

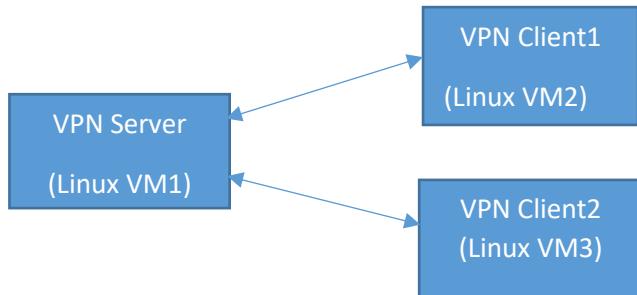
VPN LAB

Introduction:

This lab is a VPN setup containing a server authenticating a connection and clients creating a connection with server over the tunnel. For this lab, I have set up 3 VMs to carry out this implementation.

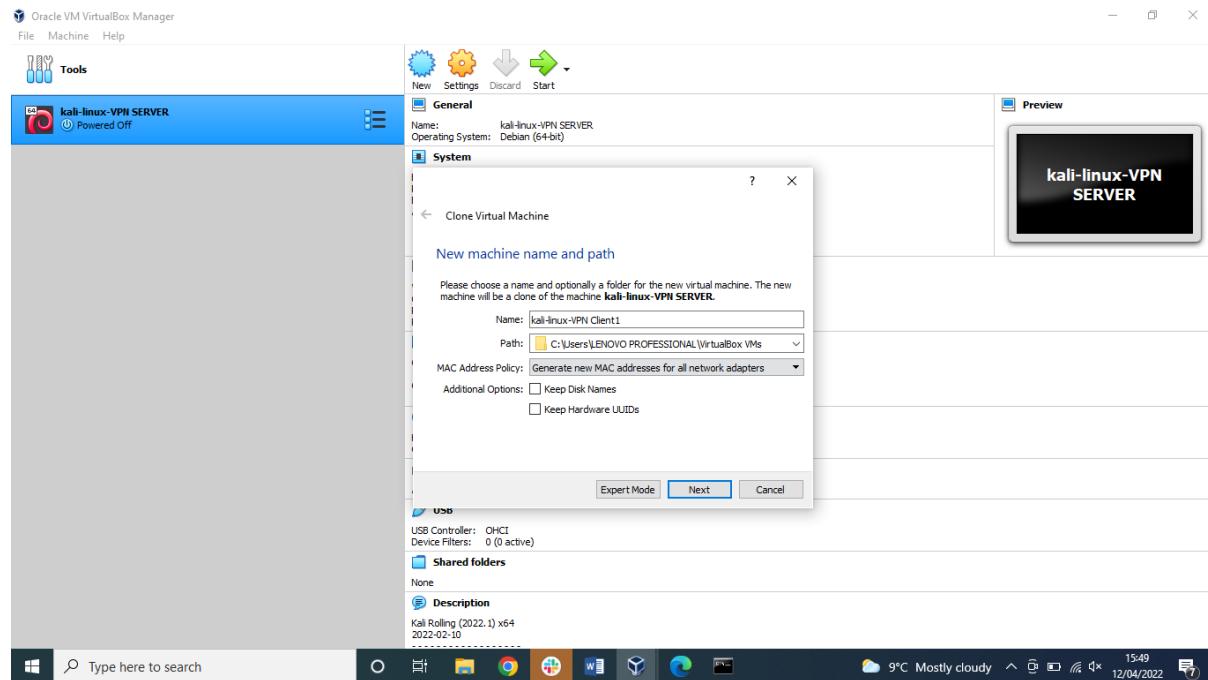
Design:

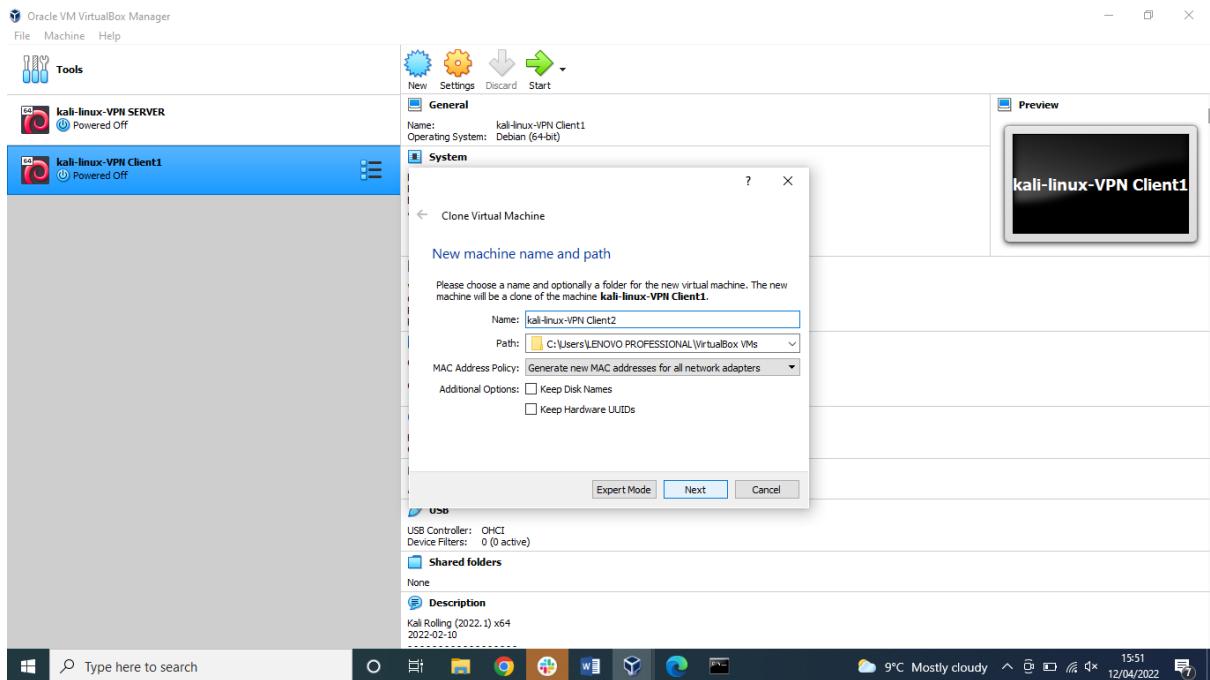
The diagram illustrating the configuration is presented below:



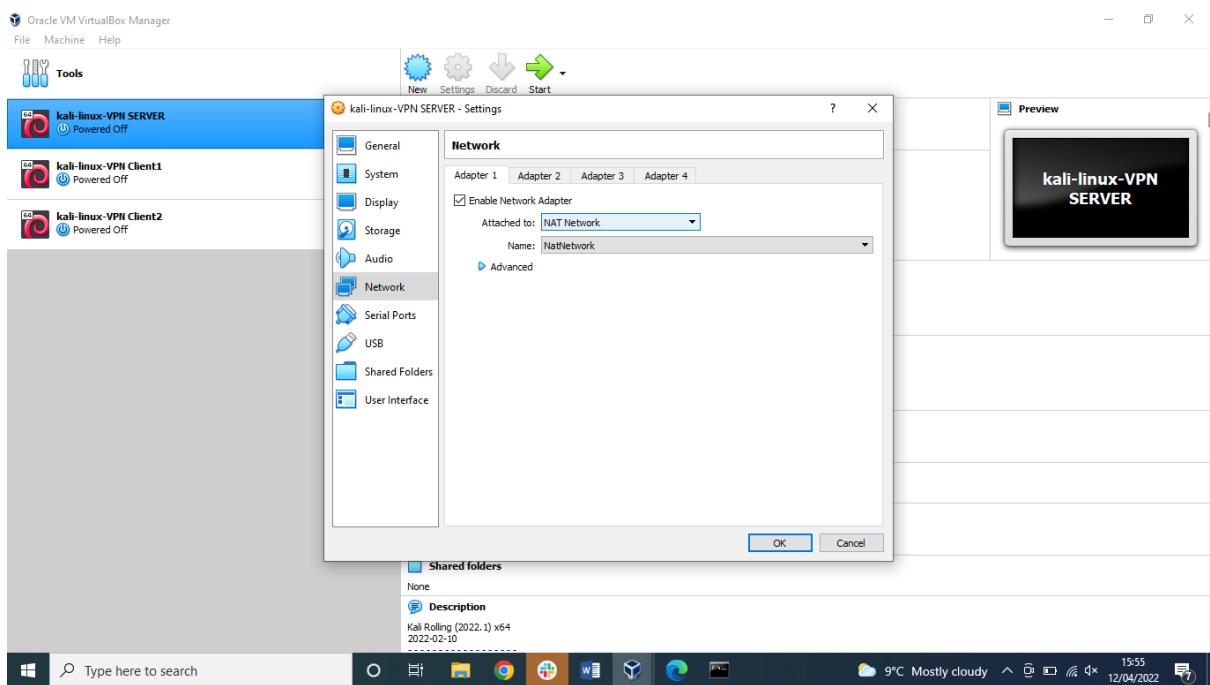
Step 1

Create 3 Kali-Linux VMs as shown in the figure above (When cloning ensure you select the generate new MAC addresses for network adapter so that we don't get the same IP address on all the VMs)





Step 2: Set up the network adapter. Use NAT Network adapter for all VMs in this setup



Step3 Power the VMs

Step 4: Install OpenVPN on the VMs- Confirm if OpenVPN is preinstalled on your version of kali-linux by typing the command below

```
# apt-get update
```

```
# apt-get install openvpn
```

kali-linux-VPN SERVER [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

kali@kali:~

```

File Actions Edit View Help
(kali㉿kali)-[~]
$ sudo apt-get update
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [30.6 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [18.2 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [41.9
MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [15
5 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [212 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [1
,004 kB]
Fetched 61.6 MB in 39s (1,568 kB/s)
Reading package lists... Done

(kali㉿kali)-[~]
$ sudo apt-get install openvpn
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  resolvconf openvpn-systemd-resolved
Recommended packages:
  easy-rsa

```

16:10 12/04/2022

Next install easy-rsa, which installs a set of script used for Public Key Infrastructure (PKI) management.

apt-get install easy-rsa

kali-linux-VPN SERVER [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

kali@kali:~

```

File Actions Edit View Help
openvpn.service is a disabled or a static unit not running, not starting it.
Processing triggers for kali-menu (2021.4.2) ...
Processing triggers for man-db (2.9.4-4) ...
(kali㉿kali)-[~]
$ sudo apt-get install easy-rsa
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  liblzcid libpccslite1 opensc opensc-pkcs11 pscd
Suggested packages:
  pscd
The following NEW packages will be installed:
  easy-rsa liblzcid opensc opensc-pkcs11 pscd
The following packages will be upgraded:
  liblzcid libpccslite1 opensc opensc-pkcs11 pscd
1 upgraded, 5 newly installed, 0 to remove and 825 not upgraded.
Need to get 1,830 kB of additional disk space will be used.
After this operation, 5,633 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kali.download/kali kali-rolling/main amd64 libpccslite1 amd64 1.9.5-3 [80.1 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 liblzcid amd64 1.5.0-2 [360 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 pscd amd64 1.9.5-3 [93.9 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 opensc amd64 0.22.0-2 [892 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 opensc-pkcs11 amd64 0.22.0-2 [892 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 opensc-amd64 0.22.0-2 [374 kB]
Fetched 1,830 kB in 2s (1,108 kB/s)
(Reading database ... 2,000,000 files and directories currently installed.)
Preparing to unpack .../libpccslite1_1.9.5-3_amd64.deb ...
Unpacking libpccslite1:amd64 (1.9.5-3) over (1.9.5-1) ...
Selecting previously unselected package liblzcid.
Preparing to unpack .../liblzcid_1.5.0-2_amd64.deb ...
Unpacking liblzcid (1.5.0-2) ...
Selecting previously unselected package pscd.
Preparing to unpack .../pscfd_1.9.5-3_amd64.deb ...
Unpacking pscd (1.9.5-3) ...
Selecting previously unselected package easy-rsa.
Preparing to unpack .../easy-rsa_3.0.8-1_all.deb ...
Unpacking easy-rsa (3.0.8-1) ...
Selecting previously unselected package opensc-pkcs11:amd64.
Preparing to unpack .../opensc-pkcs11_0.22.0-2_amd64.deb ...
Unpacking opensc-pkcs11:amd64 (0.22.0-2) ...
Selecting previously unselected package opensc.
Preparing to unpack .../opensc_0.22.0-2_amd64.deb ...
Unpacking opensc (0.22.0-2) ...

```

16:15 12/04/2022

Note: Perform the same commands in step 4 on the other VMs

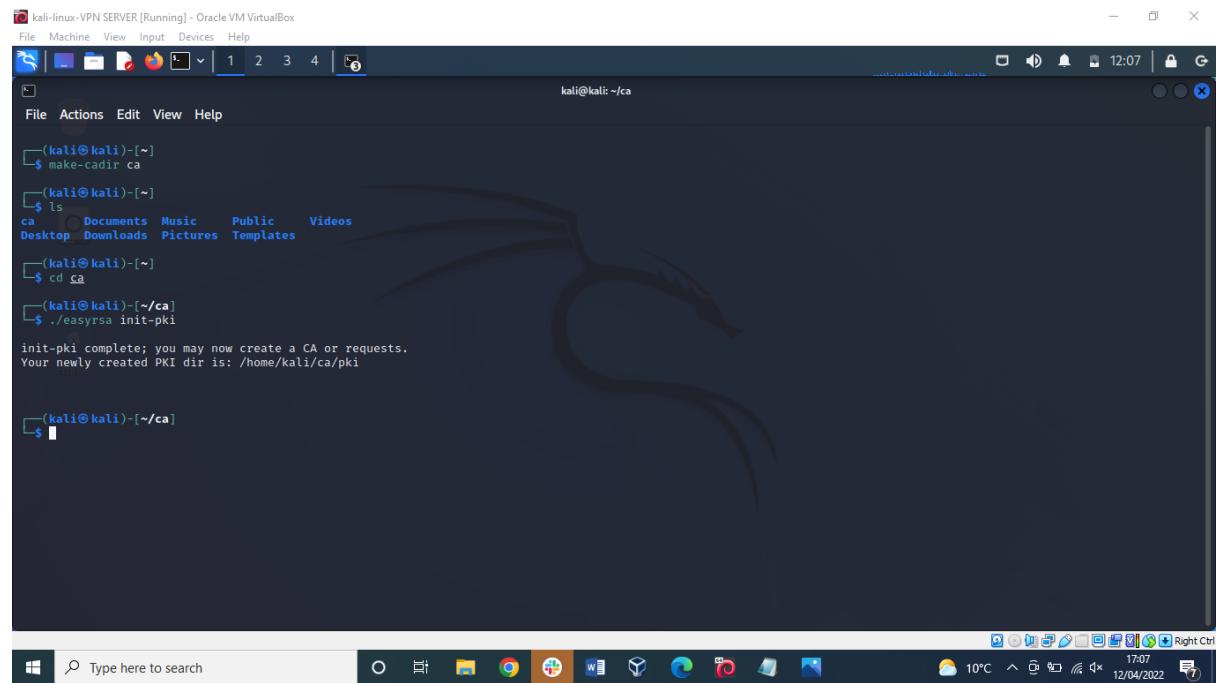
Step 5: Set up the Certificate Authority (CA) and generate certificates and keys for servers and the clients by carrying out the follow steps on one VM (here we are using the VPN server)

