

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

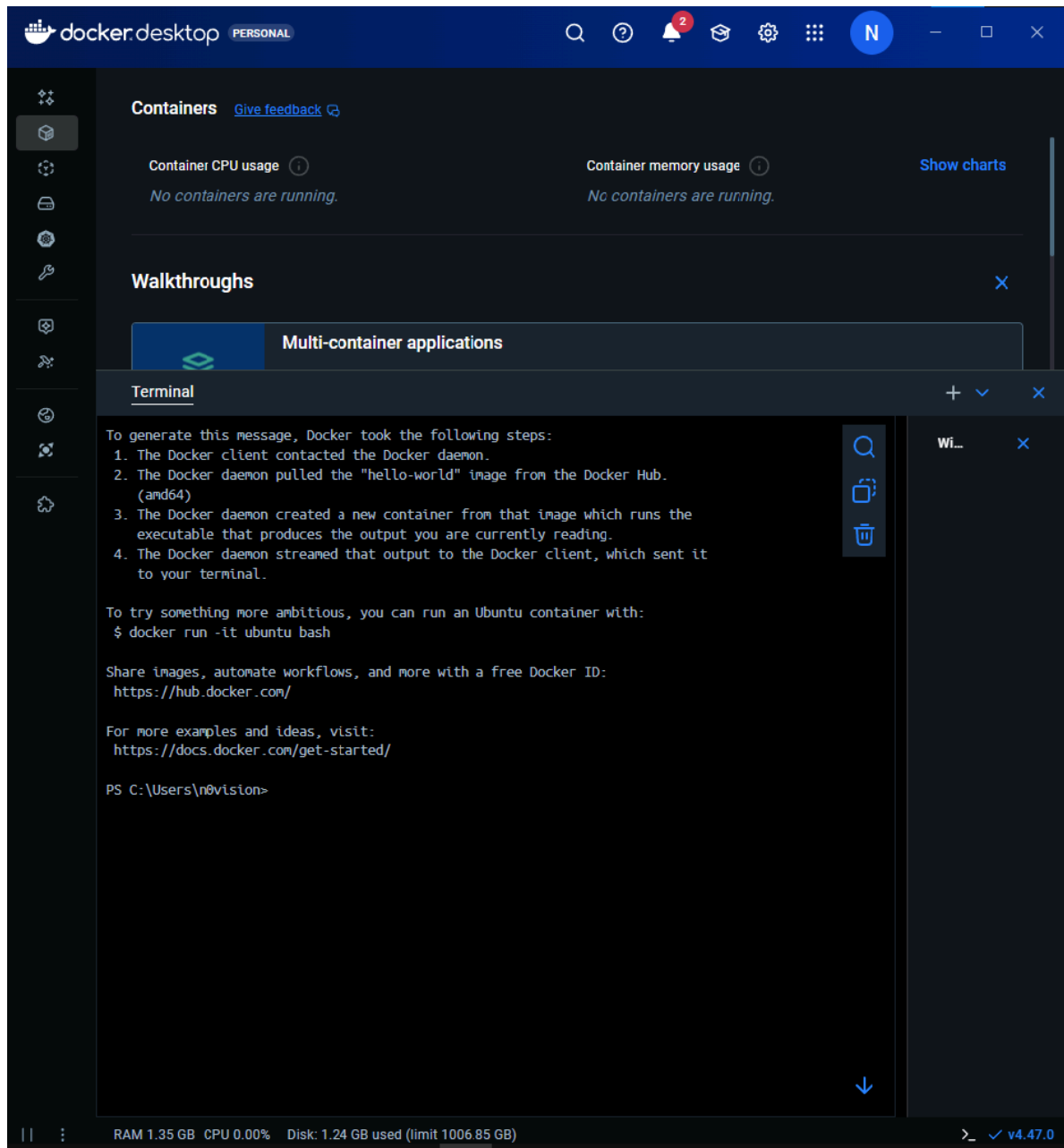
Цель работы: Освоить фундаментальные концепции и базовые операции Docker: создание образов, запуск контейнеров, управление ими, работа с сетями и томами. На практике закрепить навыки, запустив изолированную базу данных PostgreSQL и подключившись к ней извне.

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.

