**OpenVPN** **L3**

**Оглавление**

[**Справка и первоначальные настройки сети** 2](#_Toc31036068)

[**Настройка** 3](#_Toc31036069)

[1. Создание пользователя на будущем сервере 3](#_Toc31036070)

[2. Обновление пакетов 3](#_Toc31036071)

[3. Создание директории центра сертификации 3](#_Toc31036072)

[4. Настройка центра сертификации 3](#_Toc31036073)

[5. Создание центра сертификации 4](#_Toc31036074)

[6. Создание сертификата, ключа и файлов шифрования для сервера 5](#_Toc31036075)

[7. Создание сертификата и пары ключей для клиента 6](#_Toc31036076)

[8. Настройка сервиса OpenVPN 6](#_Toc31036077)

[9. Настройка сетевой конфигурации сервера 8](#_Toc31036078)

[10. Включение сервиса OpenVPN 10](#_Toc31036079)

[11. Создание инфраструктуры настройки клиентов 11](#_Toc31036080)

[12. Генерация конфигураций клиентов 14](#_Toc31036081)

[13. Настройка клиентского устройства 14](#_Toc31036082)

# **Справка и первоначальные настройки сети**

**OpenVPN** — свободная реализация технологии виртуальной частной сети (VPN) с открытым исходным кодом для создания зашифрованных каналов типа точка-точка или сервер-клиенты между компьютерами. Она позволяет устанавливать соединения между компьютерами, находящимися за NAT и сетевым экраном, без необходимости изменения их настроек.

Подробнее о работе и преимуществах и недостатках OpenVPN можно прочитать [тут](https://www.cactusvpn.com/ru/beginners-guide-to-vpn/what-is-openvpn/#how-works).

Для настройки устройств используется простейшая сеть из двух устройств Ubuntu 16.04 и роутера с выходом в интернет. Для удобства настройки используется образ с графической оболочкой и на сервере, и на клиенте.

Первоначально настроены исключительно IP-адреса на интерфейсах.

Router:

int e0/0: 192.168.100.1/30

int e0/1: 192.168.200.1/30

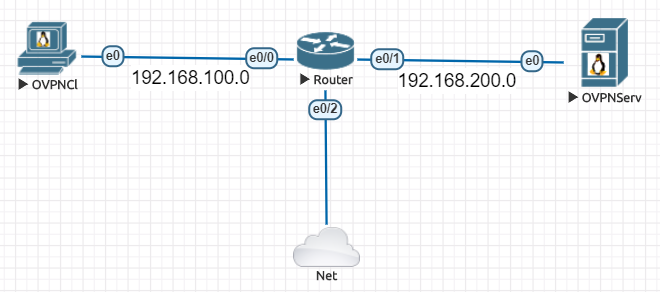
int e0/2: ip addres dhcp

OvpnCl:

eth0: 192.168.100.2

OvpnServ:

eth0: 192.168.200.2



# **Настройка**

Все команды начинаются с символа #. При настройке его использовать не нужно. То, что в командах выделено красным, необходимо менять на свои данные.

## 1. Создание пользователя на будущем сервере

На сервере создаем не рутового пользователя с sudo-привелегиями

# adduser user

Тут нас попросят ввести и повторить пароль для этого пользователя. Затем ввести некоторые данные. Просто нажимаем Enter, оставляя их пустыми

# usermod -aG sudo user

Даем ему привилегии sudo. Следующей командой логинимся под созданным пользователем

# su - user

## 2. Обновление пакетов

Обновим список пакетов сервера и установим необходимые пакеты следующими командами:

# sudo apt-get update

# sudo apt-get install openvpn easy-rsa

Необходимое программное обеспечение установлено и готово к настройке.

## 3. Создание директории центра сертификации

OpenVPN это виртуальная частная сеть, использующая TLS/SSL. Это означает, что OpenVPN использует сертификаты для шифрования трафика между сервером и клиентами. Для выпуска доверенных сертификатов (trusted certificates) нам потребуется создать наш собственный центр сертификации.

