1. Установить в виртуальную машину или VDS Docker, настроить набор контейнеров через docker compose по инструкции по ссылке: https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-docker-compose-ru. Часть с настройкой certbot и HTTPS опустить, если у вас нет настоящего домена и белого IP.
   - sudo apt install docker.io
   - sudo apt install docker-compose
   - sudo apt install yamllint
   - sudo mkdir wordpress
   - cd wordpress/
   - sudo mkdir nginx-conf/
   - cd nginx-conf/
   - sudo nano nginx.conf

```
 GNU nano 6.2                                 nginx.conf
server {
        listen 80;
        listen [::]:80;

        server_name _;

        index index.php index.html index.htm;

        root /var/www/html;

        location ~ /.well-known/acme-challenge {
                allow all;
                root /var/www/html;
        }

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

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

        location ~ /\.ht {
                deny all;
        }

        location = /favicon.ico {
                log_not_found off; access_log off;
        }
        location = /robots.txt {
                log_not_found off; access_log off; allow all;
        }
        location ~* \.(css|gif|ico|jpeg|jpg|js|png)$ {
                expires max;
                log_not_found off;
        }
}
```

   - sudo nano .env

```
 GNU nano 6.2                                  .env
MYSQL_ROOT_PASSWORD=12345678
MYSQL_USER=test
MYSQL_PASSWORD=12345678
```

   - sudo git init
   - sudo nano .gitignore

```
 GNU nano 6.2
.env
```

   - sudo nano .dockerignore

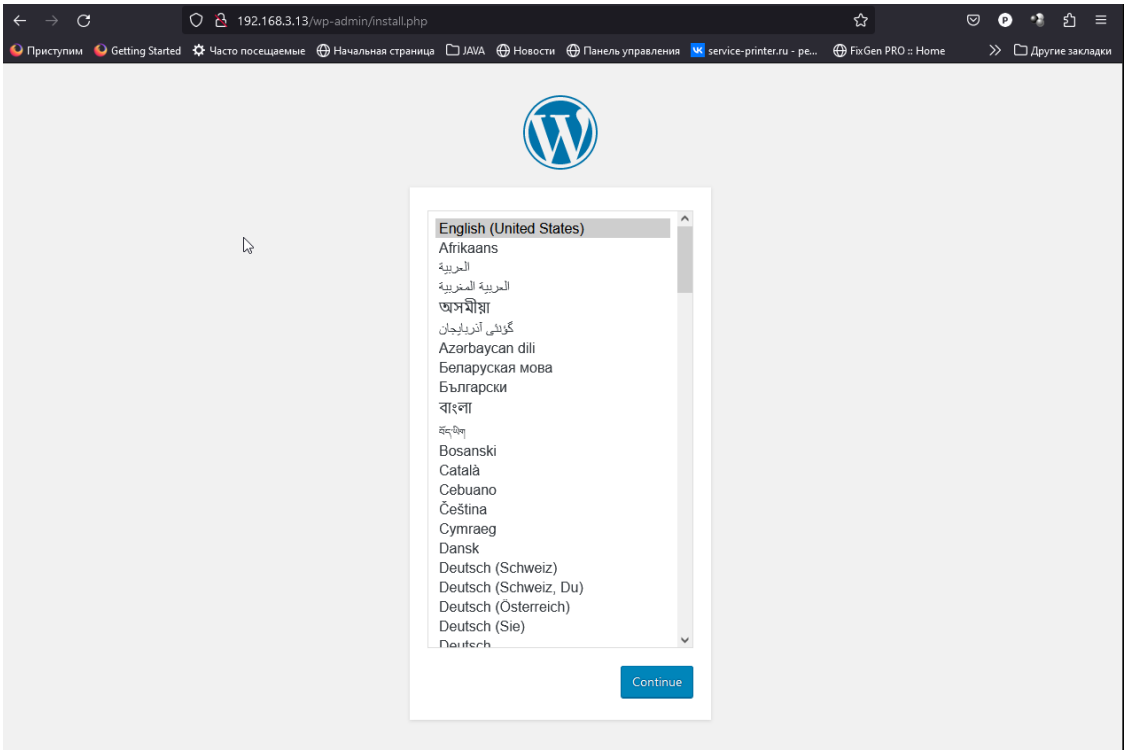
```
 GNU nano 6.2
.env
.git
docker-compose.yml
.dockerignore
```

