В современном интернете вопрос анонимности встает с каждым годом все острее. Запреты на доступ к контенту вынуждают пользователей искать обходные пути, одним из которых является использование приватных тоннелей. Установка на CentOS openvpn сервера для подключения удаленных клиентов является одной из реализаций шифрованного vpn канала. Этой задачей мы и займемся в текущей статье — поднимем шифрованный тоннель и подключим клиентов.

Содержание:

Данная статья является частью единого цикла статьей про сервер Centos.

Введение — что такое vpn server?

Упомянутое во вступлении применение технологии vpn и openvpn сервера в частности не ограничивается созданием каналов для анонимного трафика пользователей. Более того, я думаю это не основная сфера применения данных технологий. Давайте поподробнее познакомимся с этими вещами, чтобы иметь полное представление о том, что мы будем настраивать.

VPN — набор технологий, которые позволяют организовать логическую сеть поверх других. Чаще всего в роли других сетей выступает Интернет. Если простыми словами, то с помощью VPN можно организовать единую локальную сеть разделенных интернетом сегментов сети. Так как Интернет — общедоступная сеть, то трафик внутри созданной логической сети шифруется различными средствами для организации защиты передаваемых данных.

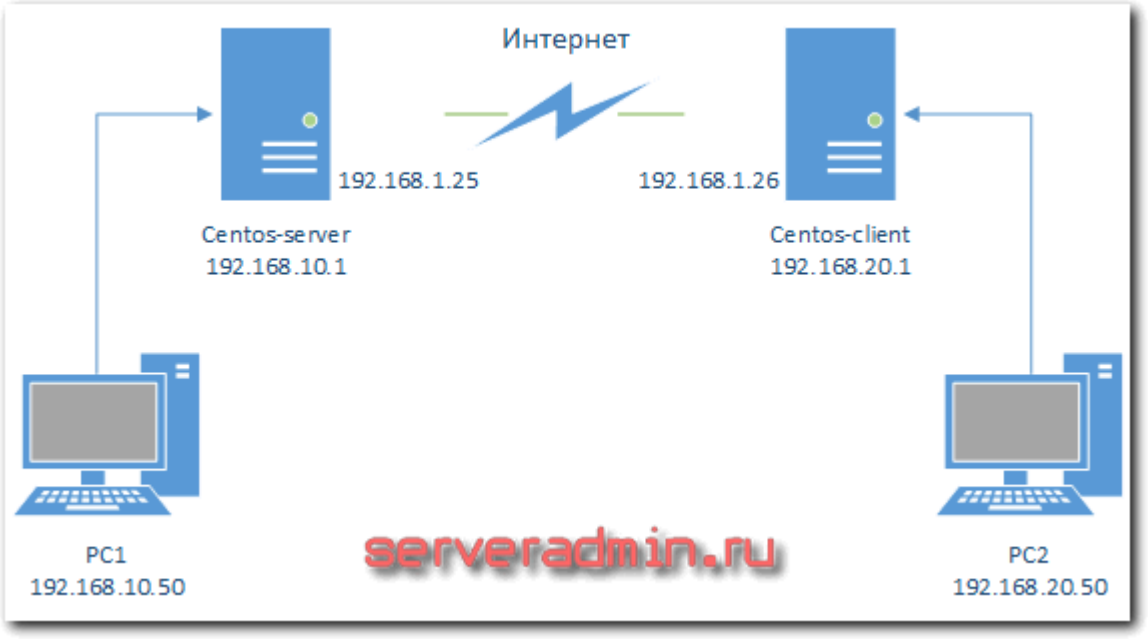
OpenVPN — одна из реализаций технологии VPN с открытым исходным кодом, а значит бесплатная. С ее помощью можно объединять в единую сеть компьютеры в том числе и находящиеся за NAT, что очень удобно. Openvpn поддерживает все популярные на сегодняшний день операционные системы, в том числе и Windows.

Среди малого и среднего бизнеса сервер openvpn очень популярен благодаря своей бесплатности, кроссплатформенности, быстроте и гибкости настроек. Лично я предпочитаю именно его для объединения удаленных локальных сетей. Его же предпочитают использовать vpn-провайдеры для оказания своих услуг по организации анонимного серфинга в интернете.

В своей статье я рассмотрю не абстрактную установку и настройку сервера, а приведу конкретный пример соединения локальных сетей двух офисов в единую логическую сеть с совместным доступом к ресурсам друг друга.

Объединение офисов с помощью openvpn

У нас имеется офис с шлюзом CentOS 7, на который мы будем устанавливать openvpn сервер. И есть филиал с таким же шлюзом, где будет установлен openvpn client для подключения сети филиала к офису:



Описание схемы сети

Имя	Centos-server	PC1	Centos-client	PC2
Внешний ip	192.168.1.25		192.168.1.26	
Локальный ip	192.168.10.1	192.168.10.50	192.168.20.1	192.168.20.50
Комментарий	Сервер openvp и шлюз в офисе	Компьютер с Windows 7 в офисе	Клиент openvp и шлюз в филиале	Компьютер с Windows 7 в филиале

В данном случае сервер в филиале может быть без внешнего белого IP адреса, это не принципиально, все будет работать и так. Нам необходим только один внешний IP адрес на сервере. Все остальные клиенты могут быть за NAT, это не помешает успешному объединению локальных сетей.

Наша задача в данном случае будет сводиться к тому, чтобы компьютеры PC1 и PC2 увидели друг друга и могли совместно использовать свои сетевые ресурсы в обе стороны. То есть как за сервером, так и за клиентом openvpn мы должны видеть сеть.

Хочу отметить, что конфигурация openvpn кроссплатформенная и отлично переносится с одной системы на другую, необходимо только пути проверить, так как в каждой системе они свои. Моя инструкция подойдет для настройки openvpn сервера на любой операционной системе, отличаться будут только специфичные для каждой системы команды установки и проверки, но сама суть настройки vpn будет такой же.

Данный материал я создавал на тестовом стенде, который специально собрал для написания статьи. Но все настройки взяты с реально работающих серверов, причем разных систем, в том числе и freebsd. Для практического применения нужно просто поменять IP адреса на свои. У меня на стенде 192.168.1.25 и 192.168.1.26 по сути внешние IP адреса, которые смотрят в интернет.

Где скачать openvpn

Прежде чем приступить к установке и настройке, давайте посмотрим, где можно скачать все, что нам понадобится в нашей работе. Свежие и актуальные версии всегда можно найти на страничке downloads официального сайта.

Где можно скачать openvpn

Исходники	openvpn-2.4.5.zip
Openvpn client Windows XP 32 bit	openvpn-install-2.3.18-I002-i686.exe
Openvpn client Windows XP 64 bit	openvpn-install-2.3.18-I002-x86_64.exe
Openvpn client Windows Vista, 7, 8, 10	openvpn-install-2.4.5-I601.exe
Openvpn Portable	OpenVPNPortable_1.8.2.paf.exe

Сразу же прокомментирую по поводу **portable** версии openvpn. На текущий момент актуальной портированной версии openvpn не существует. Та версия, на которую я дал ссылку была выпущена 2014-03-26 и проект с тех пор закрыт. В некоторых случаях эта версия работает, но не всегда и не везде. Я специально по этому поводу почитал форум Community Openvpn и нашел там неутешительные ответы. Разработчики говорят, что portable версия openvpn не поддерживается и выпускать ее они даже не собираются. А жаль, это было бы удобно.

Для Linux систем дистрибутив проще всего получить в системных репозиториях и устанавливать с помощью стандартных установщиков пакетов.

Установка openvpn на CentOS 7

Теперь вернемся к нашей задаче. Я подразумеваю, что у вас уже есть установленный и настроенный сервер CentOS 7. Если еще нет, то можете воспользоваться моими материалами. В них раскрыты следующие темы:

- Подробное описание установки CentOS 7 с разбором всех этапов и параметров.
- Настройка CentOS 7 шаг за шагом — базовая конфигурация сервера.

Устанавливаем openvpn на оба наших сервера, которые являются шлюзами в своих сегментах сети. Первым делом подключаем репозиторий epel к centos 7:

```
# yum -y install epel-release
```

Выполняем непосредственно установку:

```
# yum -y install openvpn
```

Создание сертификатов

Для работы сервера openvpn необходимо создать соответствующие сертификаты. Для этого воспользуемся программой Easy-RSA, которая генерирует сертификаты с помощью утилиты openssl. Все работы в этом разделе выполняем только на centos-server.