Для начала скопируем шаблонную директорию easy-rsa в нашу домашнюю директорию с помощью команды make-cadir:

# make-cadir ~/openvpn-ca

Далее зайдём в эту директорию для начала настройки центра сертификации:

# cd ~/openvpn-ca

## 4. Настройка центра сертификации

Для настройки переменных нашего центра сертификации нам необходимо отредактировать файл vars. Откройте этот файл в вашем текстовом редакторе:

# nano vars

Внутри файла вы найдёте переменные, которые можно отредактировать, и которые задают параметры сертификатов при их создании. Нам нужно изменить всего несколько переменных.

Перейдите ближе к концу файла и найдите настройки полей, используемые по умолчанию при создании сертификатов. Они должны выглядеть примерно так:

~/openvpn-ca/vars

. . .

export KEY\_COUNTRY="US"

export KEY\_PROVINCE="CA"

export KEY\_CITY="SanFrancisco"

export KEY\_ORG="Fort-Funston"

export KEY\_EMAIL="me@myhost.mydomain"

export KEY\_OU="MyOrganizationalUnit"

. . .

Замените значения, выделенные красным, на что-нибудь другое, не оставляйте их не заполненными:

~/openvpn-ca/vars

. . .

export KEY\_COUNTRY="US"

export KEY\_PROVINCE="NY"

export KEY\_CITY="New York City"

export KEY\_ORG="Mirea"

export KEY\_EMAIL="admin@example.com"

export KEY\_OU="Community"

. . .

Пока мы в этом файле, отредактируем значение KEY\_NAME чуть ниже, которое заполняет поле субъекта сертификатов. Для простоты зададим ему название server:

~/openvpn-ca/vars

export KEY\_NAME="server"

Сохраните и закройте файл.

## 5. Создание центра сертификации

Теперь мы можем использовать заданные нами переменные и утилиты easy-rsa для создания центра сертификации.

Убедитесь, что вы находитесь в директории центра сертификации и используйте команду source к файлу vars:

# cd ~/openvpn-ca

# source vars

Вы должны увидеть следующий вывод:

NOTE: If you run ./clean-all, I will be doing a rm -rf on /home/sammy/openvpn-ca/keys

Убедимся, что мы работаем в “чистой среде” выполнив следующую команду:

# ./clean-all

Теперь мы можем создать наш корневой центр сертификации командой:

# ./build-ca

Эта команда запустит процесс создания ключа и сертификата корневого центра сертификации. Поскольку мы задали все переменные в файле vars, все необходимые значения будут введены автоматически. Нажимайте ENTER для подтверждения выбора.

Теперь у нас есть центр сертификации, который мы сможем использовать для создания всех остальных необходимых нам файлов.

## 6. Создание сертификата, ключа и файлов шифрования для сервера

Далее создадим сертификат, пару ключей и некоторые дополнительные файлы, используемые для осуществления шифрования, для нашего сервера.

Начнём с создания сертификата OpenVPN и ключей для сервера. Это можно сделать следующей командой:

**Внимание:**

Если ранее вы выбрали имя, отличное от server, вам придётся немного изменить некоторые инструкции. Например, при копировании созданных файлов в директорию /etc/openvpn вам придётся заменить имена на заданные вами. Вам также придётся изменить файл /etc/openvpn/server.conf для того, чтобы он указывал на корректные .crt и .key файлы.

# ./build-key-server server

Вывод опять будет содержать значения по умолчанию, переданные этой команде (server), а также значения из файла vars.

Согласитесь со всеми значениями по умолчанию, нажимая ENTER. Не задавайте challenge password. В конце процесса два раза введите y для подписи и подтверждения создания сертификата.

Далее создадим оставшиеся файлы. Мы можем сгенерировать сильные ключи протокола Диффи-Хеллмана, используемые при обмене ключами, командой:

#./build-dh

Для завершения этой команды может потребоваться несколько минут.