1. Create a new PKI

\$ make-cadir ca

```
$cd ca
```

```
$./easysrsa init-pki
```

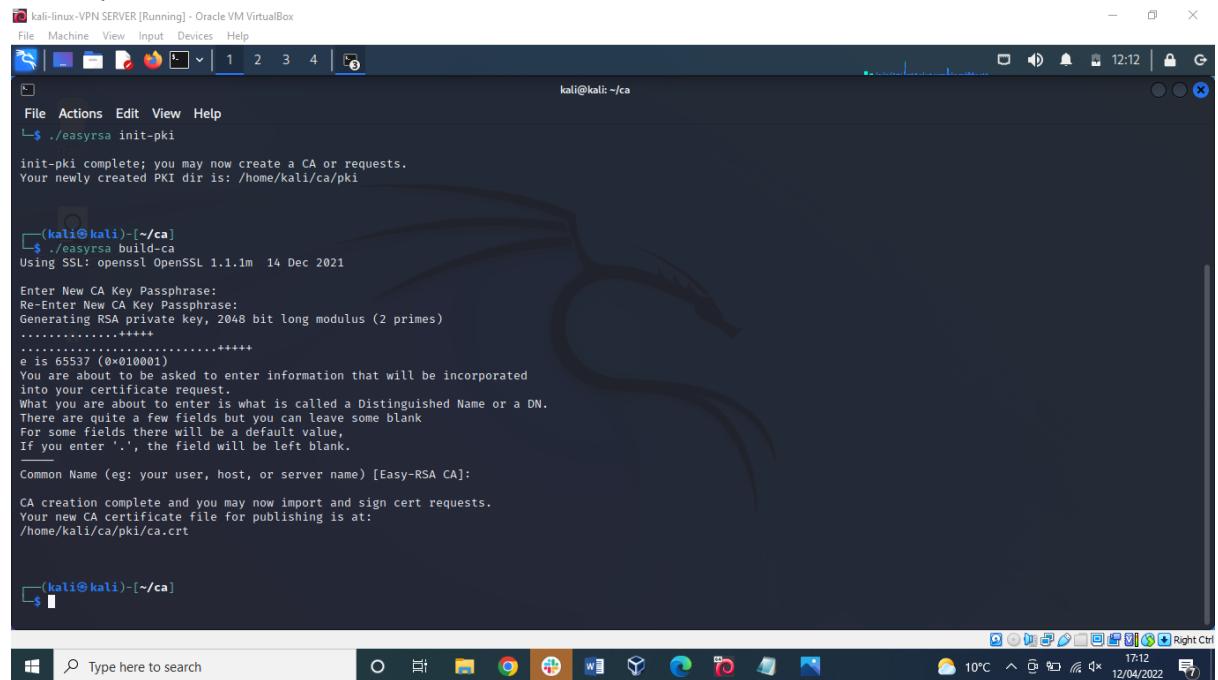


```
kali@kali: ~]$ make-cadir ca
(kali㉿kali)-[~]
$ ls
ca  Documents  Music  Public  Videos
Desktop  Downloads  Pictures  Templates
(kali㉿kali)-[~]
$ cd ca
(kali㉿kali)-[~/ca]
$ ./easysrsa init-pki
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/kali/ca/pki

(kali㉿kali)-[~/ca]
```

2. Build the CA, set up the passphrase for the CA

```
$ ./easysrsa build-ca
```



```
kali@kali: ~]$ ./easysrsa init-pki
init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: /home/kali/ca/pki

(kali㉿kali)-[~/ca]
$ ./easysrsa build-ca
Using SSL: openssl OpenSSL 1.1.1m 14 Dec 2021

Enter New CA Key Passphrase:
Re-Enter New CA Key Passphrase:
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
e is 55537 (0x010001)
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]:
CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/home/kali/ca/pki/ca.crt

(kali㉿kali)-[~/ca]
```

3. Generate certificate and private key for server

```
$ ./easysrsa build-server-full server
```

```
(kali㉿kali)-[~/ca]
$ ./easysrsa build-server-full server
Using SSL: openssl OpenSSL 1.1.1m 14 Dec 2021
Generating a RSA private key
+++++
writing new private key to '/home/kali/ca/pki/easy-rsa-19990.KA3yEE/tmp.q50keaa'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

Using configuration from /home/kali/ca/pki/easy-rsa-19990.KA3yEE/tmp.VH5JxW
Enter pass phrase for /home/kali/ca/pki/private/ca.key:
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 Jul 15 16:16:05 2024 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

(kali㉿kali)-[~/ca]
$
```

4. Generate certificate and private –key for client1

\$./easysrsa build-client-full client1

```
(kali㉿kali)-[~/ca]
$ ./easysrsa build-client-full client1
Using SSL: openssl OpenSSL 1.1.1m 14 Dec 2021
Generating a RSA private key
+++++
.....+=====
writing new private key to '/home/kali/ca/pki/easy-rsa-21548.oAIYEK/tmp.VxhZYw'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

Using configuration from /home/kali/ca/pki/easy-rsa-21548.oAIYEK/tmp.ARlkvt
Enter pass phrase for /home/kali/ca/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'client1'
Certificate is to be certified until Jul 17 13:13:34 2024 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

(kali㉿kali)-[~/ca]
$
```

5. Generate certificate and private –key for client2

\$./easysrsa build-client-full client2

kali-linux-VPN SERVER [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
(kali㉿kali)-[~/ca]
$ ./easyrsa build-client-full client2
Using SSL: openssl OpenSSL 1.1.1m 14 Dec 2021
Generating a RSA private key
.....................................................................+=====
writing new private key to '/home/kali/ca/pki/easy-rsa-22393.9T0vX2/tmp.AhY46Y'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
_____
Using configuration from /home/kali/ca/pki/easy-rsa-22393.9T0vX2/tmp.YTaSEQ
Enter pass phrase for /home/kali/ca/pki/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :ASN.1 12:'client2'
Certificate is to be certified until Jul 17 13:16:13 2024 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

(kali㉿kali)-[~/ca]
$
```

6. Generate DH (Diffie Hellman) parameters

```
$ ./easyrsa gen-dh
```

7. Generate TLS authentication key

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

```

kali-linux-VPN SERVER [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
.
.
.
DH parameters of size 2048 created at /home/kali/ca/pki/dh.pem
(kali㉿kali)-[~/ca]
$ openvpn --genkey secret ta.key
(kali㉿kali)-[~/ca]
$ ls
easyrsa openssl-easyrsa.cnf pki ta.key vars x509-types
(kali㉿kali)-[~/ca]
$ 
(kali㉿kali)-[~/ca]
$ 
(kali㉿kali)-[~/ca]
$ 

```

Step 6: Copy the generated files

1. Copy the files below from ca to your /etc/openvpn/server on the server VM

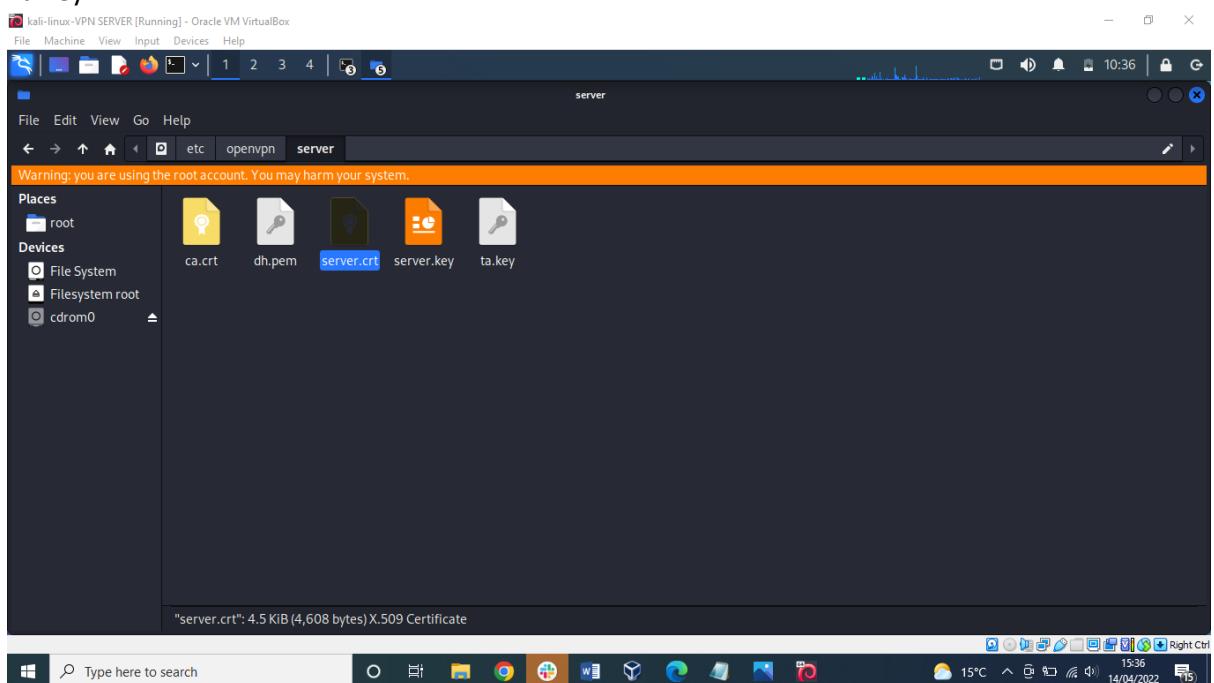
Ca.crt

Dh.pem

Server.crt

Server.key

Ta.key



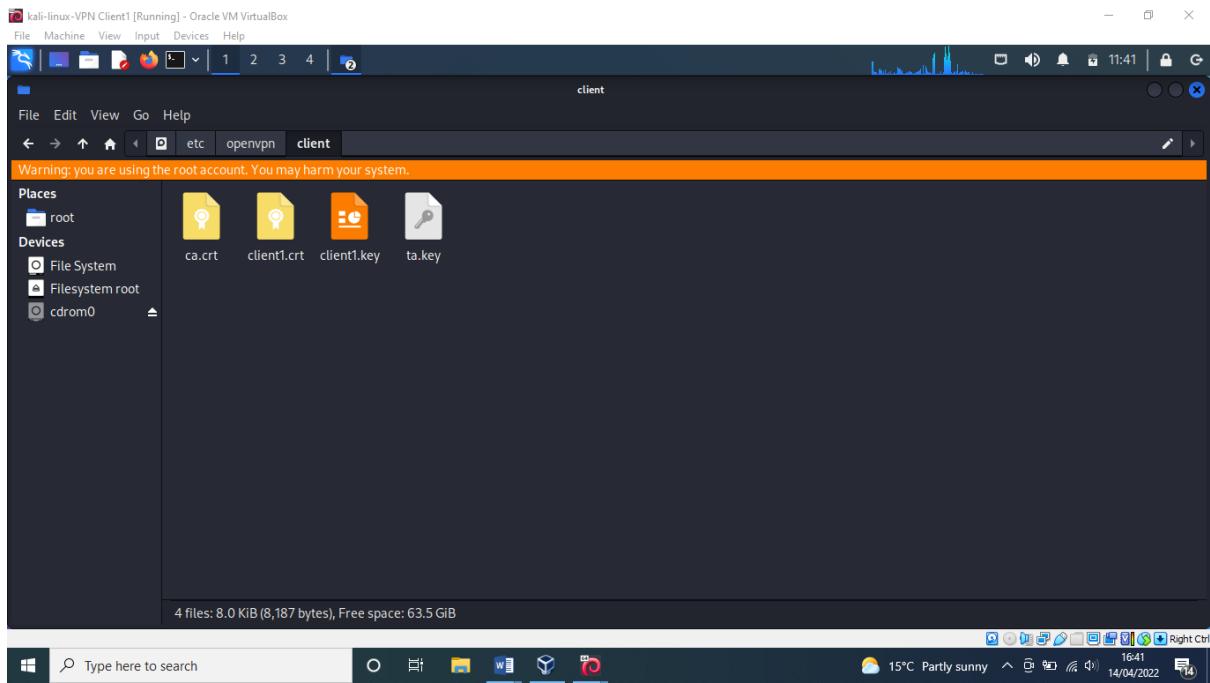
2. Copy the files below from ca to your /etc/openvpn/client on the client1 VM

Ca.crt

Client1.crt

Client1.key

Ta.key



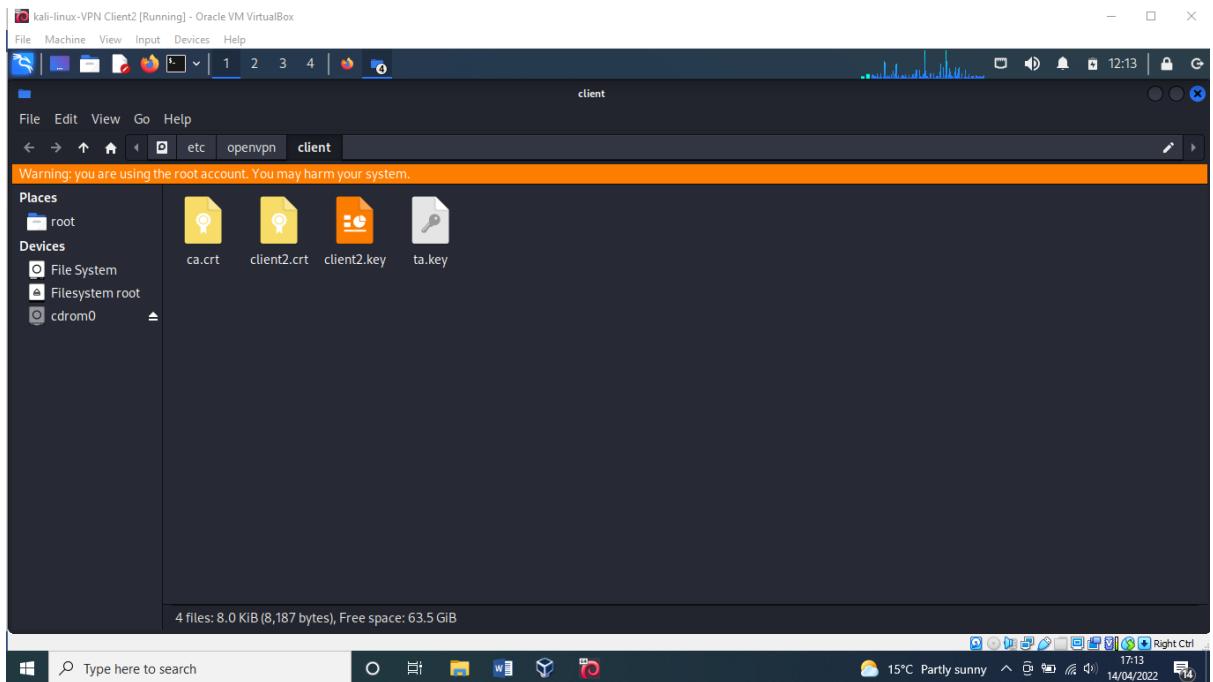
3. Copy files below from ca to your /etc/openvpn/client on the client2 VM

Ca.crt

Client2.crt

Client2.key

Ta.key



Step 7: Edit server config

Type in the commands as shown on screenshot