Создаем директорию для ключей:

```
# mkdir /etc/openvpn/keys
```

Устанавливаем архиватор zip:

```
# yum -y install unzip zip
```

Скачиваем и устанавливаем утилиту Easy-RSA:

```
# cd /etc/openvpn/keys
# wget https://github.com/OpenVPN/easy-rsa/archive/master.zip
```

Если получаете сообщение:

```
-bash: wget: command not found
```

То устанавливаете **wget**:

```
# yum -y install wget
```

Продолжаем:

```
# unzip master.zip
# cd /etc/openvpn/keys/easy-rsa-master/easyrsa3
```

Создаем структуру публичных PKI ключей:

```
# mv vars.example vars
# ./easyrsa init-pki
Note: using Easy-RSA configuration from: ./vars
init-pki complete; you may now create a CA or requests.
 Your newly created PKI dir is: /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki
```

Создайте удостоверяющий центр CA:

```
# ./easyrsa build-ca
```

```
Note: using Easy-RSA configuration from: ./vars
 Generating a 2048 bit RSA private key
 .+++
 ..............................................................................+++
 writing new private key to '/etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/private/ca.key.GKwCGouHpy'
 Enter PEM pass phrase:
 Verifying - Enter PEM pass phrase:
 – ---
 You are about to be asked to enter information that will be incorporated
 into your certificate request.
 What you are about to enter is what is called a Distinguished Name or a DN.
 There are quite a few fields but you can leave some blank
 For some fields there will be a default value,
 If you enter '.', the field will be left blank.
 – ---
 Common Name (eg: your user, host, or server name) [Easy-RSA CA]:serveradmin.ru
CA creation complete and you may now import and sign cert requests.
 Your new CA certificate file for publishing is at:
 /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/ca.crt
```

Не забудьте указанный пароль. Его нужно будет вводить каждый раз при создании нового сертификата openvpn.

Мы получили 2 ключа:

- /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/private/**ca.key**
- /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/**ca.crt**

Первый ключ секретный, его нужно оставить на сервере и никому не отдавать. Второй — открытый, его мы будем вместе с пользовательскими сертификатами передавать клиентам.

Создаем запрос сертификата для сервера без пароля с помощью опции **nopass**, иначе придется вводить пароль с консоли при каждом запуске сервера:

```
# ./easyrsa gen-req server nopass
```

```
Note: using Easy-RSA configuration from: ./vars
 Generating a 2048 bit RSA private key
 ........................+++
 ..................................................................................+++
 writing new private key to '/etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/private/server.key.vCUQuueIih'
 — ---
 You are about to be asked to enter information that will be incorporated
 into your certificate request.
 What you are about to enter is what is called a Distinguished Name or a DN.
 There are quite a few fields but you can leave some blank
 For some fields there will be a default value,
 If you enter '.', the field will be left blank.
 — ---
 Common Name (eg: your user, host, or server name) [server]:
Keypair and certificate request completed. Your files are:
 req: /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/reqs/server.req
 key: /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/private/server.key
```

Подписываем запрос на получение сертификата у нашего CA:

```
# ./easyrsa sign-req server server
```

```
Note: using Easy-RSA configuration from: ./vars
You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 3650 days:

subject=
commonName = server
Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
Using configuration from /etc/openvpn/keys/easy-rsa-master/easyrsa3/openssl-1.0.cnf
Enter pass phrase for /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :PRINTABLE:'server'
Certificate is to be certified until Sep 10 00:31:21 2025 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated

Certificate created at: /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/issued/server.crt
```

В процессе работы скрипта вводим пароль от CA, который указывали раньше и отвечаем на вопрос yes. Мы получили подписанный удостоверяющим центром сертификат для сервера — /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/issued/**server.crt**

Нам еще пригодится ключ Диффи-Хелмана, генерируем его:

```
# ./easyrsa gen-dh
```

По завершению работы скрипта получаем файл dh сертификата — /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/**dh.pem**.

Копируем в папку /etc/openvpn все необходимые для работы openvpn сервера ключи:

```
# cp pki/ca.crt /etc/openvpn/ca.crt
# cp pki/dh.pem /etc/openvpn/dh.pem
# cp pki/issued/server.crt /etc/openvpn/server.crt
# cp pki/private/server.key /etc/openvpn/server.key
```

Создадим ключ для клиента openvpn:

```
# ./easyrsa gen-req client nopass
# ./easyrsa sign-req client client
```

Процедура аналогична созданию сертификата для сервера. Так же вводим пароль, отвечаем yes. В результате получаем подписанный сертификат клиента:

- /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/issued/**client.crt**
- /etc/openvpn/keys/easy-rsa-master/easyrsa3/pki/private/**client.key**

Клиенту, которым у нас является шлюз филиала нужно будет передать следующий набор файлов — **client.crt, client.key, ca.crt**.

Настройка openvpn на CentOS 7

Теперь приступаем к настройке. Создаем файл конфигурации openvpn:

```
# mcedit /etc/openvpn/server.conf
```

```
port 13555 # я предпочитаю использовать нестандартные порты для работы
proto udp # протокол может быт и tcp, если есть необходимость в этом
dev tun

ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key
dh /etc/openvpn/dh.pem

server 10.0.0.0 255.255.255.0 # подсеть для туннеля, может быть любой
route 192.168.20.0 255.255.255.0 # указываем подсеть, к которой будем обращаться через vpn
push "route 192.168.20.0 255.255.255.0" # передаем маршрут клиентам

ifconfig-pool-persist ipp.txt # файл с записями соответствий clinet - ip
client-to-client # позволяет клиентам openvpn подключаться друг к другу
client-config-dir /etc/openvpn/ccd # директория с индивидуальными настройками клиентов

keepalive 10 120
comp-lzo
persist-key
persist-tun
status /var/log/openvpn/openvpn-status.log
log /var/log/openvpn/openvpn.log
verb 3
```

Создаем необходимые директории:

```
# mkdir /etc/openvpn/ccd && mkdir /var/log/openvpn
```

Создаем файл конфигурации клиента в папке, указанной в параметре client-config-dir :

```
mcedit /etc/openvpn/ccd/client
```

```
iroute 192.168.20.0 255.255.255.0
```

Здесь **client** — имя сертификата пользователя. Параметр **iroute** означает, что за подсеть 192.168.20.0/24 отвечает именно этот клиент. Если в openvpn не передать эту настройку, то сеть, находящуюся за клиентом будет не видно, при этом сам клиент будет видеть всю сеть, которую обслуживает сервер. Такой вариант подходит для подключения удаленных сотрудников.

Выбор устройства openvpn — TAP или TUN

Чуть подробнее остановлюсь на этом моменте. В моей конфигурации я использую **tun** интерфейс. В чем отличие tun от tap можно прочитать на википедии. Прокомментирую своими словами. Если вам нужно объединить две разные локальные сети в одну условно общую, но с разной адресацией, то вам нужен tun. То есть в нашем случае мы объединяем две сети 192.168.10.0/24 и 192.168.20.0/24 для взаимного совместного доступа.

Если же у вас стоит задача объединить 2 удаленные сети в единое адресное пространство, например сделать и в офисе и в филиале единую сеть 192.168.10.0/24, то тогда бы мы использовали **tap** интерфейс и указывали бы на компьютерах в обоих сетях не пересекающиеся адреса из одной подсети. В таком состоянии openvpn работает в режиме моста. По мне так удобнее первый вариант. Я еще не сталкивался с задачей, где бы мне нужен был tap. Вернемся к настройке.

Запускаем сервер:

```
# systemctl start openvpn@server
```

Если сервер не запустился, а в логе ошибка:

```
TCP/UDP: Socket bind failed on local address [undef]: Permission denied
```

Значит вам нужно либо правильно настроить, либо отключить SELinux. В данном материале я не хочу касаться настройки SELinux, поэтому просто отключаем его:

```
# mcedit /etc/sysconfig/selinux
```

меняем значение

```
SELINUX=disabled
```

Чтобы изменения вступили в силу, перезагружаемся:

```
# reboot
```

Пробуем снова запустить openvpn сервер:

```
# systemctl start openvpn@server
```

Проверяем, запустился или нет:

```
# netstat -tulnp | grep 13555
```

```
udp 0 0 0.0.0.0:13555 0.0.0.0:* 2472/openvpn
```

Отлично, запустился на указанном порту.