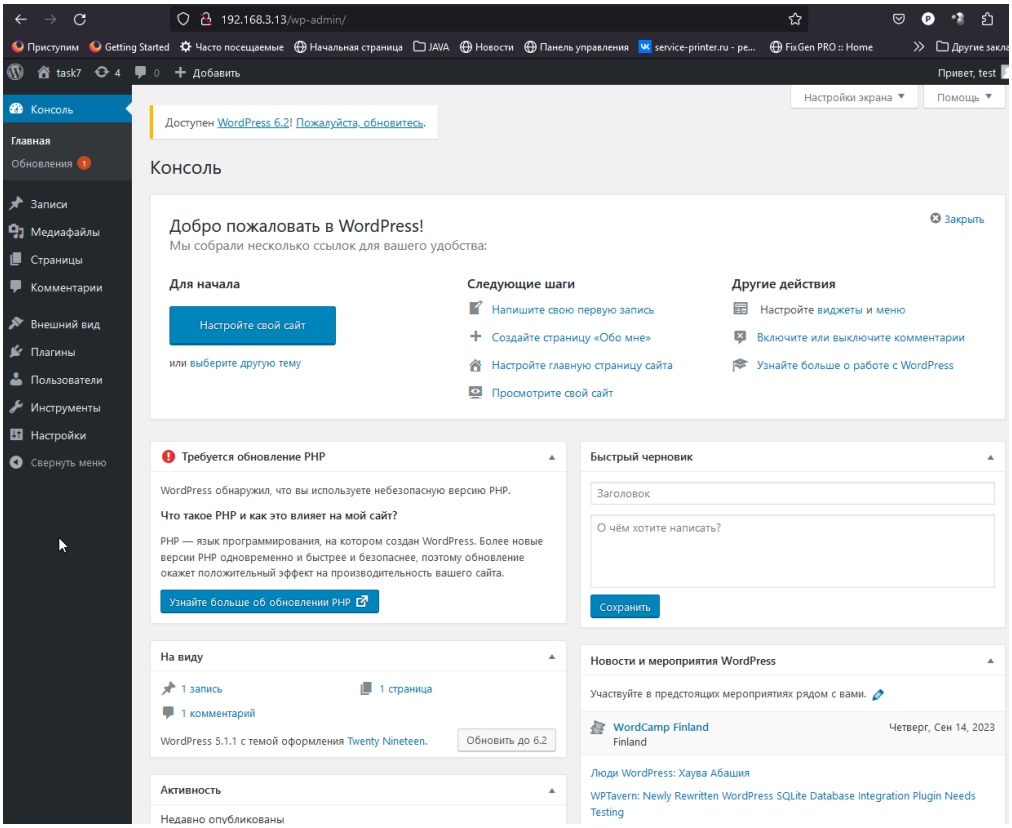
   - sudo nano docker-compose.yml

```
  GNU nano 6.2                                docker-compose.yml
version: '3'

services:
  db:
    image: mysql:8.0
    container_name: db
    restart: unless-stopped
    env_file: .env
    environment:
      - MYSQL_DATABASE=wordpress
    volumes:
      - dbdata:/var/lib/mysql
    command: '--default-authentication-plugin=mysql_native_password'
    networks:
      - app-network

  wordpress:
    depends_on:
      - db
    image: wordpress:5.1.1-fpm-alpine
    container_name: wordpress
    restart: unless-stopped
    env_file: .env
    environment:
      - WORDPRESS_DB_HOST=db:3306
      - WORDPRESS_DB_USER=$MYSQL_USER
      - WORDPRESS_DB_PASSWORD=$MYSQL_PASSWORD
      - WORDPRESS_DB_NAME=wordpress
    volumes:
      - wordpress:/var/www/html
    networks:
      - app-network

  webserver:
    depends_on:
      - wordpress
    image: nginx:1.15.12-alpine
    container_name: webserver
    restart: unless-stopped
    ports:
      - "80:80"
    volumes:
      - wordpress:/var/www/html
      - ./nginx-conf:/etc/nginx/conf.d
    networks:
      - app-network

volumes:
  wordpress:
  dbdata:

networks:
  app-network:
    driver: bridge
```

- sudo docker-compose up -d
- sudo docker-compose ps

```
sl80707@gb:~/wordpress$ sudo docker-compose ps
  Name                 Command               State                Ports
-------------------------------------------------------------------------------------
db          docker-entrypoint.sh --def ...   Up      3306/tcp, 33060/tcp
webserver   nginx -g daemon off;             Up      0.0.0.0:80->80/tcp,:::80->80/tcp
wordpress   docker-entrypoint.sh php-fpm     Up      9000/tcp
sl80707@gb:~/wordpress$
```





2. Запустить два контейнера, связанные одной сетью (используя документацию). Первый контейнер БД (например, образ mariadb:10.8), второй контейнер — phpmyadmin. Получить доступ к БД в первом контейнере через второй контейнер (веб-интерфейс phpmyadmin).
   - sudo su
   - mkdir phpmyadmin
   - cd phpmyadmin
   - mkdir nginx-conf/
   - cd nginx-conf/
   - nano nginx.conf

```
GNU nano 6.2                                    nginx.conf
server {
        root /code;
        index index.php;

        server_name _;
    location / {
        index index.php;
    }
        location ~ \.php$ {
        try_files $uri /index.php =404;
        fastcgi_pass php:9000;
        fastcgi_index index.php;
        fastcgi_buffers 16 16k;
        fastcgi_buffer_size 32k;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

- cd ..
- nano .env