```
$ cd /etc/openvpn/server
```

```
$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf server.conf
```

```
(kali㉿kali)-[~]
$ cd /etc/openvpn/server
(kali㉿kali)-[/etc/openvpn/server]
$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf server.conf
[sudo] password for kali:
(kali㉿kali)-[/etc/openvpn/server]
$ nano server.conf
(kali㉿kali)-[/etc/openvpn/server]
$ nano server.conf
(kali㉿kali)-[/etc/openvpn/server]
```

To edit the server.conf file it does not allow us to write, so we need to change the permissions on the file

```
kali@kali: /etc/openvpn/server
File Machine View Input Devices Help
└$ sudo chmod 775 server.conf
(kali㉿kali)-[~/etc/openvpn/server]
└$ nano server.conf
(kali㉿kali)-[~/etc/openvpn/server]
└$ sudo chmod 775 -R /etc/openvpn
(kali㉿kali)-[~/etc/openvpn/server]
└$ nano server.conf
(kali㉿kali)-[~/etc/openvpn/server]
└$ vi server.conf
(kali㉿kali)-[~/etc/openvpn/server]
└$ sudo chmod 777 -R /etc/openvpn
(kali㉿kali)-[~/etc/openvpn/server]
└$ nano server.conf
(kali㉿kali)-[~/etc/openvpn/server]
└$ vi server.conf
(kali㉿kali)-[~/etc/openvpn/server]
└$ 
(kali㉿kali)-[~/etc/openvpn/server]
└$ 
(kali㉿kali)-[~/etc/openvpn/server]
```

Now you can edit the server.conf

Step 8 Edit client1 config

First confirm the IP address of the server using ifconfig on the VPN server VM (Here it is: 10.0.2.4)

```
$ cd /etc/openvpn/client
```

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

```
kali@kali: /etc/openvpn/client
File Machine View Input Devices Help
└$ cd /etc/openvpn/client
(kali㉿kali)-[~/etc/openvpn/client]
└$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf client.conf
[sudo] password for kali:
(kali㉿kali)-[~/etc/openvpn/client]
└$ vi client.conf
(kali㉿kali)-[~/etc/openvpn/client]
└$ sudo chmod 777 -R /etc/openvpn
(kali㉿kali)-[~/etc/openvpn/client]
└$ vi client.conf
(kali㉿kali)-[~/etc/openvpn/client]
└$ 
(kali㉿kali)-[~/etc/openvpn/client]
```

Now edit the rules

Step 9 Edit client2 config

First confirm the IP address of the server using ifconfig on the VPN server VM (Here it is: 10.0.2.4)

```
$ cd /etc/openvpn/client
```

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

The screenshot shows a terminal window titled 'kali@kali: /etc/openvpn/client'. The user runs several commands to prepare the client configuration:

```
(kali㉿kali)-~$ cd /etc/openvpn/client
(kali㉿kali)-~/etc/openvpn/client$ sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf client.conf
[sudo] password for kali:
(kali㉿kali)-~/etc/openvpn/client$ sudo chmod 777 -R /etc/openvpn
(kali㉿kali)-~/etc/openvpn/client$ ls
```

Step 10: Start VPN server and clients

For Linux VPN server

The screenshot shows a terminal window titled 'root@kali: ~'. The user runs the ifconfig command to check the network interfaces:

```
root@kali: /etc/openvpn/server$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.4 brd 10.0.2.255 netmask 255.255.255.0 broadcast 10.0.2.255
        ether 08:00:27:95:bd:54 txqueuelen 1000 (Ethernet)
            RX packets 103 bytes 15186 (14.8 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 52 bytes 7335 (7.1 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 brd 127.0.0.1 netmask 255.0.0.0
        ether 00:00:00:00:00:00 txqueuelen 1000 (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 10.8.0.1 brd 10.8.0.2 netmask 255.255.255.252 destination 10.8.0.2
        ether fe80::ec22:3188%tun0 brd fe80::ff:fe22:3188%tun0 txqueuelen 500 (UNSPEC)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 3 bytes 144 (144.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

For the Clients; see screenshot below showing that the VPN has started

For Client 1 , Add the default route to ensure that it uses the server IP 10.0.2.4 as gateway using the command below

```
kali-linux-VPN Client1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
[root@kali:~]
# route add default gw 10.0.2.4
[root@kali:~]
# route
Kernel IP routing table (mpnypn/31:root)
Destination Gateway Gemask Flags Metric Ref Use Iface
default 10.0.2.4 0.0.0.0 UG 0 0 0 eth0
default 10.0.2.1 mpnypn/0.0.0.0 UG 100 0 0 eth0
10.0.2.0 0.0.0.0 255.255.255.0 U 100 0 0 eth0

[root@kali:~]
# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:7f:f1:01
          inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
              inet 10.0.2.1 brd 10.0.2.1 mcast 224.0.2.151
                  bcast 10.0.2.15
          inet 127.0.0.1 netmask 255.255.255.0
              loop  Link encap:Loopback
          inet 127.0.0.1 brd 127.0.0.1
                  broadcast 127.0.0.1
          RX packets 0 bytes 0 (0.0 B)
          TX packets 0 bytes 0 (0.0 B)
          errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo       Link encap:Local Loopback
          inet 127.0.0.1 netmask 255.255.255.0
              loop  Link encap:Local Loopback
          inet 127.0.0.1 brd 127.0.0.1
                  broadcast 127.0.0.1
          RX packets 0 bytes 0 (0.0 B)
          TX packets 0 bytes 0 (0.0 B)
          errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@kali:~]
# /etc/openvpn/client

Windows Taskbar:
Type here to search
15°C Partly sunny 19:07
14/04/2022
```

