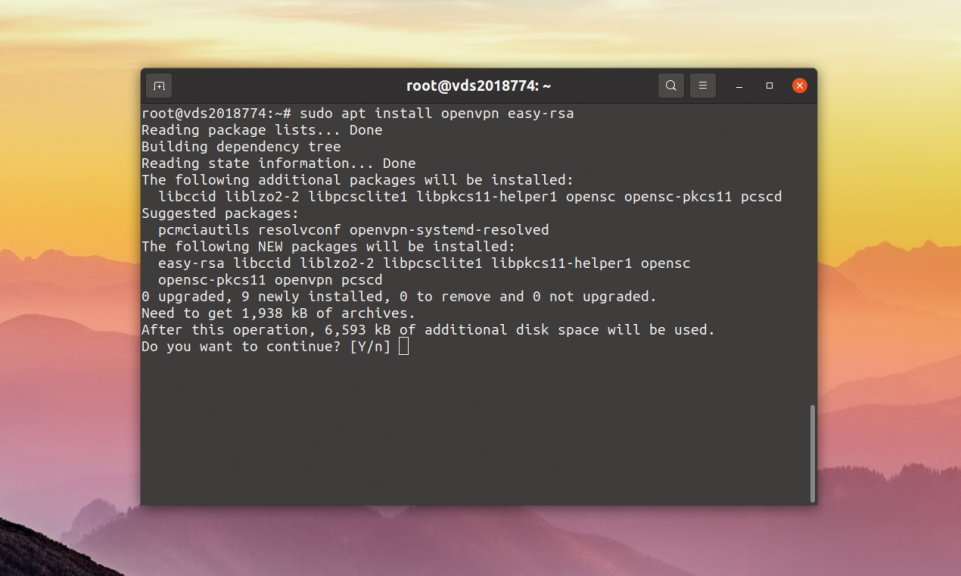
**Установка OpenVPN в Ubuntu**

**1. Установка пакетов**

На обоих машинах должен быть установлен OpenVPN, это довольно популярная программа, поэтому вы можете установить ее из официальных репозиториев. Также нам понадобится Easy-RSA для работы с секретными ключами. Для установки программ в Ubuntu используйте такую команду:

sudo apt install openvpn easy-rsa

[](https://losst.pro/wp-content/uploads/2017/04/openvpn.png)

Оба пакеты должны быть установлены как на сервере, так и на клиенте. Они понадобятся для настройки программы. Первый этап статьи установка и настройка openvpn завершен.

**2. Настройка центра сертификации**

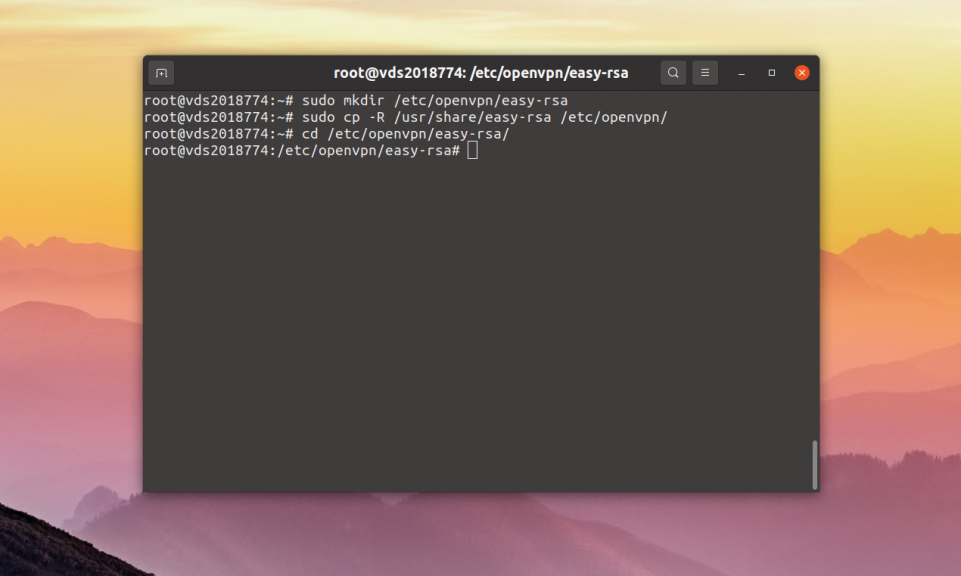
Первое что нужно сделать, это создать правильную инфраструктуру для генерации открытых ключей на сервере. Сервером мы считаем ту машину, к которой будут подключаться пользователи. Обратите внимание, что все секретные ключи должны находится в надежном месте. В OpenVPN открытый ключ называется сертификатом и имеет расширение **.crt**, а закрытый ключ так и называется ключом, его расширение - **.key**. Обслуживать всё это мы будем с помощью набора скриптов Easy-RSA.

Для того чтобы после обновления системы все ваши сертификаты и настройки не были стёрты, надо скопировать набор скриптов из каталога **/usr/share/easy-rsa** куда-нибудь, например, в **/etc/openvpn/**:

sudo mkdir /etc/openvpn/easy-rsa

Затем скопируем в эту папку все необходимые скрипты easy-rsa:

sudo cp -R /usr/share/easy-rsa /etc/openvpn/

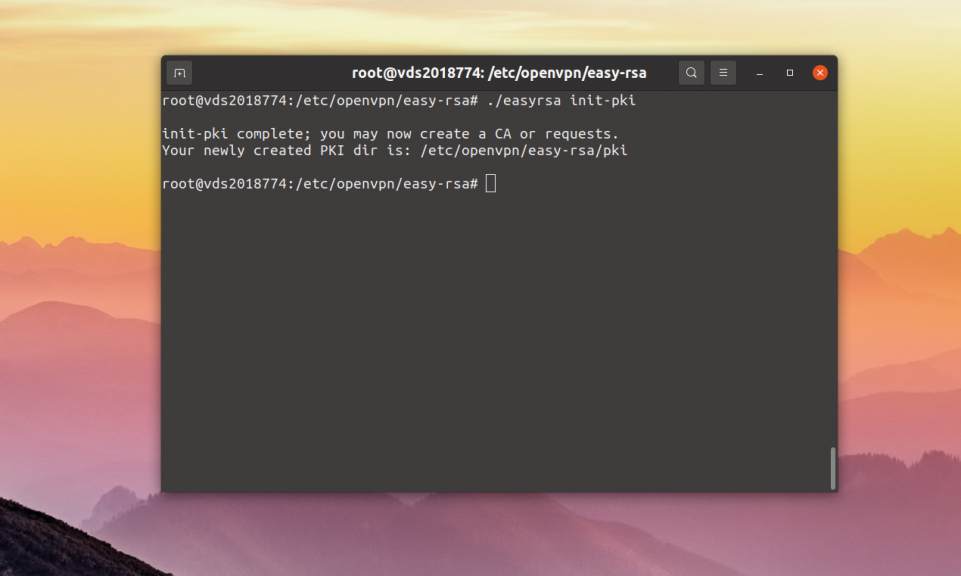
[](https://losst.pro/wp-content/uploads/2017/04/openvpn1.png)

Далее нам нужно создать центр сертификации в этой папке. Для этого сначала перейдите в неё:

cd /etc/openvpn/easy-rsa/

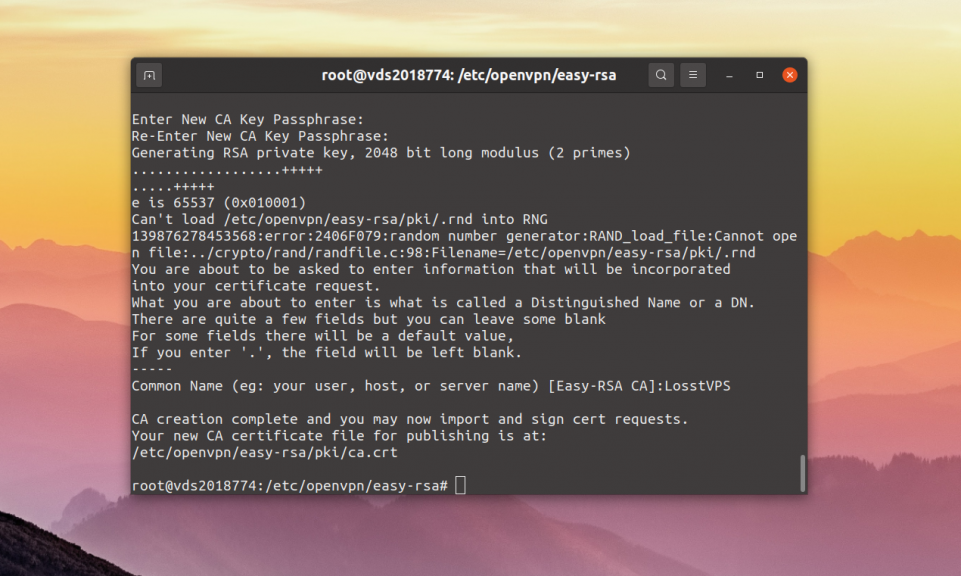
Эта команда создаст папку pki и и необходимые файлы для генерации сертификатов. Алгоритм шифрования можно настраивать, но с параметрами по умолчанию тоже всё будет работать:

sudo ./easyrsa init-pki

[](https://losst.pro/wp-content/uploads/2017/04/openvpn2.png)

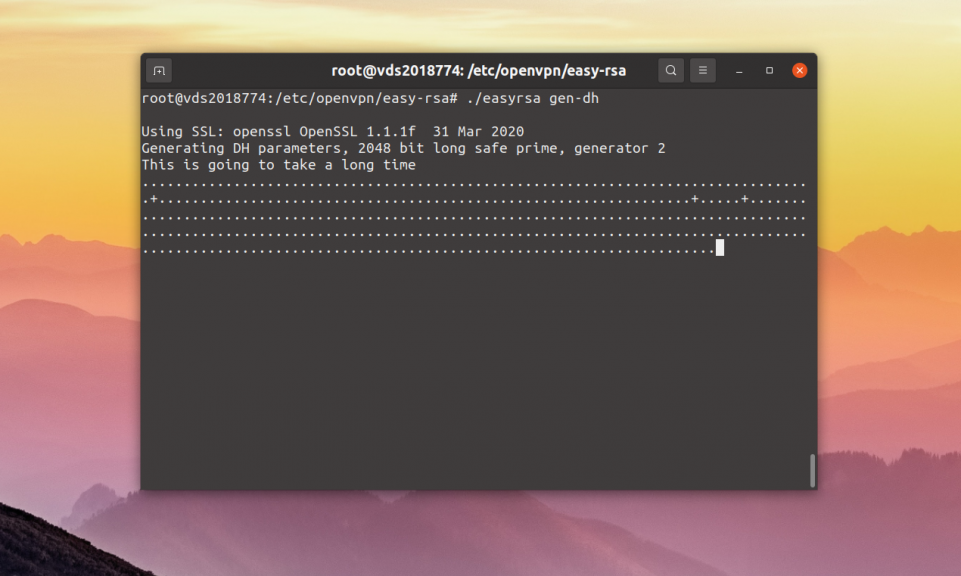
Следующая команда создаёт ключ центра сертификации, для него понадобится придумать пароль:

sudo ./easyrsa build-ca

[](https://losst.pro/wp-content/uploads/2017/04/openvpn3.png)

Далее надо создать ключи Диффи-Хафмана, которые используются при обмене ключами между клиентом и сервером. Для этого выполните:

sudo ./easyrsa gen-dh

[](https://losst.pro/wp-content/uploads/2017/04/openvpn4.png)

Команда создаст файл **/etc/openvpn/easy-rsa/pki/dh.pem**. Если вы хотите использовать TLS авторизацию, то вам ещё понадобится ключ Hash-based Message Authentication Code (HMAC). Он используется для предотвращения DoS атаки при использовании протокола UDP. Для его создания выполните:

sudo openvpn --genkey --secret /etc/easy-rsa/pki/ta.key

Для отзыва уже подписанных сертификатов нам понадобится сертификат отзыва. Для его создания выполните команду:

sudo ./easyrsa gen-crl

Будет создан файл **./pki/crl.pem**.

**3. Создание сертификатов сервера**

Для создания сертификатов, которые будут использоваться сервером надо выполнить команду:

sudo ./easyrsa build-server-full server nopass

Здесь **server** - это имя нашего сервера, а опция **nopass** отключает использование пароля. Теперь все полученные ключи надо скопировать в папку **/etc/openvpn**:

cp ./pki/ca.crt /etc/openvpn/ca.crt  
cp ./pki/dh.pem /etc/openvpn/dh.pem  
cp ./pki/crl.pem /etc/openvpn/crl.pem  
cp ./pki/ta.key /etc/openvpn/ta.key  
cp ./pki/issued/server.crt /etc/openvpn/server.crt  
cp ./pki/private/server.key /etc/openvpn/server.key

Все эти сертификаты надо будет использовать позже, при создании конфигурационного файла сервера.

**4. Конфигурационный файл сервера**

По умолчанию, конфигурационных файлов OpenVPN нет. Их нужно создать самостоятельно в зависимости от того, что планируется настраивать, сервер или клиент. Нужный файл конфигурации OpenVPN можно найти по адресу **/usr/share/doc/openvpn/examples/sample-config-files/**. Давайте создадим конфигурационный файл для сервера:

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

Здесь вам нужно настроить несколько параметров. Самые важные - **port** и **proto** - порт и протокол, по которым будет работать программа;

port 1194  
proto udp

Все созданные ключи нужно прописать в конфигурационном файле. Наши ключи хранятся по адресу /etc/openvpn, в самой папке с конфигурационным файлом, поэтому можно не прописывать к ним полный путь:

ca ca.crt  
cert server.crt  
key server.key  
dh dh.pem  
tls-auth ta.key 0

Настраиваем диапазон адресов для виртуальной сети, наш сервер будет доступен по первому из них - 10.8.0.1:

server 10.8.0.0 255.255.255.0

Следующая директива позволяет раздавать одним и тем же клиентам одни и те же IP адреса:

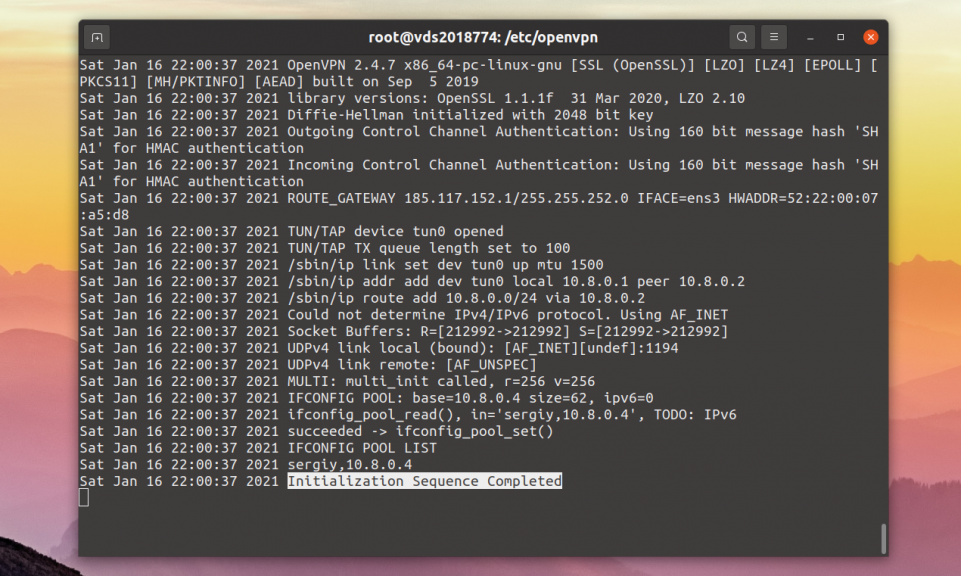
ifconfig-pool-persist /var/log/openvpn/ipp.txt

После завершения настройки сохраните изменения в файле, вы можете либо вставить всю эту конфигурацию себе или отредактировать файл с примером. Если убрать все комментарии получится такой конфигурационный файл:

port 1194  
proto udp  
dev tun  
ca ca.crt  
cert server.crt  
key server.key  
dh dh.pem  
server 10.8.0.0 255.255.255.0  
ifconfig-pool-persist /var/log/openvpn/ipp.txt  
push "redirect-gateway def1 bypass-dhcp"  
push "dhcp-option DNS 8.8.8.8"  
push "dhcp-option DNS 8.8.8.8"  
keepalive 10 120  
tls-auth ta.key 0  
cipher AES-256-CBC  
persist-key  
persist-tun  
status /var/log/openvpn/openvpn-status.log  
verb 3  
explicit-exit-notify 1

Настройка сервера OpenVPN завершена. Дальше необходимо запустить OpenVPN сервер. Это можно сделать прямо из командной строки, просто укажите адрес конфигурационного файла:

sudo openvpn /etc/openvpn/server.conf

[](https://losst.pro/wp-content/uploads/2017/04/openvpn6.png)

Если вы увидели сообщение Initialization Sequence Completed значит всё хорошо и можно двигаться дальше. Если возникли ошибки, исправьте их, скорее всего вы сделали что-то не то в конфигурационном файле. После этого остановите запущенный сервер и запустите его снова уже с помощью systemd:

sudo systemctl start openvpn@server

**5. Настройка форвардинга пакетов**

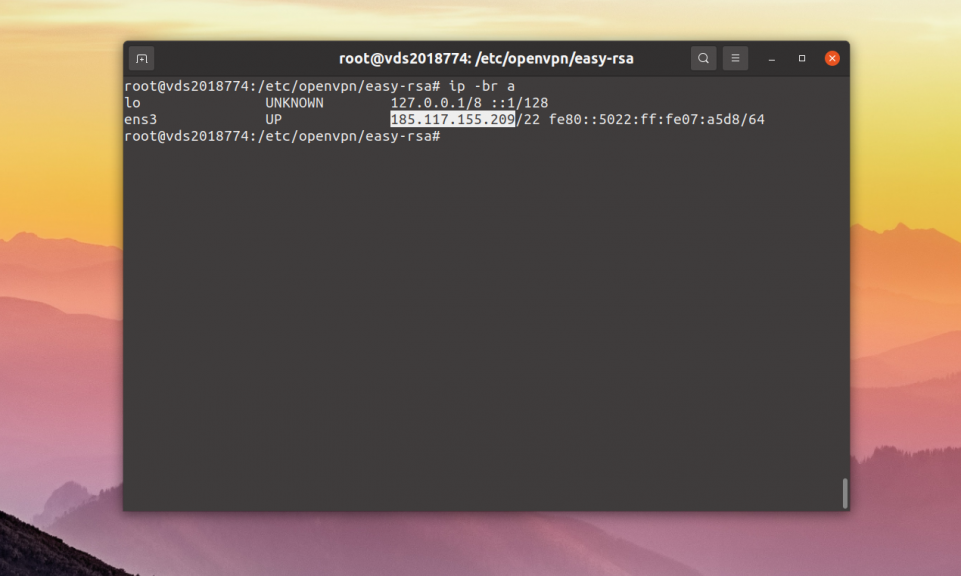
Для того чтобы OpenVPN сервер мог предоставить клиентам доступ к внешней сети сервера надо включить параметр ip\_forwarding. Для этого откройте файл для этого выполните такую команду:

sysctl -w net.ipv4.ip\_forward=1

**6. Настройка брандмауэра**

Теперь надо разрешить прохождение пакетов в брандмауэре. Давайте будем использовать iptables. Но сначала надо посмотреть сетевой интерфейс, который используется для доступа к внешней сети. Это можно сделать с помощью команды:

ip -br a

[](https://losst.pro/wp-content/uploads/2017/04/openvpn8.png)

В данном случае - это **ens3**. Поэтому правила iptables будут выглядеть вот так:

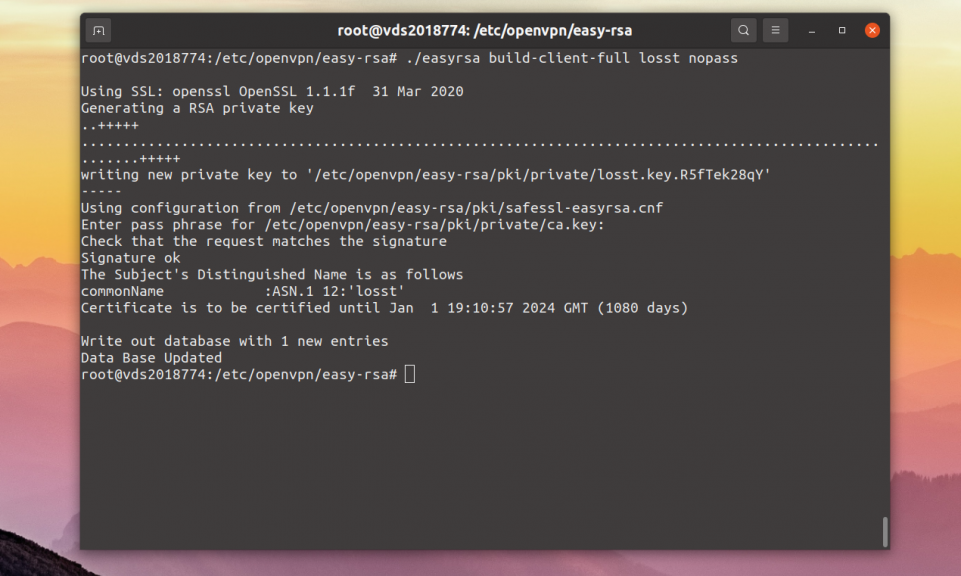
sudo iptables -I FORWARD -i tun0 -o ens3 -j ACCEPT  
sudo iptables -I FORWARD -i ens3 -o tun0 -j ACCEPT  
sudo iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE

Настройка OpenVPN Ubuntu на сервере завершена, теперь осталось подготовить всё необходимое для подключения клиентов.

**7. Создание сертификатов для клиента**

Чтобы не скидывать ключи на сервер, а потом обратно, лучше создавать ключи на сервере. Для этого в той же директории **/etc/openvpn/easy-rsa/** выполните такую команду:

sudo ./easyrsa build-client-full losst nopass

[](https://losst.pro/wp-content/uploads/2017/04/openvpn7.png)

Вместо **losst** используйте нужное вам имя пользователя, а опция nopass аналогично генерации ключей для сервера отключает использование пароля.

**8. Конфигурационный файл клиента**

Давайте создадим папку **/etc/openvpn/clients**, куда будем складывать все ключи и конфигурационные файлы клиентов:

sudo mkdir /etc/openvpn/clients

sudo mkdir /etc/openvpn/clients/losst

Затем перейдите в папку клиента:

cd /etc/openvpn/clients/losst

Затем туда надо скопировать **ca.crt**, **ta.key** и ключи клиента **losst.crt** и **losst.key:**

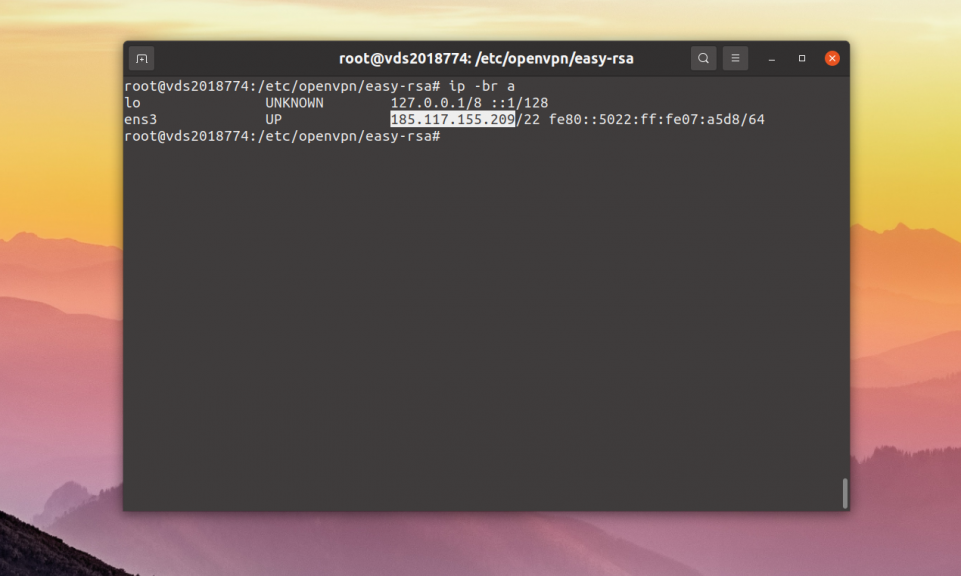
cp /etc/openvpn/easy-rsa/pki/ca.crt /etc/openvpn/clients/losst/  
cp /etc/openvpn/easy-rsa/pki/ta.key /etc/openvpn/clients/losst/  
cp /etc/openvpn/easy-rsa/pki/issued/losst.crt /etc/openvpn/clients/losst/  
cp /etc/openvpn/easy-rsa/pki/private/losst.key /etc/openvpn/clients/losst/

Теперь давайте создадим конфигурационный файл клиента, его тоже можно скопировать из директории с примерами:

sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf ./losst.conf

Перед тем, как редактировать конфигурационный файл надо узнать публичный IP вашего сервера. Для этого можно воспользоваться командой:

ip -br a

[](https://losst.pro/wp-content/uploads/2017/04/openvpn8.png)

Теперь можно открыть скопированный конфигурационный файл:

sudo vi ./losst.conf

Здесь обязательно надо заменить в этой строке **my-server-1** на IP адрес сервера, который вы узнали ранее. Например:

remote 185.117.155.209 1194

Всё остальное можно оставить по умолчанию, сертификаты должны лежать в той же папке, в которой будет находится конфигурационный файл. В результате, если убрать все комментарии получится вот так:

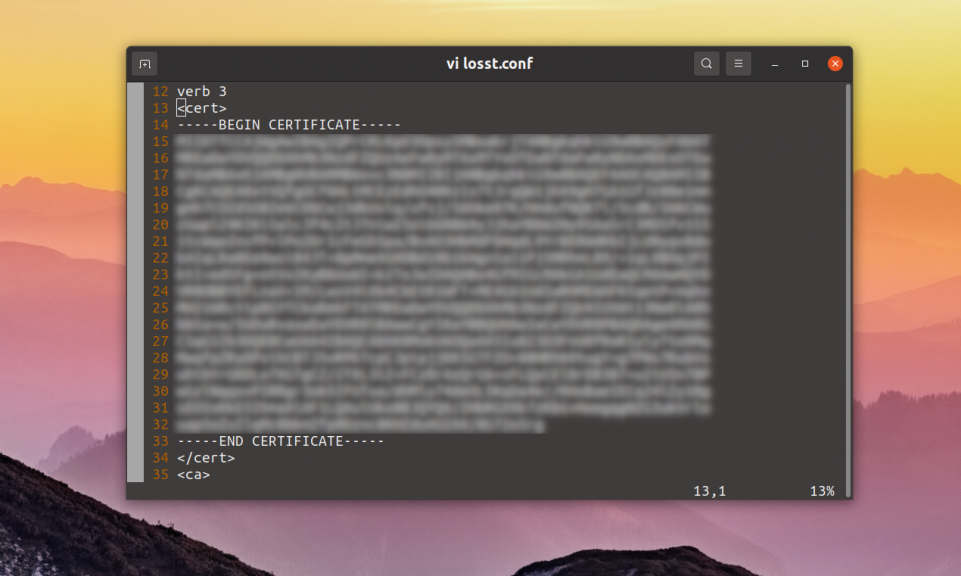
client  
dev tun  
proto udp  
remote 185.117.155.209 1194  
resolv-retry infinite  
nobind  
persist-key  
persist-tun  
ca ca.crt  
cert losst.crt  
key losst.key  
remote-cert-tls server  
tls-auth ta.key 1  
cipher AES-256-CBC  
verb 3

Обратите внимание, что конфигурационные файлы должны максимально совпадать, отсутствие определенных опций в одном из файлов может привести к ошибкам. Это не значит, что файлы будут идентичны, но основные параметры openvpn должны быть одинаковыми. Например, если вы включили опцию **comp-lzo** для сервера, то она должна быть включена и для клиента.

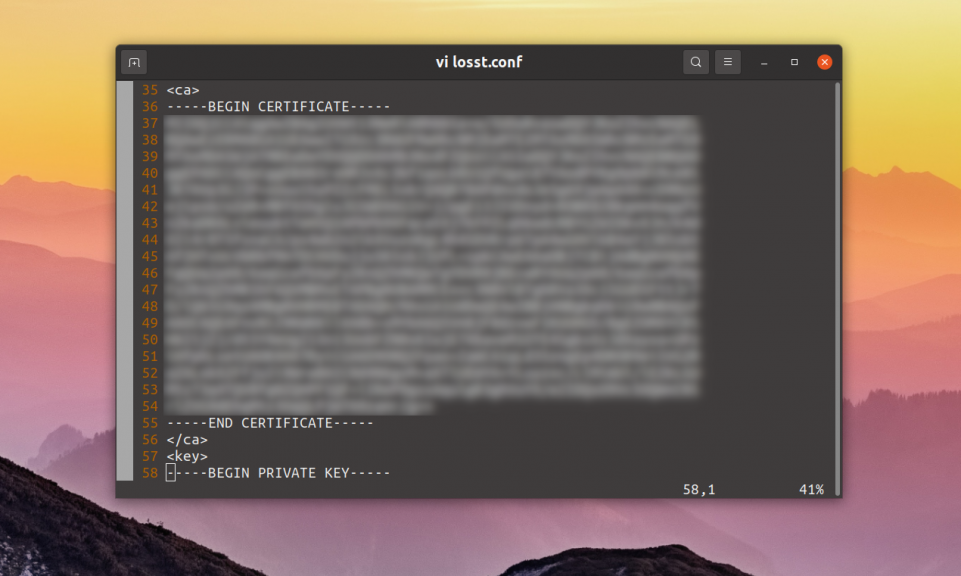
Далее осталось всё это упаковать в архив и отправить клиенту. Но можно поступить по другому. Ключи для опций **ca, cert, key** и **tls-auth** можно упаковать в сам файл конфигурации. Например, для опции **cert** используется такой синтаксис:

<cert>  
-----BEGIN CERTIFICATE-----  
текст\_сертификата  
-----END CERTIFICATE-----  
</cert>

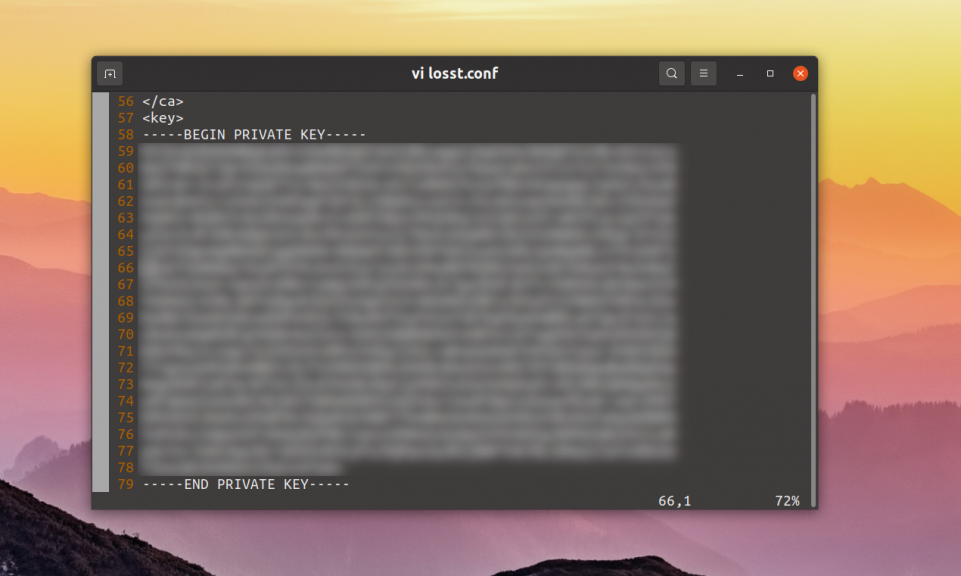
Текст ключа надо взять из файла **losst.crt** между **BEGIN CERTIFICATE** и **END CERTIFICATE**. Например:

[](https://losst.pro/wp-content/uploads/2017/04/openvpn9.png)

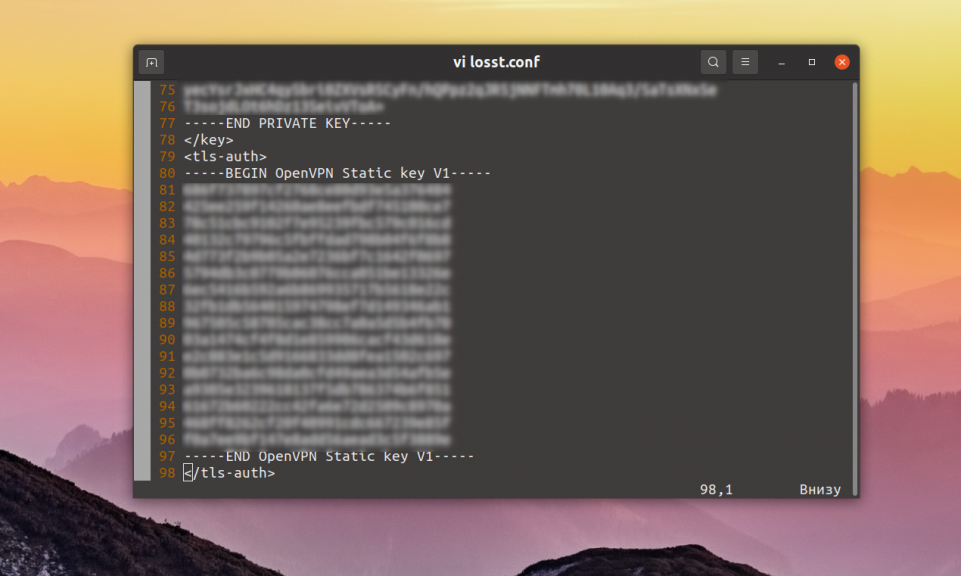
Для **ca.crt** всё будет выглядеть аналогично:

[](https://losst.pro/wp-content/uploads/2017/04/openvpn10.png)

Для приватного ключа **losst.key** надо использовать немного другой синтаксис:

[](https://losst.pro/wp-content/uploads/2017/04/openvpn11.png)

И для **tls-auth** тоже:

[](https://losst.pro/wp-content/uploads/2017/04/openvpn12.png)

После этого директивы, указывающие на файлы можно удалить, и передать клиенту только один этот файл.

**9. Проверка подключения**

Вы можете скачать конфигурационный файл клиента с сервера, например, с помощью scp:

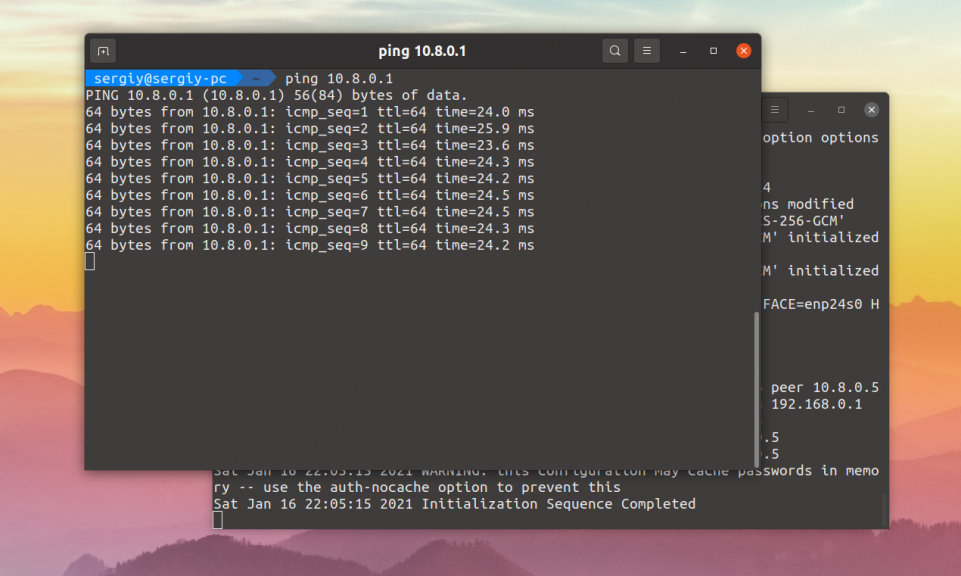
scp root@ip\_сервера:/etc/openvpn/clients/losst/losst.conf /etc/openvpn/

Вам осталось запустить OpenVPN на этой машине используя этот конфигурационный файл:

openvpn /etc/openvpn/losst.conf

Готово, теперь все работает, если вы выполните ifconfig, то увидите что был добавлен интерфейс tun0. Также вы можете попробовать выполнить ping адреса 10.8.0.1, именно этот адрес мы настроили для нашего сервера OpenVPN, пакеты ping будут нормально отправляться.

ping 10.8.0.1

[](https://losst.pro/wp-content/uploads/2017/04/openvpn13.png)

Если пакеты не идут, или еще что-то не работает, обратите внимание на вывод обоих программ, возможно, возникли какие-либо ошибки или предупреждения, также убедитесь, что брандмауэр сервера разрешает доступ извне по udp для порта 1194. Еще можно запустить сервер или клиент, настроив в конфигурационном файле уровень подробности на максимум **verb 9**. Очень часто это помогает понять почему что-то не работает. Но вы еще не можете направлять трафик через туннель.