

Задание:

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
```

```
mkdir wordpress && cd wordpress
```

```
mkdir nginx-conf
```

```
nano nginx-conf/nginx.conf
```

```
server {
    listen 80;
    listen [::]:80;

    server_name example1.com www.example1.com;

    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;
    }
}

```

nano .env

```

MYSQL_ROOT_PASSWORD=1
MYSQL_USER=1
MYSQL_PASSWORD=1

```

nano 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
      - certbot-etc:/etc/letsencrypt
    networks:
      - app-network

certbot:
  depends_on:
    - webserver
  image: certbot/certbot
  container_name: certbot
  volumes:
    - certbot-etc:/etc/letsencrypt
    - wordpress:/var/www/html
  command: certonly --webroot --webroot-path=/var/www/html --email
sammy@example1.com --agree-tos --no-eff-email --staging -d example1.com -
d www.example1.com

volumes:
  certbot-etc:
  wordpress:
  dbdata:

networks:
  app-network:
    driver: bridge

docker-compose up

apt install docker-compose

docker-compose up -d

docker-compose ps

docker-compose exec webserver ls -la /etc/letsencrypt

nano docker-compose.yml

Нали раздел файла с определением службы certbot и замените флаг --staging
в параметрах command на флаг --force-renewal

docker-compose up --force-recreate --no-deps certbot

docker-compose stop webserver

curl -sLo nginx-conf/options-ssl-nginx.conf
https://raw.githubusercontent.com/certbot/certbot/master/certbot-
nginx/certbot_nginx/_internal/tls_configs/options-ssl-nginx.conf

rm nginx-conf/nginx.conf

```

```

server {
    listen 80;
    listen [::]:80;

    server_name example1.com www.example1.com;

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

    location / {
        rewrite ^ https://$host$request_uri? permanent;
    }
}

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name example1.com www.example1.com;

    index index.php index.html index.htm;

    root /var/www/html;

    server_tokens off;

    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key
/etc/letsencrypt/live/example.com/privkey.pem;

    include /etc/nginx/conf.d/options-ssl-nginx.conf;

    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-XSS-Protection "1; mode=block" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header Referrer-Policy "no-referrer-when-downgrade" always;
    add_header Content-Security-Policy "default-src * data: 'unsafe-
eval' 'unsafe-inline'" always;
    # add_header Strict-Transport-Security "max-age=31536000;
includeSubDomains; preload" always;
    # enable strict transport security only if you understand the
implications

    location / {
        try_files $uri $uri/ /index.php$sis_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;
    }
}

```

nano docker-compose.yml

распределение порта 443

docker-compose up -d --force-recreate --no-deps webserver

docker-compose ps

\* 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).

Результат:

Текст команд, которые применялись при выполнении задания. При наличии: часть конфигурационных файлов, которые решают задачу. Скриншоты результата запуска контейнеров (веб-интерфейс). Присылаем в формате текстового документа: задание и команды для решения (без вывода). Формат — PDF (один файл на все задания).