We would need to delete the 10.0.2.1 route from the route table

```
kali-linux-VPN Client1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
[root@kali:~]
# route del default gw 10.0.2.1
[root@kali:~]
# route
Kernel IP routing table (mpnypn/31:root)
Destination Gateway Gemask Flags Metric Ref Use Iface
default 10.0.2.1 0.0.0.0 UG 100 0 0 eth0
10.0.2.0 0.0.0.0 255.255.255.0 U 100 0 0 eth0

[root@kali:~]
# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:7f:f1:01
          inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
              inet 10.0.2.1 brd 10.0.2.1 mcast 224.0.2.151
                  bcast 10.0.2.15
          inet 127.0.0.1 netmask 255.255.255.0
              loop  Link encap:Loopback
          inet 127.0.0.1 brd 127.0.0.1
                  broadcast 127.0.0.1
          RX packets 0 bytes 0 (0.0 B)
          TX packets 0 bytes 0 (0.0 B)
          errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo       Link encap:Local Loopback
          inet 127.0.0.1 netmask 255.255.255.0
              loop  Link encap:Local Loopback
          inet 127.0.0.1 brd 127.0.0.1
                  broadcast 127.0.0.1
          RX packets 0 bytes 0 (0.0 B)
          TX packets 0 bytes 0 (0.0 B)
          errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@kali:~]
# /etc/openvpn/client

Windows Taskbar:
Type here to search
15°C Partly sunny 19:12
14/04/2022
```

Now start the VPN client on Client1

```

root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
                inet6 fe80::2a0:2ff:fe00:15%eth0  brd fe80::ff:fe00:15
                        ether 08:00:27:b7:84:b1  txqueuelen 1000  (Ethernet)
                        RX packets 187  bytes 29744 (29.9 KiB)
                        RX errors 0  dropped 0  overruns 0  frame 0
                        TX packets 196  bytes 26216 (25.6 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 1000  (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 10.8.0.6  netmask 255.255.255.255  destination 10.8.0.5
                inet6 fe80::2aa:2ff:fe00:6%tun0  brd fe80::ff:fe00:6
                        ether fe80::2aa:2ff:fe00:6  txqueuelen 500  (UNSPEC)
                        RX packets 0  bytes 0 (0.0 B)
                        RX errors 0  dropped 0  overruns 0  frame 0
                        TX packets 3  bytes 144 (144.0 B)
                        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

```

Step 11: Test VPN connection

Ping 10.8.0.1

```

root@kali:~# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=0.876 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=1.44 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=0.875 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=64 time=1.50 ms
64 bytes from 10.8.0.1: icmp_seq=5 ttl=64 time=0.793 ms
64 bytes from 10.8.0.1: icmp_seq=6 ttl=64 time=0.761 ms
64 bytes from 10.8.0.1: icmp_seq=7 ttl=64 time=1.58 ms
64 bytes from 10.8.0.1: icmp_seq=8 ttl=64 time=0.729 ms
64 bytes from 10.8.0.1: icmp_seq=9 ttl=64 time=1.79 ms
64 bytes from 10.8.0.1: icmp_seq=10 ttl=64 time=1.17 ms
64 bytes from 10.8.0.1: icmp_seq=11 ttl=64 time=1.54 ms
64 bytes from 10.8.0.1: icmp_seq=12 ttl=64 time=1.51 ms
64 bytes from 10.8.0.1: icmp_seq=13 ttl=64 time=1.76 ms

```

Step 12: Set IP forwarding and masquerading (On Linux VPN server) – These commands show below are important as it enables access to the internet from the VPN client

On the VPN server , type:

```
# echo "1" >/proc/sys/net/ipv4/ip_forward
# iptables -t NAT -A POSTROUTING -o eth0 -j MASQUERADE
```

Then check the route

```
root@kali:~# rtt min/avg/max/mdev = 0.355/0.663/3.957/0.365 ms
[root@kali]# echo "1" > /proc/sys/net/ipv4/ip_forward
[root@kali]# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
[root@kali]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe95:bd54 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:95:bd:54 brd ff:ff:ff:ff:ff:ff
            RX packets 40020 bytes 5724600 (5.4 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 35396 bytes 4982600 (4.7 MB)
            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 1000 (Local Loopback)
            RX packets 53 bytes 3832 (3.7 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 53 bytes 3832 (3.7 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.8.0.1 netmask 255.255.255.255 destination 10.8.0.2
        inet6 fe80::c2146:b904:4722 prefixlen 64 scopeid 0x20<link>
            unscope=00:00:00:00:00:00 txqueuelen 500 (UNSPEC)
            RX packets 21697 bytes 1608965 (1.7 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 16222 bytes 2042839 (1.9 MB)
```

```
# route
```

Step 13: Verify that VPN is encrypted

Use Wireshark to capture traffic on the VPN server or on one client. For example, open a web page in the client and look at the traffic on eth0. The protocol of the packets should be OpenVPN: