

Лабораторная работа №4

Соколов А.Д. Б20-505

Dockerfile

```
Dockerfile
1 FROM ubuntu:latest
2
3 LABEL maintainer="Sokolov Alexandr Dm"
4 LABEL mail="koxed@kox.com"
5 LABEL description="Containered WBDashboard service - platform to view all wireguard configurations and manage them in an easier way"
6 LABEL version="0.0"
7
8 RUN apt update && apt -y install sudo wireguard net-tools python3-pip git
9 WORKDIR /service
10 RUN useradd user01
11 RUN echo "user01 ALL=(ALL) NOPASSWD:ALL" | tee /etc/sudoers.d/user01
12 RUN chown -R user01:user01 /service
13 USER user01
14
15 RUN git clone -b v3.0.6 https://github.com/donaldzou/WGDashboard.git wgdashboard
16 WORKDIR /service/wgdashboard/src
17 RUN sudo chmod u+x wgd.sh
18 RUN sudo ./wgd.sh install
19 RUN sudo chmod -R 755 /etc/wireguard
20 CMD ["sudo", "./wgd.sh", "debug"]
```

ubuntu - так как пакеты свежее, чем в дебиане, хоть и образ чуть чуть тяжелее.

CMD - так как при запуске можно будет выбрать между debug и start

Конфиг вайргарда:

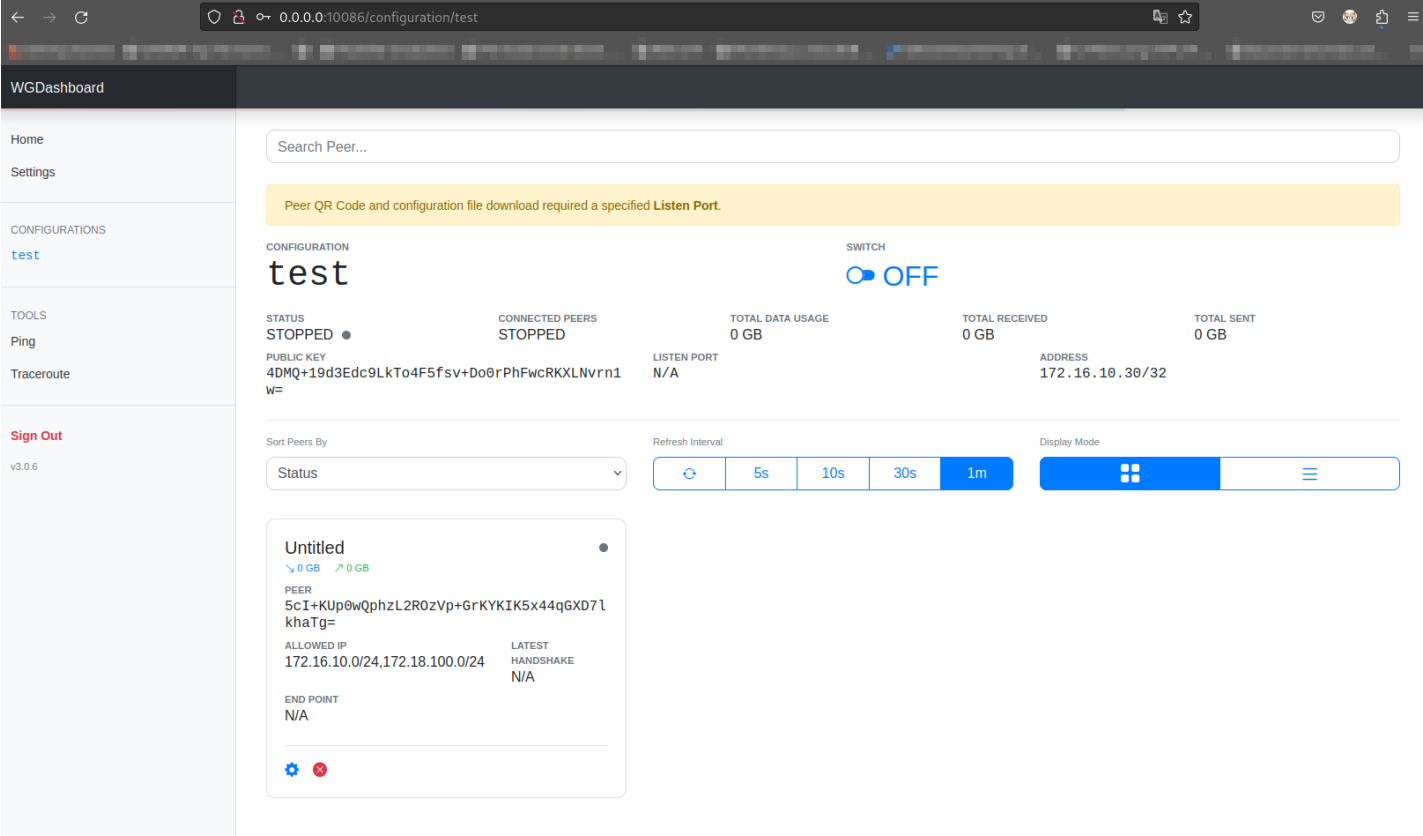
```
[Interface]
PrivateKey = gMQT3OdFbEHufawCRV4fzCL4nOiKgbSG201fkWgLaE8=
Address = 172.16.10.30/32
DNS = 8.8.8.8
MTU = 1400

[Peer]
PublicKey = 5cI+KUp0wQphzL2ROzVp+GrKYKIK5x44qGXD7lkhaTg=
AllowedIPs = 172.16.10.0/24,172.18.100.0/24
Endpoint = 85.143.113.117:6000
PersistentKeepalive = 21
PresharedKey = OilupIiaec+LhcLgc12BiyytBffg87YInuBaEkJyLNQ=%
```

Запуск с использованием докерфайла:

```
vagrant@ubuntu-jammy:~$ sudo docker run -p 10086:10086 -v /home/vagrant/wireguard:/etc/wireguard -it test
-----
| Starting WGDashboard in the foreground. |
| * Serving Flask app 'WGDashboard' |
| * Debug mode: off |
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:10086
* Running on http://172.17.0.2:10086
Press CTRL+C to quit
User not signed in - Attempted access: get_conf
10.0.2.2 -- [21/Dec/2023 08:04:00] "GET /get_config/test?search= HTTP/1.1" 302 -
10.0.2.2 -- [21/Dec/2023 08:04:00] "GET /signin?redirect=0.0.0.0:10086/get_config/test?search= HTTP/1.1" 200 -
User not signed in - Attempted access: configuration
10.0.2.2 -- [21/Dec/2023 08:04:01] "GET /configuration/test HTTP/1.1" 302 -
10.0.2.2 -- [21/Dec/2023 08:04:01] "GET /signin?redirect=0.0.0.0:10086/configuration/test HTTP/1.1" 200 -
10.0.2.2 -- [21/Dec/2023 08:04:01] "GET /static/js/tools.min.js HTTP/1.1" 304 -
10.0.2.2 -- [21/Dec/2023 08:04:01] "GET /static/img/logo.png HTTP/1.1" 304 -
10.0.2.2 -- [21/Dec/2023 08:04:01] code 400, message Bad request version ('v\x0fK\x00'\x13\x01\x13\x03\x13\x02\u0000/\x0f'\u0000\u0000')
10.0.2.2 -- [21/Dec/2023 08:04:01] "v\x16\x03\x01\x02z\x01\x00\x02v\x03\x03f\x94 \x149Y0\x82\u0000h\x15G\x1a\x8aui\x00\x9b0z'\x8bM'E\x09Rn\x99< )%Yn*\x83v0\x99/4Qf-çQK990-40
+_*3\x85v\x0fK\x00'\x13\x01\x13\x03\x13\x02\u0000/\x0f'\u0000\u0000' HTTPStatus.BAD_REQUEST -
10.0.2.2 -- [21/Dec/2023 08:04:01] "GET /static/img/192x192ios.png HTTP/1.1" 304 -
|
```

Проверка, что работает:



docker-compose.yml

```
services:
  wg-dashboard:
    build:
      dockerfile: ./Dockerfile
    ports:
      - 10086:10086
      - 51820:51820
    cap_add:
      - NET_ADMIN
    networks:
      - wg-networks
    volumes:
      - ./wireguard:/etc/wireguard
    restart: unless-stopped

    deploy:
      resources:
        limits:
          cpus: "1"
          memory: 512M

networks:
  wg-networks:
```

unless-stopped - так как другие опции(always, no, on-failure) не подходят с идеологической точки зрения.
deploy - сервис очень легкий, поэтому ему не нужно много вычислительных мощностей.