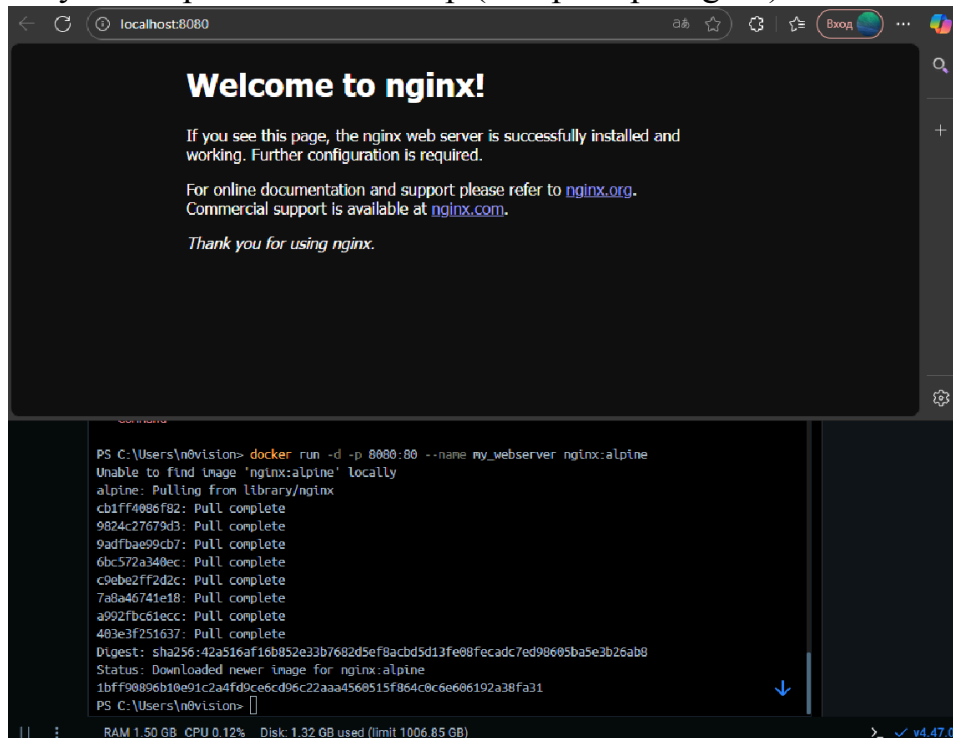
Ход работы:

Часть 0: Установка и проверка Docker

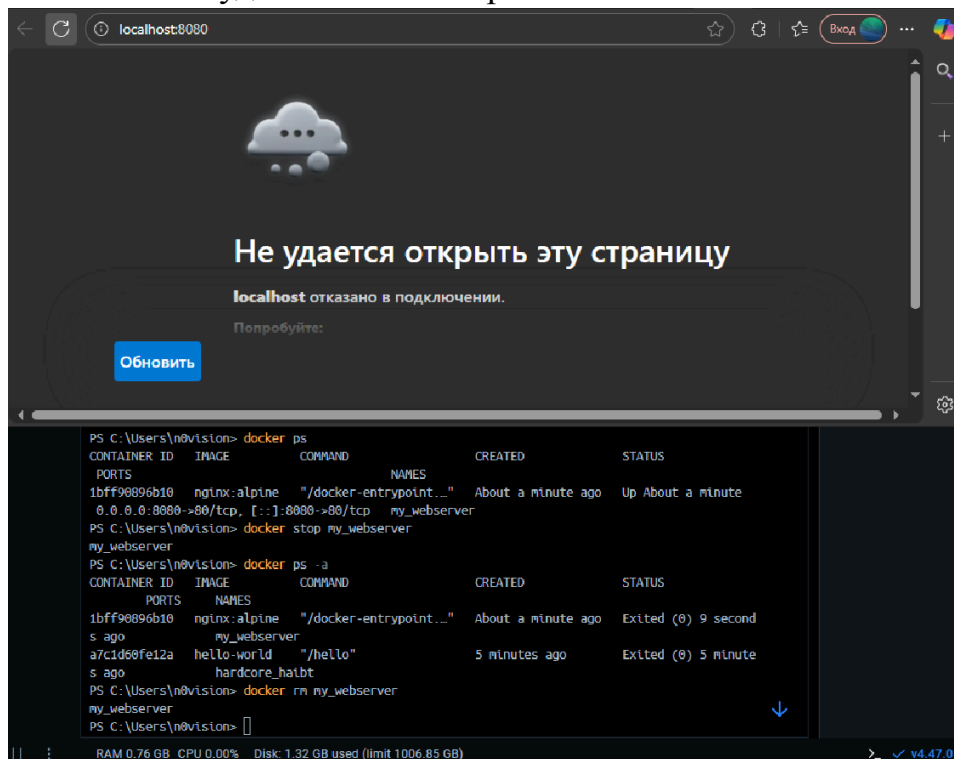


Часть 1: Базовые команды Docker. Работа с образами и контейнерами

Запустим простой контейнер (на примере Nginx):