Далее мы можем сгенерировать подпись HMAC для усиления способности сервера проверять целостность TSL:

# openvpn --genkey --secret keys/ta.key

## 7. Создание сертификата и пары ключей для клиента

Далее мы можем сгенерировать сертификат и пару ключей для клиента. Вообще это можно сделать и на клиентской машине и затем подписать полученный ключ центром сертификации сервера, но в этой статье для простоты мы сгенерируем подписанный ключ на сервере.

В этой статье мы создадим ключ и сертификат только для одного клиента. Если у вас несколько клиентов, вы можете повторять этот процесс сколько угодно раз. Просто каждый раз передавайте уникальное значение скрипту.

Поскольку мы можем вернуться к этому шагу позже, мы повторим команду source для файла vars. Мы будем использовать параметр client1 для создания первого сертификата и ключа.

Для создания файлов без пароля для облегчения автоматических соединений используйте команду build-key:

# cd ~/openvpn-ca

# source vars

# ./build-key client1

При настройке мы это не использовали, но вы можете для создания файлов, защищённых паролем, использовать команду build-key-pass:

#./build-key-pass client1

В ходе процесса создания файлов все значения по умолчанию будут введены, вы можете нажимать ENTER. Не задавайте challenge password и введите y на запросы о подписи и подтверждении создания сертификата.

## 8. Настройка сервиса OpenVPN

**Копирование файлов в директорию OpenVPN**

Нам необходимо скопировать нужные нам файлы в директорию /etc/openvpn.

Сначала скопируем созданные нами файлы. Они находятся в директории ~/openvpn-ca/keys, в которой они и были созданы. Нам необходимо скопировать сертификат и ключ центра сертификации, сертификат и ключ сервера, подпись HMAC и файл Diffie-Hellman:

# cd ~/openvpn-ca/keys

# sudo cp ca.crt ca.key server.crt server.key ta.key dh2048.pem /etc/openvpn

Далее нам необходимо скопировать и распаковать файл-пример конфигурации OpenVPN в конфигурационную директорию, мы будем использовать этот файл в качестве базы для наших настроек:

# gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz | sudo tee /etc/openvpn/server.conf

**Настройка конфигурации OpenVPN**

Теперь, когда наши файлы находятся на своём месте, займёмся настройкой конфигурационного файла сервера:

# sudo nano /etc/openvpn/server.conf

**Базовая настройка**

Сначала найдём секцию HMAC поиском директивы tls-auth. Удалите “;” для того, чтобы раскомментировать строку с tls-auth. Далее добавьте параметр key-direction и установите его значение в “0”:

/etc/openvpn/server.conf

tls-auth ta.key 0 # This file is secret

key-direction 0

Далее найдём секцию шифрования, нас интересуют закомментированные строки cipher. Шифр AES-128-CBC обеспечивает хороший уровень шифрования и широко поддерживается другими программными продуктами. Удалите “;” для раскомментирования строки AES-128-CBC:

/etc/openvpn/server.conf

cipher AES-128-CBC

Под этой строкой добавьте строку auth и выберите алгоритм HMAC. Хорошим выбором будет SHA256:

/etc/openvpn/server.conf

auth SHA256

Наконец, найдите настройки user и group и удалите “;” для раскомментирования этих строк:

/etc/openvpn/server.conf

user nobody

group nogroup

**(Опционально)** Проталкивание изменений DNS для перенаправления всего трафика через VPN

Сделанные нами настройки создают VPN соединение между двумя машинами, но они не заставляют эти машины использовать VPN соединение. Если вы хотите использовать VPN соединение для всего своего трафика, вам необходимо протолкнуть (push) настройки DNS на клиентские машины.

Для этого вам необходимо раскомментировать несколько директив. Найдите секцию redirect-gateway и удалите “;” из начала строки для расскоментирования redirect-gateway:

/etc/openvpn/server.conf

push "redirect-gateway def1 bypass-dhcp"

Чуть ниже находится секция dhcp-option. Удалите “;” для обеих строк:

/etc/openvpn/server.conf

push "dhcp-option DNS 208.67.222.222"

push "dhcp-option DNS 208.67.220.220"

Это позволит клиентам сконфигурировать свои настройки DNS для использования VPN соединения в качестве основного.

