

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И.
ВЕРНАДСКОГО»
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра компьютерной инженерии и моделирования

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4
«Установка и настройка систем управления базами данных»
по дисциплине «Системное администрирование»
студента 3 курса группы ИВТ-б-о-222(1)
Гоголев Виктора Григорьевича

09.03.01 «Информатика и вычислительная техника»

Симферополь, 2025

1. Установите из пакетов вашего дистрибутива сервер и клиент СУБД Mysql/MariaDB
2. Настройте Mysql/MariaDB в соответствии со следующими требованиями:
 - (a) Принимать соединения только на локальный сокет-файл и на локальный сетевой интерфейс (127.0.0.1).
 - (b) Хранить файлы баз данных в директории /home/db/mysql
 - (c) Кодировка по умолчанию UTF-8
 - (d) Максимальное количество соединений: 20
 - (e) Таймаут соединения: 10 сек
3. Проверьте, что СУБД функционирует от имени непривелигированного пользователя, обычно,mysql
4. Создайте группу db
5. Включите пользователя СУБД mysql в данную группу.
6. Установите из пакетов вашего дистрибутива сервер и клиент СУБД PostgreSQL
7. Настройте PostgreSQL в соответствии со следующими требованиями:
 - (a) Принимать соединения только на локальный сокет-файл и на локальный сетевой интерфейс (127.0.0.1).
 - (b) Хранить файлы баз данных в директории /home/db/postgres
 - (c) Кодировка по умолчанию UTF-8
 - (d) Максимальное количество соединений: 20
 - (e) Таймаут соединения: 10 сек
8. Проверьте, что СУБД PostgreSQL функционирует от имени непривелигированного пользователя, обычно, postgres
9. Включите пользователя СУБД postgres в группу db.
10. Для двух установленных СУБД выполните следующие действия:

(a) Данные операции необходимо выполнить с помощью консольных клиентов к

настроенным СУБД (mysql и psql соответственно)

(b) Поключитесь к СУБД с использованием аккаунта администратора БД

(c) Добавьте пользователя testuser (установите ему безопасный пароль)

(d) Данный пользователь должен быть владельцем базы testdb. Данную базу можно создать на этапе создания пользователя

(e) Измените пароль данного пользователя на любой другой

(f) Выполните процедуру сброса пароля администратора к соответствующей СУБД.

11. Выполните установку любой выбранной системы управления контентом(см. ниже) в любую

```
▷Run All Services
services:
  ▷Run Service
  mariadb:
    image: mariadb:latest
    container_name: lab_mariadb
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: testdb
      MYSQL_USER: testuser
      MYSQL_PASSWORD: testpassword
    volumes:
      - ./home/db/mysql:/var/lib/mysql
      - ./config/mysql:/etc/mysql/conf.d
    ports:
      - "127.0.0.1:3306:3306"
    networks:
      - db_network
    command:
      - --character-set-server=utf8mb4
      - --collation-server=utf8mb4_unicode_ci
      - --max_connections=20
      - --wait_timeout=10
      - --bind-address=127.0.0.1
    user: "1000:1000"
  ▷Run Service
```

Рисунок 1 — конфигурация контейнера lab_mariadb(СУБД mairadb)

```
▷ Run Service
postgres:
  image: postgres:latest
  container_name: lab_postgres
  environment:
    POSTGRES_PASSWORD: rootpassword
    POSTGRES_USER: postgres
    POSTGRES_DB: testdb
  volumes:
    - ./home/db/postgres:/var/lib/postgresql/data
    - ./config/postgres:/etc/postgresql
  ports:
    - "127.0.0.1:5432:5432"
  networks:
    - db_network
  command:
    - -c
    - max_connections=20
    - -c
    - idle_in_transaction_session_timeout=10000
    - -c
    - listen_addresses=127.0.0.1
  user: "1000:1000"
```

