

Департамент образования и науки города Москвы
Государственное автономное образовательное учреждение высшего
образования города Москвы
«Московский городской педагогический университет»
Институт цифрового образования
Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

**Интеграция и развертывание программного обеспечения с помощью
контейнеров**

Лабораторная работа 1.1

Установка и настройка Docker. Работа с контейнерами в Docker

Выполнила: Ванярина Ю.А., группа: АДЭУ-211

Преподаватель: Босенко Т.М.

Москва

2025

Цель работы: освоить процесс установки и настройки Docker, научиться работать с контейнерами и образами Docker.

Задачи:

- Установить Docker на локальный компьютер.
- Проверить корректность установки Docker.
- Ознакомиться с основными командами Docker CLI для работы с образами и контейнерами.
- Выполнить индивидуальное задание.

ПОДГОТОВКА К РАБОТЕ

1. Проверен ip адрес в виртуальной машине. (192.168.1.81)
2. Выполнено подключение к виртуальной машине по удаленному доступу

ХОД РАБОТЫ

1. Установка Docker.

```
sudo apt update
sudo apt install apt-transport-https ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io
```

Docker был успешно установлен, проверка прошла с использованием ***docker --version*** (Рисунок 1)

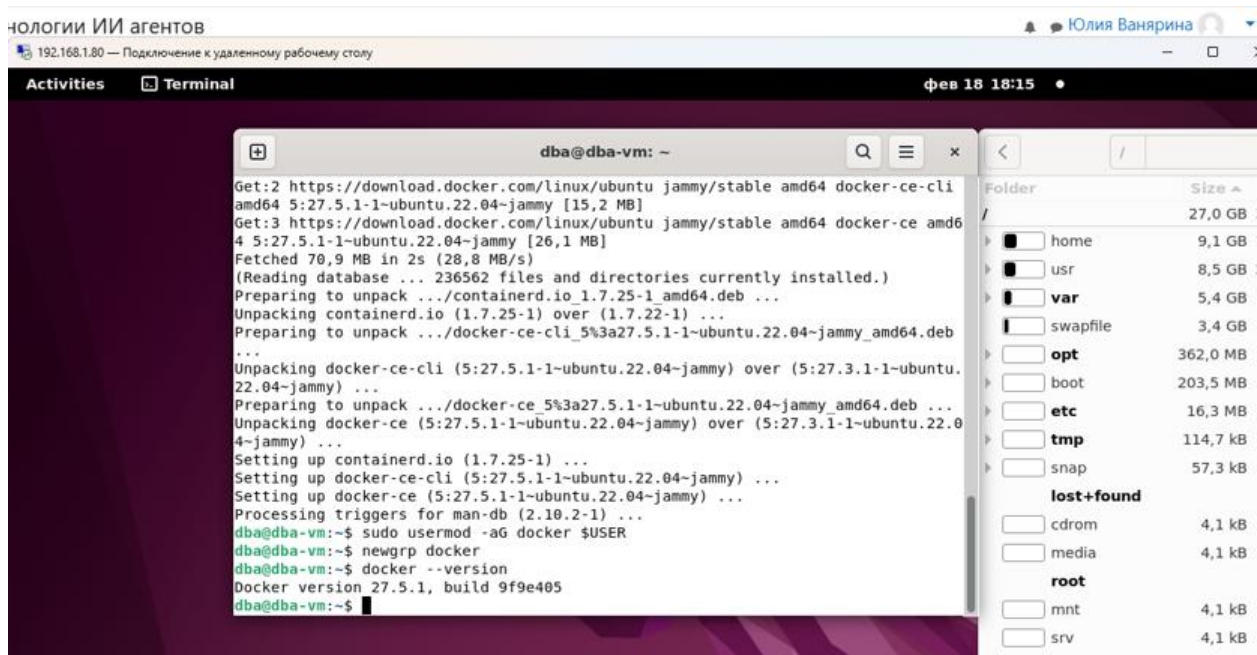


Рисунок 1 – проверка версии Docker

2. Выполните команду `docker run hello-world`, чтобы проверить, что Docker запущен и может загружать образы и запускать контейнеры.

