HACETTEPE
University

Computer Science and Engineering Department

Group Name&Surname:	Mehmet Taha USTA & Burcu ÖZTAŞ
Identity Numbers:	21527472 & 21483435
Course:	BBM-465 Information Security Lab.
Experiment:	Assignment 5
Subject:	VPNs
Due Date:	10/1/2020 - 23:59
Advisor:	Dr. Ahmet Selman BOZKIR

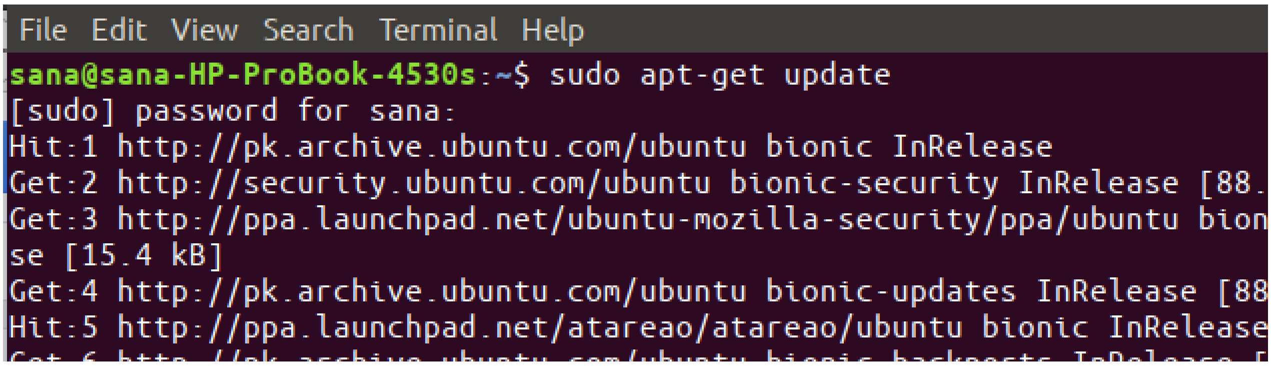
1. OpenSSH Server

Server and client must be installed to provide SSH Connection.

1.1 Update Repository Index

In order to install the latest available version of software from the Internet repositories, your local repository index needs to be in line with them. Run the following command as sudo in order to update your local repository index:

```
$ sudo apt-get update
```



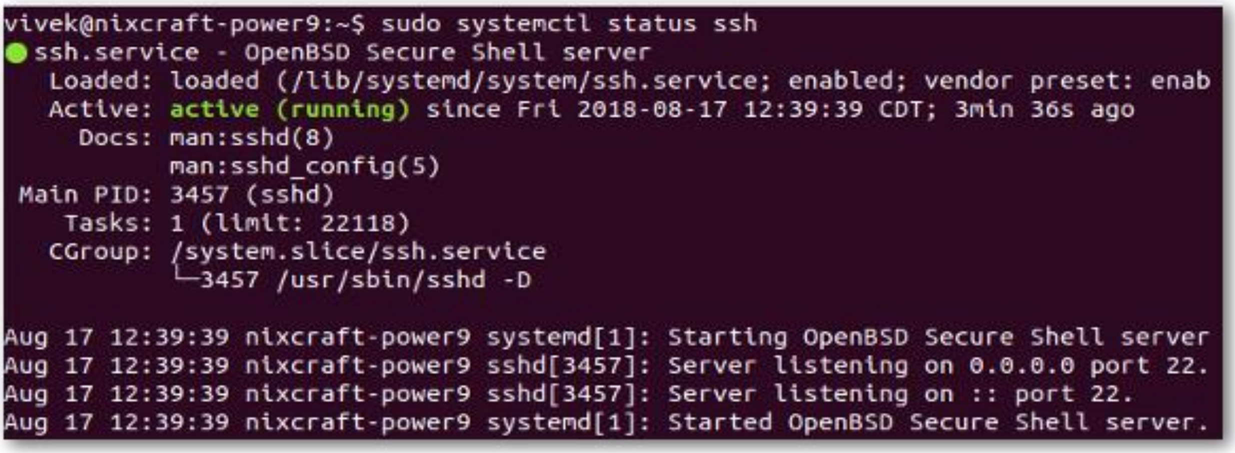
```
sana@sana-HP-ProBook-4530s:~$ sudo apt-get update
[sudo] password for sana:
Hit:1 http://pk.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.
Get:3 http://ppa.launchpad.net/ubuntu-mozilla-security/ppa/ubuntu bion
se [15.4 kB]
Get:4 http://pk.archive.ubuntu.com/ubuntu bionic-updates InRelease [88
Hit:5 http://ppa.launchpad.net/atareao/atareao/ubuntu bionic InRelease
Get:6 http://pk.archive.ubuntu.com/ubuntu bionic-backports InRelease [
```

1.2 Install OpenSSh-Server

```
$ sudo apt install openssh-server
```

1.3 Verify That SSH Service Running

```
$ sudo systemctl status ssh
```



```
vivek@nixcraft-power9:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
   Active: active (running) since Fri 2018-08-17 12:39:39 CDT; 3min 36s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 3457 (sshd)
    Tasks: 1 (limit: 22118)
   CGroup: /system.slice/ssh.service
           └─3457 /usr/sbin/sshd -D

Aug 17 12:39:39 nixcraft-power9 systemd[1]: Starting OpenBSD Secure Shell server
Aug 17 12:39:39 nixcraft-power9 sshd[3457]: Server listening on 0.0.0.0 port 22.
Aug 17 12:39:39 nixcraft-power9 sshd[3457]: Server listening on :: port 22.
Aug 17 12:39:39 nixcraft-power9 systemd[1]: Started OpenBSD Secure Shell server.
```

If not running enable the ssh server and start it as follows by typing the systemctl command:

```
$ sudo systemctl enable ssh

$ sudo systemctl start ssh
```

1.4 Password Reset

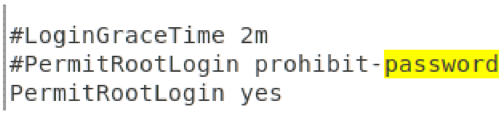
```
$ sudo passwd
```

1.5 Allow Root Authentication

```
$ sudo gedit /etc/ssh/sshd_config
```

add command to file:

```
PermitRootLogin yes
```



```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
PermitRootLogin yes
```

```
$ sudo service ssh restart
```

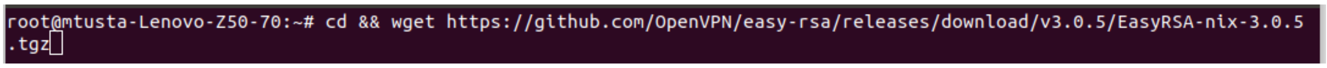
2. OpenVPN

2.1 Building CA with EasyRSA

Perform the following steps on your CA machine

First, download the EasyRSA from the project Github repository with the following wget command:

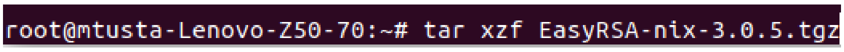
```
cd && wget https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.5/
EasyRSA-nix-3.0.5.tgz
```



```
root@mtusta-Lenovo-Z50-70:~# cd && wget https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.5/EasyRSA-nix-3.0.5
.tgz
```

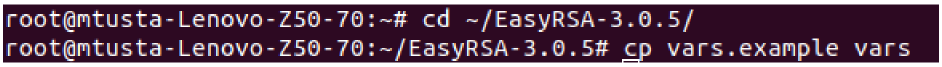
Extract the archive:

```
tar xzf EasyRSA-nix-3.0.5.tgz
```



```
root@mtusta-Lenovo-Z50-70:~# tar xzf EasyRSA-nix-3.0.5.tgz
```

