

# Настройка WordPress и MySQL

Для настройки сайта и БД будем использовать Compose-файл, в котором опишем конфигурацию:

```
version: '3.3'
services:
  wordpress:
    image: wordpress:latest
    restart: always
    links:
      - db:mysql
    ports:
      - "80:80"
    working_dir: /var/www/html
    volumes:
      - "/opt/wp-content:/var/www/html/wp-content"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
  db:
    image: mysql:5.7
    restart: always
    volumes:
      - "/opt/mysql:/var/lib/mysql"
    environment:
      MYSQL_ROOT_PASSWORD: secret
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
```

Создадим каталог **wordpress** и разместим в нем **docker-compose.yml** с вышеприведенным содержимым.

Создадим каталоги для локального хранения файлов плагинов, контента и базы данных MySQL:

```
sudo mkdir /opt/mysql
sudo mkdir /opt/wp-content
sudo chmod 777 /opt/wp-content # права нужно изменить
                                # после завершения установки на
                                # 755 (sudo chmod 755 wp-content)
```

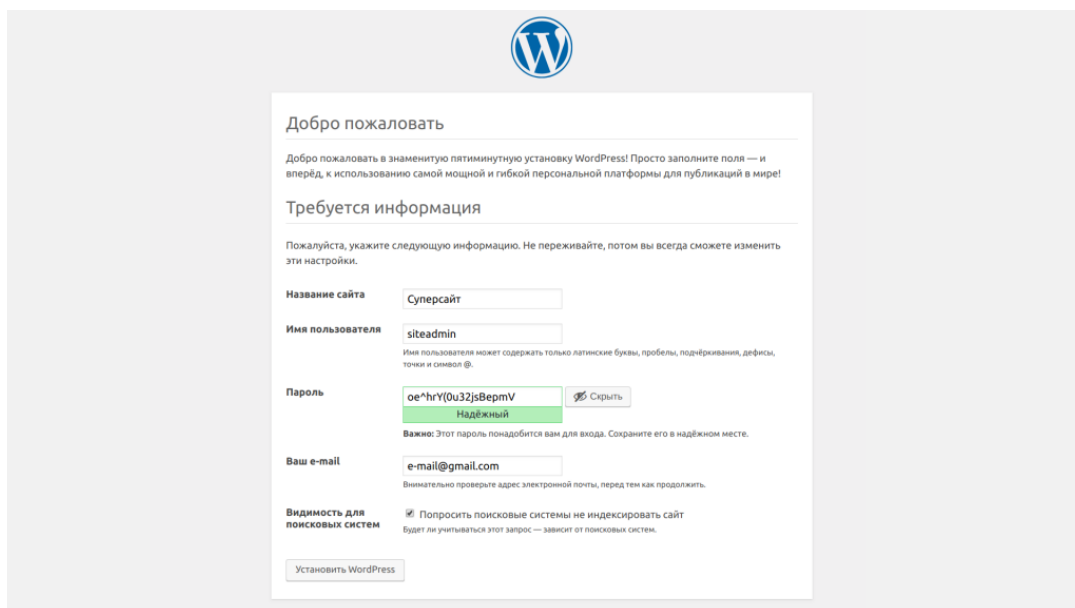
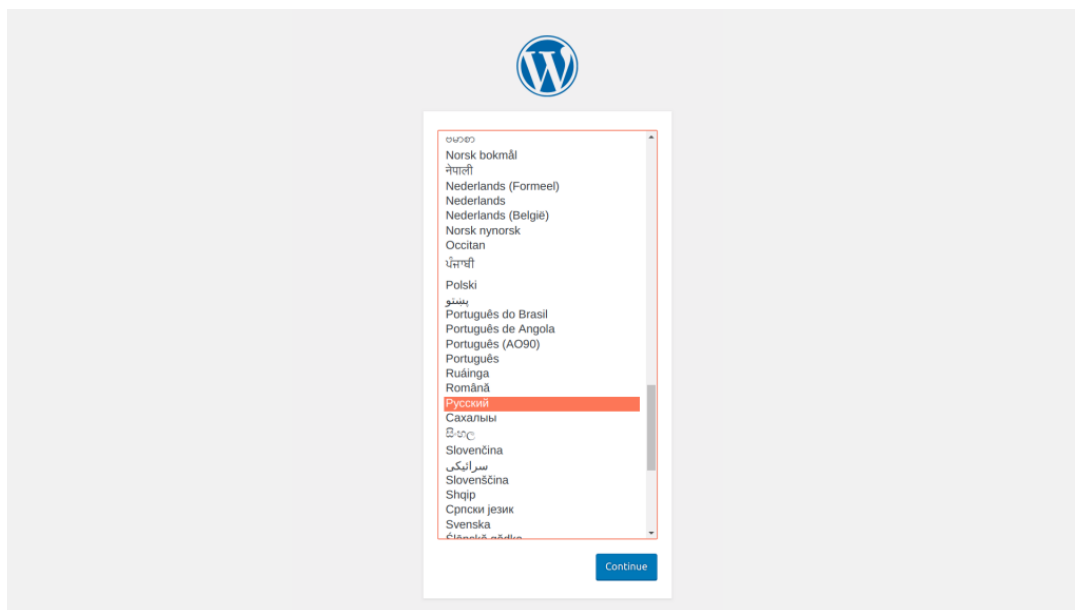
Запускаем Compose:

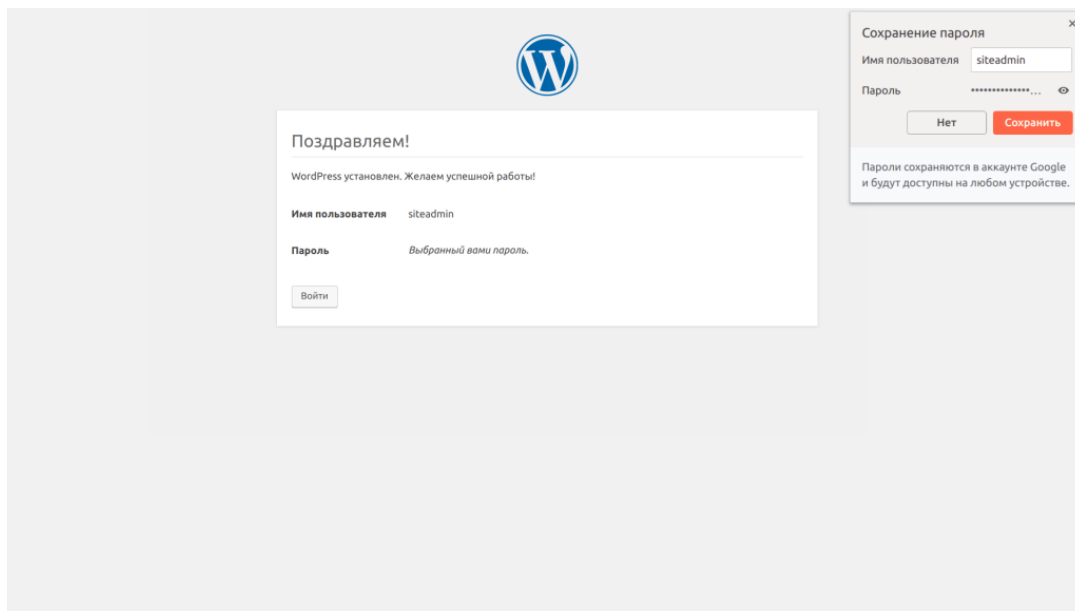
```
cd wordpress && sudo docker-compose up -d
```

После старта в выводе **sudo docker ps** должны присутствовать два сервиса:

```
sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS            PORTS              NAMES
79bcdeb504c6       wordpress:latest   "docker-entrypoint.s..." 6
minutes ago       Up 6 minutes       0.0.0.0:80->80/tcp        wp_wordpress_1
3d4a9d2d5b17       mysql:5.7          "docker-entrypoint.s..." 6
minutes ago       Up 6 minutes       3306/tcp, 33060/tcp      wp_db_1
```

Теперь вы можете открыть в браузере сайт и выполнить настройку WordPress.





## Добавление Nginx с сертификатом Let's Encrypt

Если вы хотите использовать внешний Nginx для управления трафиком к контейнеру, то замените: «80:80» на «127.0.0.1:8080:80» и выполните остановку контейнеров и измените Compose-файл:

```
sudo docker-compose down
sudo nano docker-compose.yml
sudo docker-compose up -d
```

Теперь доступ к вашему контейнеру извне будет закрыт.

Перейдите к нашему руководству по настройке проксирования с помощью Nginx с поддержкой сертификата Let's Encrypt: [Ubuntu 16.04 и 18.04](#), [Debian 9 и 10](#), [CentOS 7](#). для завершения настройки вашего блога измените в Nginx секции «**upstream**» и «**location /**», чтобы они приобрели следующий вид:

```
upstream wordpress {
    server 127.0.0.1:8080;
}
...
server {
    ....
    location / {
        proxy_pass http://wordpress/;
    }
    ...
}
```

Проверьте корректность настроек и перезапустите nginx: