Практическое задание №4.Wireguard

Для выполнения практических занятий необходимо переключиться на проект [GROUP]:[team]-lab:sandbox.

Пользовательская установка wireguard

Для начала надо развернуть новый инстанс (при ограничении ресурсов может потребоваться удалить все предыдущие инстансы) так, как это было сделанно в первой практической работе.

Зайти в режим привилегированного пользователя

```
sudo su
```

установить нужные пакеты:

```
apt update
apt install -y wireguard qrencode
```

Настройка системы

Разрешить перенаправление сетевых пакетов на уровне ядра. Для этого откройте файл /etc/sysctl.conf и добавьте в конец такие строки:

```
vi /etc/sysctl.conf
```

```
net.ipv4.ip_forward = 1
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.proxy_arp = 0
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

Затем необходимо выполнить команду sysctl -p чтобы система перечитала конфигурацию:

```
sysctl -p
```

Генерация ключей сервера

Для сервера надо создать приватный и публичный ключ. Эти ключи, потом надо будет записать в конфигурационный файл сервера и клиента, сами файлы ключей вам не нужны, поэтому можете создавать их где хотите, например, в домашней папке. Так же полученный ключ можно записать в переменную окружения:

```
wg genkey | sudo tee server_private.key | wg pubkey | sudo tee server_public.key
```

Ключи созданы, утилита tee запишет их в файл, а также выведет на экран, что очень удобно для сохранения значения в переменную

Генерация ключей клиента

Аналогичным образом создаём ключи для клиента. Команда та же:

```
wg genkey | sudo tee client_private.key | wg pubkey | sudo tee client_public.key
```

Конфигурационный файл сервера

Конфигурационный файл сервера необходимо разместить по пути /etc/wireguard/wg0.conf и заполнить следующим образом(обратить внимание, что значение ключей в файле необходимо заменить):

```
vi /etc/wireguard/wg0.conf
```

```
[Interface]
Address = 10.10.10.1/24
ListenPort = 51820
PrivateKey = “содержимое файла server_private.key”
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE;
ip6tables -A FORWARD -i wg0 -j ACCEPT; ip6tables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE;
ip6tables -D FORWARD -i wg0 -j ACCEPT; ip6tables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
MTU = 1420
[Peer]
PublicKey = “содержимое файла client_public.key”
AllowedIPs = 10.10.10.2/32
```

Файл разделен на две секции:

- Interface - настройка сервера;
- peer - настройка клиентов, которые могут подключаться к серверу, секций Peer может быть несколько.

В данном случае будет настроен сервер WireGuard для работы с IPv4, со следующими основными параметрами:

- Address - адрес сервера в сети VPN;
- ListenPort - порт, на котором будет ожидать подключения WireGuard;
- PrivateKey - приватный ключ сервера, сгенерированный ранее;
- PostUp - команда, которая выполняется после запуска сервера. В данном случае включается поддержка MASQUERADE для интерфейса enp0s8, а также разрешается прием пакетов на интерфейсе wg0. Сетевые интерфейсы вам придется заменить на свои.
- PostDown - выполняется после завершения работы WireGuard, в данном случае удаляет все правила, добавленные в PostUp.

Секции Peer содержат настройки клиентов, которые могут подключится к серверу:

- PublicKey - публичный ключ клиента, сгенерированный ранее;
- AllowedIPs - IP адрес, который может взять клиент. Обратите внимание, маска для IPv4 должна быть 32.

Теперь можно переходить к созданию конфигурационного файла непосредственно для клиента.

Конфигурационный файл клиента Конфигурационный файл клиента будет выглядеть примерно так:

```
vi client.conf
```

```
[Interface]
PrivateKey = “содержимое файла client_private.key”
Address = 10.10.10.2
DNS = 172.17.1.10
MTU = 1384
[Peer]
PublicKey = “содержимое файла server_public.key”
Endpoint = “ip адрес вашего инстанса”:51820
AllowedIPs = 0.0.0.0/0
```

Обратите внимание, что все ключи мы генерируем на сервере, а затем уже скидываем конфигурационный файл клиента на компьютер, который надо подключить к сети.

Рассмотрим подробнее что за что отвечает:

- PrivateKey - приватный ключ клиента, сгенерированный ранее;
- Address - IP адрес интерфейса wg0 клиента;
- DNS - серверы DNS, которые будут использоваться для разрешения доменных имён;
- PublicKey - публичный ключ сервера, к которому надо подключится.
- Endpoint - здесь надо указать IP адрес сервера, на котором установлен WireGuard и порт;
- AllowedIPs - IP адреса, трафик с которых будет перенаправляться в сеть VPN, в данном примере выбраны все адреса.

Запуск сервера

Для запуска сервера используйте такую команду:

```
sudo systemctl start wg-quick@wg0
```

С помощью systemd можно настроить автозагрузку интерфейса:

```
sudo systemctl enable wg-quick@wg0
```

Подключение клиента

Вывести в консоль qr код, для подключения к vpn. Подключаться к ВПН с использованием клиента wireguard с мобильного телефона Для этого в консоли сгенерировать qr код:

```
qrencode -t ansiutf8 < client.conf
```

Далее необходимо проверить, что установленный сервер работает.

Инструкция по проверке подключения находится в конце данного руководства.

Установка wireguard из готового контейнера

Контейнер сам по себе является операционной системой минимального размера, с установленным внутри необходимым программным обеспечением. Контейнеры могут быть преднастроенными, и всё, что необходимо с ними сделать, это установить, передав нужные аргументы. Для работы с контейнерами чаще всего используются docker контейнеры, настроенные на необходимый режим работы путём передачи в них переменных окружения.

Перед установкой Docker нужно выполнить все необходимые настройки системы.

Установка Docker

Docker является набором утилит, для работы с контейнерами. Установку лучше всего выполнять из репозиториев самого Docker. Для начала необходимо установить набор утилит, помогающих работать со сторонними репозиториями.

```
sudo apt-get update
sudo apt-get install -y apt-transport-https ca-certificates curl gnupg lsb-release
```

Далее необходимо скачать ключи доступа к репозиториям (одной командой):

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
| sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

После этого необходимо добавить нужные репозитории (одной командой):

```
echo   "deb [arch=$(dpkg --print-architecture) \
signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \
https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Добавив репозитории, можно установить все необходимые пакеты

```
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io nftables
```

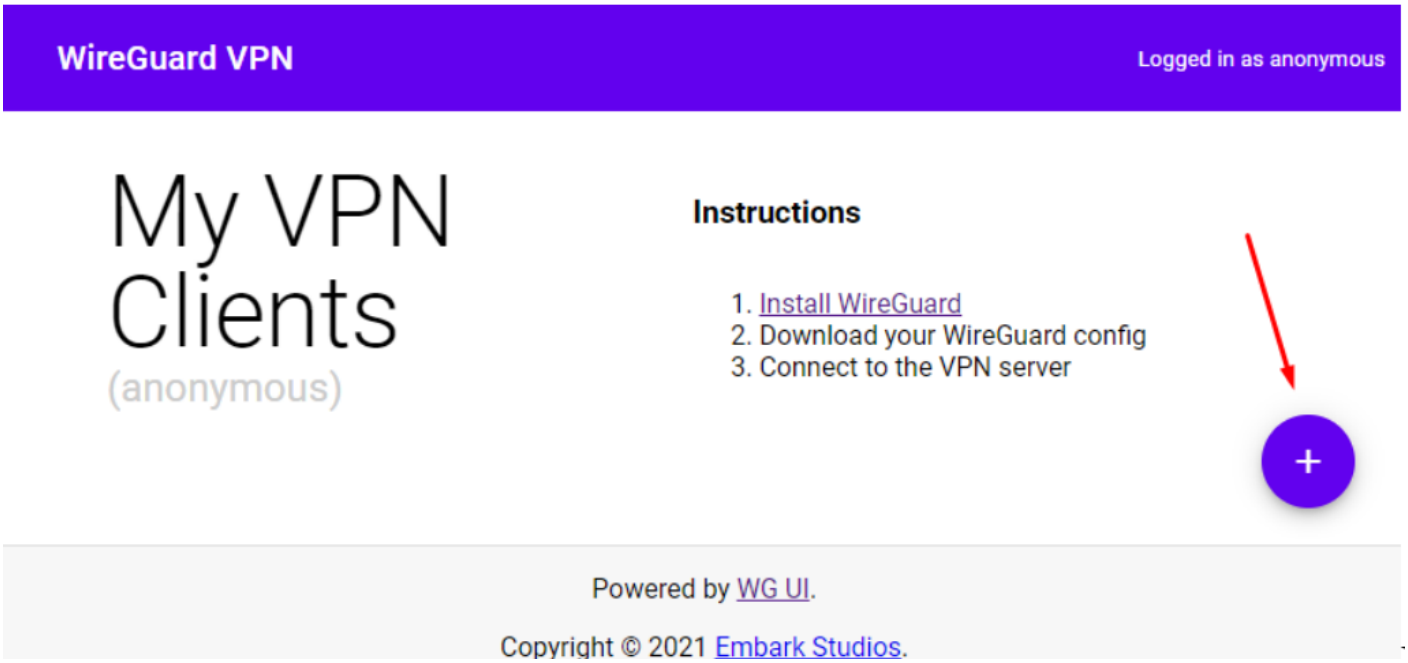
Запуск контейнера

Как только все необходимые приготовления сделаны, необходимо запустить контейнер с нужными параметрами (одной командой):

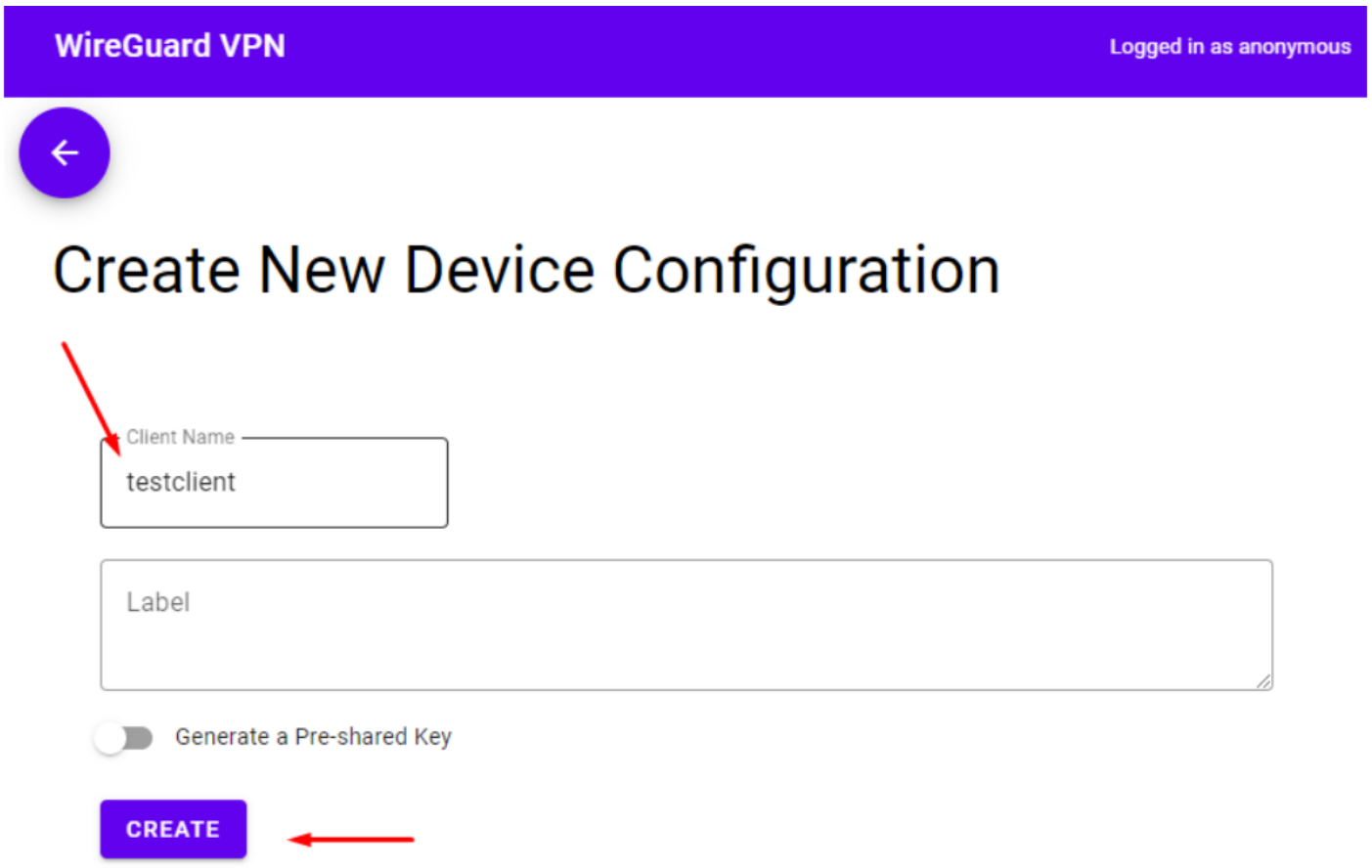
```
sudo docker run -ti -d  --restart=always --network host \
--entrypoint "/wireguard-ui" -v /tmp/wireguard-ui:/data \
--privileged embarkstudios/wireguard-ui:latest  \
--data-dir=/data --wg-listen-port=51820 \
--wg-endpoint="«ip адрес вашего виртуального сервера»:51820" \
--wg-allowed-ips=0.0.0.0/0 --wg-dns="172.17.1.10" \
--wg-device-name="wg0" --listen-address=":80" \
--nat --nat-device="eth0" --client-ip-range="10.0.8.1/24"
```

Настройка VPN тоннелей

Как только контейнер запущен, необходимо подключиться к web интерфейсу VPN сервера для того, чтобы добавить клиентов для подключения. Для этого необходимо открыть в браузере адрес: http://«ip адрес вашего виртуального сервера» В открывшемся окне нажать + в правом нижнем углу



В появившемся окне ввести имя клиента и нажать create



Клиент для подключения создан, так же создан конфигурационный файл для клиента и QR код, по которому можно скачать содержимое этого конфигурационного файла



После создания клиента VPN сервер готов к подключениям внешних клиентов, и можно переходить к проверке.