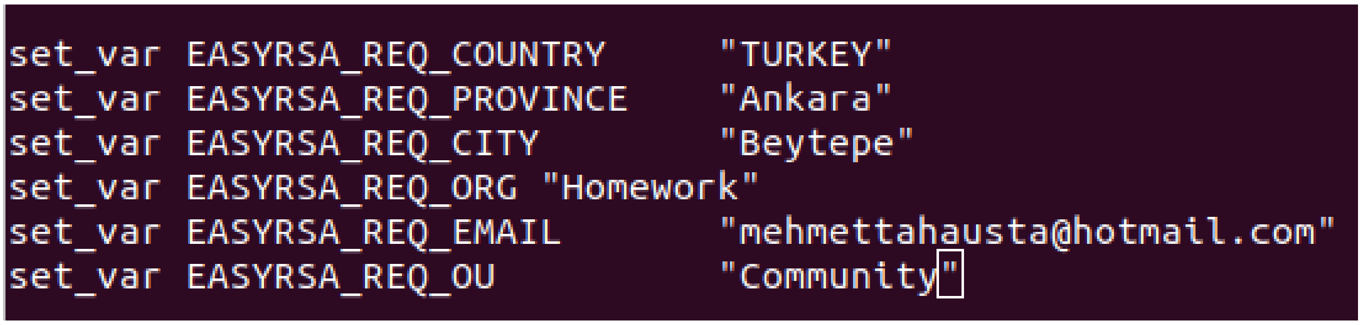
Switch to the EasyRSA directory and create a configuration file named vars by copying the vars.example file:



```
root@mtusta-Lenovo-Z50-70:~# cd ~/EasyRSA-3.0.5/
root@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5# cp vars.example vars
```

Open the file and uncomment and update the following entries to match your information.

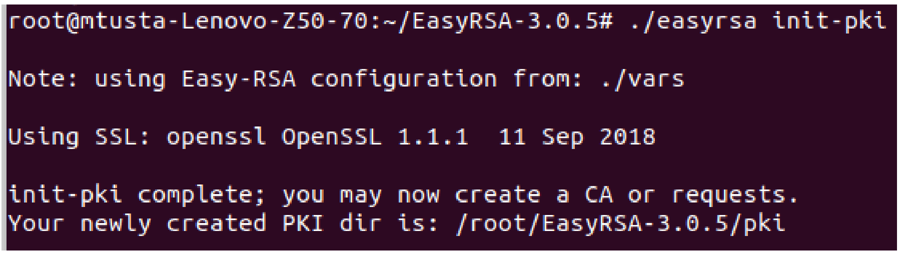
```
nano ~/EasyRSA-3.0.5/vars
```



```
set_var EASYRSA_REQ_COUNTRY     "TURKEY"
set_var EASYRSA_REQ_PROVINCE    "Ankara"
set_var EASYRSA_REQ_CITY        "Beytepe"
set_var EASYRSA_REQ_ORG "Homework"
set_var EASYRSA_REQ_EMAIL       "mehmettahausta@hotmail.com"
set_var EASYRSA_REQ_OU          "Community"
```

Before generating a CA keypair first we need to initialize a new PKI with:

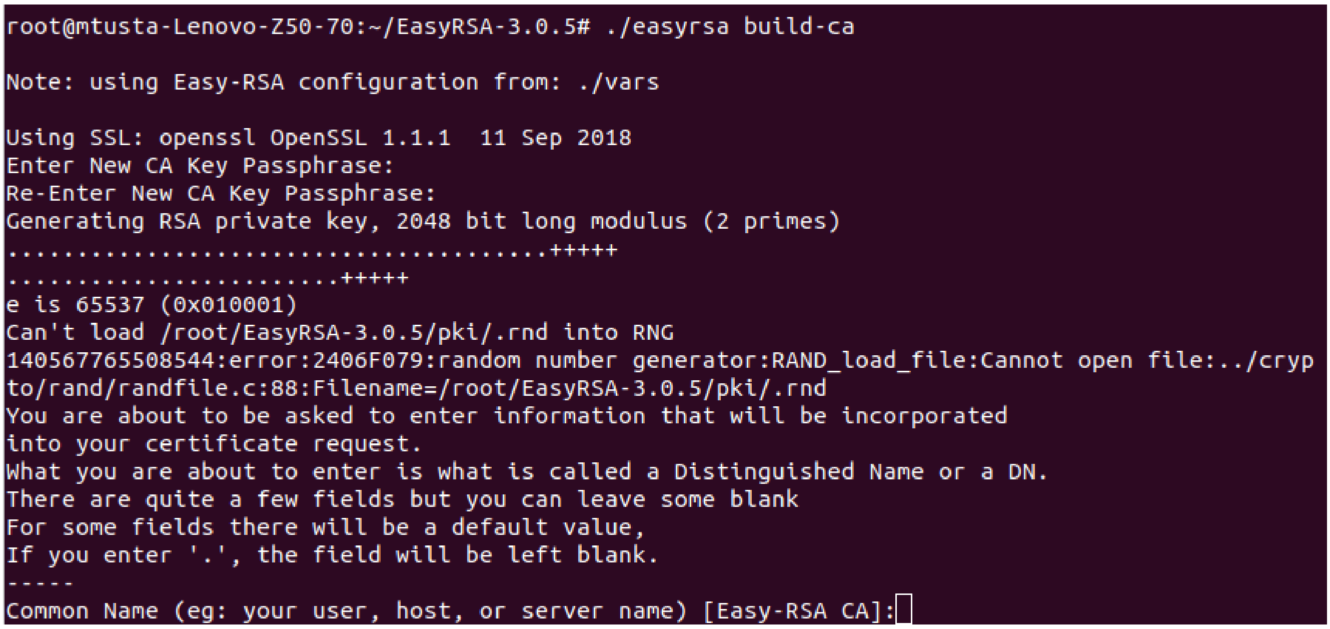
```
./easyrsa init-pki
```



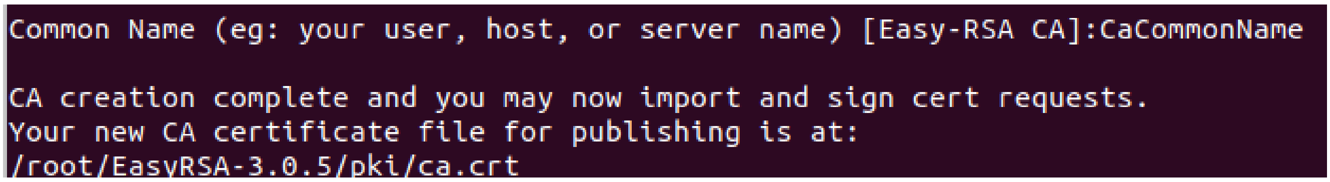
```
root@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5# ./easyrsa init-pki

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1  11 Sep 2018

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /root/EasyRSA-3.0.5/pki
```

The next step is to build the CA:

```
./easyrsa build-ca nopass(optional)
```



```
root@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5# ./easyrsa build-ca

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1  11 Sep 2018
Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................+++++
........................+++++
e is 65537 (0x010001)
Can't load /root/EasyRSA-3.0.5/pki/.rnd into RNG
140567765508544:error:2406F079:random number generator:RAND_load_file:Cannot open file:../cryp
to/rand/randfile.c:88:Filename=/root/EasyRSA-3.0.5/pki/.rnd
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]:
```



```
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:CaCommonName

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/root/EasyRSA-3.0.5/pki/ca.crt
```