Остановим и удалим контейнер:



Часть 2: Запуск PostgreSQL в контейнере

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

	Name	Container ID	Image	Port(s)	Actions
<input type="checkbox"/>	● my_postgres_db	6fb1a96fb80d	postgres:11	5432:5432	   

Terminal

```
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=#
```

Window...

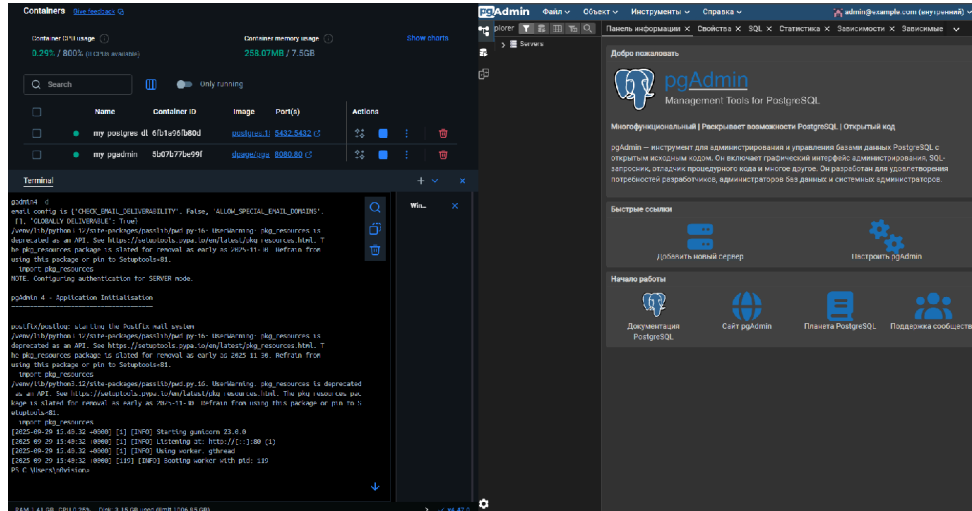
```

                                List of databases
   Name   | Owner   | Encoding | Collate  |  Ctype  | ICU Locale | Locale Provider |
-----+-----+-----+-----+-----+-----+-----+
Access privileges
test_db=# INSERT INTO users (name) VALUES ('Alice'), ('Bob');
INSERT 0 2
test_db=# SELECT * FROM ISERS;
ERROR:  relation "isers" does not exist
LINE 1: SELECT * FROM ISERS;
                        ^
test_db=# SELECT * FROM users;
 id | name
----+-----
  1 | Alice
  2 | Bob
(2 rows)

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



Настроим подключение в pgAdmin

```
List of databases
Name | Owner | Encoding | Collate | Ctype | ICU Locale | Locale Provider |
Access privileges
test_db=# INSERT INTO users (name) VALUES ('Alice'), ('Bob');
INSERT 0 2
test_db=# SELECT * FROM ISERS;
ERROR:  relation "isers" does not exist
LINE 1: SELECT * FROM ISERS;
                    ^
test_db=# SELECT * FROM users;
 id | name
----+-----
  1 | Alice
  2 | Bob
(2 rows)

test_db=#
```

Через Query Tool в pgAdmin выполним запрос:

Через Query Tool в pgAdmin выполним запрос:

The screenshot shows the pgAdmin Query Tool interface. On the left, the 'Servers' tree is expanded to 'test' > 'Databases (2)' > 'postgres' > 'test_db'. The main pane shows the 'Query' tab with the SQL query: `SELECT * FROM users;`. Below the query, the 'Data Output' tab is active, displaying the results of the query. The results are shown in a table with two columns: 'Id' (integer, primary key) and 'name' (character varying (50)). The table contains two rows: (1, Alice) and (2, Bob). The interface also includes a 'Query History' tab and a 'Scratch' tab. The 'Data Output' tab has a toolbar with icons for adding, deleting, and saving queries, as well as a 'SQL' button. The status bar indicates 'Showing rows: 1 to 2' and 'Page No: 1 of 1'.

	Id [PK] integer	name character varying (50)
1	1	Alice
2	2	Bob

Часть 4: Сохранение данных с помощью Томов (Volumes)

Остановим и удалим текущий контейнер с БД

```
PS C:\Users\n0vision> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
RTS
9a2fc9d6f09c   dpage/pgadmin4 "/entrypoint.sh"        16 minutes ago Up 15 minutes 0.
0.0.0:8080->80/tcp, [::]:8080->80/tcp   my_pgadmin
b5c1f555c0c3   postgres:15    "docker-entrypoint.s..." 22 minutes ago Up 9 minutes   0.
0.0.0:5432->5432/tcp, [::]:5432->5432/tcp my_postgres_db
PS C:\Users\n0vision> docker stop my_postgres_db
my_postgres_db
PS C:\Users\n0vision> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
9a2fc9d6f09c   dpage/pgadmin4 "/entrypoint.sh"        17 minutes ago Up 15 minutes 0.
0.0.0:8080->80/tcp, [::]:8080->80/tcp   my_pgadmin
b5c1f555c0c3   postgres:15    "docker-entrypoint.s..." 22 minutes ago Exited (0) 6 seconds ago
my_postgres_db
PS C:\Users\n0vision> docker rm my_postgres_db
my_postgres_db
```

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

```
PS C:\Users\n0vision> docker volume create postgres_data
postgres_data
PS C:\Users\n0vision> colume ls
colume : Имя "colume" не распознано как имя командлета, функции, файла сценария или выполняемой программы. Проверьте правильность написания имени, а также наличие и правильность пути, после чего повторите попытку.
строка:1 знак:1
+ colume ls
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (colume:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\n0vision> docker volume ls
DRIVER      VOLUME NAME
local       8ad2047f084028d5b5e9aacdeabce649f171bfffefab6aabd46b350571e0ecbc3
local       80b8f74ff8d1cb7c21b7592d5cfa6a81613a0a6ba15507593d1ada78c82e41f6
local       90162660fad46cc0d1a46e1a06d2241645b63bd7c726cd1f83a0c2aa68827ace
local       postgres_data
```