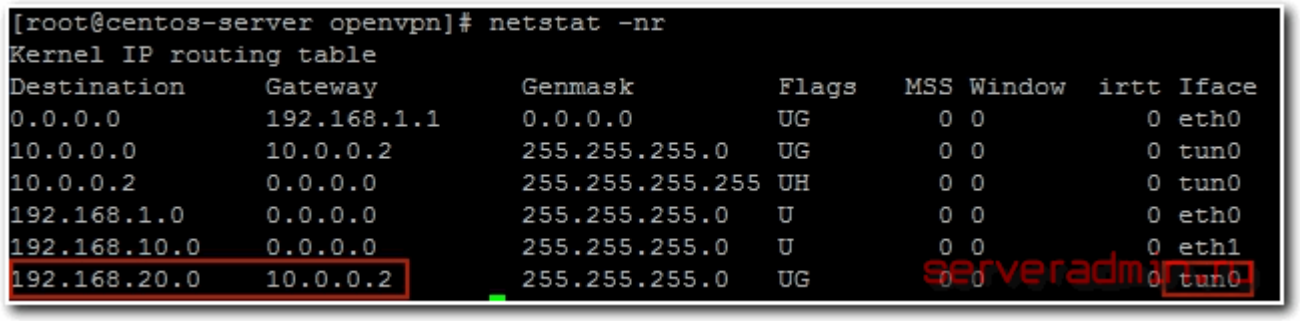
Добавляем сервер openvpn в автозагрузку:

```
# systemctl enable openvpn@server

ln -s '/usr/lib/systemd/system/openvpn@.service' '/etc/systemd/system/multi-user.target.wants/openvpn@server.service'
```

Теперь внимательно проверим корректность всех настроек на сервере. Сначала посмотрим информацию о сетевых интерфейсах:

```
[root@centos-server openvpn]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.25  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::215:5dff:fe01:f06  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:01:0f:06  txqueuelen 1000  (Ethernet)
        RX packets 264  bytes 38975 (38.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 204  bytes 85964 (83.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.1  netmask 255.255.255.0  broadcast 192.168.10.255
        inet6 fe80::215:5dff:fe01:f0d  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:01:0f:0d  txqueuelen 1000  (Ethernet)
        RX packets 26  bytes 1479 (1.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 37  bytes 4955 (4.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.255.255.255  destination 10.0.0.2
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Обращаем внимание на адреса туннеля vpn. Теперь проверяем статические маршруты:



Тут тоже все в порядке. Траффик из подсети филиала 192.168.20.0/24 будет маршрутизироваться в тоннель. С настройкой сервера закончили, идем теперь на клиент.

Настройка в CentOS 7 клиента openvpn

На centos-client отключаем SELinux, создаем в директории /etc/openvpn файл конфигурации client.conf:

```
# mcedit /etc/openvpn/client.conf
```

```
dev tun
proto udp
remote 192.168.1.25 13555
client
resolv-retry infinite
ca /etc/openvpn/ca.crt
cert /etc/openvpn/client.crt
key /etc/openvpn/client.key
route 192.168.10.0 255.255.255.0
```

```
persist-key
persist-tun
comp-lzo
verb 3
status /var/log/openvpn/openvpn-status.log 1
status-version 3
log-append /var/log/openvpn/openvpn-client.log
```

Не забываем скопировать в /etc/openvpn сохраненные ранее ключи ca.crt, client.crt, client.key.

Обращаю внимание на параметр **route** в данном конфиге. Его можно здесь не указывать, сделав конфиг более унифицированным для множества клиентов. Вместо этого данную настройку можно передавать с сервера openvpn, указав в файле настроек клиента параметр **push route** следующим образом:

```
push "route 192.168.10.0 255.255.255.0"
```

Создаем каталог для логов:

```
# mkdir /var/log/openvpn
```

Запускаем openvpn client:

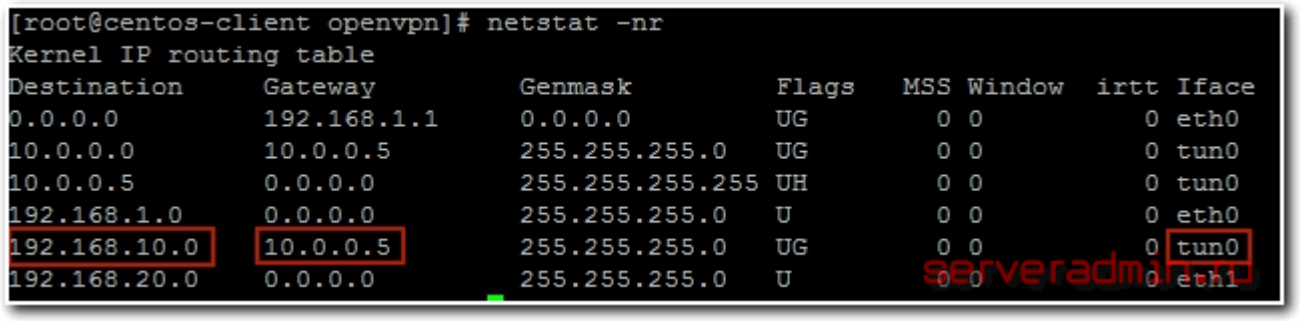
```
# systemctl start openvpn@client
```

Добавляем в автозагрузку:

```
# systemctl enable openvpn@client
```

Теперь смотрим картину настроек на клиенте:

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.26  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::215:5dff:fe01:f0b  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:01:0f:0b  txqueuelen 1000  (Ethernet)
        RX packets 4476  bytes 627556 (612.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3194  bytes 1353776 (1.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.20.1  netmask 255.255.255.0  broadcast 192.168.20.255
        inet6 fe80::215:5dff:fe01:f0e  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:01:0f:0e  txqueuelen 1000  (Ethernet)
        RX packets 311  bytes 27118 (26.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 259  bytes 24937 (24.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.0.0.6  netmask 255.255.255.255  destination 10.0.0.5
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100  (UNSPEC)
        RX packets 2  bytes 168 (168.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2  bytes 168 (168.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



Все в порядке, подключение к vpn серверу есть, маршруты прописаны верно.

В принципе, этого уже достаточно, чтобы трафик забегал в обе стороны из одной подсети в другую. Но для этого у вас должны быть соответствующим образом настроены сами шлюзы, фаерволы на них, маскарадинг, в том числе на интерфейсе **tun**. Я приведу вам свои настройки iptables с обоих серверов, чтобы вам было проще сравнить конфиги и диагностировать возможные проблемы.

Перед этим напомню важный момент, если вдруг вы с нуля настраиваете шлюз и не проверили его в реальной работе. Для роутинга трафика между сетевыми интерфейсами, необходимо добавить строку:

```
net.ipv4.ip_forward = 1
```

в файл /etc/sysctl.conf и применить настройку:

```
# sysctl -p
```

Если у вас этого не сделано, то трафик между интерфейсами ходить не будет. Ниже мои конфиги iptables.

Centos-server:

```
#!/bin/bash
#
# Объявление переменных
export IPT="iptables"

# Интерфейс который смотрит в интернет
export WAN=eth0
export WAN_IP=192.168.1.25

# Локалка
export LAN=eth1
export LAN_IP_RANGE=192.168.10.0/24

# Очистка всех цепочек iptables
$IPT -F
$IPT -F -t nat
$IPT -F -t mangle
$IPT -X
$IPT -t nat -X
$IPT -t mangle -X

# Установим политики по умолчанию для трафика, не соответствующего ни одному из правил
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP

# разрешаем трафик для loopback и локалки
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT
$IPT -A INPUT -i $LAN -j ACCEPT
$IPT -A OUTPUT -o $LAN -j ACCEPT
```

```
# разрешаем пинги
$IPT -A INPUT -p icmp — icmp-type echo-reply -j ACCEPT
$IPT -A INPUT -p icmp — icmp-type destination-unreachable -j ACCEPT
$IPT -A INPUT -p icmp — icmp-type time-exceeded -j ACCEPT
$IPT -A INPUT -p icmp — icmp-type echo-request -j ACCEPT

# Разрешаем исходящие соединения самого сервера
$IPT -A OUTPUT -o $WAN -j ACCEPT

# Разрешаем OpenVPN
$IPT -A INPUT -i tun+ -j ACCEPT
$IPT -A OUTPUT -o tun+ -j ACCEPT
$IPT -A FORWARD -i tun+ -j ACCEPT
# Разрешаем доступ из внутренней сети в vpn
$IPT -A FORWARD -i $LAN -o tun+ -j ACCEPT

# Разрешаем доступ из внутренней сети наружу
$IPT -A FORWARD -i $LAN -o $WAN -j ACCEPT

# Маскарадинг
$IPT -t nat -A POSTROUTING -o tun0 -j MASQUERADE
$IPT -t nat -A POSTROUTING -o $WAN -s $LAN_IP_RANGE -j MASQUERADE

# Состояние ESTABLISHED говорит о том, что это не первый пакет в соединении.
# Пропускать все уже инициированные соединения, а также дочерние от них
$IPT -A INPUT -p all -m state — state ESTABLISHED,RELATED -j ACCEPT
# Пропускать новые, а так же уже инициированные и их дочерние соединения
$IPT -A OUTPUT -p all -m state — state ESTABLISHED,RELATED -j ACCEPT
# Разрешить форвардинг для уже инициированных и их дочерних соединений
$IPT -A FORWARD -p all -m state — state ESTABLISHED,RELATED -j ACCEPT

# Включаем фрагментацию пакетов. Необходимо из за разных значений MTU
$IPT -I FORWARD -p tcp — tcp-flags SYN,RST SYN -j TCPMSS — clamp-mss-to-pmtu
```

```
# Отбрасывать все пакеты, которые не могут быть идентифицированы
# и поэтому не могут иметь определенного статуса.
$IPT -A INPUT -m state — state INVALID -j DROP
$IPT -A FORWARD -m state — state INVALID -j DROP

# Приводит к связыванию системных ресурсов, так что реальный
# обмен данными становится не возможным.
$IPT -A INPUT -p tcp ! — syn -m state — state NEW -j DROP
$IPT -A OUTPUT -p tcp ! — syn -m state — state NEW -j DROP

# Открываем порт для ssh
$IPT -A INPUT -i $WAN -p tcp — dport 22 -j ACCEPT

# Открываем порт для openvpn
$IPT -A INPUT -i $WAN -p udp — dport 13555 -j ACCEPT

# Записываем правила
/sbin/iptables-save > /etc/sysconfig/iptables
```

В **centos-client** настройки абсолютно такие же, кроме переменных WAN, LAN и LAN_IP_RANGE.

Подробнее о настройке iptables читайте в соответствующем материале.

Если ваш openvpn клиент не подключается, то в первую очередь проверяйте настройки firewall — входящие подключения к серверу и исходящие клиента. Обратите внимание на номер порта и тип (TCP или UDP) если вы их меняли. У меня были затупы, когда никак не мог разобраться, почему нет соединения. Оказывалось, что я менял порт с UDP на TCP, но по привычке на фаерволе оставлял UDP.

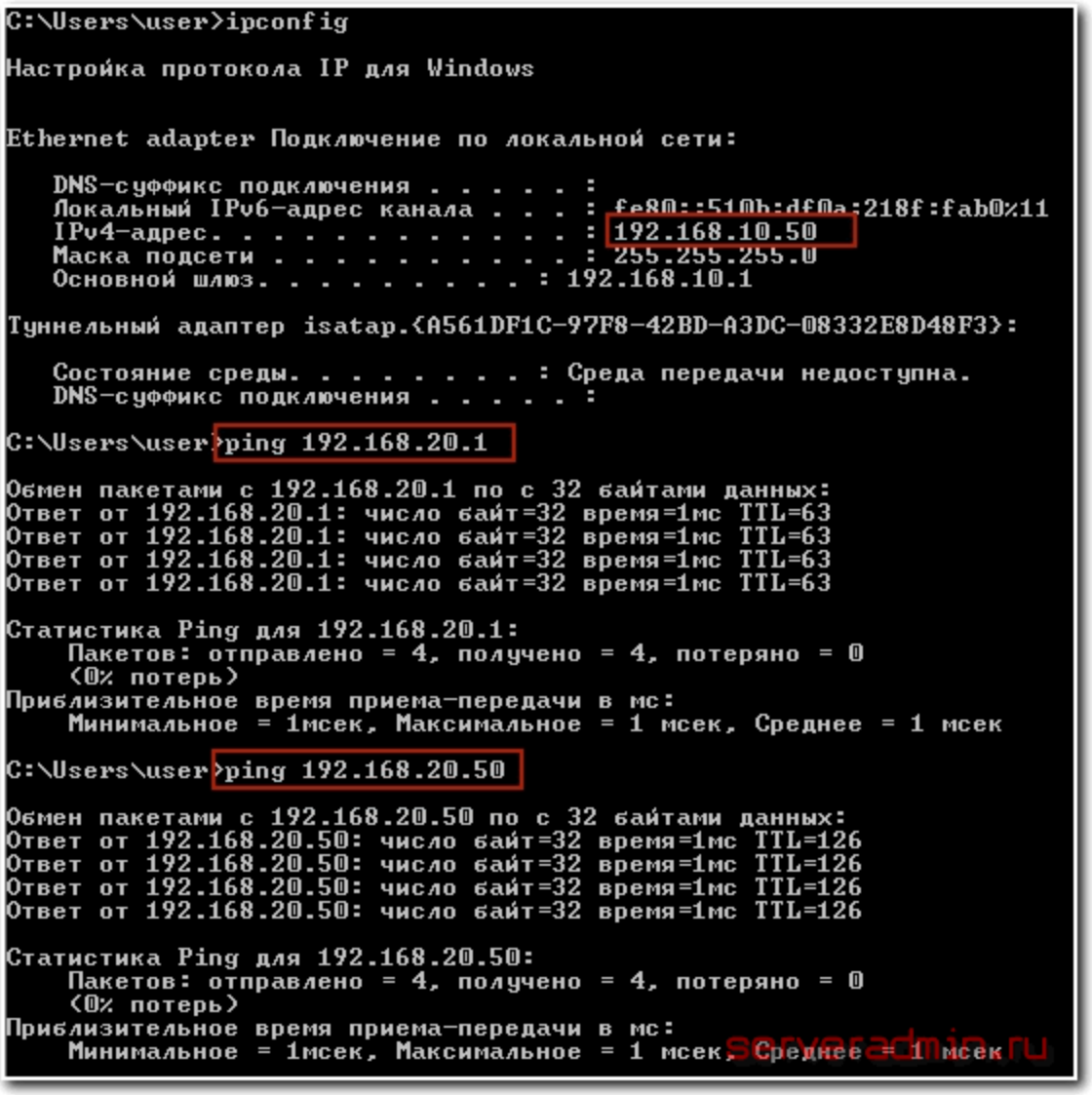
Теперь давайте проверим, как бегают пакеты в нашей vpn сети. Заходим на centos-client (192.168.20.1) и пингуем centos-server (192.168.10.1) и pc1 (192.168.10.50):

```
[root@centos-client etc]# ping -c 4 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.942 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.955 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=1.28 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=0.712 ms

--- 192.168.10.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.712/0.973/1.286/0.208 ms
[root@centos-client etc]# ping -c 4 192.168.10.50
PING 192.168.10.50 (192.168.10.50) 56(84) bytes of data.
64 bytes from 192.168.10.50: icmp_seq=1 ttl=127 time=1.24 ms
64 bytes from 192.168.10.50: icmp_seq=2 ttl=127 time=0.903 ms
64 bytes from 192.168.10.50: icmp_seq=3 ttl=127 time=0.930 ms
64 bytes from 192.168.10.50: icmp_seq=4 ttl=127 time=1.66 ms

--- 192.168.10.50 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3018ms
rtt min/avg/max/mdev = 0.903/1.186/1.668/0.311 ms
[root@centos-client etc]#
```

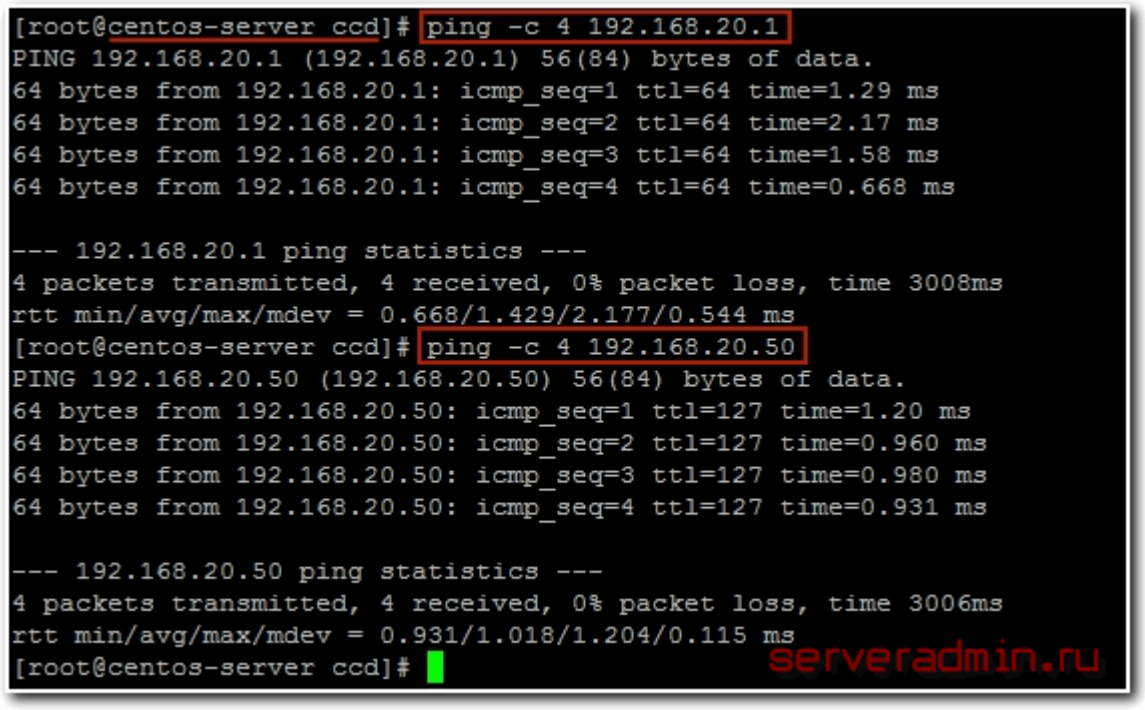

Заходим на pc2 (192.168.20.50) и пингуем centos-server и pc1:

```
C:\Users\user>ipconfig

Настройка протокола IP для Windows


Ethernet adapter Подключение по локальной сети:

   DNS-суффикс подключения . . . . . :
   Локальный IPv6-адрес канала . . . : fe80::b49f:650a:56d0:97c6%11
   IPv4-адрес. . . . . . . . . . . . : 192.168.20.50
   Маска подсети . . . . . . . . . . : 255.255.255.0
   Основной шлюз. . . . . . . . . . : 192.168.20.1

Туннельный адаптер isatap.{A561DF1C-97F8-42BD-A3DC-08332E8D48F3}:

   Состояние среды. . . . . . . . . : Среда передачи недоступна.
   DNS-суффикс подключения . . . . . :

C:\Users\user>ping 192.168.10.1

Обмен пакетами с 192.168.10.1 по с 32 байтами данных:
Ответ от 192.168.10.1: число байт=32 время=6мс TTL=63
Ответ от 192.168.10.1: число байт=32 время<1мс TTL=63
Ответ от 192.168.10.1: число байт=32 время<1мс TTL=63
Ответ от 192.168.10.1: число байт=32 время<1мс TTL=63

Статистика Ping для 192.168.10.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 6 мсек, Среднее = 1 мсек

C:\Users\user>ping 192.168.10.50

Обмен пакетами с 192.168.10.50 по с 32 байтами данных:
Ответ от 192.168.10.50: число байт=32 время=1мс TTL=126
Ответ от 192.168.10.50: число байт=32 время=6мс TTL=126
Ответ от 192.168.10.50: число байт=32 время=1мс TTL=126
Ответ от 192.168.10.50: число байт=32 время=4мс TTL=126

Статистика Ping для 192.168.10.50:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 1мсек, Максимальное = 6 мсек, Среднее = 3 мсек
```

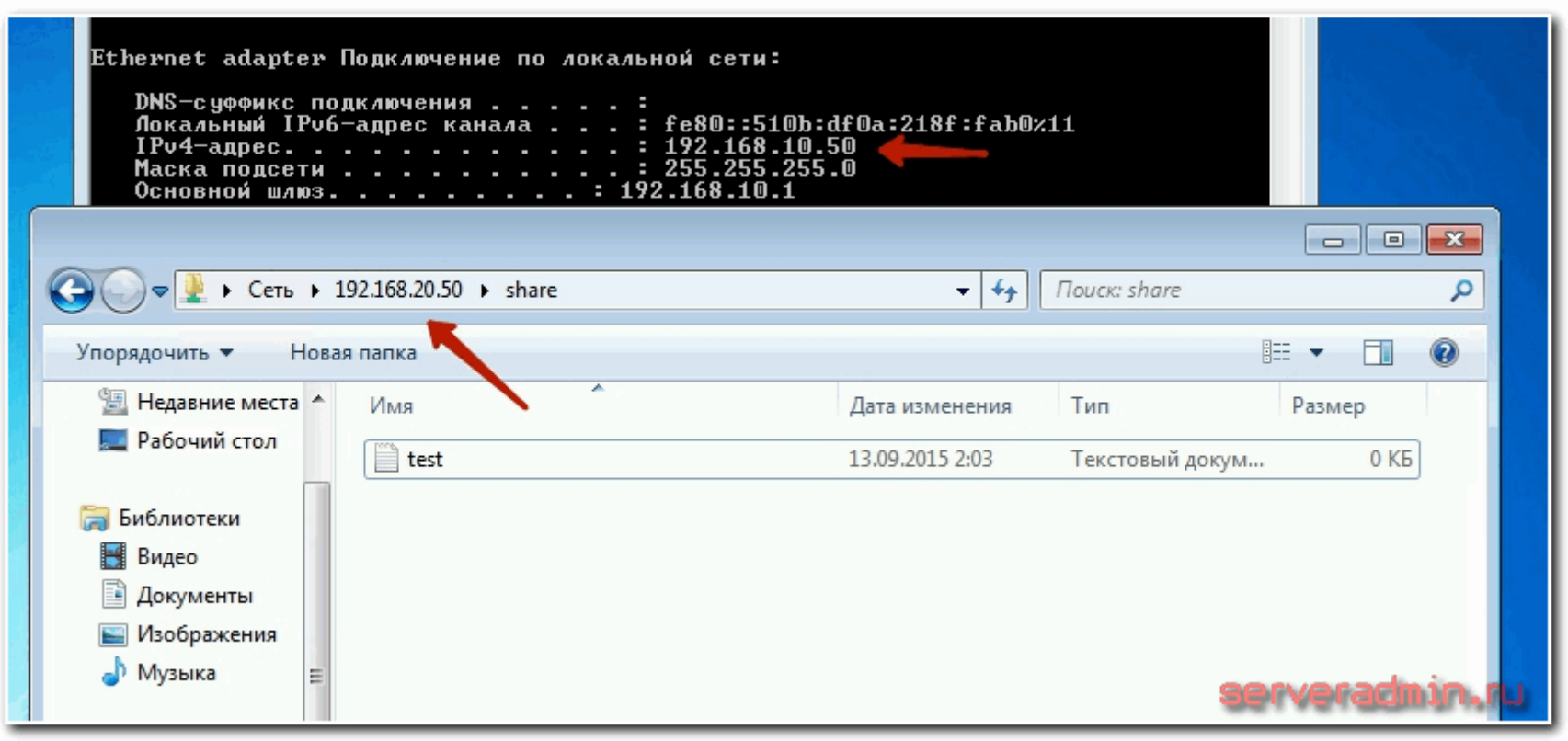
Теперь в обратную сторону. Заходим на pc1 (192.168.10.50) и пингуем centos-client (192.168.20.1) и pc2 (192.168.20.50):


C:\Users\user>ipconfig
Настройка протокола IP для Windows
Ethernet adapter Подключение по локальной сети:
DNS-суффикс подключения . . . . . :
Локальный IPv6-адрес канала . . . : fe80::510b:df0a:218f:fab0%11
IPv4-адрес. . . . . : 192.168.10.50
Маска подсети . . . . . : 255.255.255.0
Основной шлюз. . . . . : 192.168.10.1
Туннельный адаптер isatap.{A561DF1C-97F8-42BD-A3DC-08332E8D48F3}:
Состояние среды. . . . . : Среда передачи недоступна.
DNS-суффикс подключения . . . . . :
C:\Users\user>ping 192.168.20.1
Обмен пакетами с 192.168.20.1 по с 32 байтами данных:
Ответ от 192.168.20.1: число байт=32 время=1мс TTL=63
Ответ от 192.168.20.1: число байт=32 время=1мс TTL=63
Ответ от 192.168.20.1: число байт=32 время=1мс TTL=63
Ответ от 192.168.20.1: число байт=32 время=1мс TTL=63
Статистика Ping для 192.168.20.1:
Пакетов: отправлено = 4, получено = 4, потеряно = 0
(0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 1мсек, Максимальное = 1 мсек, Среднее = 1 мсек
C:\Users\user>ping 192.168.20.50
Обмен пакетами с 192.168.20.50 по с 32 байтами данных:
Ответ от 192.168.20.50: число байт=32 время=1мс TTL=126
Ответ от 192.168.20.50: число байт=32 время=1мс TTL=126
Ответ от 192.168.20.50: число байт=32 время=1мс TTL=126
Ответ от 192.168.20.50: число байт=32 время=1мс TTL=126
Статистика Ping для 192.168.20.50:
Пакетов: отправлено = 4, получено = 4, потеряно = 0
(0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 1мсек, Максимальное = 1 мсек, Среднее = 1 мсек
serveradmin.ru

И напоследок пропингуем с vpn сервера подсеть клиента:



В завершение темы объединения удаленных офисов, проверим доступ к общим сетевым ресурсам. Расшарим папку на PC2 и зайдем на нее с PC1:



Все получилось. Мы реально объединили удаленные офисы в единую логическую сеть с помощью openvpn сервера.

Настройка openvpn client в windows

Теперь рассмотрим вариант подключения к нашей логической сети удаленного сотрудника с рабочей станцией windows. Допустим, мы объединили наши офисы в единую сеть, доступ работает в обе стороны. Нам необходимо, чтобы удаленный пользователь смог подключиться либо к обоим сетям, либо выборочно только к основному офису, либо только к филиалу.

Первым делом идем на сервер и создаем для клиента сертификаты.