На рисунке 2 показаны образы после выполнения команды ***docker run hello-world***

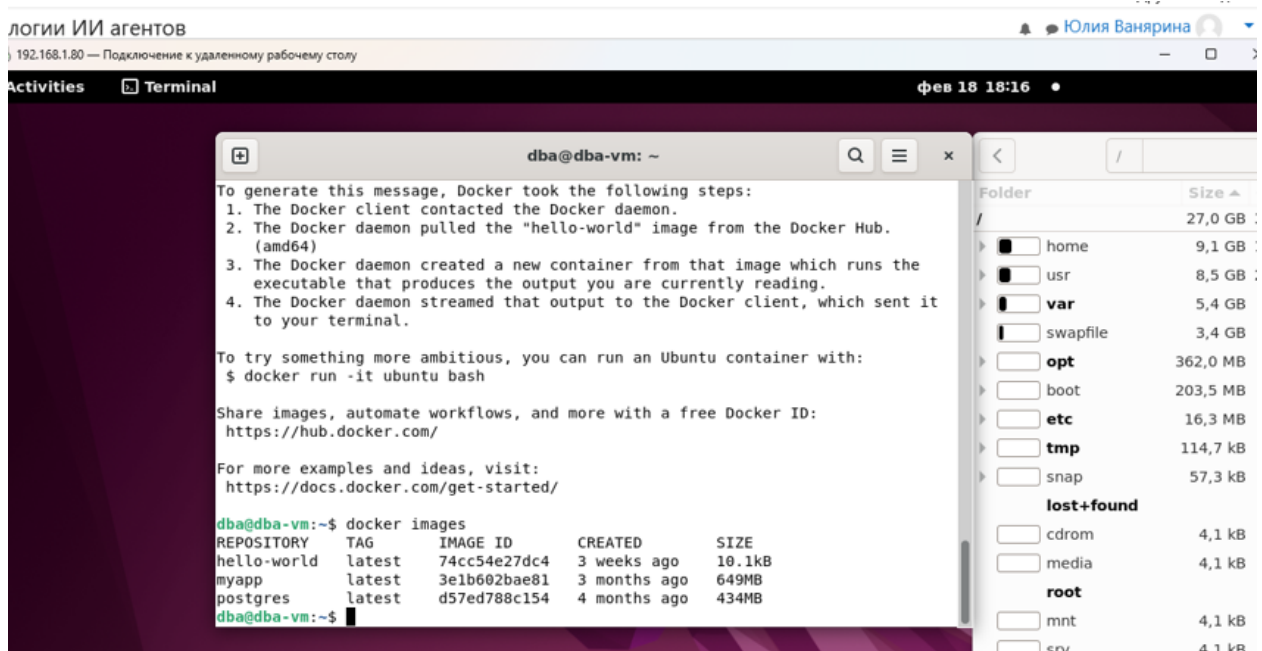


Рисунок 2 – образы в docker

Далее были выполнены команды *docker ps* и *docker ps -a*, чтобы посмотреть запущенные контейнеры и все контейнеры, включая остановленные, соответственно. (Рисунок 3)

```

dba@dba-vm:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
dba@dba-vm:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
653b021e7e43   hello-world    "/hello"                55 seconds ago Exited (0)    distracted_jemison
a457dbff5f72   myapp          "python -m uvicorn s..." 3 months ago   Exited (255)  4 weeks ago   0.0.0.0:8000->8000/tcp, :::8000->8000/tcp myapp
fc7c96c36fe7   postgres      "docker-entrypoint.s..." 3 months ago   Exited (255)  4 weeks ago   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp mydb
dba@dba-vm:~$

```

Рисунок 3 – просмотр контейнеров

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Вариант 15. Загрузить образ wordpress, запустить контейнер, настроить переменные среды для подключения к базе данных и проверить доступность веб-сайта WordPress.

Источник: <https://timeweb.cloud/tutorials/docker/kak-ustanovit-wordpress-s-pomoshchyu-docker>

Для реализации задания были созданы 3 контейнера:

- Nginx – веб-сервер
- Wordpress – система управления контентом
- Mysql – база данных

Информация о контейнерах размещена в следующих каталогах (рисунок 4)

```

mkdir -p ~/wordpress
cd ~/wordpress
mkdir -p nginx/
mkdir -p logs/
mkdir -p logs/nginx
mkdir -p data/
mkdir -p data/html
mkdir -p data/mysql

```

```

dba@dba-vm:~/wordpress$ mkdir -p nginx/
mkdir -p logs/
mkdir -p logs/nginx
mkdir -p data/
mkdir -p data/html
mkdir -p data/mysql
dba@dba-vm:~/wordpress$ ls
data logs nginx
dba@dba-vm:~/wordpress$

```

Рисунок 4 – система каталогов

Далее был создан конфигурационный файл для Nginx в ранее созданной директории, предназначенной для него (рисунок 5)

nano ~/wordpress/nginx/nginx.conf

```

GNU nano 6.2 /home/dba/wordpress/nginx/nginx.conf *
server {
    listen 80 default_server;
    server_name _;

    root /var/www/html;
    index index.php;

    access_log /var/log/nginx/site-access.log;
    error_log /var/log/nginx/site-error.log;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass wordpress:9000;
        fastcgi_index index.php;
        include fastcgi_params;
    }
}

```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
 ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line

Рисунок 5 – содержимое конфигурационного файла

Далее был создан файл с описанием зависимостей между контейнерами (рисунок 6)

sudo nano docker-compose.yml

```
dba@dba-vm:~/wordpress$ cat docker-compose.yml
version: '3.3'

services:
  nginx:
    image: nginx:latest
    volumes:
      - ./nginx:/etc/nginx/conf.d
      - ./data/html:/var/www/html
      - ./logs/nginx:/var/log/nginx
    ports:
      - "8080:80"
    links:
      - wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    volumes:
      - ./data/html:/var/www/html
    ports:
      - "80:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: mgpu_ico_etl_15
      WORDPRESS_DB_PASSWORD: K8DQAKTc
      WORDPRESS_DB_NAME: mgpu_ico_etl_15

  db:
    image: mysql:5.7
    volumes:
      - ./data/mysql:/var/lib/mysql
    restart: always
    environment:
      MYSQL_DATABASE: mgpu_ico_etl_15
      MYSQL_USER: mgpu_ico_etl_15
      MYSQL_PASSWORD: K8DQAKTc
```

Рисунок 6 – содержимое конфигурационного файла

Следующим шагом был запущен контейнер с помощью **docker compose up -d** (рисунок 7)

```
192.168.1.80 — Подключение к удаленному рабочему столу
dba@dba-vm:~/wordpress$ docker compose up -d
WARN[0000] /home/dba/wordpress/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid pote
[+] Running 17/42
   wordpress [#####] Pulling
   ✓ 814b6ecb84b0 Download complete
   : a4e58aa84c36 Downloading [=====] 31.72MB/104.3MB
   ✓ b545bb7ff18e Download complete
   : 8ca47539e139 Downloading [=====] 6.144MB/20.12MB
   : ea823f46cc3c Waiting
   : bcbeeb454049 Waiting
   : 68d70c2b9fc9 Waiting
   : b9903ecbcf0b Waiting
   : f473bcb0e44 Waiting
   : d8b79b64a9d5 Waiting
   : 5c36aa47b3f5 Waiting
   : c6834909cd19 Waiting
   : 4f4fb700ef54 Waiting
   : beeeb2e72514 Waiting
   : b7fd27be5b20 Waiting
   : 1475f50d3e62 Waiting
   : 5ac3fe5e5e7e Waiting
   : e2a530cb2e77 Waiting
   : ed50d9a76bab Waiting
   : 08dc3bd6914b Waiting
   : 63eaf96fd2ee Waiting
   db [#####] Pulling
   : 20e4dcae4c69 Extracting [=====] 10.49MB/50.5MB
   ✓ 1c56c3d4ce74 Download complete
   ✓ e9f03a1c24ce Download complete
   ✓ 68c3898c2015 Download complete
   ✓ 6b95a940e7b6 Download complete
   ✓ 90986bb8de6e Download complete
   ✓ ae71319cb779 Download complete
   ✓ ffc89e9dfd88 Download complete
   ✓ 43d05e938198 Download complete
   : 064b2d298fba Waiting
   ✓ df9a4d85569b Download complete
   nginx [#####] 71.06MB / 72.17MB Pulling
   : c29f5b76f736 Extracting [=====] 19.17MB/28.21MB
   ✓ e19db8451adb Download complete
   ✓ 24ff42a0d907 Download complete
   ✓ c558df217949 Download complete
   ✓ 976e8fb25dd Download complete
   ✓ 6c78b0b1a32 Download complete
   ✓ 84cade77a831 Download complete
```

Рисунок 7 – запуск контейнеров

Далее были выявлены ошибки (с помощью команд **docker logs KONTAINER_NAME**) в подключении, контейнер db ругался на отсутствие пароля root, а контейнер wordpress не видел базу данных.

Исправление ошибок:

1. Сброс пароля для root в контейнере db , контейнер был запущен отдельно в безопасном режиме для установки нового пароля root (рисунок 8)

Источник: <https://qna.habr.com/q/1142580>

```
docker run --name dump_mysql \
-v "/.mysql:/var/lib/mysql" \ # Отправляем локальную папку с настройками в
контейнер
-e MYSQL_ALLOW_EMPTY_PASSWORD=yes \ # Отключаем пароль
-p 3306:3306 \ # Настраиваем порт
-d mysql:5.7 --skip-grant-tables # Пропускаем основные таблицы
```

```

dba@dba-vm:~/wordpress$ docker run --rm -d --name mysql-recovery -e MYSQL_ALLOW_EMPTY_PASSWORD=yes -v ./data/mysql:/var/lib/mysql mysql:5.7 --skip-grant-tables
6126127422d7c82cfea6f75fccc18a34e1e307d8639672320d70d9d593f00f74
dba@dba-vm:~/wordpress$ docker exec -it mysql-recovery mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> UPDATE user SET authentication_string=PASSWORD('12') WHERE user='root';
Query OK, 1 row affected, 1 warning (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 1

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> EXIT;
Bye

```

Рисунок 8 – установка нового пароля

2. Добавление пользователя mgpu_ico_etl_15

С помощью команды **exec -it wordpress-db-1 mysql -u root -p** работа выполнялась внутри контейнера db с правами root

С помощью **Select host, user from mysql.user** были найдены пользователи и их хосты (рисунок 9)

```

dba@dba-vm:~/wordpress$ docker exec -it wordpress-db-1 mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT Host, User FROM mysql.user;
+-----+-----+
| Host      | User          |
+-----+-----+
| localhost | mysql.session |
| localhost | mysql.sys     |
| localhost | root          |
+-----+-----+

```

Рисунок 9 – пользователи в бд

Далее был добавлен пользователь с паролем и любым подключением хостов (рисунок 10-11)

```

CREATE USER 'mgpu_ico_etl_15'@ '%' IDENTIFIED BY 'K8DQAKTc';
GRANT ALL PRIVILEGES ON mgpu_ico_etl_15.* TO 'mgpu_ico_etl_15'@ '%';
FLUSH PRIVILEGES;

```



```
mysql> CREATE USER 'mgpu_ico_etl_15'@'%' IDENTIFIED BY 'K8DQAKTc';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON mgpu_ico_etl_15.* TO 'mgpu_ico_etl_15'@'%;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
```

Рисунок 10 – добавление нового пользователя

```
dba@dba-vm:~/wordpress$ docker exec -it wordpress-db-1 mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT Host, User FROM mysql.user;
+-----+-----+
| Host      | User                |
+-----+-----+
| %         | mgpu_ico_etl_15    |
| localhost | mysql.session      |
| localhost | mysql.sys          |
| localhost | root                |
+-----+-----+
4 rows in set (0.00 sec)
```

Рисунок 11 – проверка добавления

3. Создание базы данных для wordpress (рисунок 12-13)

```

dba@dba-vm:~/wordpress$ docker exec -it wordpress-db-1 mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE mgpu_ico_etl_15;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON mgpu_ico_etl_15.* TO 'mgpu_ico_etl_15'@'%';
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> EXIT;
Bye

```

Рисунок 12 – создание бд для wordpress

```

dba@dba-vm:~/wordpress$ docker exec -it wordpress-db-1 mysql -u mgpu_ico_etl_15 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database                |
+-----+
| information_schema       |
| mgpu_ico_etl_15         |
+-----+
2 rows in set (0.00 sec)

mysql> exit
Bye

```

Рисунок 13 – проверка созданной бд

После исправления ошибок и перезапуска контейнеров получилось открыть wordpress на localhost (рисунок 14-15)

```
dba@dba-vm:~/wordpress$ docker compose restart
WARN[0000] /home/dba/wordpress/docker-compose.yml: the at
[+] Restarting 3/3
✓ Container wordpress-db-1           Started
✓ Container wordpress-wordpress-1    Started
✓ Container wordpress-nginx-1        Started
dba@dba-vm:~/wordpress$
```

Рисунок 14 – перезапуск контейнеров

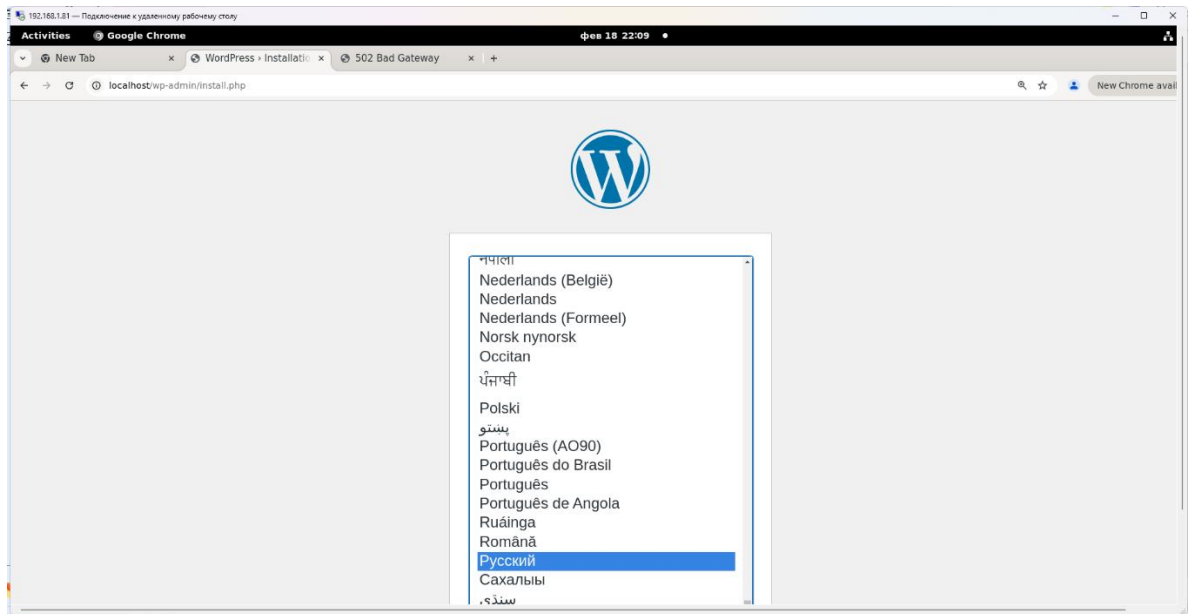


Рисунок 15 – успешное подключение к wordpress

Создан аккаунт Vanyarina Julia на wordpress, изучен функционал CMS (рисунок 16-17)