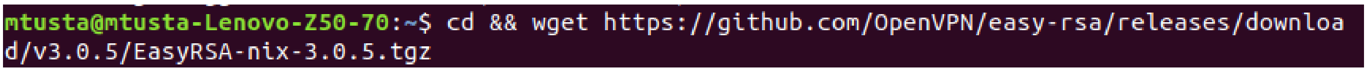
2.2 Installing OpenVPN and EasyRSA

Our next step is to install the OpenVPN package which is available in Ubuntu's repositories and download the latest version of EasyRSA.

```
sudo apt update

sudo apt install openvpn
```

Download the EasyRSA from the project Github repository with the following wget command:

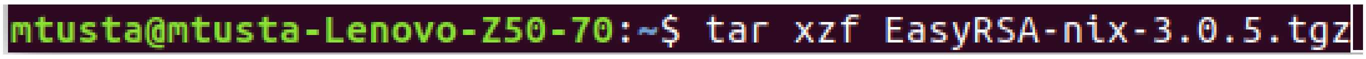
```
cd && wget https://github.com/OpenVPN/easy-rsa/releases/download/v3.0.5/
EasyRSA-nix-3.0.5.tgz
```



```
mtusta@mtusta-Lenovo-Z50-70:~$ cd && wget https://github.com/OpenVPN/easy-rsa/releases/downloa
d/v3.0.5/EasyRSA-nix-3.0.5.tgz
```

Extract the archive:

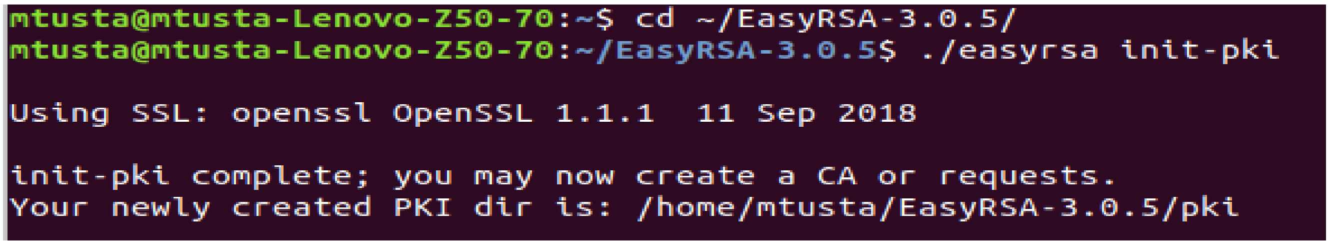
```
tar xzf EasyRSA-nix-3.0.5.tgz
```



```
mtusta@mtusta-Lenovo-Z50-70:~$ tar xzf EasyRSA-nix-3.0.5.tgz
```

we need to create a new PKI on the OpenVPN server

```
cd ~/EasyRSA-3.0.5/

./easyrsa init-pki
```

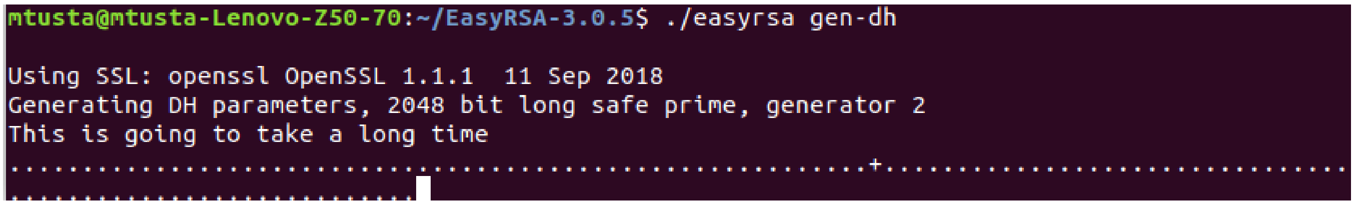


```
mtusta@mtusta-Lenovo-Z50-70:~$ cd ~/EasyRSA-3.0.5/
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ ./easyrsa init-pki

Using SSL: openssl OpenSSL 1.1.1  11 Sep 2018

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/mtusta/EasyRSA-3.0.5/pki
```

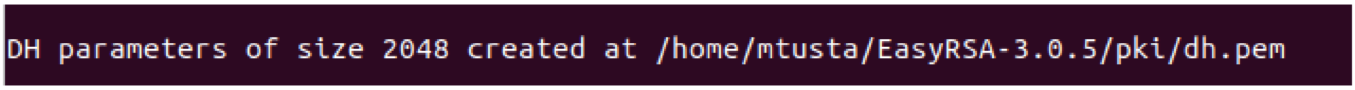
2.3 Creating Diffie-Hellman and HMAC keys

First navigate to the EasyRSA directory on your OpenVPN server and Generate a Diffie-Hellman key:

```
cd ~/EasyRSA-3.0.5/

./easyrsa gen-dh
```



```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ ./easyrsa gen-dh

Using SSL: openssl OpenSSL 1.1.1  11 Sep 2018
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
..............................................................+...................................
..............................
```



```
DH parameters of size 2048 created at /home/mtusta/EasyRSA-3.0.5/pki/dh.pem
```

Copy the dh.pem file to the /etc/openvpn directory:

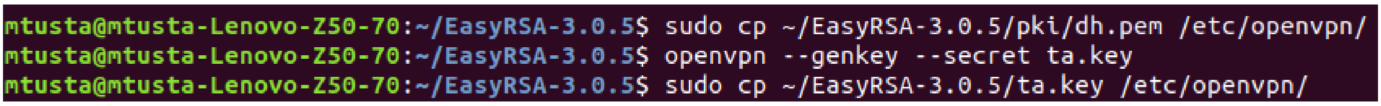
```
sudo cp ~/EasyRSA-3.0.5/pki/dh.pem /etc/openvpn/
```

Generate a HMAC signature:

```
openvpn --genkey --secret ta.key
```

Once completed copy the ta.key file to the /etc/openvpn directory:

```
sudo cp ~/EasyRSA-3.0.5/ta.key /etc/openvpn/
```



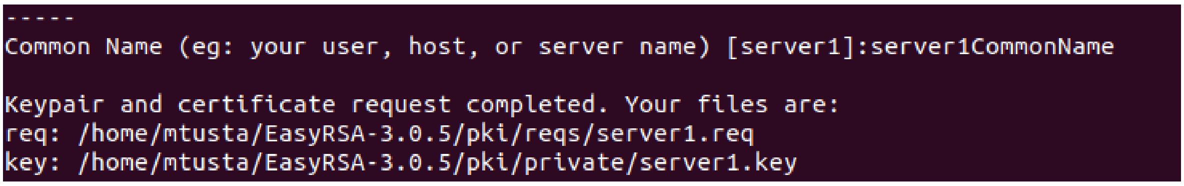
```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo cp ~/EasyRSA-3.0.5/pki/dh.pem /etc/openvpn/
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ openvpn --genkey --secret ta.key
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo cp ~/EasyRSA-3.0.5/ta.key /etc/openvpn/
```

2.4 Creating Server Certificate and Private Key

This section describes how to generate a private key and certificate request for the OpenVPN server.

