**Установка Docker**

sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"

sudo apt update

apt-cache policy docker-ce

sudo apt install docker-ce

sudo systemctl status docker

sudo usermod -aG docker ${USER}

su - ${USER}

id –nG

sudo usermod -aG docker user1

**Установка Docker Compose**

sudo curl -L https://github.com/docker/compose/releases/download/1.29.2/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

**Установка WordPress**

Установка и настройка

mkdir wordpress && cd wordpress

mkdir nginx-conf

nano nginx-conf/nginx.conf

---

server {

listen 80;

listen [::]:80;

server\_name example.com [www.example.com](http://www.example.com) localhost;

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=your\_root\_password

MYSQL\_USER=your\_wordpress\_database\_user

MYSQL\_PASSWORD=your\_wordpress\_database\_password

---

git init

nano .gitignore

---

.env

---

nano .dockerignore

---

.env

.git

docker-compose.yml

.dockerignore

---

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@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

---

docker-compose up –d

docker-compose ps