

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 docker-compose – установка

systemctl status docker – проверка работоспособности

sudo su – переход в root

mkdir wordpress – создаем папку

ll – смотрим файлы

cd wordpress – переход в папку

cat > .env – создание файла

MYSQL_ROOT_PASSWORD=dfkljs_d324fD_klj

MYSQL_USER=wp

MYSQL_PASSWORD=KMJ23f9sad_80ds

cat > .dockerignore – создание файла

.env

cat > docker-compose.yml – создание файла

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:6.0.1-php8.0-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: 1.22.0-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@example.com --agree-tos --no-eff-email --staging -d example.com -d www.example.com

volumes:

certbot-etc:

wordpress:

dbdata:

networks:

app-network:

driver: bridge

nano docker-compose.yml – коррекция файла

mkdir nginx-conf – создание папки

cd nginx-conf/ - переход в новую папку

cat > nginx.conf – создание файла

server {

listen 80;

listen [::]:80;

```
server_name example.com www.example.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;
}
}

```

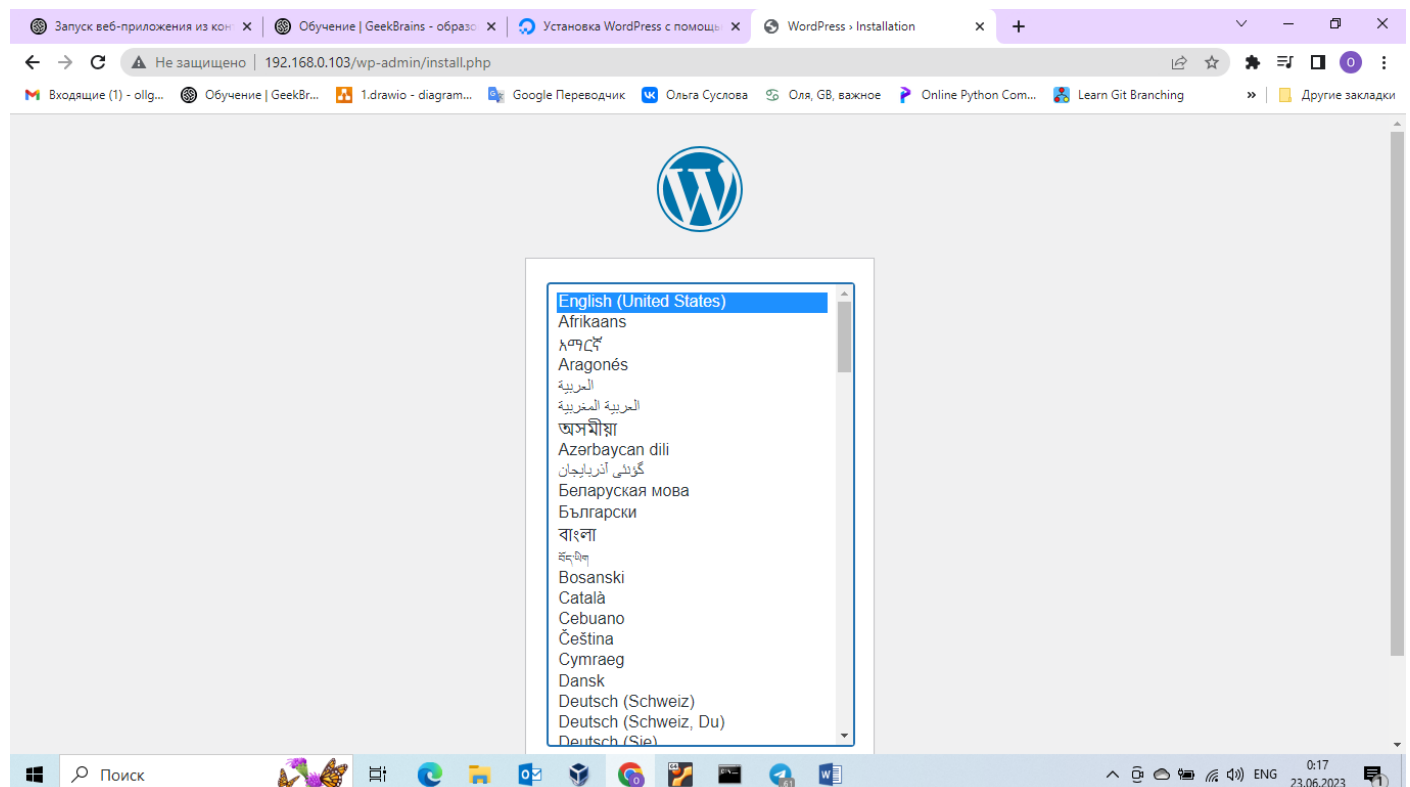
cd .. – переход в папку wordpress

docker-compose up -d – создание контейнера

root@olgas-VirtualBox:~/wordpress# docker-compose ps - список

Name	Command	State	Ports

db	docker-entrypoint.sh --def ...	Up	3306/tcp, 33060/tcp
webserver	/docker-entrypoint.sh nginx ...	Up	0.0.0.0:80->80/tcp, :::80->80/tcp
wordpress	docker-entrypoint.sh php-fpm	Up	9000/tcp



* 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).

```
~/wordpress# docker network create newnetwork
```

```
fdc18bdcd0ce01ab938ef8fb1e94f6d2e4ac99e74120d5336bd33ee667312e02
```

```
docker network ls
```

NETWORK ID	NAME	DRIVER	SCOPE
5a1a370f78ff	bridge	bridge	local
e6f79d66014f	docker_default	bridge	local
cef48855c419	host	host	local
fdc18bdcd0ce	newnetwork	bridge	local
0b91047d49db	none	null	local
ecdf3f625c89	wordpress_app-network	bridge	local

```
docker run --name mariadb -e MYSQL_ROOT_PASSWORD=12345 --network newnetwork -d mariadb:10.8
```

```
sudo docker run --name phpmyadmin -d --network newnetwork -e PMA_HOST=mariadb -p 8080:80  
phpmyadmin
```

```
root@olgas-VirtualBox:~/wordpress# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e44b23777401	phpmyadmin	"/docker-entrypoint...."	13 minutes ago	Created		phpmyadmin
c0c4418fa522	mariadb:10.8	"docker-entrypoint.s..."	17 minutes ago	Up 17 minutes	3306/tcp	mariadb

```
root@olgas-VirtualBox:~/wordpress# docker network inspect newnetwork
```

```
[  
  {  
    "Name": "newnetwork",  
    "Id": "fdc18bdcd0ce01ab938ef8fb1e94f6d2e4ac99e74120d5336bd33ee667312e02",  
    "Created": "2023-06-23T00:45:35.363325682+03:00",  
    "Scope": "local",  
    "Driver": "bridge",
```

```
"EnableIPv6": false,
"IPAM": {
  "Driver": "default",
  "Options": {},
  "Config": [
    {
      "Subnet": "172.20.0.0/16",
      "Gateway": "172.20.0.1"
    }
  ]
},
"Internal": false,
"Attachable": false,
"Ingress": false,
"ConfigFrom": {
  "Network": ""
},
"ConfigOnly": false,
"Containers": {
  "c0c4418fa5229486cd831818e52608d3d8683897178c2726829157a527db3e26": {
    "Name": "mariadb",
    "EndpointID": "0c788c9fba8d49863131f8e17be17d53fadd2e8298e910e934bb09e55d6672f2",
    "MacAddress": "02:42:ac:14:00:02",
    "IPv4Address": "172.20.0.2/16",
    "IPv6Address": ""
  }
},
"Options": {},
"Labels": {}
}
```

]