```
GNU nano 6.2
MYSQL_ROOT_PASSWORD=root
PMA_HOST=db
PMA_USER=root
PMA_PASSWORD=root
```

- git init
- nano .gitignore

```
GNU nano 6.2
.env
```

- nano .dockerignore

```
GNU nano 6.2
.env
.git
docker-compose.yml
.dockerignore
```

- nano docker-compose.yml

```
GNU nano 6.2                                    docker-compose.yaml
version: "3"

services:
    nginx:
        depends_on:
            - phpmyadmin
        restart: unless-stopped
        image: nginx
        ports:
            - "8080:80"
        volumes:
            - ./code:/code
            - ./nginx-conf/nginx.conf:/etc/nginx/conf.d/default.conf
        networks:
            - app-network
    php:
        restart: unless-stopped
        image: php:fpm
        volumes:
            - ./code:/code
        networks:
            - app-network
    db:
        image: mariadb:10.8
        env_file: .env
        restart: unless-stopped
        volumes:
            - ./mariadb:/var/lib/mysql
        ports:
            - "3306:3306"
        networks:
            - app-network
    phpmyadmin:
        depends_on:
            - db
        image: phpmyadmin/phpmyadmin
        env_file: .env
        ports:
            - "8081:80"
        restart: unless-stopped
        networks:
            - app-network

networks:
  app-network:
    driver: bridge
```

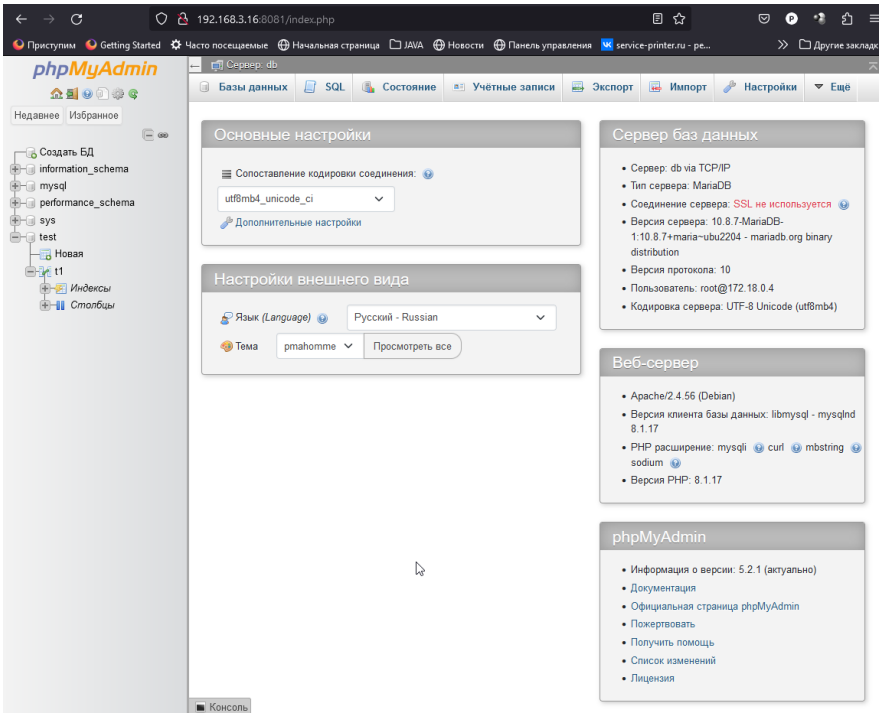
- mkdir code
- cd code
- nano index.php

```
  GNU nano 6
<?php
phpinfo();
```

- docker-compose up -d
- docker-compose ps

```
root@gb:/home/s180707/phpmyadmin/code# docker-compose ps
         Name                       Command               State                  Ports
-------------------------------------------------------------------------------------------------------
phpmyadmin_db_1           docker-entrypoint.sh mariadbd    Up      0.0.0.0:3306->3306/tcp,:::3306->3306/tcp
phpmyadmin_nginx_1        /docker-entrypoint.sh ngin ...   Up      0.0.0.0:8080->80/tcp,:::8080->80/tcp
phpmyadmin_php_1          docker-php-entrypoint php-fpm    Up      9000/tcp
phpmyadmin_phpmyadmin_1   /docker-entrypoint.sh apac ...   Up      0.0.0.0:8081->80/tcp,:::8081->80/tcp
```

- http://192.168.3.16:8081/



- http://192.168.3.16:8080/