```
# cd /etc/openvpn/keys/easy-rsa-master/easyrsa3
# ./easyrsa gen-req user1 nopass
# ./easyrsa sign-req client user1
```

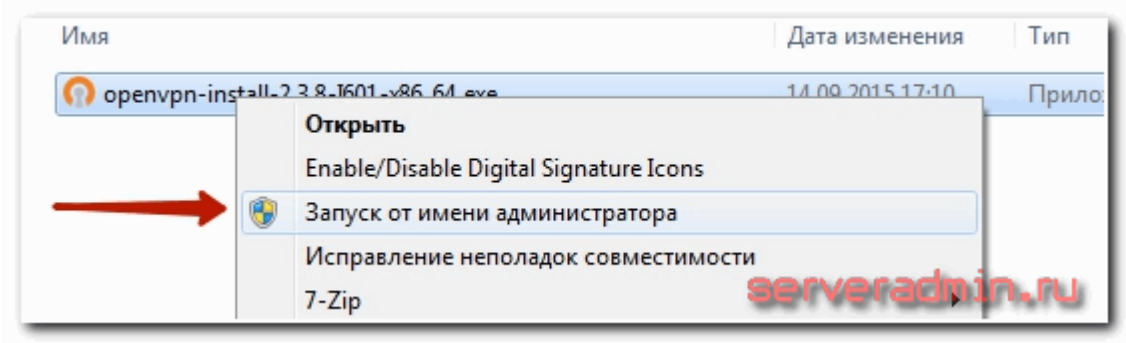
Процедура такая же, как и при создании первого сертификата клиента, который мы уже сгенерировали ранее. На выходе имеем два файла: **user1.key** и **user1.crt**. Добавляем сюда ключ **ca.crt** и передаем пользователю на компьютер.

Дальше создаем файл конфигурации для этого клиента:

```
# cd /etc/openvpn/ccd
# mcedit user1
push "route 192.168.10.0 255.255.255.0"
push "route 192.168.20.0 255.255.255.0"
```

Этими параметрами мы передаем клиенту маршруты к обоим сетям офисов. Если нужно подключать клиента только к какой-то одной сети, то оставляйте одну сеть, вторую удаляйте.

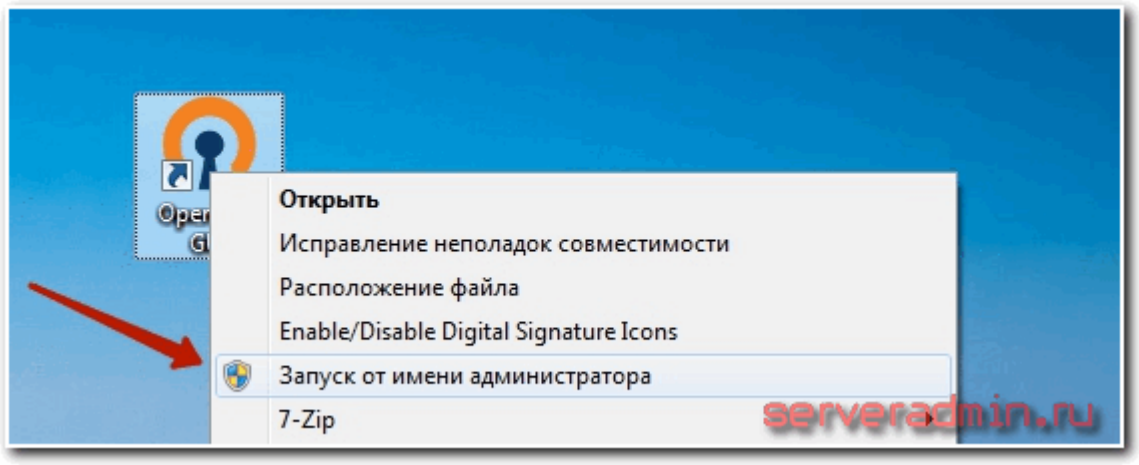
Теперь нужно скачать openvpn client под нашу версию windows. Ссылки для скачивания я давал в самом начале статьи. Дальше выполняем установку клиента. Обращаю внимание, что запускать установщик нужно обязательно с правами администратора:



Приступаем к настройке клиента в windows. Для этого нам понадобится файл конфигурации. Можно взять уже готовый config, который мы создавали ранее и немного изменить его. Файл конфигурации openvpn клиента должен выглядеть вот так:

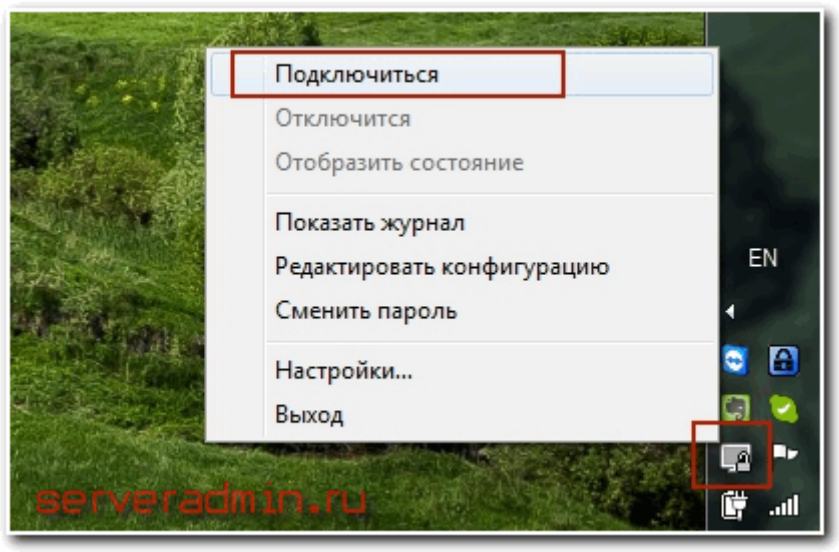
```
dev tun
proto udp
remote 192.168.1.25
port 13555
client
resolv-retry infinite
ca ca.crt
cert user1.crt
key user1.key
persist-key
persist-tun
comp-lzo
```

У нас нет задачи видеть сеть за клиентом, да и он скорее всего не будет являться шлюзом, поэтому параметра **route** в конфиге нет. Плюс убраны пути для логов — openvpn будет создавать их в папке по-умолчанию. Сохраняем конфигурацию под именем openvpn.ovpn в папку C:\Program Files\OpenVPN\config, туда же копируем файлы сертификатов и запускаем OpenVPN GUI от имени **администратора**!



Это важно, без прав администратора openvpn client для правильной маршрутизации не сможет прописать необходимые маршруты.

Ищем серый мониторчик в трее, нажимаем правой кнопкой мыши на него и выбираем «Подключиться».



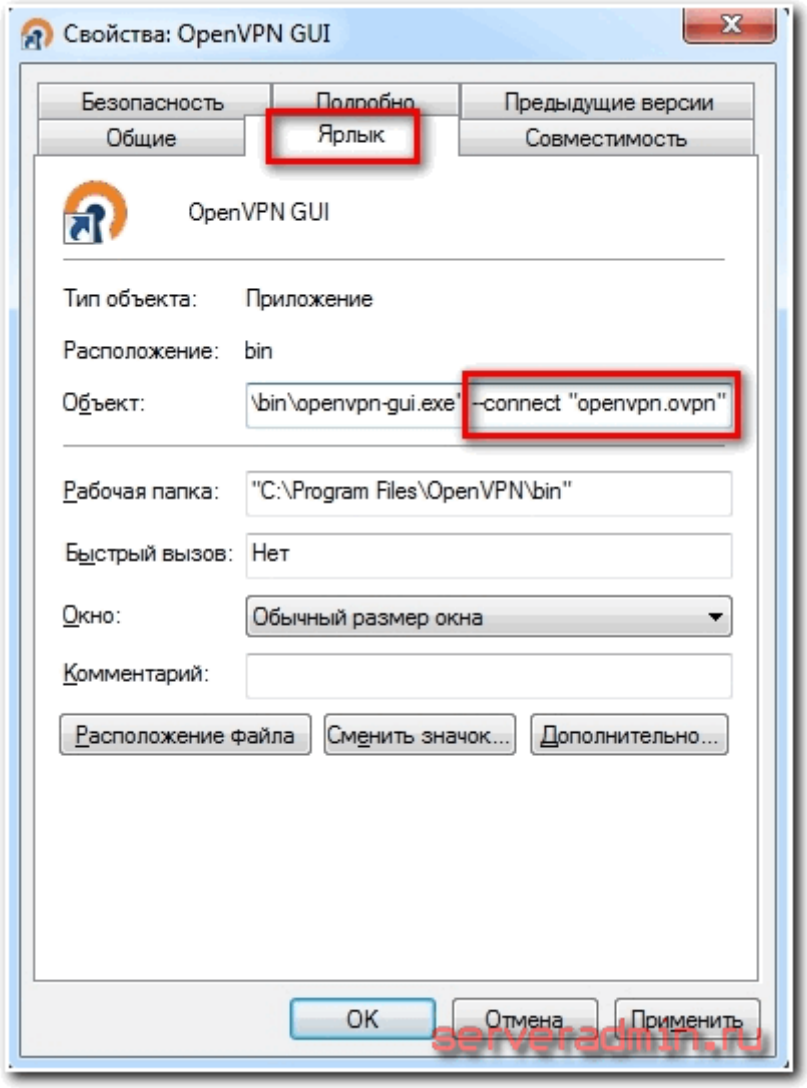
Во время подключения мониторчик будут гореть желтым цветом. Как только подключение будет установлено, цвет сменится на зеленый. Это означает, что openvpn клиент успешно создал туннель и можно начинать работать по vpn.

Для того, чтобы openvpn client автоматически подключался при запуске, нужно создать для него ярлык с параметрами:

```
--connect "openvpn.ovpn"
```

Для этого выбираем ярлык на рабочем столе OpenVPN GUI, выбираем его и нажимаем правой кнопкой мыши, открываем «Свойства». На вкладке «Ярлык» в поле «Объект» в самом конце дописываем указанные параметры. Вся строка должна выглядеть следующим образом:

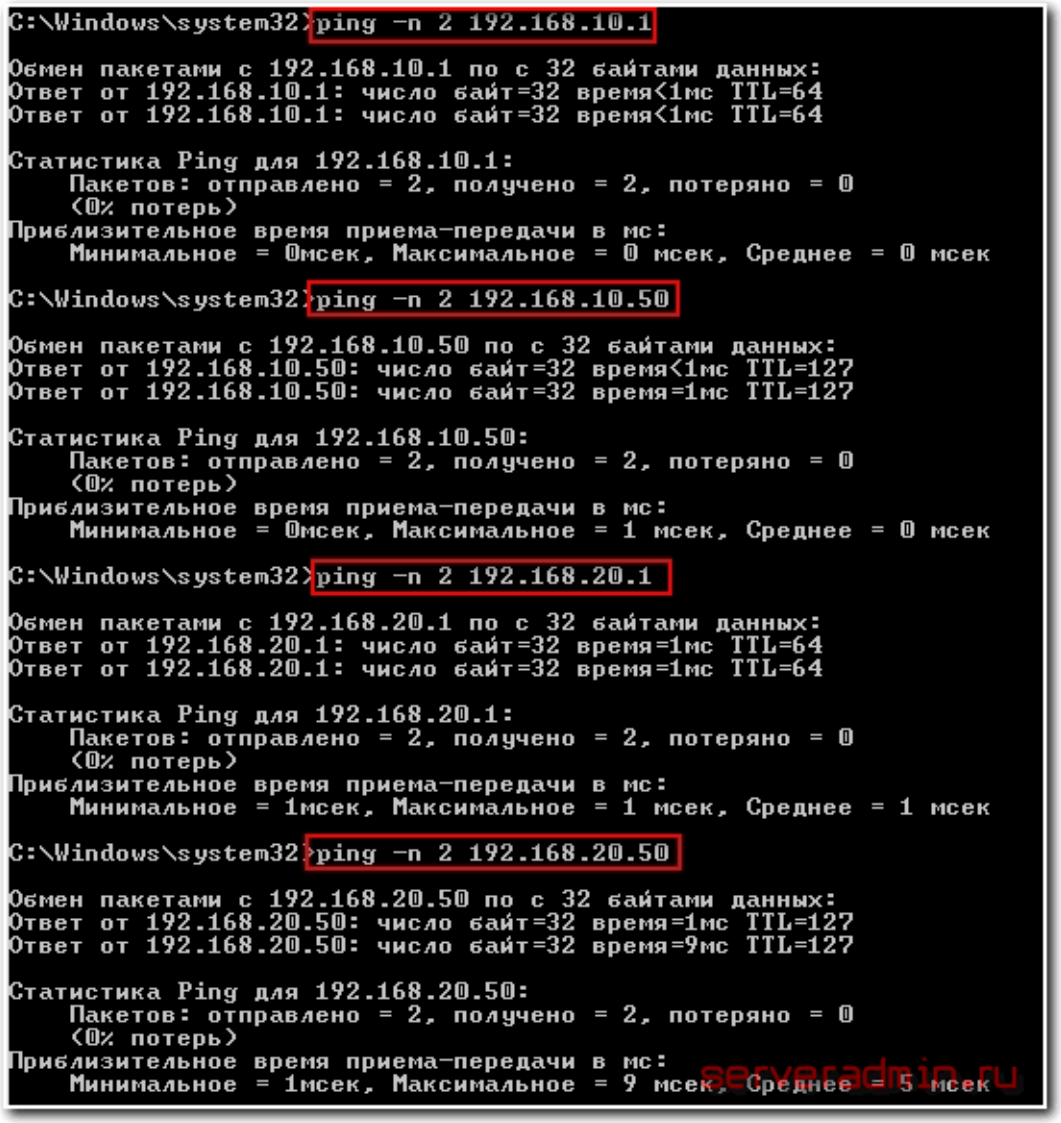
```
"C:\Program Files\OpenVPN\bin\openvpn-gui.exe" --connect "openvpn.ovpn"
```


Свойства: OpenVPN GUI
Безопасность
Подробно
Предыдущие версии
Общие
Ярлык
Совместимость
OpenVPN GUI
Тип объекта: Приложение
Расположение: bin
Объект: \bin\openvpn-gui.exe" --connect "openvpn.ovpn"
Рабочая папка: "C:\Program Files\OpenVPN\bin"
Быстрый вызов: Нет
Окно: Обычный размер окна
Комментарий:
Расположение файла
Сменить значок...
Дополнительно...
OK
Отмена
Применить
serveradmin.ru

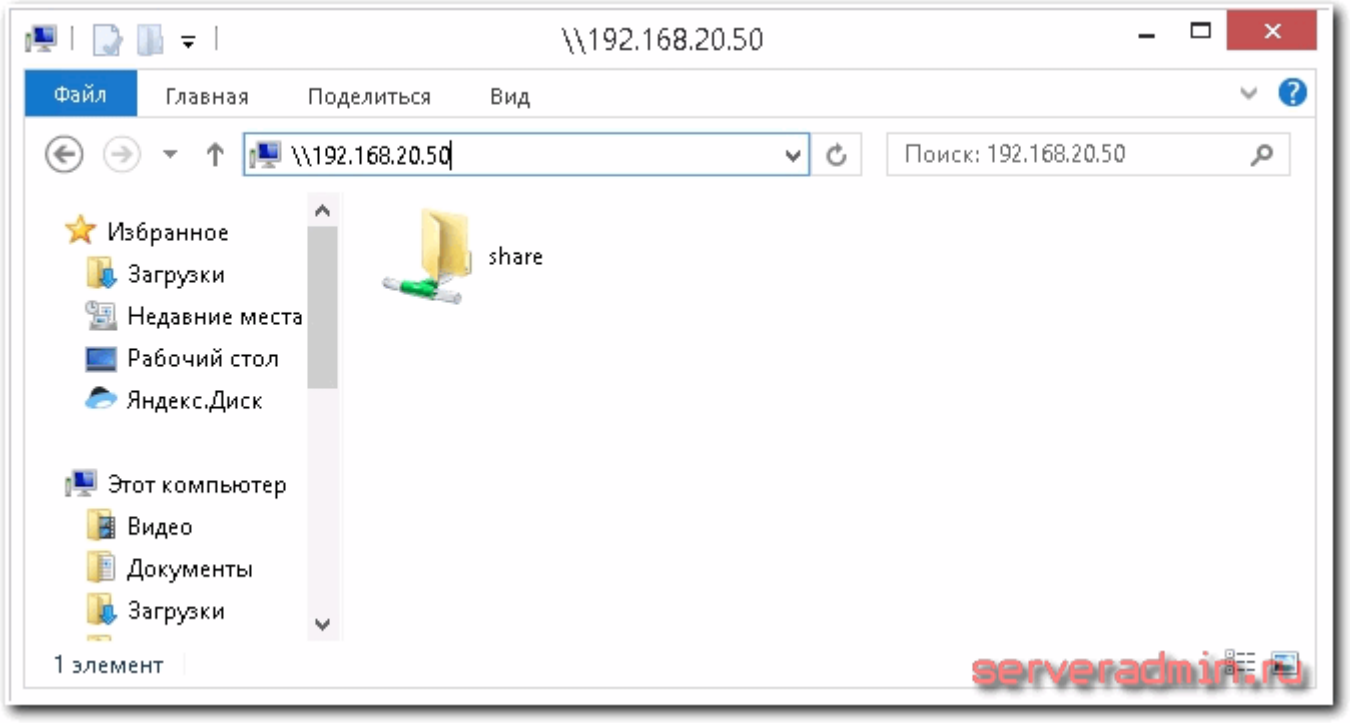
Теперь при запуске ярлыка openvpn будет автоматически подключаться к серверу и устанавливать vpn соединение.

Мы подключились к корпоративной vpn сети, объединяющую 2 офиса. Давайте попробуем получить доступ к компьютерам внутри этой сети. Пингуем все машины из нашей схемы:

192.168.10.1 192.168.10.50 192.168.20.1 192.168.20.50


C:\Windows\system32>ping -n 2 192.168.10.1
Обмен пакетами с 192.168.10.1 по с 32 байтами данных:
Ответ от 192.168.10.1: число байт=32 время<1мс TTL=64
Ответ от 192.168.10.1: число байт=32 время<1мс TTL=64
Статистика Ping для 192.168.10.1:
Пакетов: отправлено = 2, получено = 2, потеряно = 0
(0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
C:\Windows\system32>ping -n 2 192.168.10.50
Обмен пакетами с 192.168.10.50 по с 32 байтами данных:
Ответ от 192.168.10.50: число байт=32 время<1мс TTL=127
Ответ от 192.168.10.50: число байт=32 время=1мс TTL=127
Статистика Ping для 192.168.10.50:
Пакетов: отправлено = 2, получено = 2, потеряно = 0
(0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 0мсек, Максимальное = 1 мсек, Среднее = 0 мсек
C:\Windows\system32>ping -n 2 192.168.20.1
Обмен пакетами с 192.168.20.1 по с 32 байтами данных:
Ответ от 192.168.20.1: число байт=32 время=1мс TTL=64
Ответ от 192.168.20.1: число байт=32 время=1мс TTL=64
Статистика Ping для 192.168.20.1:
Пакетов: отправлено = 2, получено = 2, потеряно = 0
(0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 1мсек, Максимальное = 1 мсек, Среднее = 1 мсек
C:\Windows\system32>ping -n 2 192.168.20.50
Обмен пакетами с 192.168.20.50 по с 32 байтами данных:
Ответ от 192.168.20.50: число байт=32 время=1мс TTL=127
Ответ от 192.168.20.50: число байт=32 время=9мс TTL=127
Статистика Ping для 192.168.20.50:
Пакетов: отправлено = 2, получено = 2, потеряно = 0
(0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 1мсек, Максимальное = 9 мсек, Среднее = 5 мсек
serveradmin.ru

Отлично, связь есть. Теперь попробуем зайти на сетевой ресурс, который мы создали ранее на компьютере сети филиала PC2:



Доступ есть, все в порядке.

Заключение

Теперь подведем итоги того, что мы сделали:

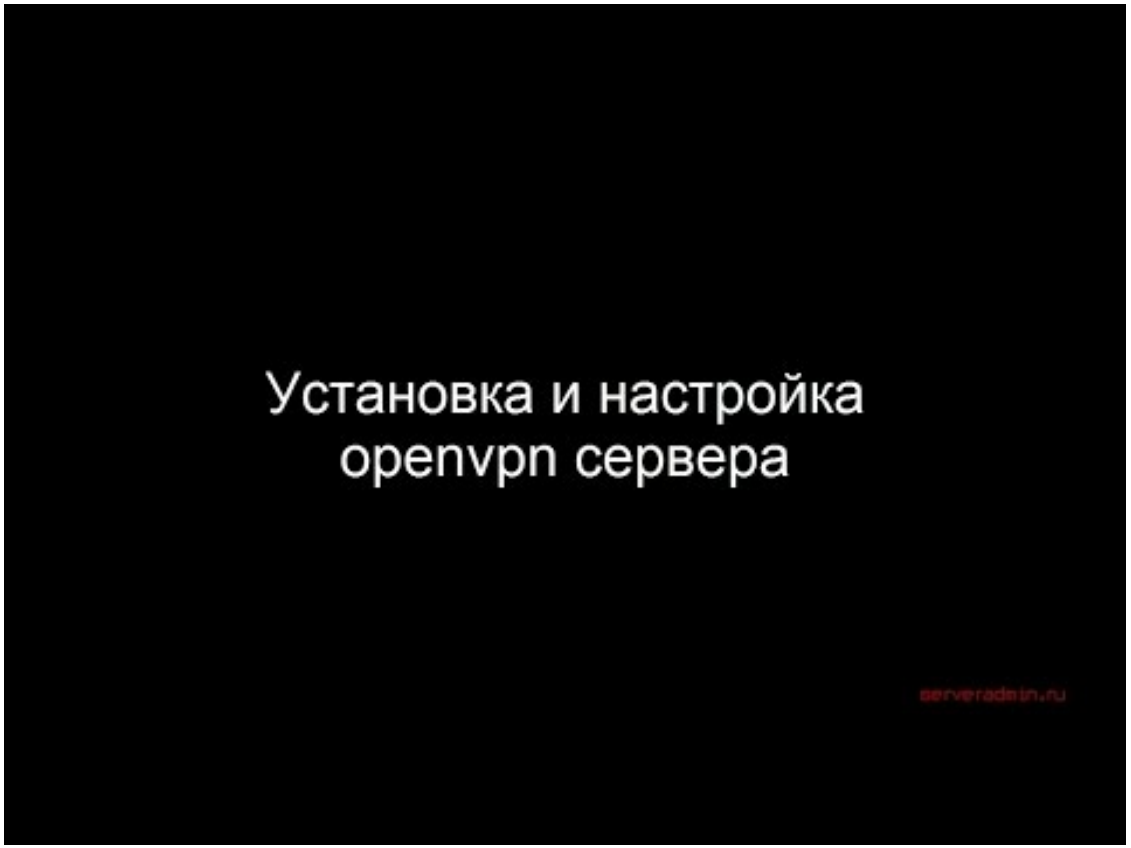
1. В первую очередь настроили сервер openvpn на CentOS 7. Для этого создали инфраструктуру для удостоверяющего центра CA, с помощью которого мы создаем сертификаты.
2. Затем с помощью этого центра сформировали сертификаты для сервера и клиента, в роли которых выступает головной офис компании и его

филиал.
3. Потом мы настроили openvpn сервер в офисе компании и подключили к нему в качестве клиента сервер филиала. Проверили это соединение, взаимную доступность узлов обоих сетей.
4. В завершении сформировали сертификат для удаленного сотрудника и настроили ему подключение openvpn клиента в windows. Проверили доступность всех узлов обоих сетей.

По этой схеме можно добавить любое количество филиалов или удаленных сотрудников к общей сети. Я с крупными сетями не работал, но лично подключал до 5-ти удаленных филиалов к головной сети офисов, плюс удаленных пользователей к ним. У пользователей маршрутами настраивал доступ к той или иной сети. Единственная проблема, которая тут может возникнуть, это если нумерация подключаемых сетей будет совпадать. Но тут уже ничего не поделаешь, придется ее где-то менять.

Напоминаю, что данная статья является частью единого цикла статьей про сервер Centos.

Видео



Watch this video on YouTube

Онлайн курс по Linux

Если вы хотите стать специалистом по отказоустойчивым виртуальным и кластерным средам, рекомендую познакомиться с онлайн-курсом **Администратор Linux. Виртуализация и кластеризация** в OTUS. Курс не для новичков, для поступления нужны хорошие знания по Linux. Обучение длится 5 месяцев, после чего успешные выпускники курса смогут пройти собеседования у партнеров. Что даст вам

этот курс:

- Умение строить отказоустойчивые кластера виртуализации для запуска современных сервисов, рассчитанных под высокую нагрузку.
- Будете разбираться в современных технологиях кластеризации, оркестрации и виртуализации.
- Научитесь выбирать технологии для построения отказоустойчивых систем под высокую нагрузку.
- Практические навыки внедрения виртуализации KVM, oVirt, Xen.
- Кластеризация сервисов на базе pacemaker,k8s, nomad и построение дисковых кластеров на базе ceph, glaster, linstore.

Проверьте себя на вступительном тесте и смотрите подробнее программу по .

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.