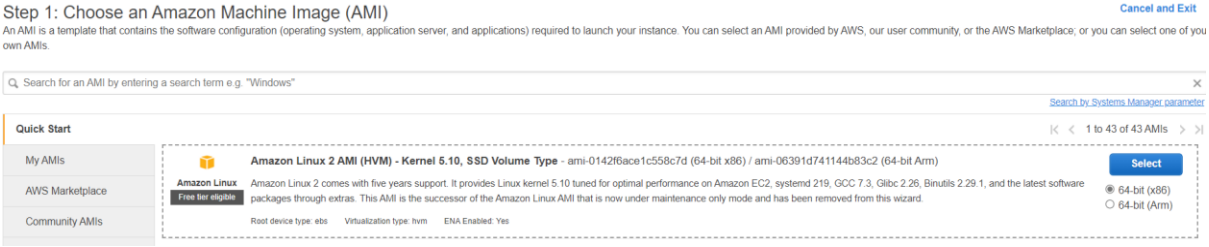
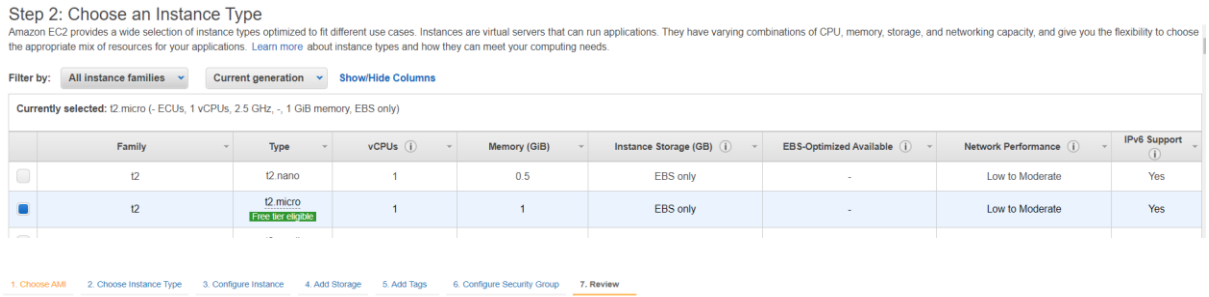
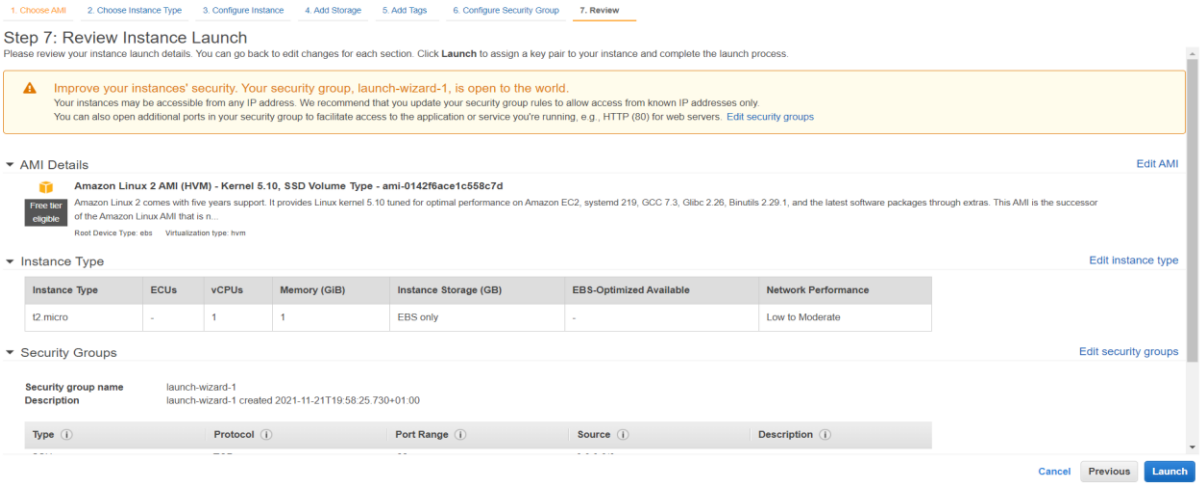
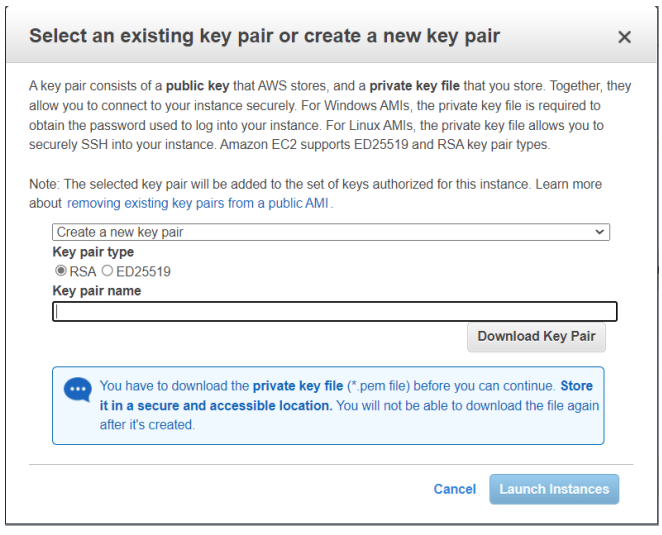
Setting up a VPN server using AWS

1. To set up a VPN server, at first we need to launch an EC2 instance. Which we are going to do in the AWS console. Navigate to EC2, select launch instance. I chose to leave the default settings as they are. So I chose for the default AMI (Amazon Linux 2) and the default instance type (t2.micro). Then I selected review and launch, while launching, make sure you choose to create a new keypair and download the keypair, so you can use it to connect with the ssh client.

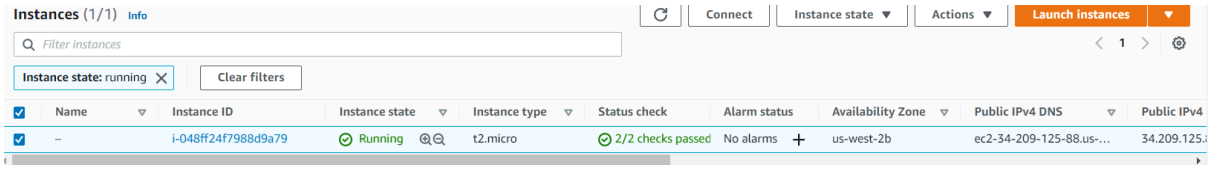


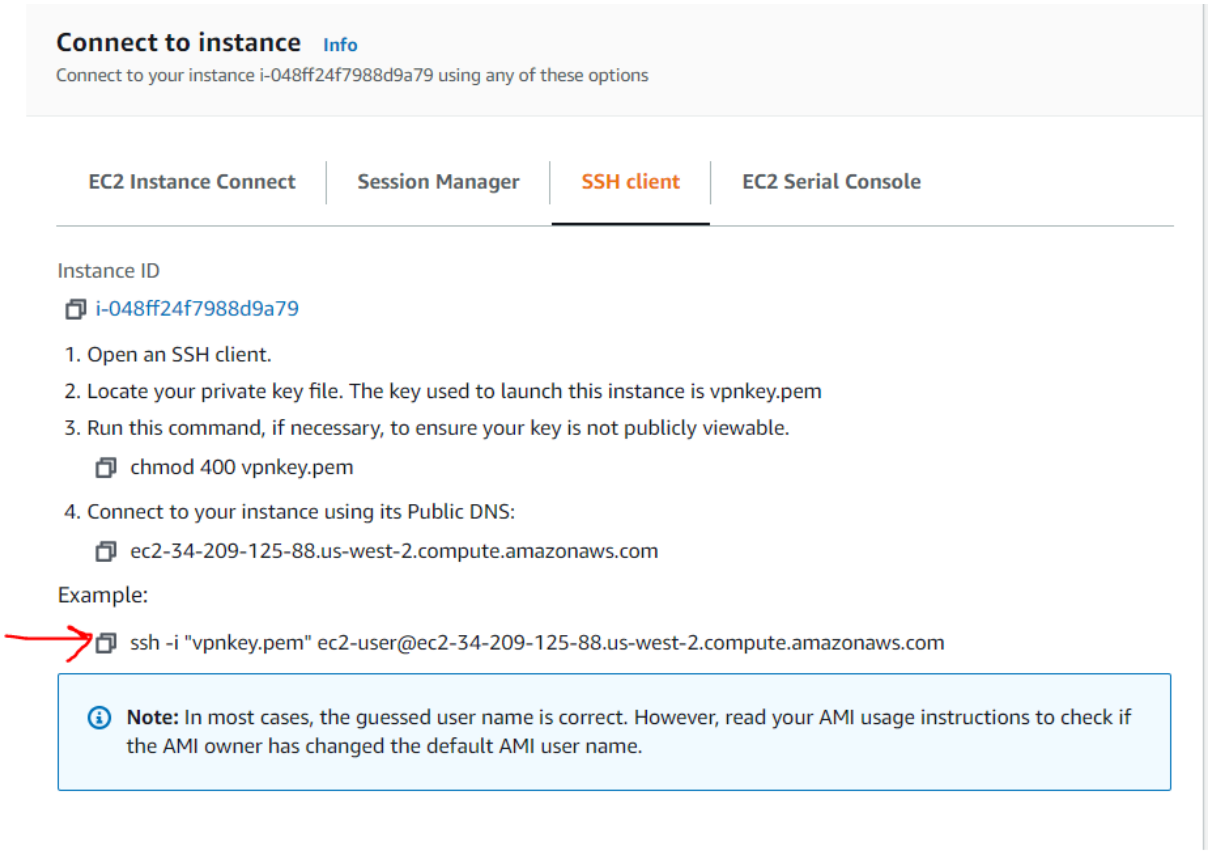






2. Wait till the status checks have passed, select the instance and then click on connect. Click on the tab ssh client, copy and paste the last link in the ssh client to connect with the instance.





3. I opened command prompt, navigated to the directory in which the keypair was downloaded, then pasted the example link and then pressed enter to make connection. Type 'yes' when asked if you're sure to continue connecting.

```
C:\Users\steel>cd downloads

C:\Users\steel\Downloads>ssh -i "vpnkey.pem" ec2-user@ec2-34-209-125-88.us-west-2.compute.amazonaws.com
The authenticity of host 'ec2-34-209-125-88.us-west-2.compute.amazonaws.com (34.209.125.88)' can't be established.
ECDSA key fingerprint is SHA256:GfZAtxRthRLOOcKZfbUxFJUN92LHNPfWxEgsGAcXeLs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-209-125-88.us-west-2.compute.amazonaws.com,34.209.125.88' (ECDSA) to the list of know
n hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-23-212 ~]$
```

4. Before we do anything else, we are going to switch to the root user to make sure that we have all permissions to run any necessary command. We can do this by using the following command: $ sudo su root

I prefer to start in the root directory, to navigate to the root directory use command:
$ cd

```
[ec2-user@ip-172-31-23-212 ~]$ sudo su root
[root@ip-172-31-23-212 ec2-user]# cd
[root@ip-172-31-23-212 ~]# yum update -y
```

5. Now we are going to make sure that all repositories and packages we might need are up to date by using this command:
   $ yum update -y

   We will need the OpenVPN package to do this exercise, which can not be downloaded from the default repositories, it is available in the EPEL (Extra Packages for Enterprise Linux) repository. To enable the EPEL repository we run this command:
   $ yum install epel-release -y
   By using this command I found out there was no epel-release package available. Reading the output I found out the following command should work:
   $ sudo amazon-linux-extras install epel
   And indeed, it did work!

   Then update all repositories and packages again:
   $ yum update -y

   Finally install OpenVPN with the following command:
   $ yum install -y openvpn

```
Transaction test succeeded
Running transaction
  Installing : pkcs11-helper-1.11-3.el7.x86_64                                  1/3
  Installing : lzo-2.06-8.amzn2.0.4.x86_64                                      2/3
  Installing : openvpn-2.4.11-1.el7.x86_64                                      3/3
  Verifying  : lzo-2.06-8.amzn2.0.4.x86_64                                      1/3
  Verifying  : openvpn-2.4.11-1.el7.x86_64                                      2/3
  Verifying  : pkcs11-helper-1.11-3.el7.x86_64                                  3/3

Installed:
  openvpn.x86_64 0:2.4.11-1.el7

Dependency Installed:
  lzo.x86_64 0:2.06-8.amzn2.0.4                 pkcs11-helper.x86_64 0:1.11-3.el7

Complete!
[root@ip-172-31-23-212 ec2-user]#
```

   As you can see the OpenVPN package is installed!

6. In the next step we are going to install easy rsa, which we will need to generate a SSL key pair and a PKI Certificate Authority (CA). Use the following command to install easy rsa:
   $ yum install -y wget
   Then download the latest version of the CLI utility, which is 3.0.8 at the moment. You can do this by using the following command:
   $ wget https://github.com/OpenVPN/easy-rsa/archive/v3.0.8.tar.gz

```
--2021-11-21 19:58:14--  https://codeload.github.com/OpenVPN/easy-rsa/tar.gz/v3.0.8
Resolving codeload.github.com (codeload.github.com)... 192.30.255.120
Connecting to codeload.github.com (codeload.github.com)|192.30.255.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'v3.0.8.tar.gz'

    [   <=>                                                                 ] 3,864,366   8.04MB/s   in 0.5s

2021-11-21 19:58:15 (8.04 MB/s) - 'v3.0.8.tar.gz' saved [3864366]

[root@ip-172-31-23-212 ec2-user]#
```

In the output you can check if the archive file is successfully saved or not. Now we need to extract the file, which is possible by using the following command:

$ tar -xf v3.0.8.tar.gz

7. In the next step we will create a new directory (openvpn) and move into it straightaway. Use command:

   $ cd /etc/openvpn/

   Then create a subdirectory in directory openvpn by using command:

   $ mkdir /etc/openvpn/easy-rsa

   By listing it with command: $ ll you can check if the directory was created successful.

```
[root@ip-172-31-23-212 ~]# tar -xf v3.0.8.tar.gz
[root@ip-172-31-23-212 ~]# cd /etc/openvpn/
[root@ip-172-31-23-212 openvpn]# mkdir /etc/openvpn/easy-rsa
[root@ip-172-31-23-212 openvpn]# ll
total 0
drwxr-x--- 2 root openvpn 6 Apr 21  2021 client
drwxr-xr-x 2 root root    6 Nov 21 20:15 easy-rsa
drwxr-x--- 2 root openvpn 6 Apr 21  2021 server
[root@ip-172-31-23-212 openvpn]#
```

   And then move the earlier extracted directory into /etc/openvpn/easy-rsa by using command:

   $ mv /root/easy-rsa-3.0.8 /etc/openvpn/easy-rsa

   Move into directory easy rsa with command: $ cd easy-rsa

   Then use $ ll again to list en check if the directory was moved successfully.

```
[root@ip-172-31-23-212 openvpn]# mv /root/easy-rsa-3.0.8 /etc/openvpn/easy-rsa
[root@ip-172-31-23-212 openvpn]# cd easy-rsa
[root@ip-172-31-23-212 easy-rsa]# ll
total 0
drwxrwxr-x 8 root root 311 Sep 16  2020 easy-rsa-3.0.8
[root@ip-172-31-23-212 easy-rsa]#
```

8. Now we are going to configure the OpenVPN. Before moving on, make sure you return to the root directory by using command $ cd. The first step is to copy the sample server.conf file from OpenVPN's documentation directory:

   $ cp /usr/share/doc/openvpn-2.4.9/sample/sample-config-files/server.conf /etc/openvpn

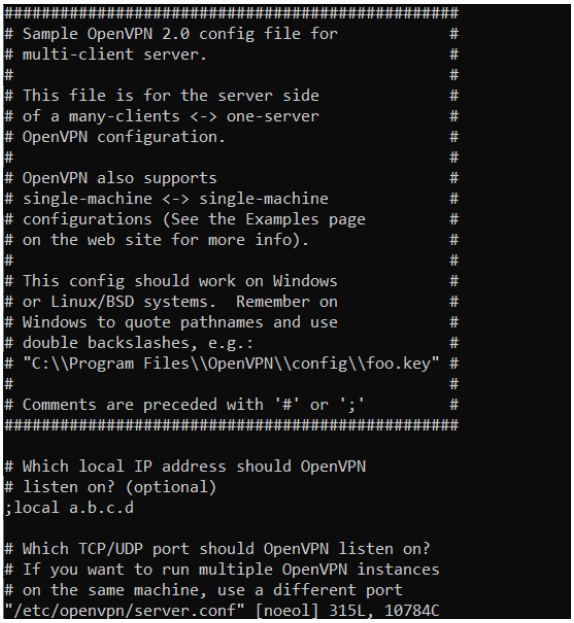
   If you cannot find the OpenVPN sample configuration file, search for its location using the find command:

   $ find / -name server.conf

   Then, open the copied configuration file with a text editor of your choice:

   $ vi etc/openvpn/server.conf

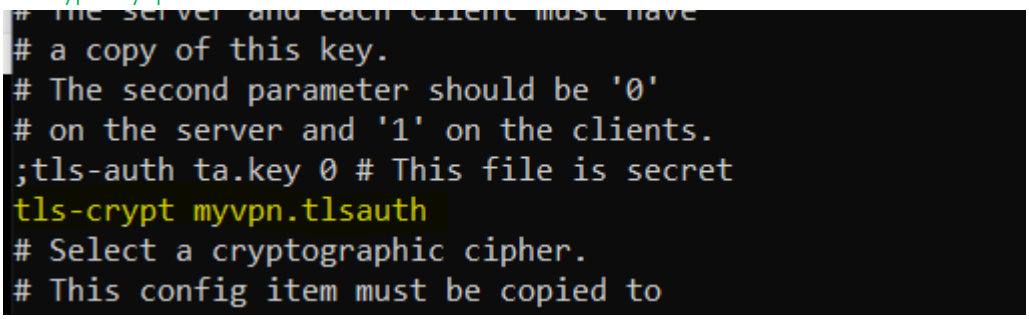
   You should see the following in the file:



To set up the basic configurations, we need to uncomment a few lines in this file. To uncomment just remove the ; (semicolons) or # hashtags.

- topology subnet (makes the OpenVPN installation function as a subnetwork)
- push "redirect-gateway def1 bypass-dhcp" (instructs the client to redirect traffic through the OpenVPN server)
- push "dhcp-option DNS 208.67.222.222" (uses an OpenDNS resolver to connect to OpenVPN)
- push "dhcp-option DNS 208.67.220.220" (uses an OpenDNS resolver to connect to OpenVPN)
- user nobody (runs OpenVPN with no privileges)
- group nobody (runs OpenVPN with no privileges)

Then, generate a static encryption key to enable TLS authentication. To do that, locate the line tls-auth ta.key 0 and comment it by adding ; in front of it. Then, add a new line under it: tls-crypt myvpn.tlsauth



Save the changes and exit the file. Now we can generate the static encryption key which we just specified in the configuration file. Use command:

$ openvpn --genkey --secret /etc/openvpn/myvpn.tlsauth

9. Now we have configured the OpenVPN we are going to generate keypair and certificate. At first we are creating a vars configuration file using vars.example which is stored in the easy-rsa/easy-rsa-3.0.8/easyrsa3. To move into the right directory use:
$ cd /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3

Use command: $ ls to check if there is any file named vars.example

```
[root@ip-172-31-23-212 ~]# cd /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3
[root@ip-172-31-23-212 easyrsa3]# ls
easyrsa  openssl-easyrsa.cnf  vars.example  x509-types
[root@ip-172-31-23-212 easyrsa3]#
```

Copy the sample file vars.example under the name vars:
$ cp vars.example vars

```
[root@ip-172-31-23-212 easyrsa3]# cp vars.example vars
[root@ip-172-31-23-212 easyrsa3]# ls
easyrsa  openssl-easyrsa.cnf  vars  vars.example  x509-types
[root@ip-172-31-23-212 easyrsa3]#
```

Open the vars file in a text editor:
$ vi vars
The file looks like this:

```
# Easy-RSA 3 parameter settings

# NOTE: If you installed Easy-RSA from your distro's package manager, don't edit
# this file in place -- instead, you should copy the entire easy-rsa directory
# to another location so future upgrades don't wipe out your changes.

# HOW TO USE THIS FILE
#
# vars.example contains built-in examples to Easy-RSA settings. You MUST name
# this file 'vars' if you want it to be used as a configuration file. If you do
# not, it WILL NOT be automatically read when you call easyrsa commands.
#
# It is not necessary to use this config file unless you wish to change
# operational defaults. These defaults should be fine for many uses without the
# need to copy and edit the 'vars' file.
#
```

Scroll through it, find the following lines, uncomment them by removing the # and replace the default values with your own information.
#set_var EASYRSA_REQ_COUNTRY "US"
#set_var EASYRSA_REQ_PROVINCE "California"
#set_var EASYRSA_REQ_CITY "San Francisco"
#set_var EASYRSA_REQ_ORG "Copyleft Certificate Co"
#set_var EASYRSA_REQ_EMAIL "me@example.net"
#set_var EASYRSA_REQ_OU "My Organizational Unit"

Then add the following lines:
export KEY_NAME="server"
export KEY_CN=openvpn.yourdomain.com

Save your changes and exit the file.

Now we have adjusted the configuration file, we can start generating the key-pair and certificate. But first we will clean up any previous keys. Use command:
$ ./easyrsa clean-all

```
[root@ip-172-31-23-212 easyrsa3]# ./easyrsa clean-all

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/vars

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/pki


[root@ip-172-31-23-212 easyrsa3]#
```

Now we can move on to building the certificate authority with the built-ca script. Use command:
$./ easyrsa build-ca

You will be asked to set a CA Key Passphrase and a common name for your CA. To skip this part use the following command instead of the command above:
$ ./easyrsa build-ca nopass

```
[root@ip-172-31-23-212 easyrsa3]# ./easyrsa build-ca nopass

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/vars
Using SSL: openssl OpenSSL 1.0.2k-fips  26 Jan 2017
Generating RSA private key, 2048 bit long modulus
.....................+++
...................+++
e is 65537 (0x10001)
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]:VPNSA

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/pki/ca.crt


[root@ip-172-31-23-212 easyrsa3]#
```

Create a key and certificate for the server:
$ ./easyrsa build-server-full server

You will be asked to fill in a passphrase.
Now let us generate a Diffie-Hellman key exchange file by using the command:
$ ./easyrsa gen-dh

We also need a certificate for each client. You can generate them on the server and then copy them to the client machine. Create a certificate and key for client1 (replace with any name) with the following command:
$ ./easyrsa build-client-full client1

```
[root@ip-172-31-23-212 easyrsa3]# ./easyrsa build-server-full server

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/vars
Using SSL: openssl OpenSSL 1.0.2k-fips  26 Jan 2017
Generating a 2048 bit RSA private key
.....................+++
.............................................+++
writing new private key to '/etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/pki/easy-rsa-463.n4fUMu/tmp.o7EEPh'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
Verify failure
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
Using configuration from /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/pki/easy-rsa-463.n4fUMu/tmp.Bvw66X
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'server'
Certificate is to be certified until Feb 24 21:17:49 2024 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

[root@ip-172-31-23-212 easyrsa3]#
```

```
.....+.............................+.....................................................
.........................+.....................................................................+..
.....................................................................................+.....
.....................................................+.....+.........................................
.....................................................................................+.................++*++*

DH parameters of size 2048 created at /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/pki/dh.pem


[root@ip-172-31-23-212 easyrsa3]#
```

```
[root@ip-172-31-23-212 easyrsa3]# ./easyrsa build-client-full Shagoefta

Note: using Easy-RSA configuration from: /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/vars
Using SSL: openssl OpenSSL 1.0.2k-fips  26 Jan 2017
Generating a 2048 bit RSA private key
..............................+++
..+++
writing new private key to '/etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/pki/easy-rsa-624.wDqF3a/tmp.q78VDD'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
Using configuration from /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/pki/easy-rsa-624.wDqF3a/tmp.Jk8GXK
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'Shagoefta'
Certificate is to be certified until Feb 24 21:29:17 2024 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

[root@ip-172-31-23-212 easyrsa3]#
```

After creating the keys and certificates, copy them from the pki directory into the openvpn directory. First navigate to the pki directory:

$ cd /etc/openvpn/easy-rsa/easyrsa3/pki

We need to copy the following 4 files:

- ca.crt
- dh.pem
- ca.key
- server.key

The first two files are stored in the pki directory and the other two files are in the subdirectory pki/private.

Copy ca.crt and dh.pem first, use command:
$ cp ca.crt dh.pem /etc/openvpn
Then move into the private directory:
$ cd private
Copy ca.key and server.key to the openvpn directory:
$ cp ca.key server.key /etc/openvpn

```
[root@ip-172-31-23-212 easyrsa3]# cd /etc/openvpn/easy-rsa/easy-rsa-3.0.8/easyrsa3/pki
[root@ip-172-31-23-212 pki]# ls
ca.crt           index.txt          index.txt.old       private  revoked              serial.old
certs_by_serial  index.txt.attr     issued              renewed  safessl-easyrsa.cnf
dh.pem           index.txt.attr.old  openssl-easyrsa.cnf reqs     serial
[root@ip-172-31-23-212 pki]# cp ca.crt dh.pem /etc/openvpn
[root@ip-172-31-23-212 pki]# cd private
[root@ip-172-31-23-212 private]# ls
ca.key  server.key  Shagoefta.key
[root@ip-172-31-23-212 private]# cp ca.key server.key /etc/openvpn
[root@ip-172-31-23-212 private]#
```

10. Now we have generated the keys and certificates, we are moving on to the next step. The firewall and routing configuration. At first we are going to install the firewall package by using command:
$ yum install -y firewalld

```
Installed:
  firewalld.noarch 0:0.4.4.4-6.amzn2.0.1

Dependency Installed:
  dbus-glib.x86_64 0:0.100-7.2.amzn2                    dbus-python.x86_64 0:1.1.1-9.amzn2.0.2
  ebtables.x86_64 0:2.0.10-16.amzn2.0.1                 firewalld-filesystem.noarch 0:0.4.4.4-6.amzn2.0.1
  gobject-introspection.x86_64 0:1.56.1-1.amzn2         ipset.x86_64 0:6.29-1.amzn2.0.1
  ipset-libs.x86_64 0:6.29-1.amzn2.0.1                  libselinux-python.x86_64 0:2.5-12.amzn2.0.2
  python-decorator.noarch 0:3.4.0-3.amzn2               python-firewall.noarch 0:0.4.4.4-6.amzn2.0.1
  python-gobject-base.x86_64 0:3.22.0-1.amzn2.1         python-slip.noarch 0:0.4.0-4.amzn2
  python-slip-dbus.noarch 0:0.4.0-4.amzn2

Complete!
[root@ip-172-31-23-212 ~]#
```

When the installation is completed you need to enable and start the firewall service. Use the following commands:
$ systemctl enable firewalld
$ systemctl start firewalld

Then you can check the status by using command:
$ systemctl status firewalld

```
[root@ip-172-31-23-212 ~]# systemctl enable firewalld
[root@ip-172-31-23-212 ~]# systemctl start firewalld
[root@ip-172-31-23-212 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-11-21 21:49:22 UTC; 21s ago
     Docs: man:firewalld(1)
 Main PID: 866 (firewalld)
   CGroup: /system.slice/firewalld.service
           └─866 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

Nov 21 21:49:22 ip-172-31-23-212.us-west-2.compute.internal systemd[1]: Starting firewalld - dynamic firewall daemon...
Nov 21 21:49:22 ip-172-31-23-212.us-west-2.compute.internal systemd[1]: Started firewalld - dynamic firewall daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-23-212 ~]#
```

Once you made sure you have an active running firewall, we need to add the openvpn service to the list of services firewalld allows in the active zone. check the active zones by using command:

$ firewall-cmd –get-active-zone

In this case there was no output. So in the following command add the zone and directly add the openvpn service to the zone too:

$ firewall-cmd –zone=public –add-service openvpn

Then make it permanent:

$ firewall-cmd –zone=public –add-service openvpn --permanent

Check if the service is successfully added:

$ firewall-cmd –list-services –zone=public

```
[root@ip-172-31-23-212 ~]# firewall-cmd --get-active-zone
[root@ip-172-31-23-212 ~]# firewall-cmd --zone=public --add-service openvpn
success
[root@ip-172-31-23-212 ~]# firewall-cmd --zone=public --add-service openvpn --permanent
success
[root@ip-172-31-23-212 ~]# firewall-cmd --list-services --zone=public
ssh dhcpv6-client openvpn
[root@ip-172-31-23-212 ~]#
```

Then add a masquerade to the runtime instance:

$ firewall-cmd --add-masquerade

Make it permanent:

$ firewall-cmd --add-masquerade –permanent

And verify if the masquerade was added:

$ firewall-cmd --query-masquerade

```
[root@ip-172-31-23-212 ~]# firewall-cmd --add-masquerade
success
[root@ip-172-31-23-212 ~]# firewall-cmd --add-masquerade --permanent
success
[root@ip-172-31-23-212 ~]# firewall-cmd --query-masquerade
yes
[root@ip-172-31-23-212 ~]#
```

11. Now move on to routing to your OpenVPN subnet. Create a variable for the primary network interface used by your server. In the following command the variable is named VAR.