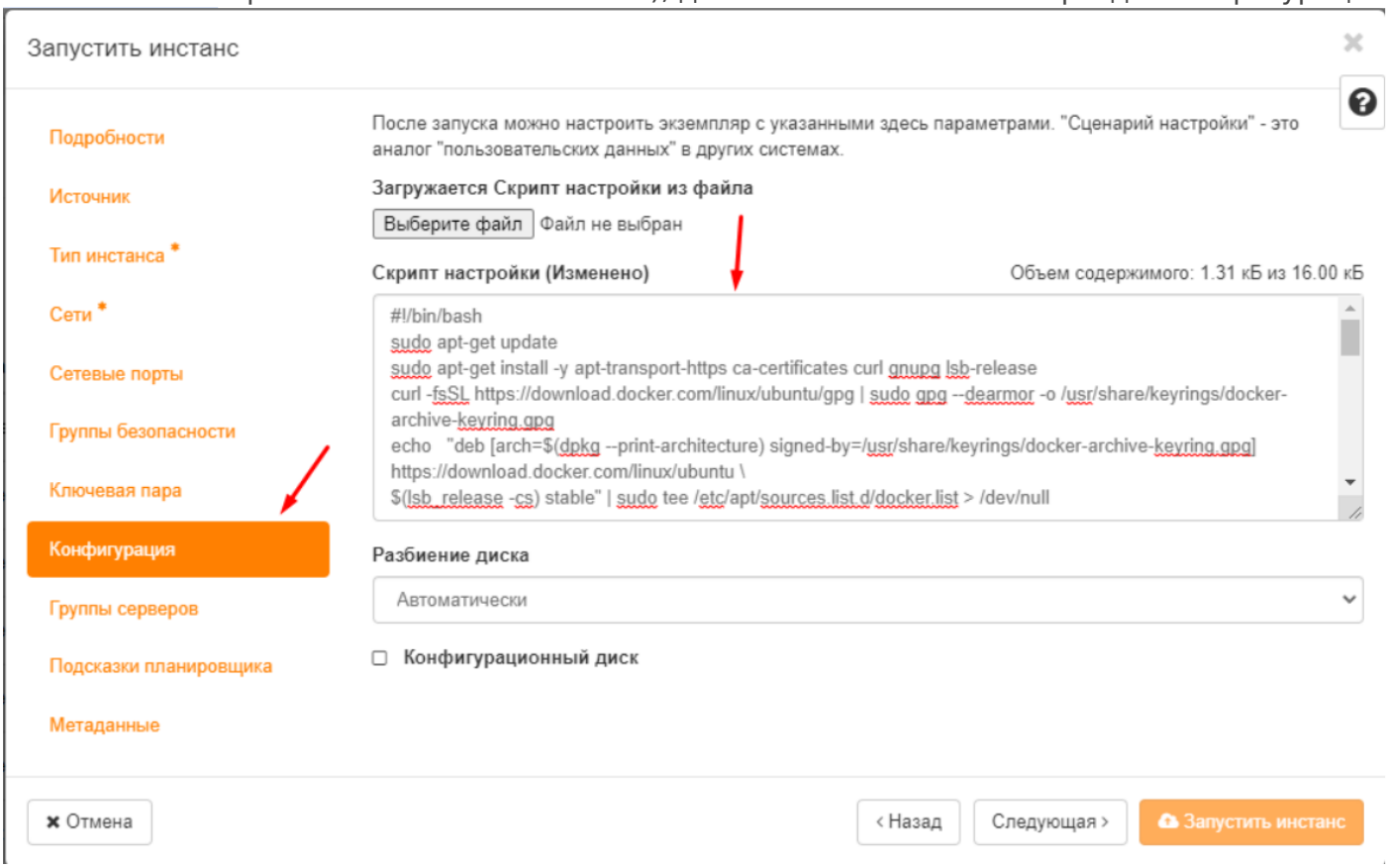
Автоматизация развёртывания облачного сервиса.

Для автоматического развёртывания облачного сервиса, необходимо при создании виртуальной машины дополнительно указать скрипт запуска и развёртывания необходимых сервисов. Для развёртывания этого сервиса необходимо создать скрипт, запускающий контейнер.

> Добавлять в контейнер установку Docker не нужно, так как вы будете запускать из виртуального инстанса с предустановленным и запущенным Docker

Создание преднастроенной виртуальной машины

Необходимо создать виртуальную машину(в данной практической работе необходимо использовать образ ubuntu-server-20:docker), дополнительно заполнив раздел конфигурация



В разделе конфигурация в текстовое поле ввести скрипт, автоматизирующий установку и развёртывание облачного сервиса. Сам скрипт автоматизации должен включать в себя

запуск wireguard контейнера, и запись необходимых параметров в системные файлы.