3anyck c docker compose

```
vagrant@ubuntu-jammy:~$ sudo docker compose up --build
[+] Building 1.2s (16/16) FINISHED                                docker:default
=> [wg-dashboard internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 871B                                                            0.0s
=> [wg-dashboard internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                                  0.0s
=> [wg-dashboard internal] load metadata for docker.io/library/ubuntu:latest                  1.1s
=> [wg-dashboard 1/12] FROM docker.io/library/ubuntu:latest@sha256:6042500cf4b44023ea1894effe7890666b0c5c7871ed83a97c36c76ae560bb9b  0.0s
=> => resolve docker.io/library/ubuntu:latest@sha256:6042500cf4b44023ea1894effe7890666b0c5c7871ed83a97c36c76ae560bb9b  0.0s
=> CACHED [wg-dashboard 2/12] RUN apt update && apt -y install sudo wireguard net-tools python3-pip git  0.0s
=> CACHED [wg-dashboard 3/12] WORKDIR /service                                                0.0s
=> CACHED [wg-dashboard 4/12] RUN useradd user01                                              0.0s
=> CACHED [wg-dashboard 5/12] RUN echo "user01 ALL=(ALL) NOPASSWD:ALL" | tee /etc/sudoers.d/user01  0.0s
=> CACHED [wg-dashboard 6/12] RUN chown -R user01:user01 /service                            0.0s
=> CACHED [wg-dashboard 7/12] RUN git clone -b v3.0.6 https://github.com/donaldzou/WGDashboard.git wgdashboard  0.0s
=> CACHED [wg-dashboard 8/12] RUN sudo chown user01:user01 /service/wgdashboard                0.0s
=> CACHED [wg-dashboard 9/12] WORKDIR /service/wgdashboard/src                               0.0s
=> CACHED [wg-dashboard 10/12] RUN sudo chmod u+x wg.sh                                       0.0s
=> CACHED [wg-dashboard 11/12] RUN sudo ./wg.sh install                                      0.0s
=> CACHED [wg-dashboard 12/12] RUN sudo chmod -R 755 /etc/wireguard                          0.0s
=> [wg-dashboard] exporting to image                                                         0.0s
=> => exporting layers                                                                          0.0s
=> => writing image sha256:0e24efbe89fele75e0127df661291df5c5e8f4317d1420e9975c40ad6b54f8e2  0.0s
=> => naming to docker.io/library/vagrant-wg-dashboard                                        0.0s
[+] Running 1/1
✔ Container vagrant-wg-dashboard-1 Recreated                                                0.2s
Attaching to vagrant-wg-dashboard-1
vagrant-wg-dashboard-1 | -----
vagrant-wg-dashboard-1 | | Starting WGDashboar in the foreground. |
vagrant-wg-dashboard-1 | * Serving Flask app 'WGDashboar' |
vagrant-wg-dashboard-1 | * Debug mode: off |
vagrant-wg-dashboard-1 | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead. |
vagrant-wg-dashboard-1 | * Running on all addresses (0.0.0.0) |
vagrant-wg-dashboard-1 | * Running on http://127.0.0.1:10086 |
vagrant-wg-dashboard-1 | * Running on http://172.20.0.2:10086 |
vagrant-wg-dashboard-1 | Press CTRL+C to quit |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:10:52] "GET /configuration/test HTTP/1.1" 302 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:10:52] "GET /signin?redirect=0.0.0.0:10086/configuration/test HTTP/1.1" 200 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:10:52] "GET /static/css/dashboard.css HTTP/1.1" 304 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:10:52] "GET /static/js/tools.min.js HTTP/1.1" 304 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:10:52] "GET /static/img/logo.png HTTP/1.1" 304 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:10:52] "GET /static/img/192x192ios.png HTTP/1.1" 304 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:10:59] "POST /auth HTTP/1.1" 200 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:10:59] "GET /configuration/test HTTP/1.1" 200 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:11:00] "GET /static/js/tools.min.js HTTP/1.1" 304 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:11:00] "GET /static/js/wireguard.min.js HTTP/1.1" 200 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:11:00] "GET /static/js/configuration.min.js HTTP/1.1" 200 - |
vagrant-wg-dashboard-1 | 10.0.2.2 - - [21/Dec/2023 08:11:00] "GET /static/img/192x192ios.png HTTP/1.1" 200 - |
```

Проверка:

WGDashboard

Home

Settings

CONFIGURATIONS

test

TOOLS

Ping

Traceroute

Sign Out

v3.0.6

0.0.0.0:10086/configuration/test

Peer QR Code and configuration file download required a specified Listen Port.

Peer QR Code and configuration file download required a specified Listen Port.

CONFIGURATION

test

SWITCH OFF

STATUS STOPPED

CONNECTED PEERS STOPPED

TOTAL DATA USAGE 0 GB

TOTAL RECEIVED 0 GB

TOTAL SENT 0 GB

PUBLIC KEY 4DMQ+19d3Edc9LkTo4F5fsv+Do0rPhFcRWKXLNvrn1

W=

LISTEN PORT N/A

ADDRESS 172.16.10.30/32

Sort Peers By Status

Refresh Interval 5s 10s 30s 1m

Display Mode

Untitled

0 GB 0 GB

PEER 5cI+Kup0wQphzL2R0zVp+GrKYKIK5x44q6XD7l

khaTg=

ALLOWED IP 172.16.10.0/24,172.18.100.0/24

LATEST HANDSHAKE N/A

END POINT N/A

Финальный vagrantfile:

```
Vagrantfile
1 Vagrant.configure("2") do |config|
2   config.vbguest.auto_update = false
3   config.vm.box = "ubuntu/jammy64"
4   config.vm.define "lab4"
5
6   config.vm.provision "shell", privileged: true, path: "https://get.docker.com"
7   config.vm.network "forwarded_port", guest: 10086, host: 10086
8
9   config.vm.provision "shell", inline: <<-SHELL
10    cp /vagrant/Dockerfile .
11    cp /vagrant/docker-compose.yml .
12    mkdir wireguard
13    cp /vagrant/wg.conf ./wireguard/wg.conf
14    sudo docker compose up --build -d
15  SHELL
16 end
```