Navigate to the EasyRSA directory on your OpenVPN server and generate a new private key for the server and a certificate request file:

```
cd ~/EasyRSA-3.0.5/

./easyrsa gen-req server1 nopass(optional)
```



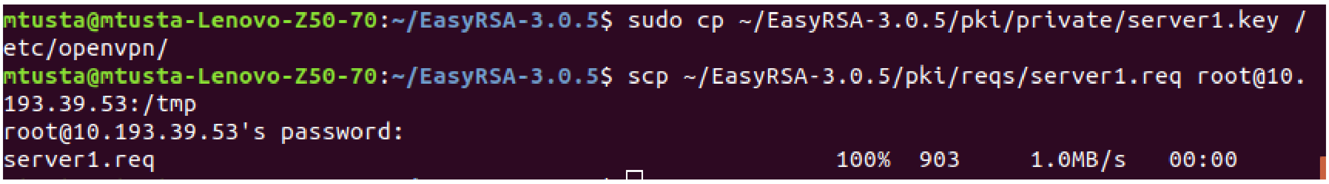
```
-----
Common Name (eg: your user, host, or server name) [server1]:server1CommonName

Keypair and certificate request completed. Your files are:
req: /home/mtusta/EasyRSA-3.0.5/pki/reqs/server1.req
key: /home/mtusta/EasyRSA-3.0.5/pki/private/server1.key
```

Copy the private key to the /etc/openvpn directory:

```
sudo cp ~/EasyRSA-3.0.5/pki/private/server1.key /etc/openvpn/
```

Transfer the certificate request file to your CA machine:

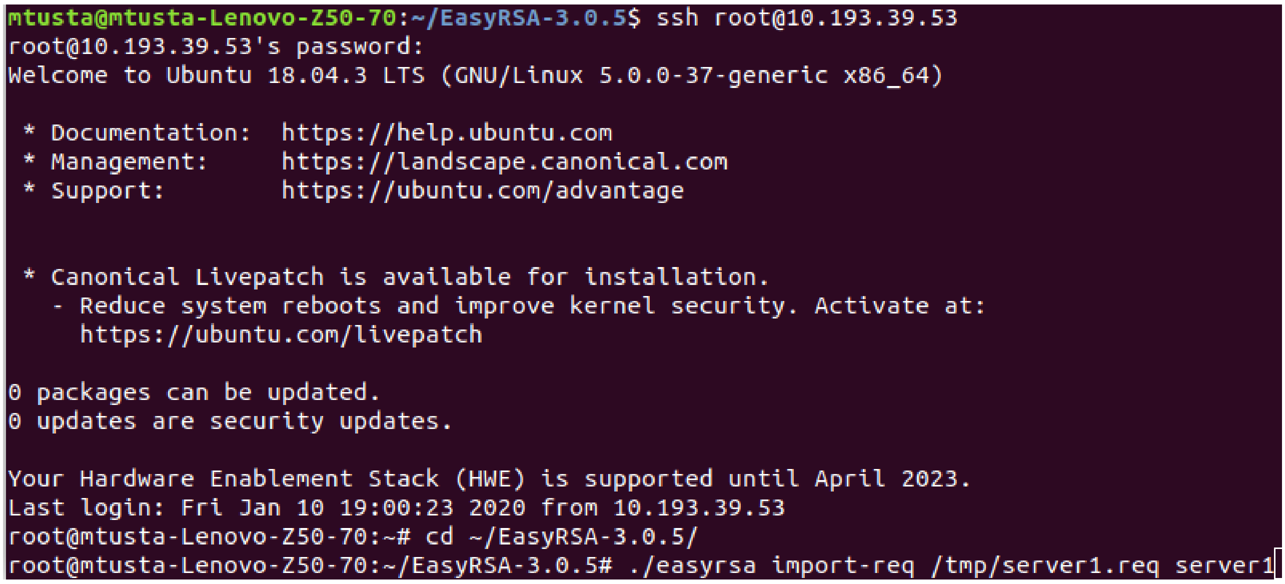
```
scp ~/EasyRSA-3.0.5/pki/reqs/server1.req root@10.193.39.53:/tmp
```



```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo cp ~/EasyRSA-3.0.5/pki/private/server1.key /
etc/openvpn/
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ scp ~/EasyRSA-3.0.5/pki/reqs/server1.req root@10.
193.39.53:/tmp
root@10.193.39.53's password:
server1.req                                        100%  903     1.0MB/s   00:00
```

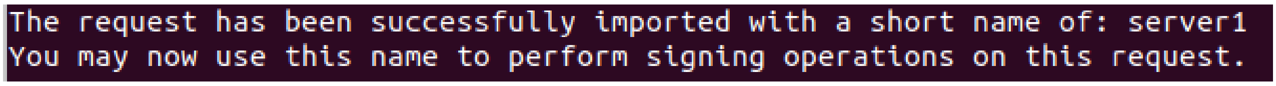
Login to your CA machine, switch to the EasyRSA directory and import the certificate request file:

```
cd ~/EasyRSA-3.0.5

./easyrsa import-req /tmp/server1.req server1
```



```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ ssh root@10.193.39.53
root@10.193.39.53's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Fri Jan 10 19:00:23 2020 from 10.193.39.53
root@mtusta-Lenovo-Z50-70:~# cd ~/EasyRSA-3.0.5/
root@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5# ./easyrsa import-req /tmp/server1.req server1
```

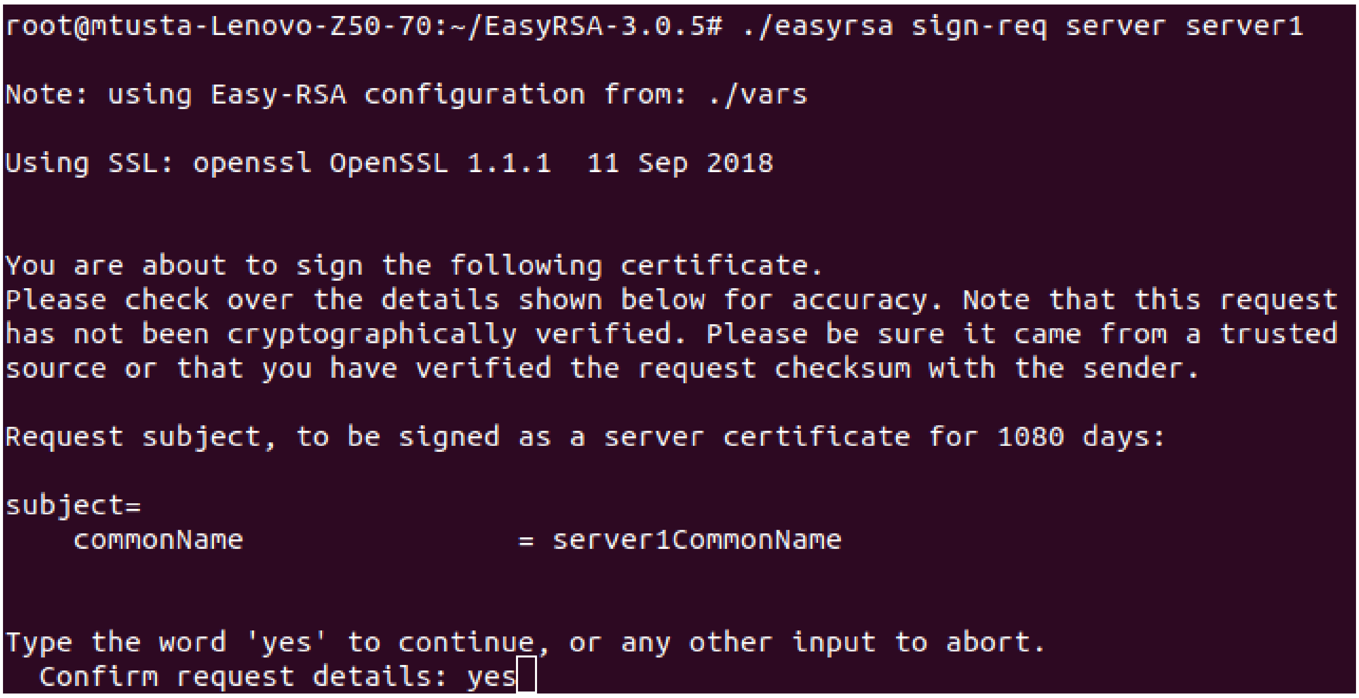
This command just copies the request file into the pki/reqs directory.



```
The request has been successfully imported with a short name of: server1
You may now use this name to perform signing operations on this request.
```