Рисунок 2 — конфигурация контейнера lab_postgres(СУБД postgres)

```
wordpress:
  image: wordpress:latest
  container_name: lab_wordpress
  depends_on:
    - mariadb
  environment:
    WORDPRESS_DB_HOST: lab_mariadb:3306
    WORDPRESS_DB_USER: testuser
    WORDPRESS_DB_PASSWORD: testpassword
    WORDPRESS_DB_NAME: testdb
  volumes:
    - wordpress_data:/var/www/html
  ports:
    - "8080:80"
  networks:
    - db_network
  restart: unless-stopped

volumes:
  wordpress_data:

networks:
  db_network:
    driver: bridge
```

Рисунок 3 — конфигурация контейнера lab_wordpress(CMS wordpress)

После запуска compose файла был написан скрипт для проверки выполнения ТЗ по конфигурированию обоих СУБД с выводом информации в терминал.

```

~ /study/3_2/SysAdmin/4/4_lab main !8 ?3 ./check.sh
=== MariaDB Проверка ===
Параметр      Значение
max_connections 20
wait_timeout   10
character_set_server utf8mb4
bind_address   127.0.0.1
testuser_exists Да
testdb_exists  Да

=== PostgreSQL Проверка ===
Параметр      | Значение
-----+-----
max_connections      | 20
idle_in_transaction_session_timeout | 10000
server_encoding      | UTF8
listen_addresses     | 127.0.0.1
testdb_exists        | Да
postgres_user_exists | Да
(6 rows)

```

Рисунок 4 — результат выполнения скрипта

```

~ /study/3_2/SysAdmin/4/4_lab main !8 ?3 ss -tulpn | grep 3306
tcp LISTEN 0 4096 127.0.0.1:3306 0.0.0.0:*
~ /study/3_2/SysAdmin/4/4_lab main !8 ?3 ss -tulpn | grep 5432
tcp LISTEN 0 4096 127.0.0.1:5432 0.0.0.0:*
~ /study/3_2/SysAdmin/4/4_lab main !8 ?3

```

Рисунок 5 — проверка что сервисы слушают соединения только на localhost

Скрипт не содержит ничего интересного, а просто два больших запросов к каждой из бд, с обращением либо к переменным окружения(как в случае maria_db), либо к базе pg_settings, для получения значений и сравнения с ТЗ

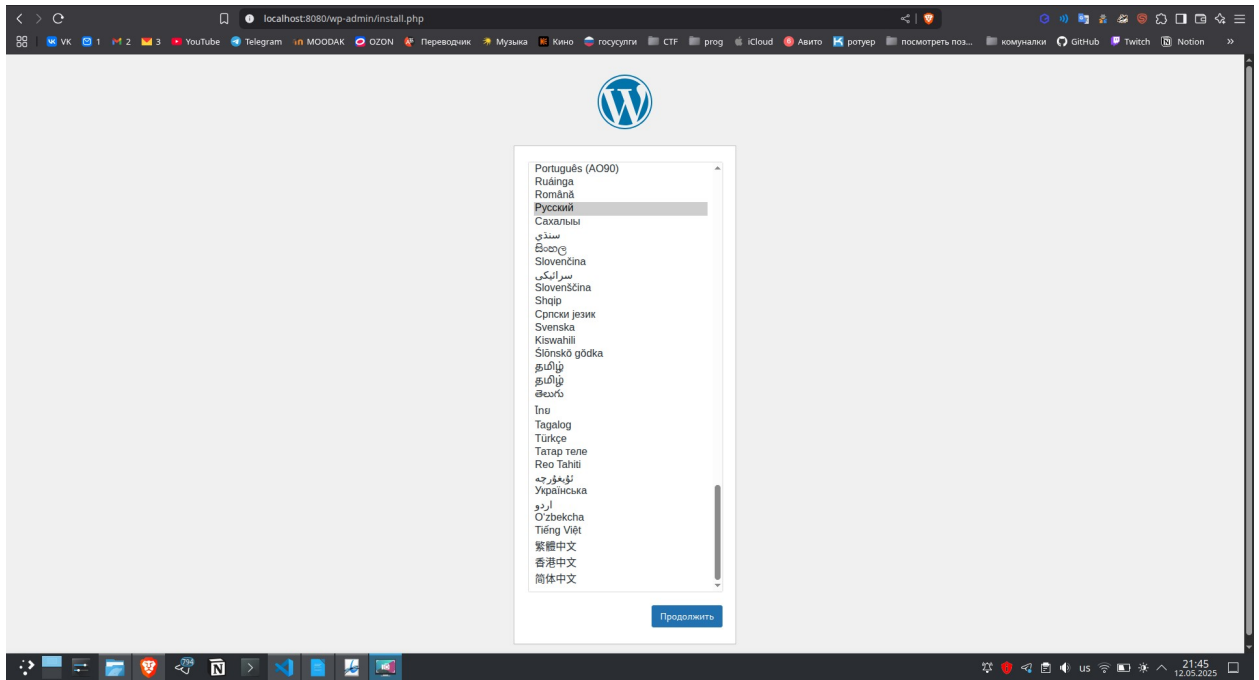


Рисунок 6 — стартовое окно wordpress

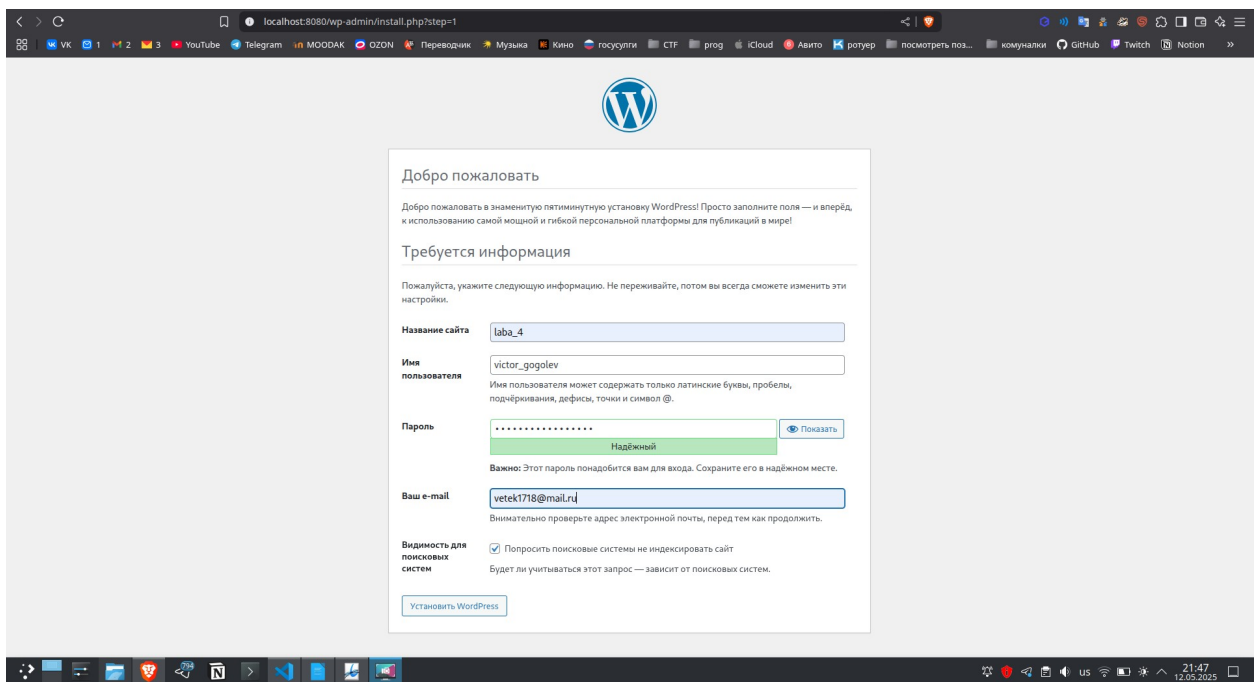


Рисунок 7 — окно регистрации сайта и создания его управляющего

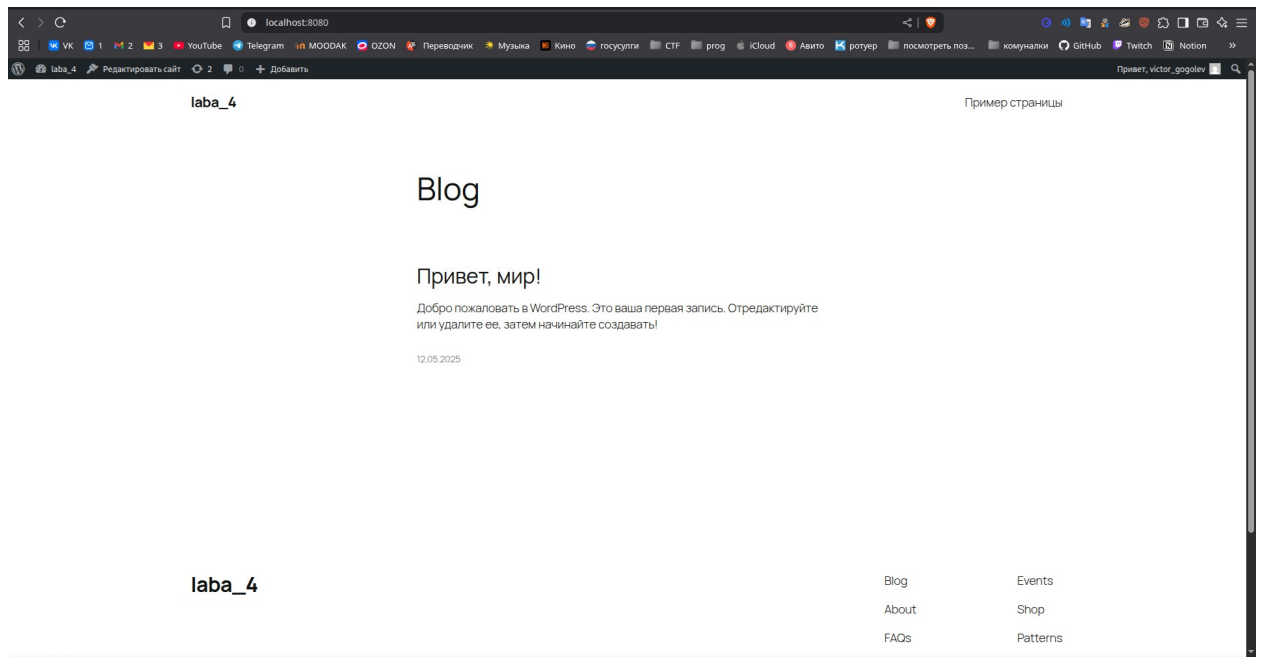


Рисунок 8 — сайт по умолчанию wordpress

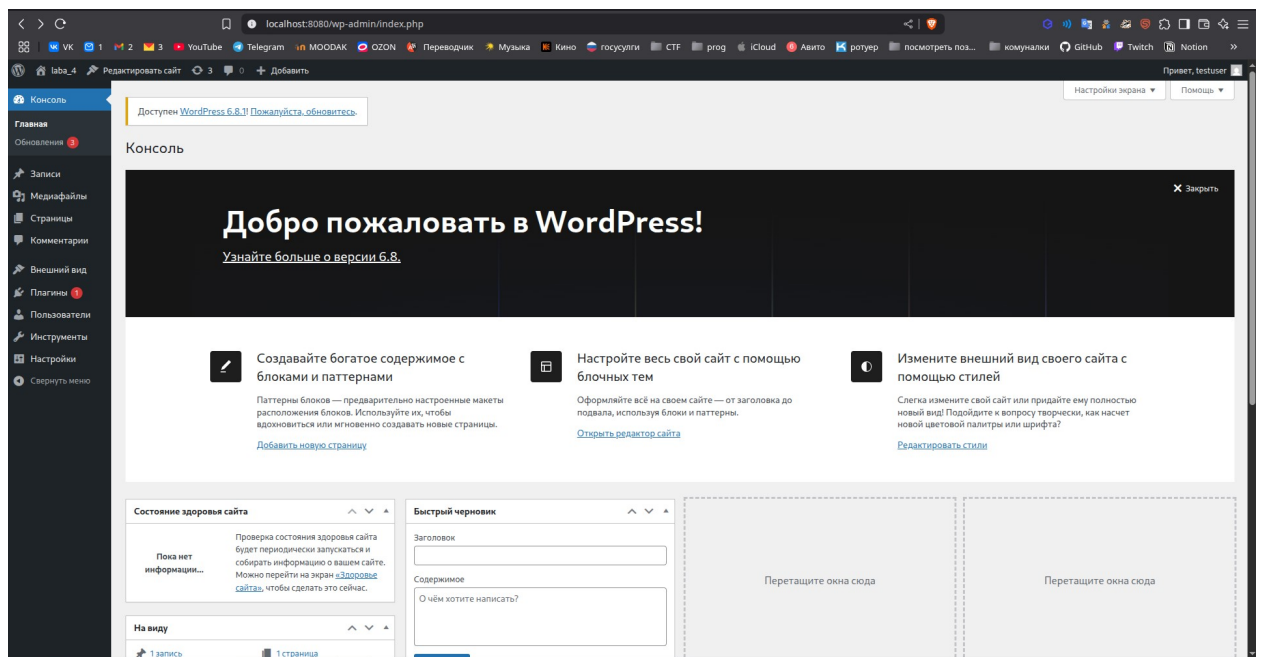


Рисунок 9 — админ панель созданного мной сайта laba_4

Соответственно для нормальной работы wordpress+СУБД(я выбрал mariadb), немного изменил конфиг из исходного ТЗ, так как просили сделать доступ только по 127.0.0.1, а в wordpress находится в другом контейнере(который хоть и находится в одной виртуальной сети),но имеет другой адрес, отличающийся от 127.0.0.1