```
my_postgres_db_persistent 23979374b715 postgres:1! 5432:5

Terminal

PS C:\Users\n0vision> docker rm my_postgres_db
my_postgres_db
PS C:\Users\n0vision> docker volume create postgres_data
postgres_data
PS C:\Users\n0vision> colume ls
colume : Имя "colume" не распознано как имя командлета, функции, файла сценария или выполня
емой программы. Проверьте правильность написания имени, а также наличие и правильность пути
, после чего повторите попытку.
строка:1 знак:1
+ colume ls
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (colume:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\n0vision> docker volume ls
DRIVER      VOLUME NAME
local       8ad2047f084028d5b5e9aacdeabce649f171bffffeab6aabd46b350571e0ecbc3
local       80b8f74ff8d1cb7c21b7592d5cfa6a81613a0a6ba15507593d1ada78c82e41f6
local       90162660fad46cc0d1a46e1a06d2241645b63bd7c726cd1f83a0c2aa68827ace
local       postgres_data
PS C:\Users\n0vision> docker run --name my_postgres_db_persistent -p 5432:5432 -e POSTGRES_U
SER=postgres -e POSTGRES_PASSWORD=postgres -e POSTGRES_DB=test_db -v postgres_data:/var/lib/
postgresql/data --network my_network -d postgres:15
23979374b7154834b4a0c72e001c8d84358117f63624e3ade517d6f522556ead
```

Проверим сохранность данных:

test_db

Casts

Catalogs

Event Triggers

Extensions

Foreign Data W

Languages

Publications

Schemas

Subscriptions

Login/Group Roles

Tablespaces

Query

Query History

Scratch

1

SELECT * FROM users;

Data Output

Messages

Notifications

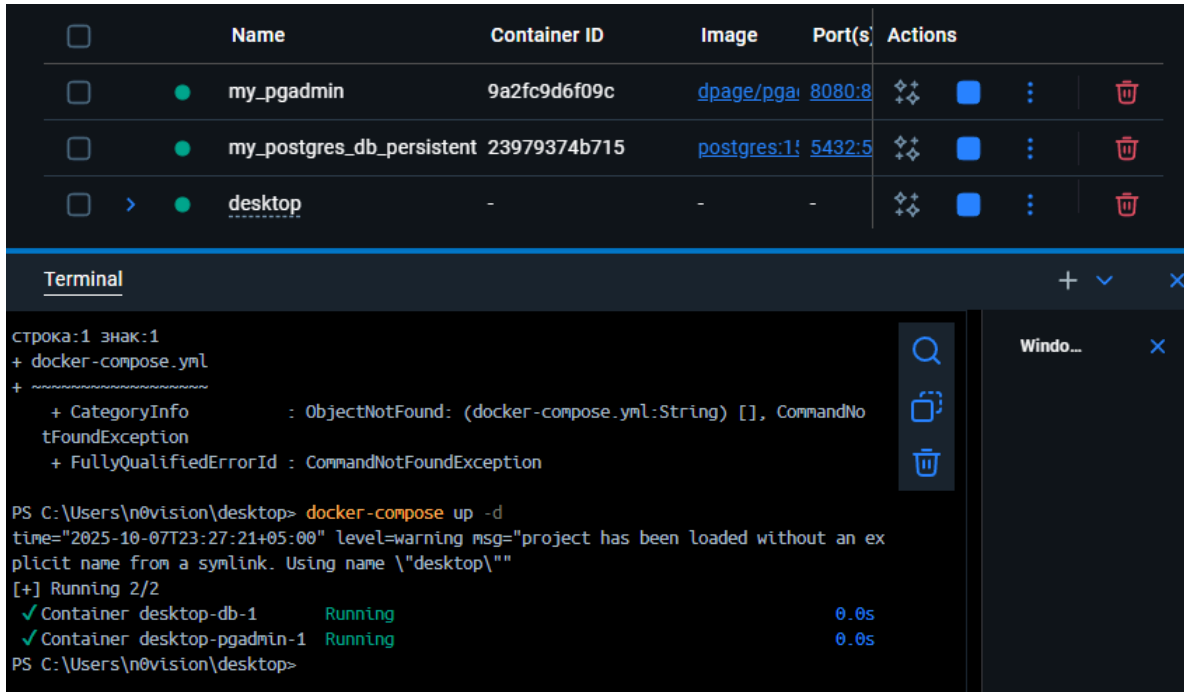
Showing rows: 1 to 2

Page No: 1

of 1

	id	name
	[PK] integer	character varying (50)
1	1	Alice
2	2	Bob

Часть 5: Перенос конфигурации контейнеров в docker-compose.yaml



The screenshot shows the Docker Desktop interface. At the top, there is a table of containers:

Name	Container ID	Image	Port(s)	Actions
my_pgadmin	9a2fc9d6f09c	dpage/pgadmin	8080:8	[Stop] [Restart] [Refresh] [Delete]
my_postgres_db_persistent	23979374b715	postgres:1	5432:5	[Stop] [Restart] [Refresh] [Delete]
desktop	-	-	-	[Stop] [Restart] [Refresh] [Delete]

Below the table is a terminal window titled "Terminal". It shows the following output:

```
строка:1 знак:1
+ docker-compose.yml
+
+ CategoryInfo          : ObjectNotFound: (docker-compose.yml:String) [], CommandNo
tFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\n0vision\desktop> docker-compose up -d
time="2025-10-07T23:27:21+05:00" level=warning msg="project has been loaded without an ex
plicit name from a symlink. Using name \"desktop\""
[+] Running 2/2
 ✓ Container desktop-db-1      Running      0.0s
 ✓ Container desktop-pgadmin-1 Running      0.0s
PS C:\Users\n0vision\desktop>
```

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.

Ответы на вопросы:

1. Что такое docker?

Платформа для автоматизации разработки, доставки и развёртывания приложений в средах с поддержкой контейнеризации.

2. Для чего нужны тома и сети docker?

Для управления взаимодействием контейнеров и хранения данных.

3. Как подключиться к контейнеру и выполнить в нём команды?

`docker exec`.

4. Для чего нужен pgAdmin?

Для управления базами данных PostgreSQL через GUI.