Пример конфига

```
#!/bin/bash

cat << EOF | sudo tee -a /etc/sysctl.conf
net.ipv4.ip_forward = 1
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.proxy_arp = 0
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
EOF

sudo sysctl -p

ip=$(ip a | grep 172.17 | awk '{print $2}' | awk -F "/" '{print $1}')

sudo docker run -ti -d -p 80:80 -p 51820:51820 --restart=always --network host --entrypoint "/wireguard-ui" -v /tmp/wireguard-ui:/data --privileged embarkstudios/wireguard-ui:latest --data-dir=/data --wg-listen-port=51820 --wg-endpoint="$ip:51820" --wg-allowed-ips=0.0.0.0/0 --wg-dns="172.17.1.10" --wg-device-name="wg0" --listen-address=":80" --nat --nat-device="eth0" --client-ip-range="10.0.8.1/24"

sudo reboot -h now
```

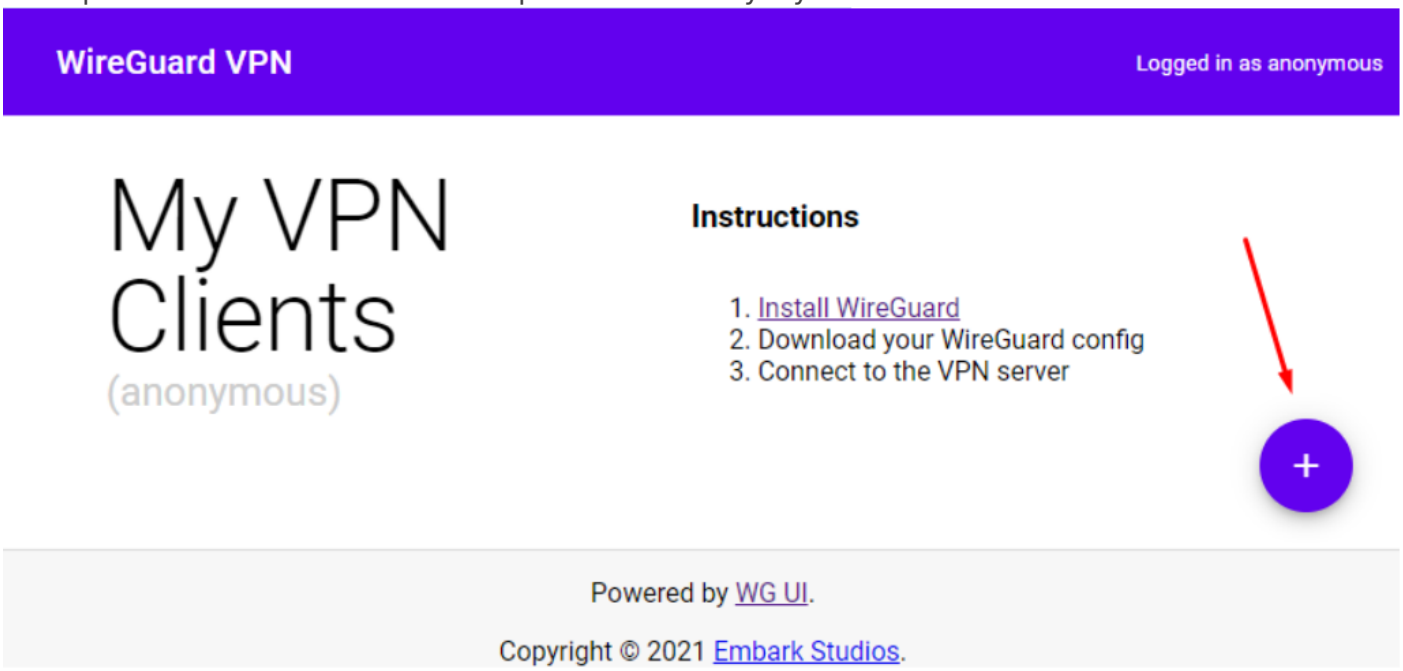
В конце данного скрипта должна быть инструкция на перезагрузку сервера
После запуска виртуальной машины с указанным скриптом необходимо дождаться запуска виртуальной машины и находящихся на ней сервисов.

Настройка VPN тоннелей

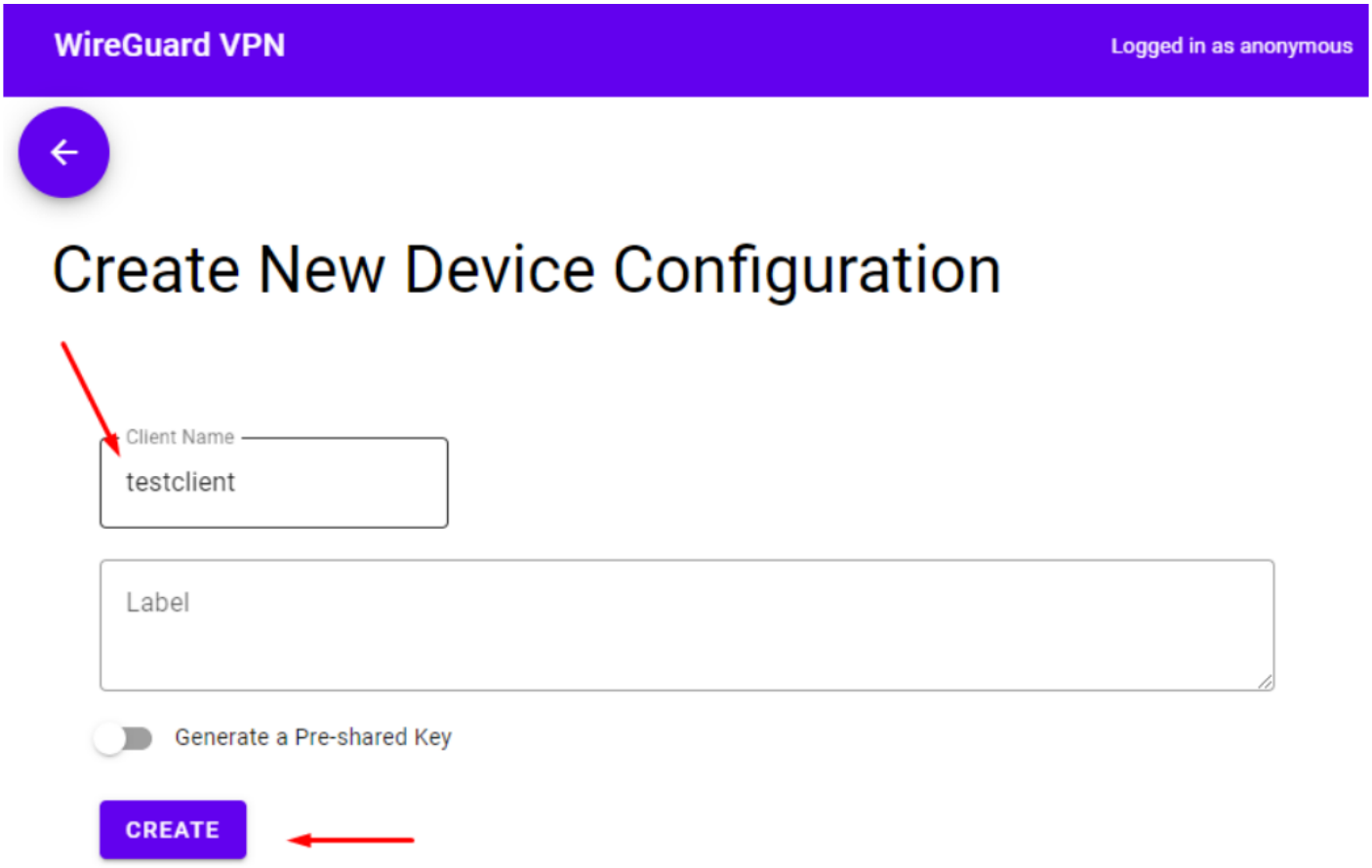
Как только контейнер запущен, необходимо подключиться к web интерфейсу VPN сервера для того, чтобы добавить клиентов для подключения. Для этого необходимо открыть в браузере адрес:

http://«ip адрес вашего виртуального сервера»

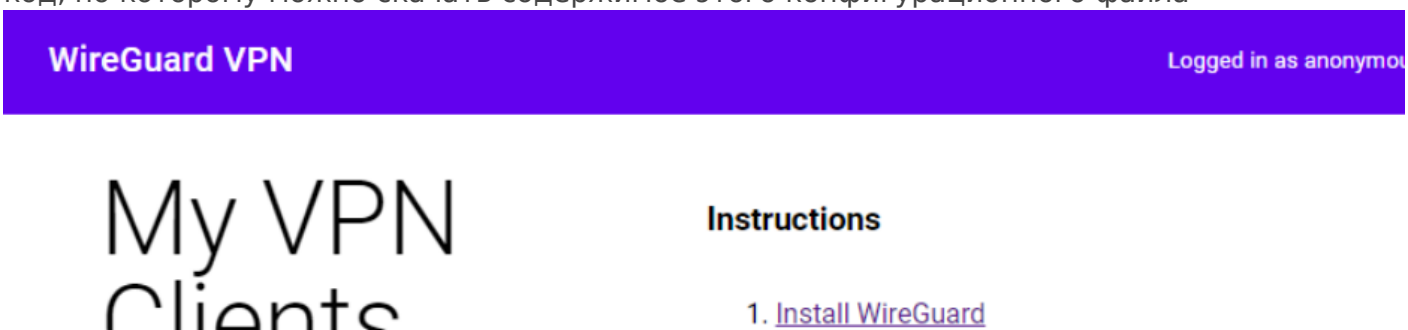
В открывшемся окне нажать + в правом нижнем углу



В появившемся окне ввести имя клиента и нажать create



Клиент для подключения создан, так же создан конфигурационный файл для клиента и QR код, по которому можно скачать содержимое этого конфигурационного файла



После создания клиента VPN сервер готов к подключениям внешних клиентов, и можно переходить к проверке.

Проверка

Для начала необходимо установить клиент для подключения к VPN на мобильный телефон:

Android	**IOS**

Проверку необходимо выполнить со своего персонального устройства. Для начала убедиться, что вы работаете из сети СПбГУТ (Необходимо быть подключенным к WiFi сети) Перед подключением к VPN серверу необходимо проверить свой текущий ip адрес, под которым вас идентифицируют внешние службы. Сделать это можно открыв сайт ifconfig.resds.ru. На этой странице будет показано, с каким ip адресом вы обращайтесь как к этой странице.

Для проверки работоспособности VPN сервера необходимо к нему подключиться, и проверить, изменился ли ваш адрес, под которым вы обращаетесь к внешним службам. Если задание практической части было сделано правильно, то вы должны обращаться к внешним службам от адреса вашего VPN сервера.
Для подключения к VPN серверу необходимо:

1. Открыть скачанное приложение WireGuard
2. Нажать Add a tunnel для добавления VPN тоннеля
3. Выбрать Create from QR code
4. Отсканировать QR код с настройками вашего тоннеля
5. В появившемся меню ввести произвольное имя тоннеля и нажать Save

6. Подключиться к созданному тоннелю, нажав на переключатель в списке подключений Теперь снова необходимо открыть сайт ifconfig.resds.ru. Если значения изменились, можно сделать вывод о том, что ваш трафик идёт через сервер WireGuard

В случае возникновения проблем у WireGuard нет подробных логов, где можно было бы посмотреть какая ошибка произошла, а причин проблем может быть очень много. Чаще всего — это несоответствующие ключи, закрытый порт или неверный адрес сервера. Для исправления этих ошибок необходимо заново проверить все выполненные настройки.

Версия #11
Артем Швидкий создал 17 ноября 2022 16:42:43
Тарабанов Илья Федорович обновил 13 декабря 2023 21:22:51