## 9. Настройка сетевой конфигурации сервера

Далее нам необходимо настроить сетевую конфигурацию сервера, чтобы OpenVPN мог корректно перенаправлять трафик.

**Настройка перенаправления IP**

Сначала разрешим серверу перенаправлять трафик. Это ключевая функциональность нашего VPN сервера.

Настроим это в файле /etc/sysctl.conf:

# sudo nano /etc/sysctl.conf

Найдите строку настройки net.ipv4.ip\_forward. Удалите “#” из начала строки, чтобы раскомментировать её:

/etc/sysctl.conf

net.ipv4.ip\_forward=1

Сохраните и закройте файл.

Для применения настроек к текущей сессии наберите команду:

# sudo sysctl -p

Настройка правил UFW для сокрытия соединений клиентов

Наберите команду:

# ip route | grep default

Публичный интерфейс должен следовать за словом “dev”. Например, в нашем случае этот интерфейс называется eth0:

Вывод

default via 203.0.113.1 dev wlp11s0 proto static metric 600

Зная название интерфейса откроем файл /etc/ufw/before.rules и добавим туда соответствующие настройки:

# sudo nano /etc/ufw/before.rules

Это файл содержит настройки UFW, которое применяются перед применением правил UFW. Добавьте в начало файла выделенные красным строки. Это настроит правила, применяемые по умолчанию, к цепочке POSTROUTING в таблице nat и будет скрывать весь трафик от VPN:

Внимание: не забудьте заменить eth0 в строке -A POSTROUTING на имя интерфейса, найденное нами ранее.

/etc/ufw/before.rules

#

# rules.before

#

# Rules that should be run before the ufw command line added rules. Custom

# rules should be added to one of these chains:

# ufw-before-input

# ufw-before-output

# ufw-before-forward

#

# START OPENVPN RULES

# NAT table rules

\*nat

:POSTROUTING ACCEPT [0:0]

# Allow traffic from OpenVPN client to eth0

-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE

COMMIT

# END OPENVPN RULES

# Don't delete these required lines, otherwise there will be errors

\*filter

. . .

Сохраните и закройте файл.

Теперь мы должны сообщить UFW, что ему по умолчанию необходимо разрешать перенаправленные пакеты. Для этого откройте файл /etc/default/ufw:

# sudo nano /etc/default/ufw

Найдите в файле директиву DEFAULT\_FORWARD\_POLICY. Мы изменим значение с DROP на ACCEPT:

/etc/default/ufw

DEFAULT\_FORWARD\_POLICY="ACCEPT"

Сохраните и закройте файл.

**Открытие порта OpenVPN и применение изменений**

Далее настроим сам файервол для разрешения трафика в OpenVPN.

Если вы не меняли порт и протокол в файле /etc/openvpn/server.conf, вам необходимо разрешить трафик UDP для порта 1194. Если вы изменили эти настройки, введите указанные вами значения.

Даже мы добавим порт SSH на случай, если вы не сделали этого ранее.

# sudo ufw allow 1194/udp

# sudo ufw allow OpenSSH

Теперь деактивируем и активируем UFW для применения внесённых изменений:

# sudo ufw disable

# sudo ufw enable

Теперь наш сервер сконфигурирован для обработки трафика OpenVPN.

## 10. Включение сервиса OpenVPN

Мы готовы включит сервис OpenVPN на нашем сервере. Мы можем сделать это с помощью systemd.