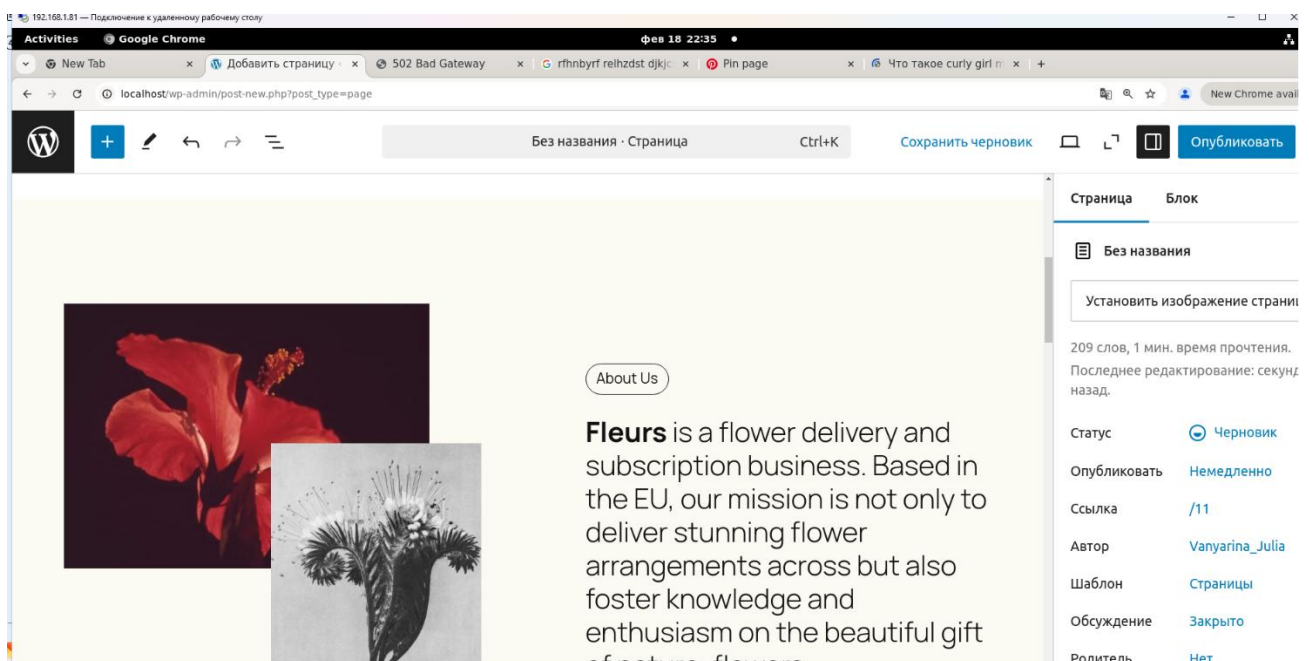


Рисунок 16 – возможность размещения тестовой страницы

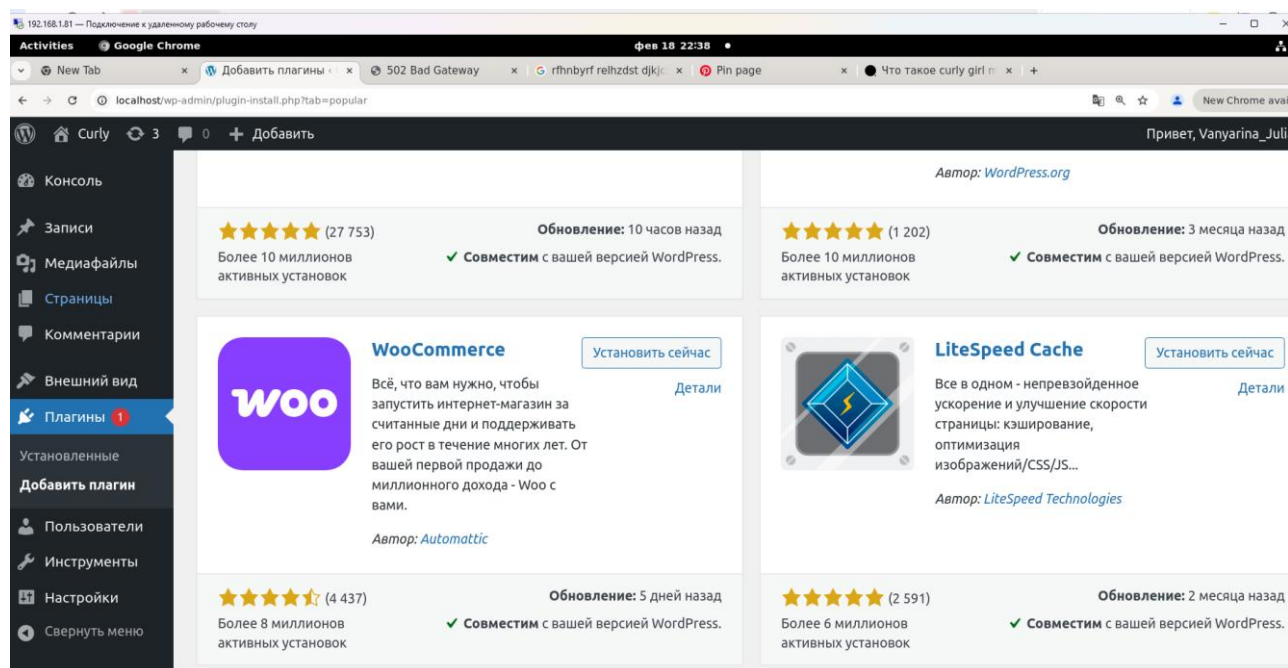


Рисунок 17 – возможность установки плагинов

Вывод: Docker является универсальным и удобным инструментом для быстрой и безопасной настройки сервисов и развертывания приложений. В процессе работы был установлен docker, изучены базовые операции. Решены проблемы с паролем пользователя, а также углубленно изучена работа внутри конкретных контейнеров. Итог работы: успешная загрузка wordpress на локальном хосте.