    $ VAR=$(ip route get 208.67.222.222 | awk 'NR==1 {print $(NF-2)}')

```
[root@ip-172-31-23-212 ~]# VAR=$(ip route get 208.67.222.222 | awk 'NR==1 {print $(NF-2)}')
[root@ip-172-31-23-212 ~]#
```

    Then permanently add the routing rule using the created variable:

    $ firewall-cmd --permanent --direct --passthrough ipv4 -t nat -A POSTROUTING -s 10.8.0.0/24 -o $VAR -j MASQUERADE

```
[root@ip-172-31-23-212 ~]# firewall-cmd --permanent --direct --passthrough ipv4 -t nat -A POSTROUTING -s 10.8.0.0/24 -o
$VAR -j MASQUERADE
success
[root@ip-172-31-23-212 ~]#
```

    Then reload firewalld so the changes can take place:

    $ firewall-cmd –reload

    Move on to routing all web traffic from the client to the server's IP address by enabling IP forwarding. Open the sysctl.conf file:

    $ vi /etc/sysctl.conf

    Add the following line at the top of the file:

    net.ipv4.ip_forward = 1

```
net.ipv4.ip_forward = 1

# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
~
```

    Now restart and check the service:

    $ systemctl restart network.service:

    $ systemctl status network.service:

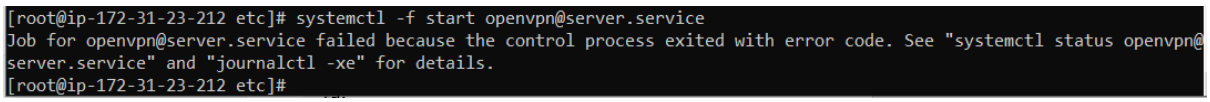
```
[root@ip-172-31-23-212 etc]# systemctl restart network.service
[root@ip-172-31-23-212 etc]# systemctl status network.service
● network.service - LSB: Bring up/down networking
   Loaded: loaded (/etc/rc.d/init.d/network; bad; vendor preset: disabled)
   Active: active (running) since Sun 2021-11-21 22:42:53 UTC; 16s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 1212 ExecStop=/etc/rc.d/init.d/network stop (code=exited, status=0/SUCCESS)
  Process: 1389 ExecStart=/etc/rc.d/init.d/network start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/network.service
           ├─1564 /sbin/dhclient -q -lf /var/lib/dhclient/dhclient--eth0.lease -pf /var/run/dhclient-eth0.pid -H ip-1...
           └─1613 /sbin/dhclient -6 -nw -lf /var/lib/dhclient/dhclient6--eth0.lease -pf /var/run/dhclient6-eth0.pid e...

Nov 21 22:42:53 ip-172-31-23-212.us-west-2.compute.internal ec2net[1582]: [get_meta] Trying to get http://169.254.1...4s
Nov 21 22:42:53 ip-172-31-23-212.us-west-2.compute.internal ec2net[1589]: [remove_aliases] Removing aliases of eth0
Nov 21 22:42:53 ip-172-31-23-212.us-west-2.compute.internal network[1389]: Determining IPv6 information for eth0.....e.
Nov 21 22:42:53 ip-172-31-23-212.us-west-2.compute.internal network[1389]: [  OK  ]
Nov 21 22:42:53 ip-172-31-23-212.us-west-2.compute.internal systemd[1]: Started LSB: Bring up/down networking.
Nov 21 22:42:53 ip-172-31-23-212.us-west-2.compute.internal dhclient[1613]: XMT: Solicit on eth0, interval 1080ms.
Nov 21 22:42:54 ip-172-31-23-212.us-west-2.compute.internal dhclient[1613]: XMT: Solicit on eth0, interval 2120ms.
Nov 21 22:42:56 ip-172-31-23-212.us-west-2.compute.internal dhclient[1613]: XMT: Solicit on eth0, interval 4160ms.
Nov 21 22:43:00 ip-172-31-23-212.us-west-2.compute.internal dhclient[1613]: XMT: Solicit on eth0, interval 7940ms.
Nov 21 22:43:08 ip-172-31-23-212.us-west-2.compute.internal dhclient[1613]: XMT: Solicit on eth0, interval 16190ms.
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-23-212 etc]#
```

12. Now it's time to start the OpenVPN and check if it is working.
    To start the OpenVPN service, run the command:
    $ systemctl -f start openvpn@server.service

    Then, enable it to start up at boot by running:
    $ systemctl -f enable openvpn@server.service

    Verify the service is active with:
    $ systemctl status openvpn@server.service



    Unfortunately I was not able to start the OpenVPN service. I tried to find out how to solve the error, but without success till now. This document will be updated after the solution is found!