While still in the EasyRSA directory on CA machine run the following command to sign the request:

```
./easyrsa sign-req server server1
```



```
root@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5# ./easyrsa sign-req server server1

Note: using Easy-RSA configuration from: ./vars

Using SSL: openssl OpenSSL 1.1.1  11 Sep 2018


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 1080 days:

subject=
    commonName                = server1CommonName


Type the word 'yes' to continue, or any other input to abort.
  Confirm request details: yes
```

Next step is to transfer the signed certificate server1.crt and ca.crt files back to your OpenVPN server. Again you can use scp, rsync or any other secure method:

```
scp ~/EasyRSA-3.0.5/pki/issued/server1.crt mtusta@10.193.39.53:/tmp

scp ~/EasyRSA-3.0.5/pki/ca.crt mtusta@10.193.39.53:/tmp
```

Login to your OpenVPN server, and move the server1.crt and ca.crt files into the /etc/openvpn/ directory:

```
sudo mv /tmp/{server1,ca}.crt /etc/openvpn/
```

Upon completing the steps outlined in this section, you should have the following new files on your OpenVPN server:

- `/etc/openvpn/ca.crt`
- `/etc/openvpn/dh.pem`
- `/etc/openvpn/ta.key`
- `/etc/openvpn/server1.crt`
- `/etc/openvpn/server1.key`

2.5 Configuring the OpenVPN Service

Start by extracting the configuration file to the /etc/openvpn/ directory:

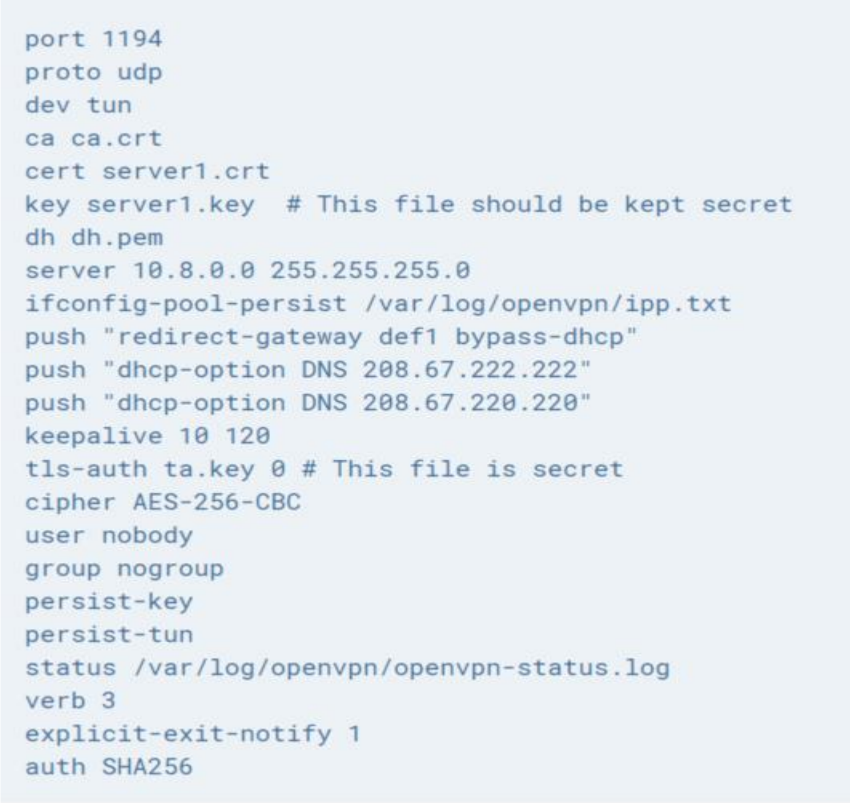
```
sudo sh -c "gunzip -c /usr/share/doc/openvpn/examples/sample-config-
files/server.conf.gz > /etc/openvpn/server1.conf"
```

Open the file with text editor:

```
sudo gedit /etc/openvpn/server1.conf
```

Delete some command line character (# or ;)

Once you are done, the server configuration file (excluding comments) should look something like this:

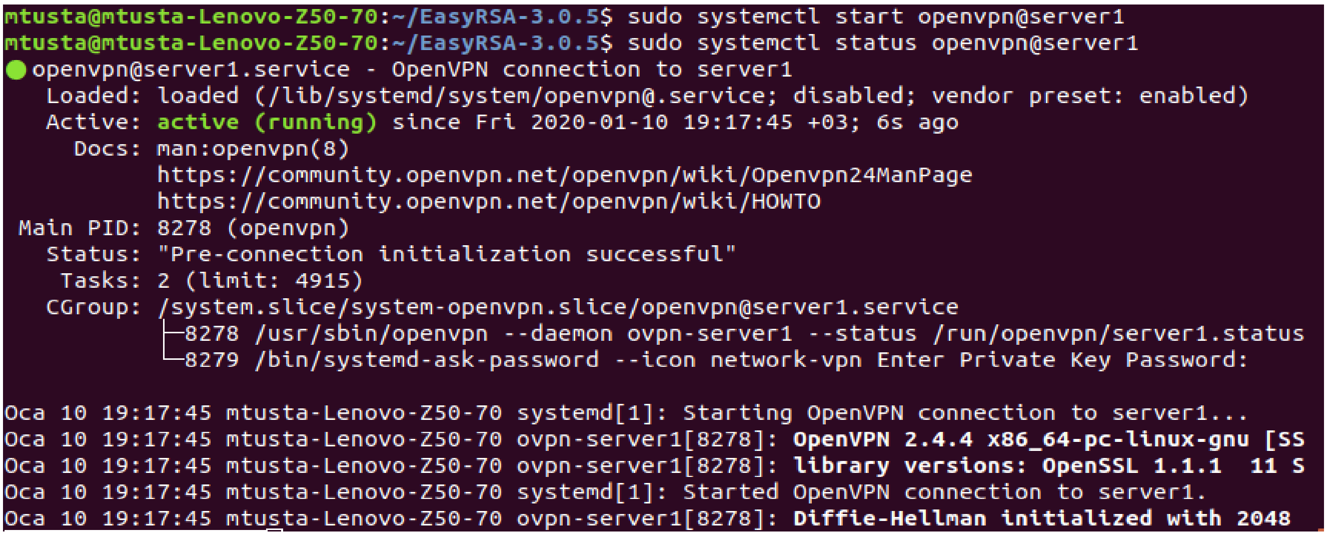


```
port 1194
proto udp
dev tun
ca ca.crt
cert server1.crt
key server1.key  # This file should be kept secret
dh dh.pem
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
keepalive 10 120
tls-auth ta.key 0 # This file is secret
cipher AES-256-CBC
user nobody
group nogroup
persist-key
persist-tun
status /var/log/openvpn/openvpn-status.log
verb 3
explicit-exit-notify 1
auth SHA256
```

2.6 Starting OpenVPN Service

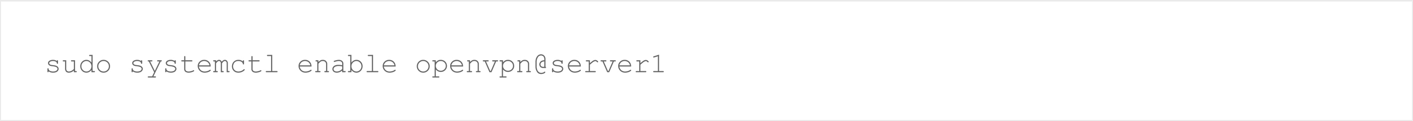
On your OpenVPN server run the following command to start the OpenVPN service:

```
sudo systemctl start openvpn@server1

sudo systemctl status openvpn@server1
```

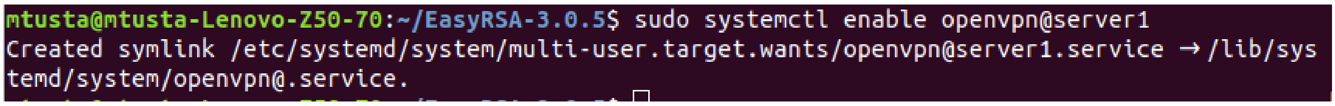


```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo systemctl start openvpn@server1
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo systemctl status openvpn@server1
● openvpn@server1.service - OpenVPN connection to server1
   Loaded: loaded (/lib/systemd/system/openvpn@.service; disabled; vendor preset: enabled)
   Active: active (running) since Fri 2020-01-10 19:17:45 +03; 6s ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
 Main PID: 8278 (openvpn)
   Status: "Pre-connection initialization successful"
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/system-openvpn.slice/openvpn@server1.service
           ├─8278 /usr/sbin/openvpn --daemon ovpn-server1 --status /run/openvpn/server1.status
           └─8279 /bin/systemd-ask-password --icon network-vpn Enter Private Key Password:

Oca 10 19:17:45 mtusta-Lenovo-Z50-70 systemd[1]: Starting OpenVPN connection to server1...
Oca 10 19:17:45 mtusta-Lenovo-Z50-70 ovpn-server1[8278]: OpenVPN 2.4.4 x86_64-pc-linux-gnu [SS
Oca 10 19:17:45 mtusta-Lenovo-Z50-70 ovpn-server1[8278]: library versions: OpenSSL 1.1.1  11 S
Oca 10 19:17:45 mtusta-Lenovo-Z50-70 systemd[1]: Started OpenVPN connection to server1.
Oca 10 19:17:45 mtusta-Lenovo-Z50-70 ovpn-server1[8278]: Diffie-Hellman initialized with 2048
```

Enable the service to automatically start on boot with:



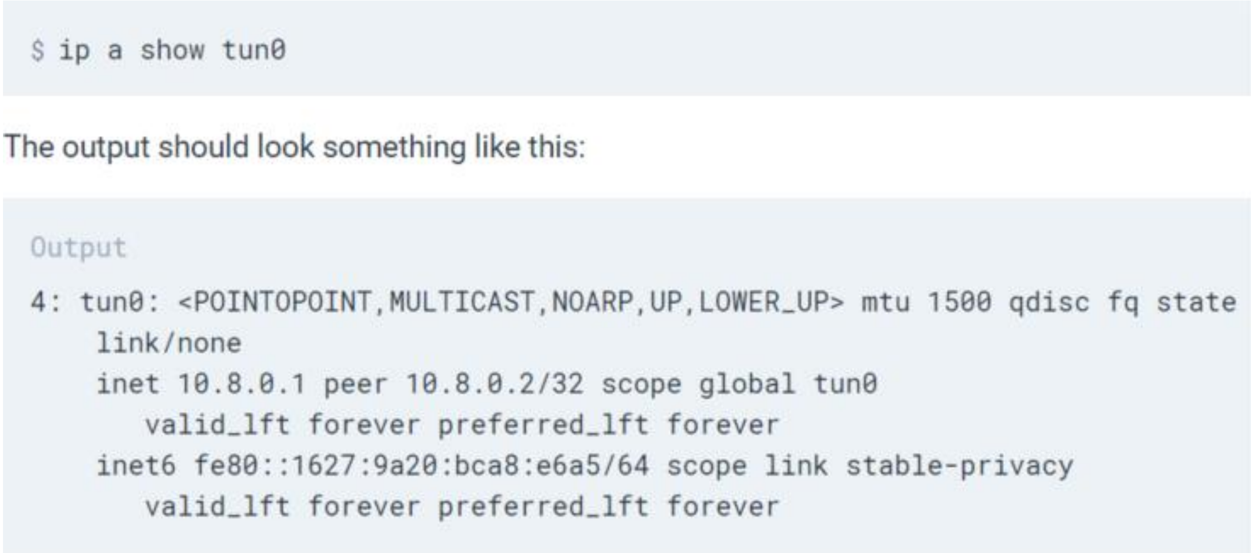
```
sudo systemctl enable openvpn@server1
```



```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo systemctl enable openvpn@server1
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn@server1.service → /lib/systemd/system/openvpn@.service.
```

If the OpenVPN service fails to start check the logs with sudo journalctl -u openvpn@server1

The OpenVPN Server will create a new tun device tun0. To check whether the device is available, use the following ip command:



```
$ ip a show tun0
```

The output should look something like this:

```
Output
4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq state
    link/none
    inet 10.8.0.1 peer 10.8.0.2/32 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::1627:9a20:bca8:e6a5/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

At this point, your OpenVPN server is configured and running properly.

2.7 Firewall and Server Networking Configuration

In order to forward network packets properly, we need to enable IP forwarding.

```
sudo gedit /etc/sysctl.conf
```

# Uncomment the next line to enable packet forwarding for IPv4

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

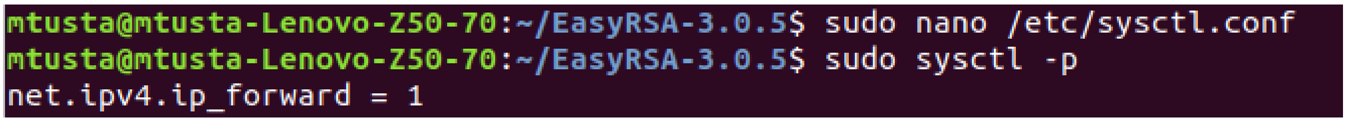
Save and close the file.

Apply the new settings by running the following command:

```
sudo sysctl -p
```

Output:

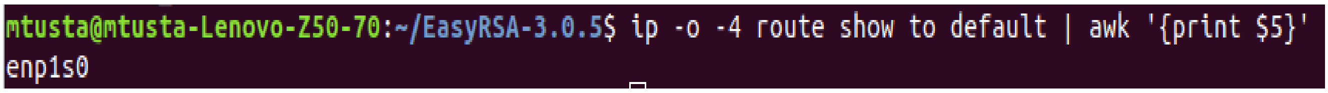
```
net.ipv4.ip_forward = 1
```



```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo nano /etc/sysctl.conf
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo sysctl -p
net.ipv4.ip_forward = 1
```

public network interface of Ubuntu OpenVPN Server.

```
ip -o -4 route show to default | awk '{print $5}'
```



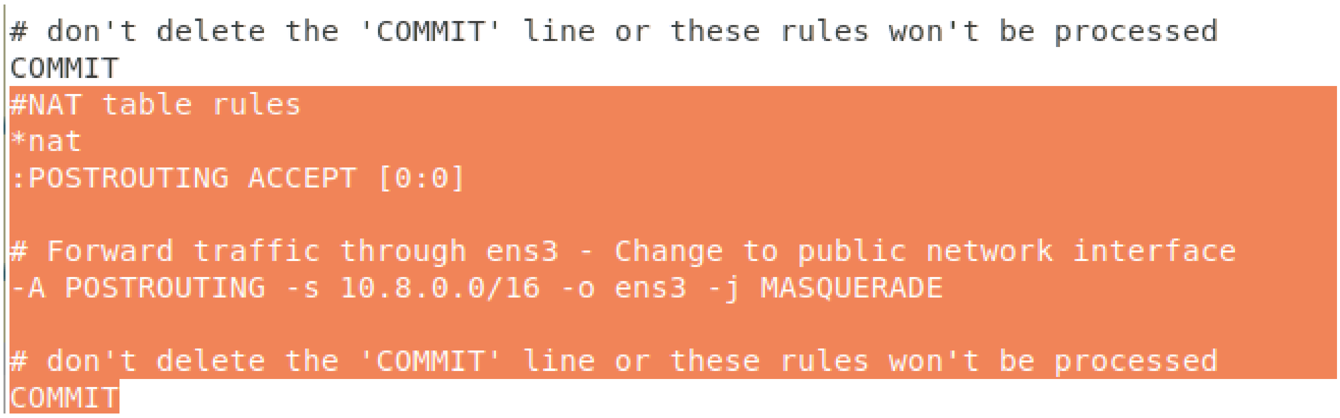
```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ ip -o -4 route show to default | awk '{print $5}'
enp1s0
```

Open the UFW configuration file, locate the DEFAULT_FORWARD_POLICY key and change the value from DROP to ACCEPT:

```
# Set the default forward policy to ACCEPT, DROP or REJECT.  Please note that

# if you change this you will most likely want to adjust your rules

DEFAULT_FORWARD_POLICY="ACCEPT"
```

Next, we need to set the default policy for the POSTROUTING chain in the nat table and set the masquerade rule.

```
sudo gedit /etc/ufw/before.rules
```



```
# don't delete the 'COMMIT' line or these rules won't be processed
COMMIT
#NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]

# Forward traffic through ens3 - Change to public network interface
-A POSTROUTING -s 10.8.0.0/16 -o ens3 -j MASQUERADE

# don't delete the 'COMMIT' line or these rules won't be processed
COMMIT
```

```
# don't delete the 'COMMIT' line or these rules won't be processed

COMMIT

#NAT table rules

*nat

:POSTROUTING ACCEPT [0:0]

# Forward traffic through ens3 - Change to public network interface

-A POSTROUTING -s 10.8.0.0/16 -o ens3 -j MASQUERADE

# don't delete the 'COMMIT' line or these rules won't be processed

COMMIT
```

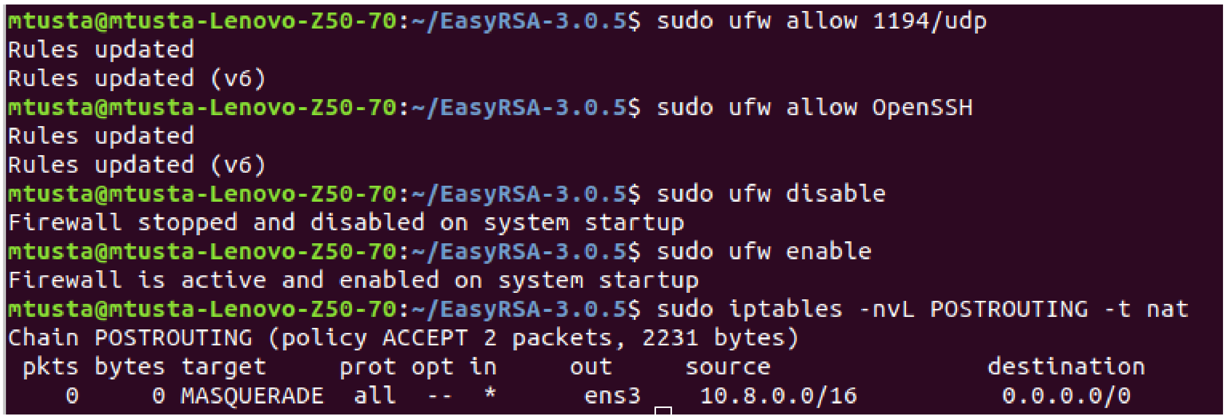
When you are done, save and close the file.

We also need to open UDP traffic on port 1194

```
sudo ufw allow 1194/udp

sudo ufw allow OpenSSH

sudo ufw disable

sudo ufw enable
```

To verify the changes run the following command to list the POSTROUTING rules:

```
sudo iptables -nvL POSTROUTING -t nat
```



```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo ufw allow 1194/udp
Rules updated
Rules updated (v6)
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo ufw allow OpenSSH
Rules updated
Rules updated (v6)
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo ufw disable
Firewall stopped and disabled on system startup
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo ufw enable
Firewall is active and enabled on system startup
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ sudo iptables -nvL POSTROUTING -t nat
Chain POSTROUTING (policy ACCEPT 2 packets, 2231 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 MASQUERADE  all  --  *      ens3    10.8.0.0/16          0.0.0.0/0
```

2.8 Creating the Client Configuration Infrastructur

Start by creating a set of directories to store the clients files:

base directory will store the base files and configuration that will be shared across all client files.
configs directory will store the generated client configuration.
files directory will store client-specific certificate/key pair.

```
mkdir -p ~/openvpn-clients/{configs,base,files}
```

Copy the ca.crt and ta.key files to the ~/openvpn-clients/base directory:

```
cp ~/EasyRSA-3.0.5/ta.key ~/openvpn-clients/base/

cp /etc/openvpn/ca.crt ~/openvpn-clients/base/
```

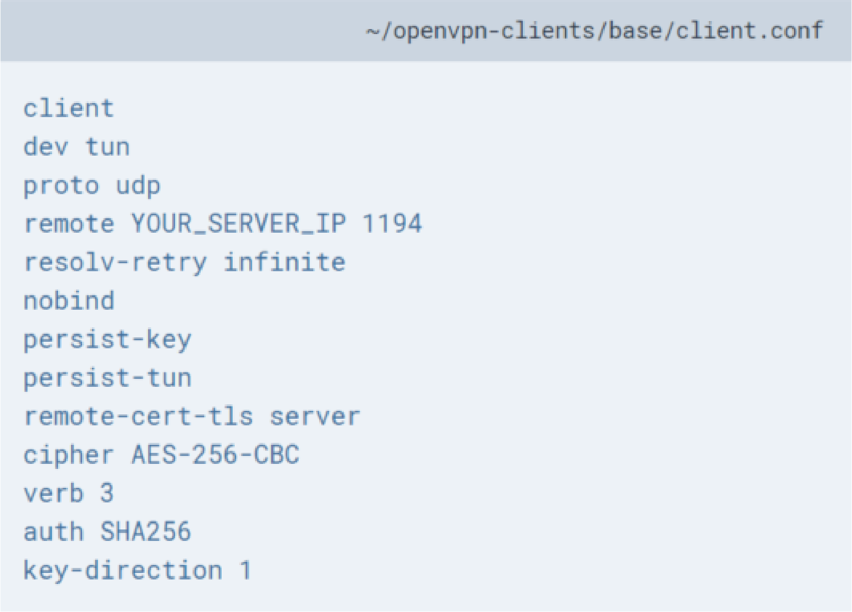
Next copy the sample VPN client configuration file into the client-~/openvpn-clients/base directory. We will use this file as a base configuration:

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

Open the configuration file with your text editor::

```
gedit ~/openvpn-clients/base/client.conf
```

The server configuration file should look something like this:



~/openvpn-clients/base/client.conf

```
client
dev tun
proto udp
remote YOUR_SERVER_IP 1194
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
cipher AES-256-CBC
verb 3
auth SHA256
key-direction 1
```

Next, create a simple bash script that will merge the base configuration and files with the client certificate and key, and store the generated configuration in the ~/openvpn-clients/configs directory.

```
nano ~/openvpn-clients/gen_config.sh
```

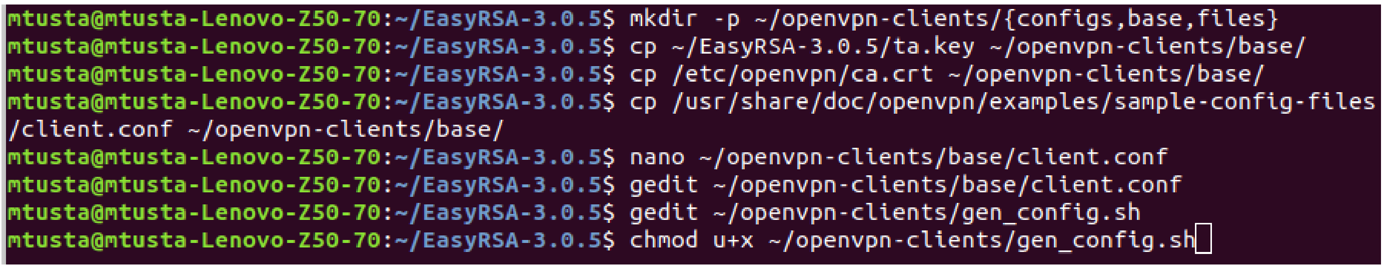
```
#!/bin/bash

FILES_DIR=$HOME/openvpn-clients/files

BASE_DIR=$HOME/openvpn-clients/base

CONFIGS_DIR=$HOME/openvpn-clients/configs



BASE_CONF=${BASE_DIR}/client.conf

CA_FILE=${BASE_DIR}/ca.crt

TA_FILE=${BASE_DIR}/ta.key



CLIENT_CERT=${FILES_DIR}/${1}.crt
```

```
CLIENT_KEY=${FILES_DIR}/${1}.key

# Test for files
for i in "$BASE_CONF" "$CA_FILE" "$TA_FILE" "$CLIENT_CERT" "$CLIENT_KEY"; do
    if [[ ! -f $i ]]; then
        echo " The file $i does not exist"
        exit 1
    fi

    if [[ ! -r $i ]]; then
        echo " The file $i is not readable."
        exit 1
    fi
done

# Generate client config
cat > ${CONFIGS_DIR}/${1}.ovpn <<EOF
$(cat ${BASE_CONF})
<key>
$(cat ${CLIENT_KEY})
</key>
<cert>
$(cat ${CLIENT_CERT})
</cert>
<ca>
$(cat ${CA_FILE})
```

```
</ca>

<tls-auth>

$(cat ${TA_FILE})

</tls-auth>

EOF
```

Save the file and make it executable by running the following chmod command:

```
chmod u+x ~/openvpn-clients/gen_config.sh
```

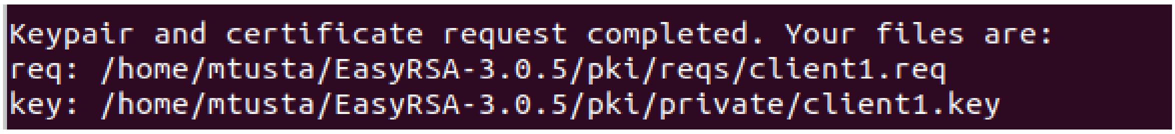


```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ mkdir -p ~/openvpn-clients/{configs,base,files}
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ cp ~/EasyRSA-3.0.5/ta.key ~/openvpn-clients/base/
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ cp /etc/openvpn/ca.crt ~/openvpn-clients/base/
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ cp /usr/share/doc/openvpn/examples/sample-config-files
/client.conf ~/openvpn-clients/base/
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ nano ~/openvpn-clients/base/client.conf
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ gedit ~/openvpn-clients/base/client.conf
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ gedit ~/openvpn-clients/gen_config.sh
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ chmod u+x ~/openvpn-clients/gen_config.sh
```

2.9 Creating Client Certificate Private Key and Configuration

Navigate to the EasyRSA directory on your OpenVPN server and generate a new private key and a certificate request file for the client:

```
cd ~/EasyRSA-3.0.5/

./easyrsa gen-req client1 nopass(optional)
```



```
Keypair and certificate request completed. Your files are:
req: /home/mtusta/EasyRSA-3.0.5/pki/reqs/client1.req
key: /home/mtusta/EasyRSA-3.0.5/pki/private/client1.key
```

Copy the private key client1.key to the ~/openvpn-clients/files directory you created in the previous section:

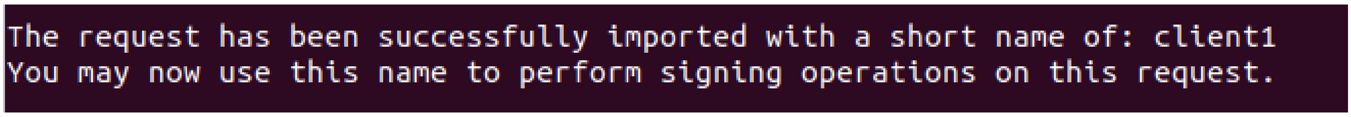
```
cp ~/EasyRSA-3.0.5/pki/private/client1.key ~/openvpn-clients/files/
```

Transfer the certificate request file to your CA machine:

```
scp ~/EasyRSA-3.0.5/pki/reqs/client1.req root@10.193.39.53:/tmp
```

Login to your CA machine, switch to the EasyRSA directory and import the certificate request file:

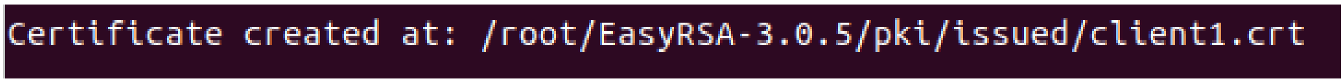
```
cd ~/EasyRSA-3.0.5

./easyrsa import-req /tmp/client1.req client1
```



```
The request has been successfully imported with a short name of: client1
You may now use this name to perform signing operations on this request.
```

From within the EasyRSA directory on CA machine run the following command to sign the request:

```
./easyrsa sign-req client client1
```



```
Certificate created at: /root/EasyRSA-3.0.5/pki/issued/client1.crt
```

Next, transfer the signed certificate client1.crt file back to your OpenVPN server. You can use scp, rsync or any other secure method:

```
scp ~/EasyRSA-3.0.5/pki/issued/client1.crt mtusta@10.193.39.53:/tmp
```

Login to your OpenVPN server, and move the client1.crt file into the ~/openvpn-clients/files directory:

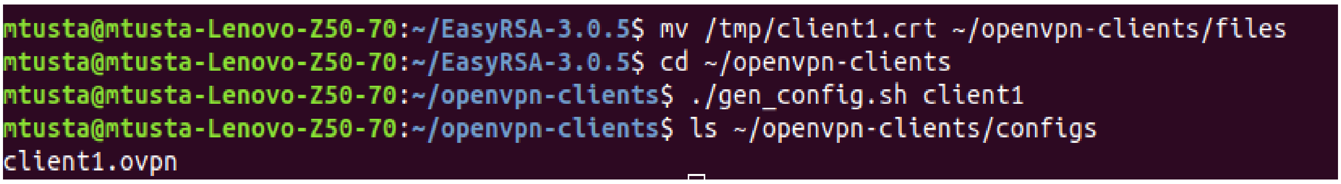
```
mv /tmp/client1.crt ~/openvpn-clients/files
```

The final step is to generate a client configuration using the gen_config.sh script. Switch to the ~/openvpn-clients directory and run the script using the client name as an argument:

```
cd ~/openvpn-clients

./gen_config.sh client1
```

The script will create a file named client1.ovpn in the ~/client-configs/configs directory. You can check by listing the directory:

```
ls ~/openvpn-clients/configs
```



```
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ mv /tmp/client1.crt ~/openvpn-clients/files
mtusta@mtusta-Lenovo-Z50-70:~/EasyRSA-3.0.5$ cd ~/openvpn-clients
mtusta@mtusta-Lenovo-Z50-70:~/openvpn-clients$ ./gen_config.sh client1
mtusta@mtusta-Lenovo-Z50-70:~/openvpn-clients$ ls ~/openvpn-clients/configs
client1.ovpn
```

At this point the client configuration is created. You can now transfer the configuration file to the device you intend to use as a client.

2.10 Connecting Clients

Install OpenVPN on Ubuntu

```
 sudo apt update

sudo apt install openvpn
```

Once the package is installed, to connect to the VPN server use the openvpn command and specify the client configuration file:

```
sudo openvpn --config client1.ovpn
```