Нам необходимо запустить сервер OpenVPN указав имя нашего файла конфигурации в качестве переменной после имени файла systemd. Файл конфигурации для нашего сервера называется /etc/openvpn/server.conf, поэтому мы добавим @server в конец имени файла при его вызове:

# sudo systemctl start openvpn@server

Убедимся, что сервис успешно запущен командой:

# sudo systemctl status openvpn@server

Если всё получилось, вывод должен выглядеть примерно следующим образом:

openvpn@server.service - OpenVPN connection to server

Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)

Active: active (running) since Tue 2016-05-03 15:30:05 EDT; 47s ago

Docs: man:openvpn(8)

https://community.openvpn.net/openvpn/wiki/Openvpn23ManPage

https://community.openvpn.net/openvpn/wiki/HOWTO

Process: 5852 ExecStart=/usr/sbin/openvpn --daemon ovpn-%i --status /run/openvpn/%i.status 10 --cd /etc/openvpn --script-security 2 --config /etc/openvpn/%i.conf --writepid /run/openvpn/%i.pid (code=exited, sta

Main PID: 5856 (openvpn)

Tasks: 1 (limit: 512)

CGroup: /system.slice/system-openvpn.slice/openvpn@server.service

└─5856 /usr/sbin/openvpn --daemon ovpn-server --status /run/openvpn/server.status 10 --cd /etc/openvpn --script-security 2 --config /etc/openvpn/server.conf --writepid /run/openvpn/server.pid

May 03 15:30:05 openvpn2 ovpn-server[5856]: /sbin/ip addr add dev tun0 local 10.8.0.1 peer 10.8.0.2

May 03 15:30:05 openvpn2 ovpn-server[5856]: /sbin/ip route add 10.8.0.0/24 via 10.8.0.2

May 03 15:30:05 openvpn2 ovpn-server[5856]: GID set to nogroup

May 03 15:30:05 openvpn2 ovpn-server[5856]: UID set to nobody

May 03 15:30:05 openvpn2 ovpn-server[5856]: UDPv4 link local (bound): [undef]

May 03 15:30:05 openvpn2 ovpn-server[5856]: UDPv4 link remote: [undef]

May 03 15:30:05 openvpn2 ovpn-server[5856]: MULTI: multi\_init called, r=256 v=256

May 03 15:30:05 openvpn2 ovpn-server[5856]: IFCONFIG POOL: base=10.8.0.4 size=62, ipv6=0

May 03 15:30:05 openvpn2 ovpn-server[5856]: IFCONFIG POOL LIST

May 03 15:30:05 openvpn2 ovpn-server[5856]: Initialization Sequence Completed

Вы также можете проверить доступность интерфейса OpenVPN tun0 следующей командой:

# ip addr show tun0

Вы должны увидеть конфигурацию интерфейса:

4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 100

link/none

inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0

valid\_lft forever preferred\_lft forever

Если всё в порядке, настроем сервис на автоматическое включение при загрузке сервера:

# sudo systemctl enable openvpn@server

## 11. Создание инфраструктуры настройки клиентов

Далее настроим систему для простого создания файлов конфигурации для клиентов.

**Создание структуры директорий конфигурации клиентов**

В домашней директории сервера создайте структуру директорий для хранения файлов:

# mkdir -p ~/client-configs/files

Поскольку наши файлы конфигурации будут содержать клиентские ключи, мы должны настроить права доступа для созданных директорий:

# chmod 700 ~/client-configs/files

**Создание базовой конфигурации**

Далее скопируем конфигурацию-пример в нашу директорию для использования в качестве нашей базовой конфигурации:

# cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ~/client-configs/base.conf

Откройте этот файл в вашем текстовом редакторе:

# nano ~/client-configs/base.conf

Сначала найдите директиву remote. Эта директива сообщает клиенту адрес нашего сервера OpenVPN. Это должен быть публичный IP адрес вашего сервера OpenVPN. Если вы изменили порт, который слушает сервер OpenVPN, измените порт по умолчанию 1194 (если ранее вы его меняли) на ваше значение:

~/client-configs/base.conf

. . .

# The hostname/IP and port of the server.

# You can have multiple remote entries

# to load balance between the servers.

remote server\_IP\_address 1194

. . .

