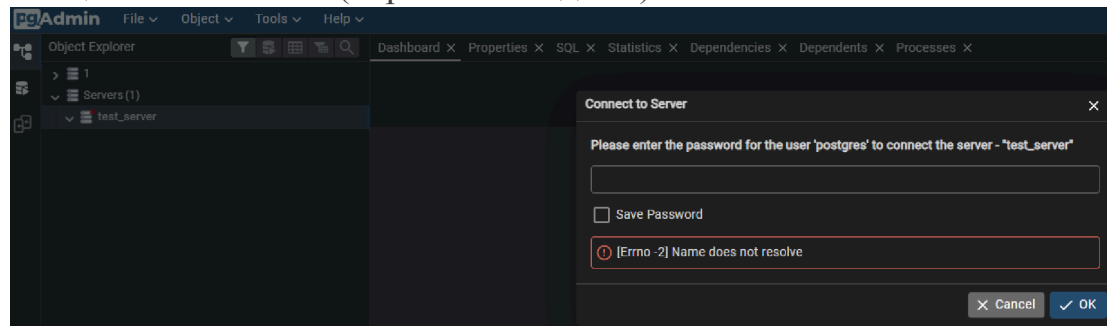
Создадим том для хранения данных БД и запустим новый контейнер с PostgreSQL, подключив том

*Далее я не понял: почему мы останавливаем и удаляем текущий контейнер с БД, после чего пытаемся запустить несуществующий контейнер?*

**Остановите и удалите текущий контейнер с БД**

- Запустите контейнеры `docker start my_postgres_db my_pgadmin`

И еще такая ошибка (пароли совпадают)



**Вывод:** В ходе данной лабораторной работы были успешно освоены основы работы с технологией контейнеризации Docker и системой управления базами данных PostgreSQL. Цель работы была достигнута, все поставленные задачи выполнены в полном объеме