```

"ConfigFrom": {
  "Network": ""
},
"ConfigOnly": false,
"Containers": {
  "3232f3026c123d3955aff92df319859bd446a7bb8dced4a2c66b780a4e6be0cc": {
    "Name": "lab_postgres",
    "EndpointID": "c40fcfbf7ec93d02f6b5bef7388aa3680a80dfffc71847c3cdb0a6eef9f7c8b70",
    "MacAddress": "5a:07:bd:9e:1c:b2",
    "IPv4Address": "172.18.0.2/16",
    "IPv6Address": ""
  },
  "922e14b5c93d95605d239fb4ef187f4cb32f65673781ecceec51b548c4009ae61": {
    "Name": "lab_mariadb",
    "EndpointID": "1abc817292e3e9cbff837b6edecf2978cdeff2075038831115bb5d72c77b8f29",
    "MacAddress": "22:37:84:b9:cf:46",
    "IPv4Address": "172.18.0.3/16",
    "IPv6Address": ""
  },
  "93763dc6e1d550096e09eeead3d06acce1fe82c64a4520fc238b44276c0ce1": {
    "Name": "lab_wordpress",
    "EndpointID": "7e5196967dce8b9b43dcc36713e8aa5ad6e60dd42fcb1e5d88a24b2d749f9d69",
    "MacAddress": "92:7a:fd:4e:52:87",
    "IPv4Address": "172.18.0.4/16",
    "IPv6Address": ""
  }
}

```

Рисунок 10 — выполнение команды `docker network` позволило увидеть, что хоть 3 процесса и в одной виртуальной подсети, но wordpress имеет адрес 172.18.0.4/16, поэтому для контейнера `mariadb` я поставил прослушивание соединения на всех адресах 0.0.0.0 для успешного подключения wordpress к mariadb