Убедитесь, что протокол совпадает с настройками сервера:

~/client-configs/base.conf

proto udp

Далее раскомментируйте директивы user и group удаляя “;”:

~/client-configs/base.conf

# Downgrade privileges after initialization (non-Windows only)

user nobody

group nogroup

Найдите директивы ca, cert и key. Закомментируйте эти директивы, так как мы будем добавлять сертификаты и ключи в самом файле:

~/client-configs/base.conf

# SSL/TLS parms.

# See the server config file for more

# description. It's best to use

# a separate .crt/.key file pair

# for each client. A single ca

# file can be used for all clients.

#ca ca.crt

#cert client.crt

#key client.key

Добавьте настройки cipher и auth согласно заданным в файле /etc/openvpn/server.conf:

~/client-configs/base.conf

cipher AES-128-CBC

auth SHA256

Далее добавьте директиву key-direction в любое место в файле. Она должна иметь значение “1” для корректной работы сервера:

~/client-configs/base.conf

key-direction 1

Наконец, добавьте несколько закомментированных строк. Мы ходим добавить эти строки в каждый файл конфигурации, но они будут включены только для клиентов на Linux, которые используют файл /etc/openvpn/update-resolv-conf. Этот скрипт использует утилиту resolvconf для обновление информации DNS на клиентах Linux.

~/client-configs/base.conf

# script-security 2

# up /etc/openvpn/update-resolv-conf

# down /etc/openvpn/update-resolv-conf

Если ваш клиент работает на Linux и использует файл /etc/openvpn/update-resolv-conf, вы должны раскомментировать эти строки в сгенерированном клиентском файле конфигурации OpenVPN.

Сохраните и закройте файл.

**Создание скрипта генерации файлов конфигурации**

Теперь создадим простой скрипт для генерации файлов конфигурации с релевантными сертификатами, ключами и файлами шифрования. Он будет помещать сгенерированные файла конфигурации в директорию ~/client-configs/files. Создайте и откройте файл make\_config.sh внутри директории ~/client-configs:

# nano ~/client-configs/make\_config.sh

Вставьте следующие текст в этот файл:

#!/bin/bash

# First argument: Client identifier

KEY\_DIR=~/openvpn-ca/keys

OUTPUT\_DIR=~/client-configs/files

BASE\_CONFIG=~/client-configs/base.conf

cat ${BASE\_CONFIG} \

<(echo -e '<ca>') \

${KEY\_DIR}/ca.crt \

<(echo -e '</ca>\n<cert>') \

${KEY\_DIR}/${1}.crt \

<(echo -e '</cert>\n<key>') \

${KEY\_DIR}/${1}.key \

<(echo -e '</key>\n<tls-auth>') \

${KEY\_DIR}/ta.key \

<(echo -e '</tls-auth>') \

> ${OUTPUT\_DIR}/${1}.ovpn

Сохраните и закройте файл. Сделайте его исполняемым файлом командой:

# chmod 700 ~/client-configs/make\_config.sh

## 12. Генерация конфигураций клиентов

Теперь мы можем легко сгенерировать файлы конфигурации клиентов.

Если вы следовали всем шагам, то у вас создан сертификат client1.crt и ключ клиента client1.key командой ./build-key client1. Вы можете сгенерировать конфигурацию для этих файлов перейдя в директорию ~/client-configs и используя только что созданный нами скрипт:

# cd ~/client-configs

# ./make\_config.sh client1

Если всё прошло успешно, мы должны получить файл client1.ovpn в директории ~/client-configs/files:

# ls ~/client-configs/files

Вывод

client1.ovpn

## 13. Настройка клиентского устройства

**Доставка конфигураций клиентам**

Теперь мы должны переместить файл конфигурации на клиентское устройство. Мы используем в качестве клиента ubuntu 16.04. На клиенте выполним следующую команду:

# sftp user@openvpn\_server\_ip:client-configs/files/client1.ovpn ~/

**Установка**

Выполним обновление и установку необходимого пакета:

# sudo apt-get update

# sudo apt-get install openvpn

**Настройка**

Сначала проверьте, содержит ли ваш дистрибутив скрипт /etc/openvpn/update-resolv-conf:

# ls /etc/openvpn

Вывод

update-resolve-conf

Далее отредактируйте полученный с сервера файл конфигурации клиента OpenVPN:

# nano client1.ovpn

Если вам удалось найти файл update-resolv-conf, раскомментируйте следующие строки файла:

client1.ovpn

script-security 2

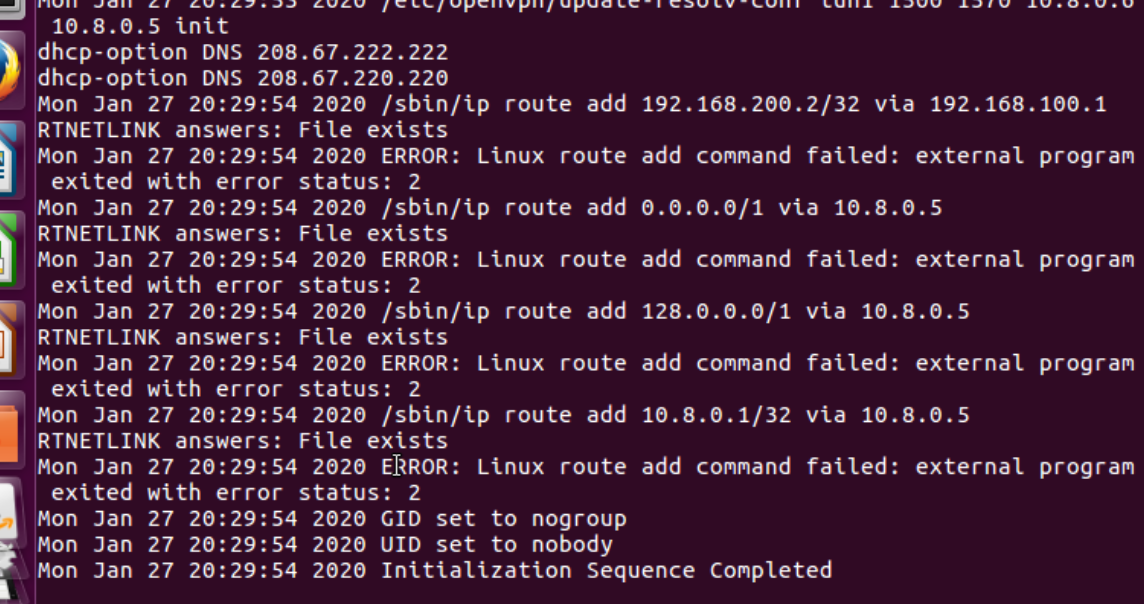
up /etc/openvpn/update-resolv-conf

down /etc/openvpn/update-resolv-conf

Теперь вы можете соединиться с VPN используя команду openvpn следующим образом:

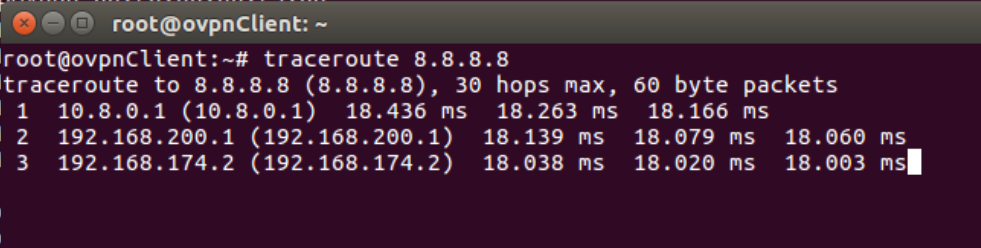
# sudo openvpn --config client1.ovpn

В результате вы подключитесь к серверу, появится вывод о подключении и его настройках, терминал становится неактивен. Для продолжения работы необходимо открыть новый.



При отсутствии графической оболочки новый терминал можно открыть сочетанием клавиш Ctrl+Alt+F1.Последняя клавиша – F<Номер термаинала>, может быть с любым номером.

Проверяем:



Трассировка идет через IP-адрес сервера, значит, всё работает успешно.