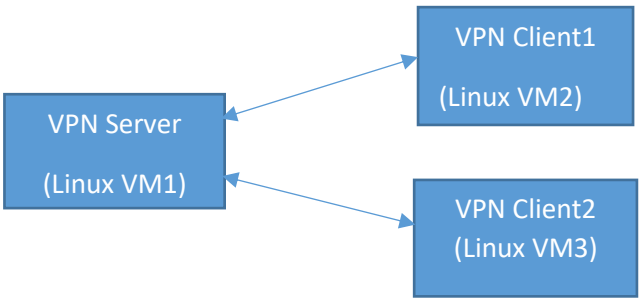
VPN LAB

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 carried pout this implementation.

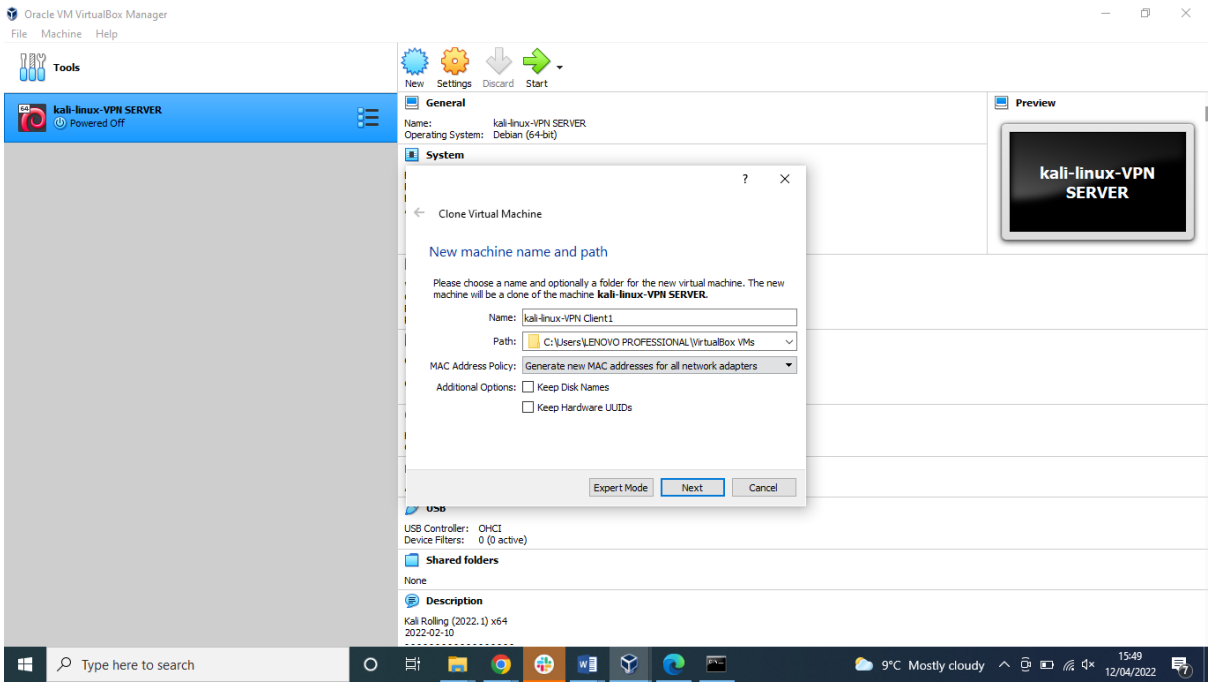
Design:

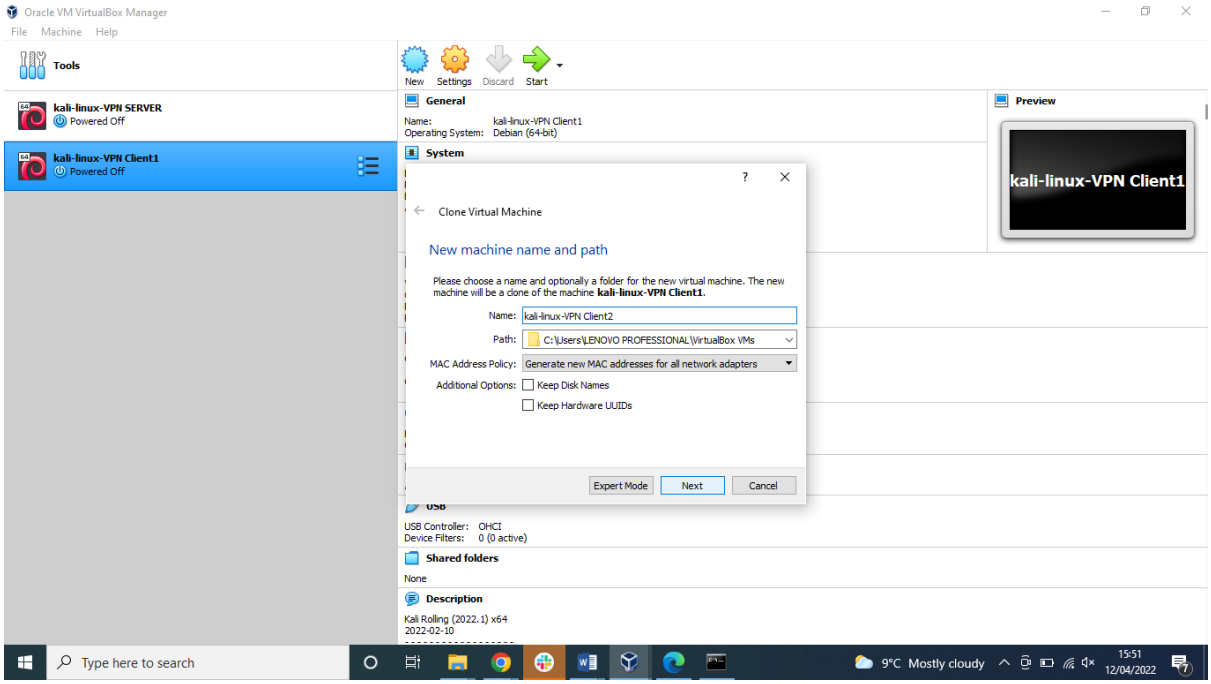
The diagram illustrating the configuration is presented below:



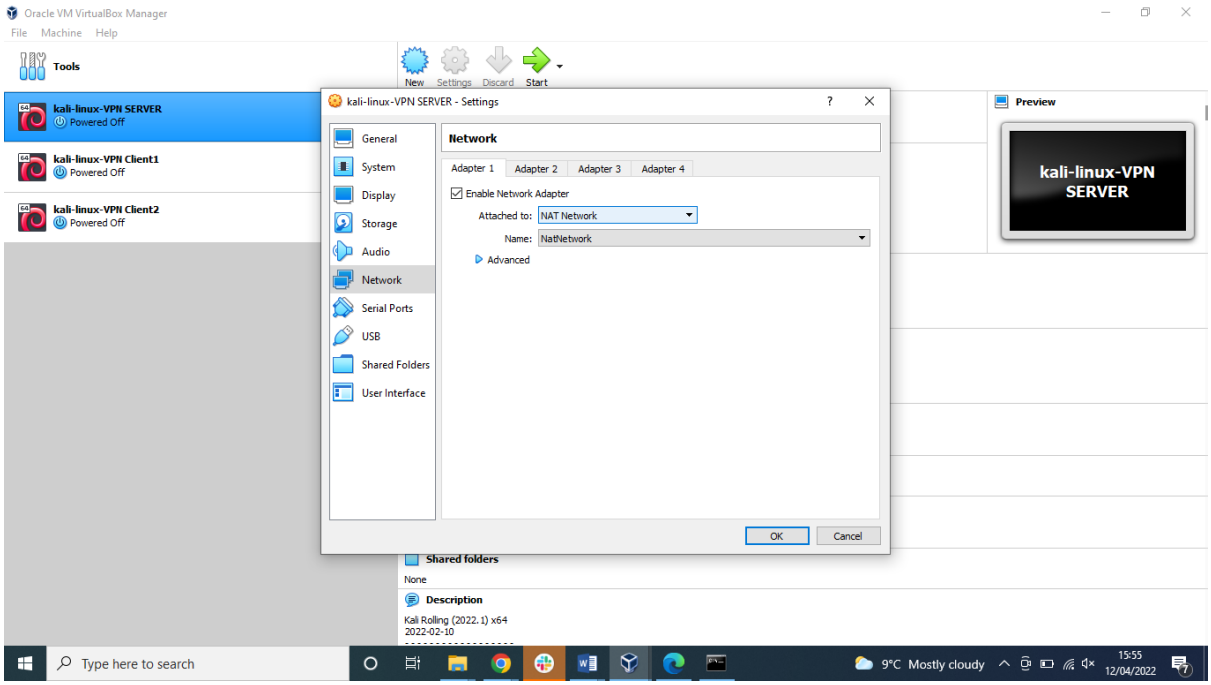
Step 1

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

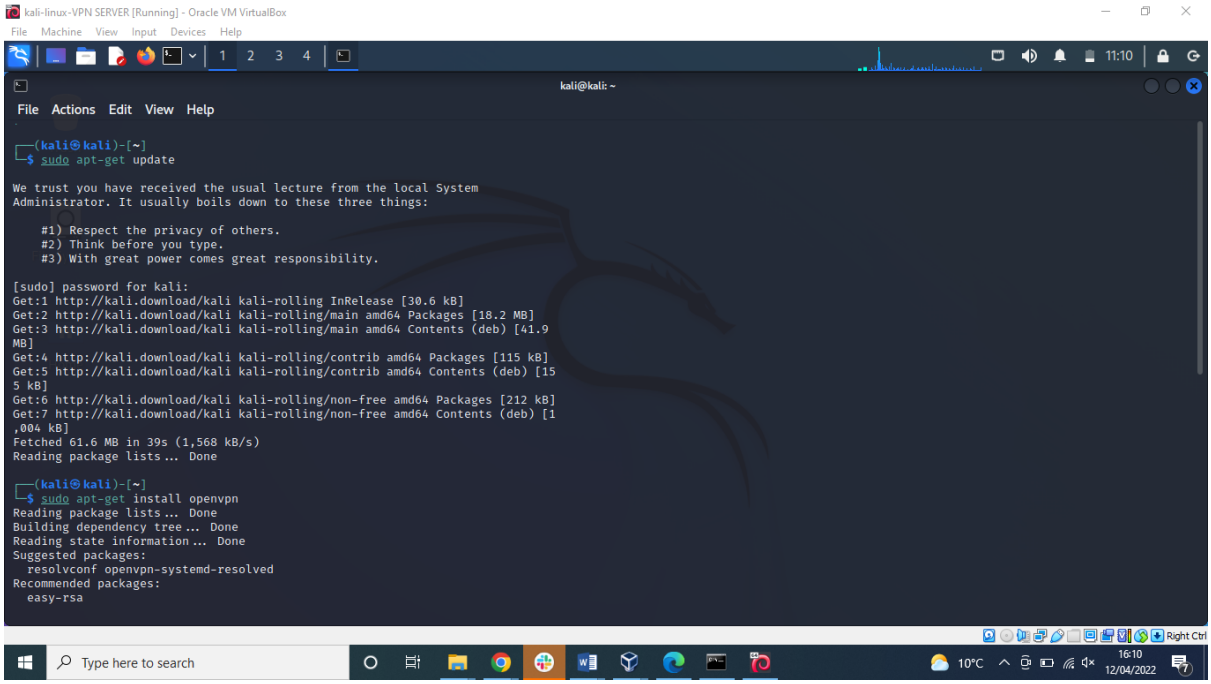


Step3 Power the VMs

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

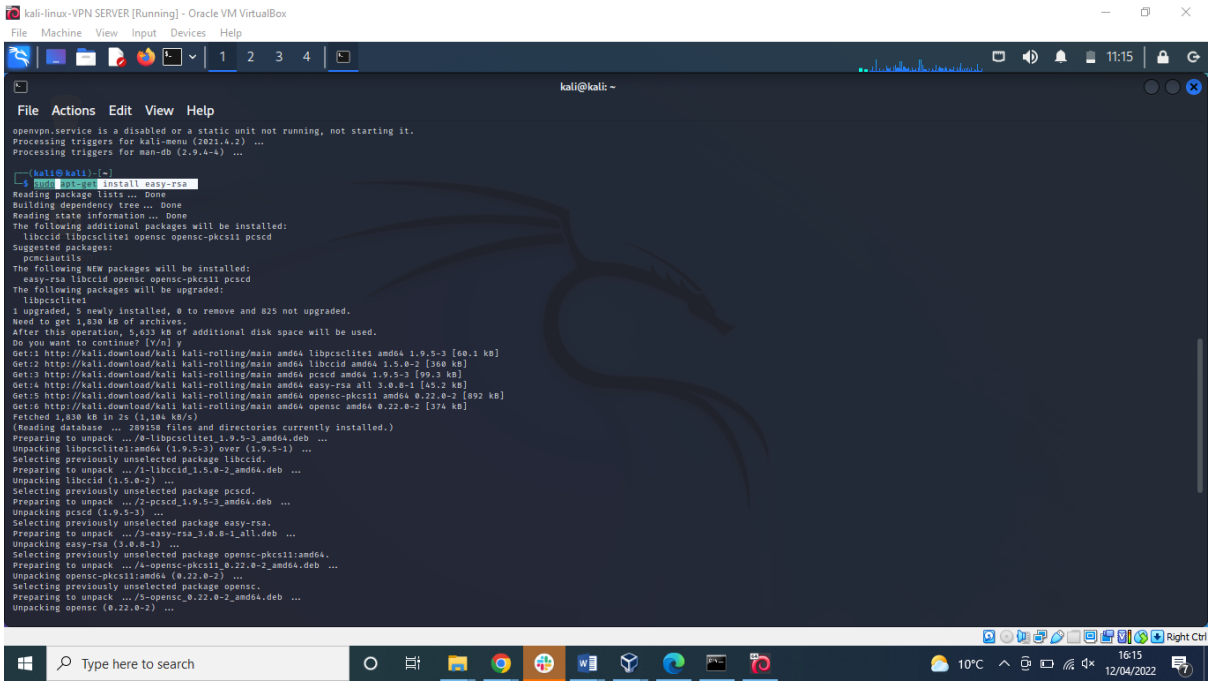
apt-get update

apt-get install openvpn



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

apt-get install easy-rsa



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

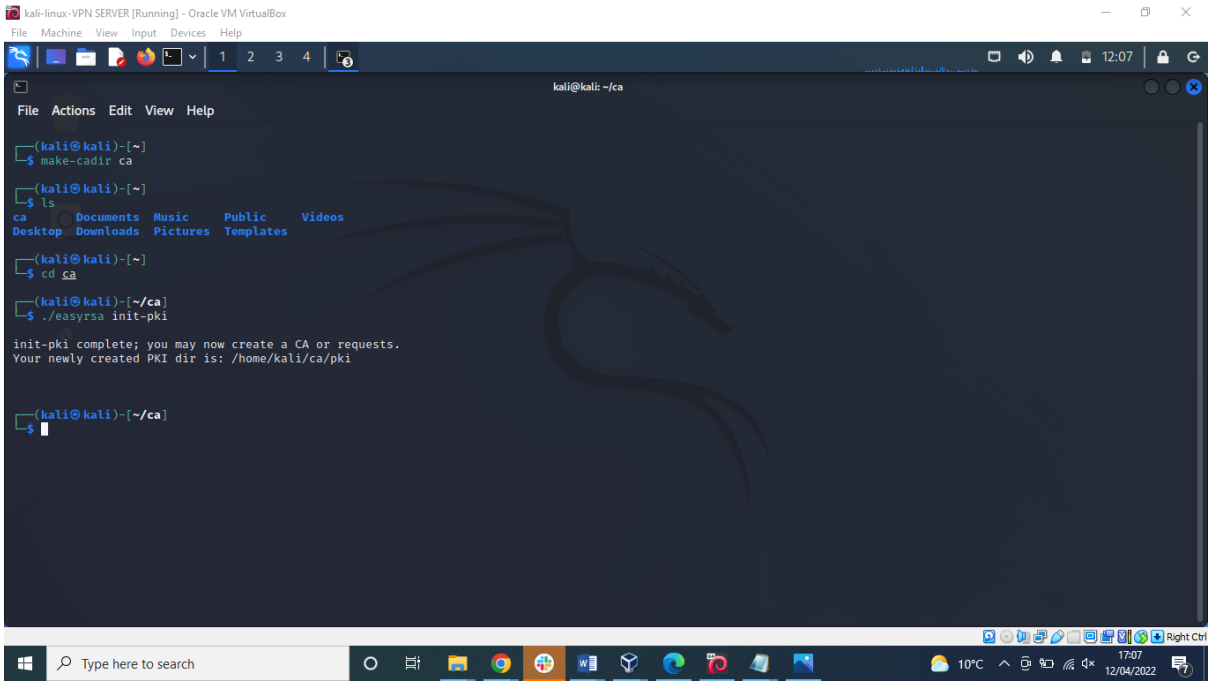
Step 5: Set up the Certificate Authority (CA) and generate certificates and keys for servers and the clients by carrying out the follow steps on one VM (here we are using the VPN server)

1. Create a new PKI

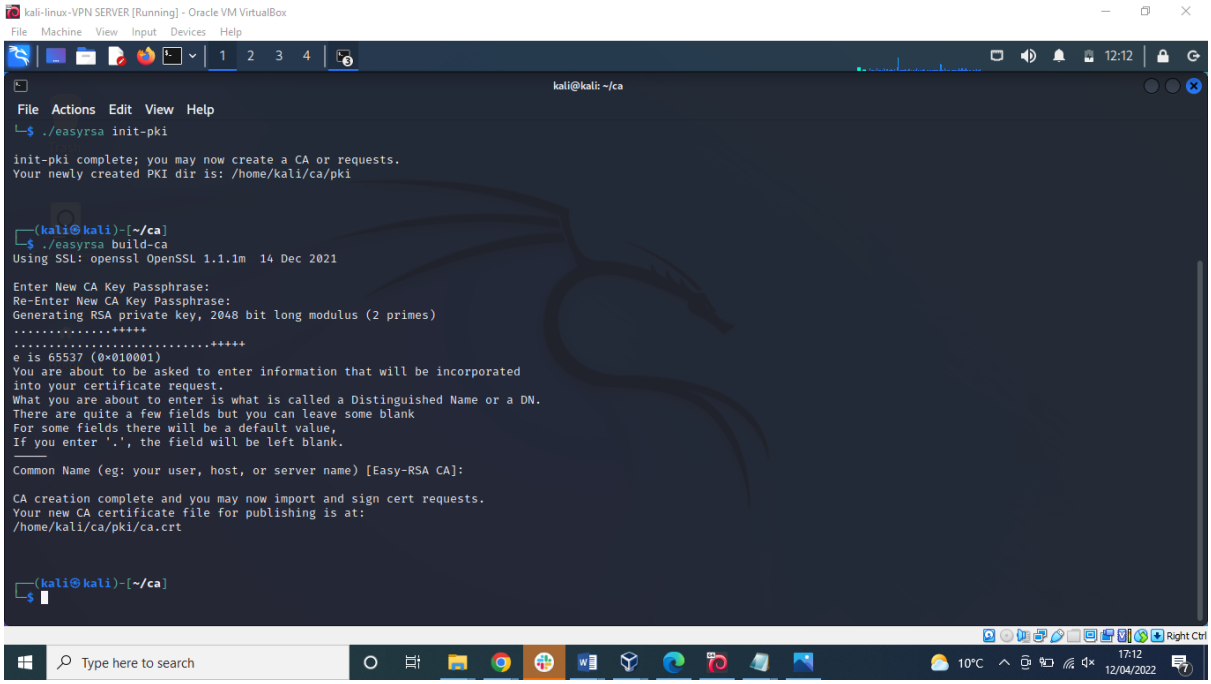
$ make-cadir ca

$cd ca

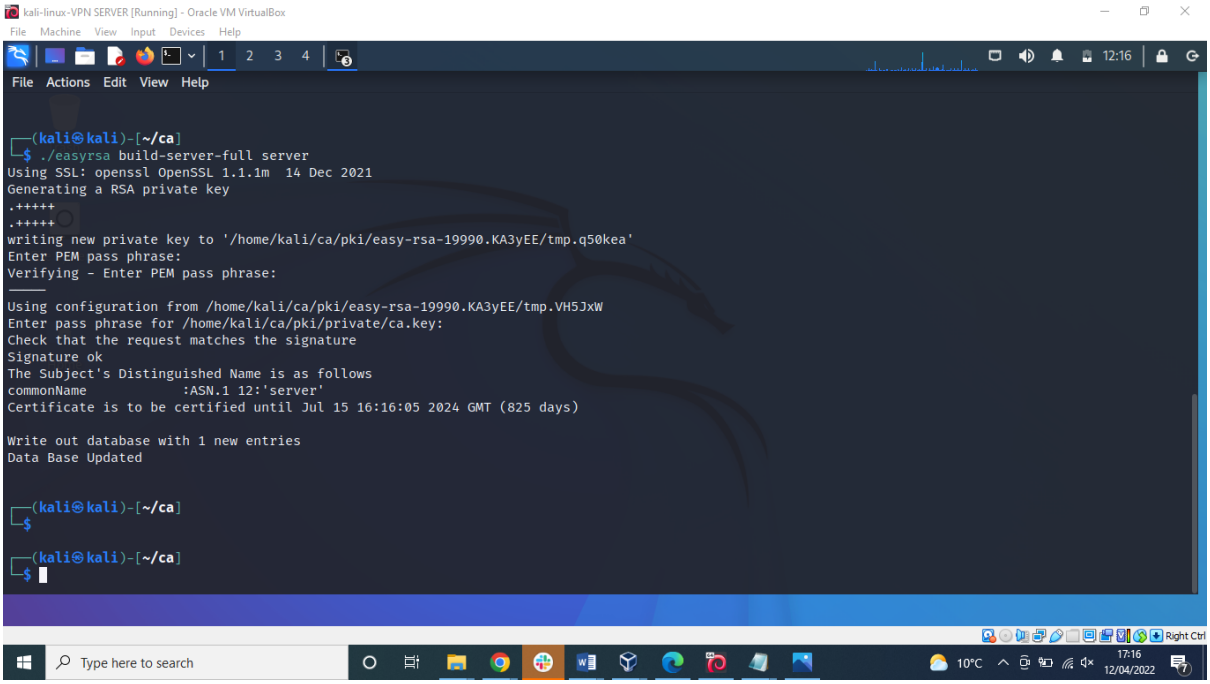
$./easyrsa init-pki



2. Build the CA, set up the passphase for the CA
   $ ./easyrsa build-ca

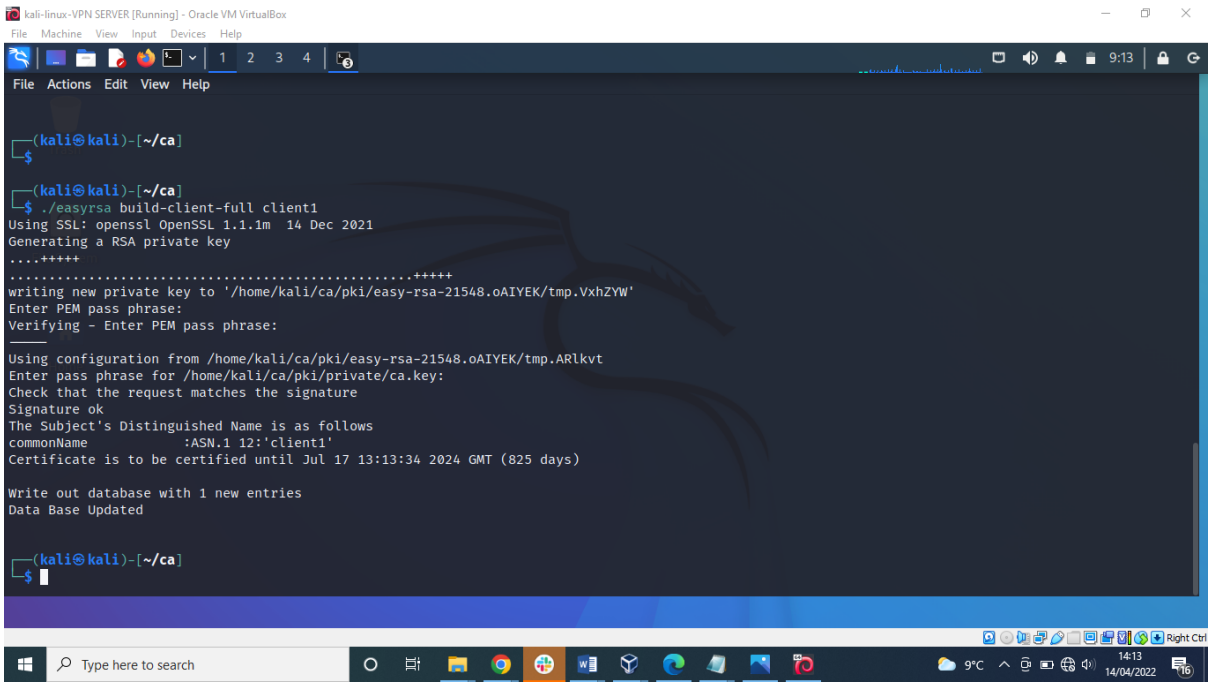


3. Generate certificate and private key for server
   $ ./easyrsa build-server-full server



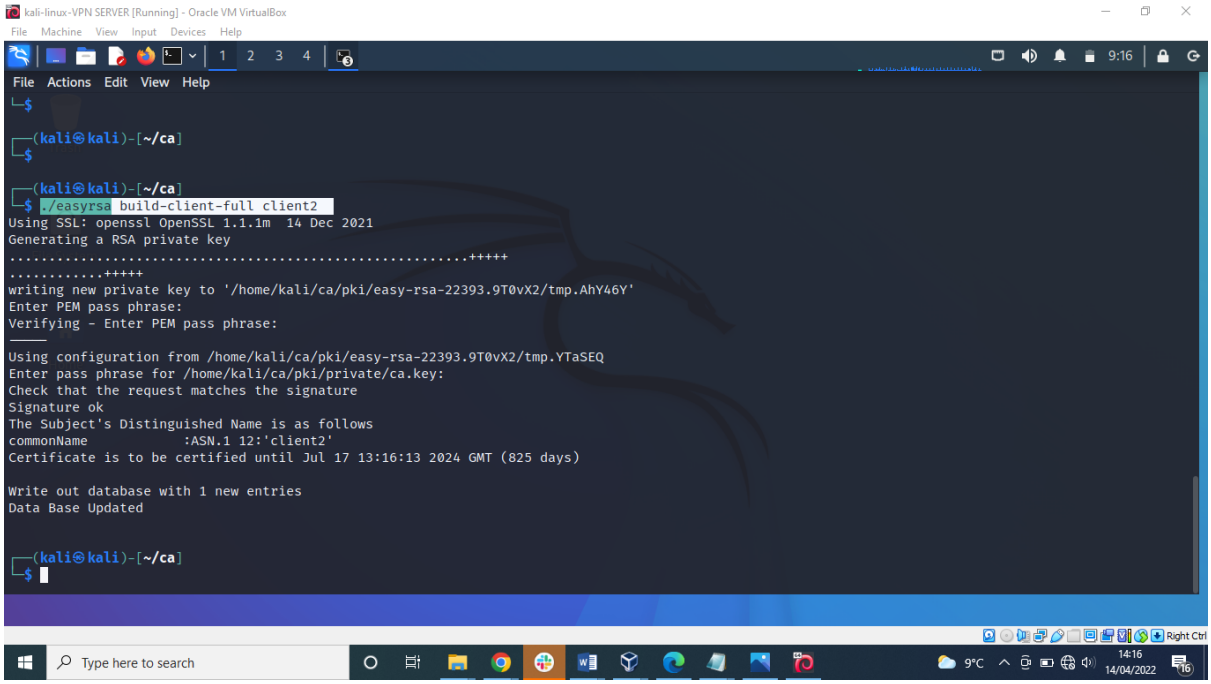
4. Generate certificate and private –key for client1

$ ./easyrsa build-client-full client1



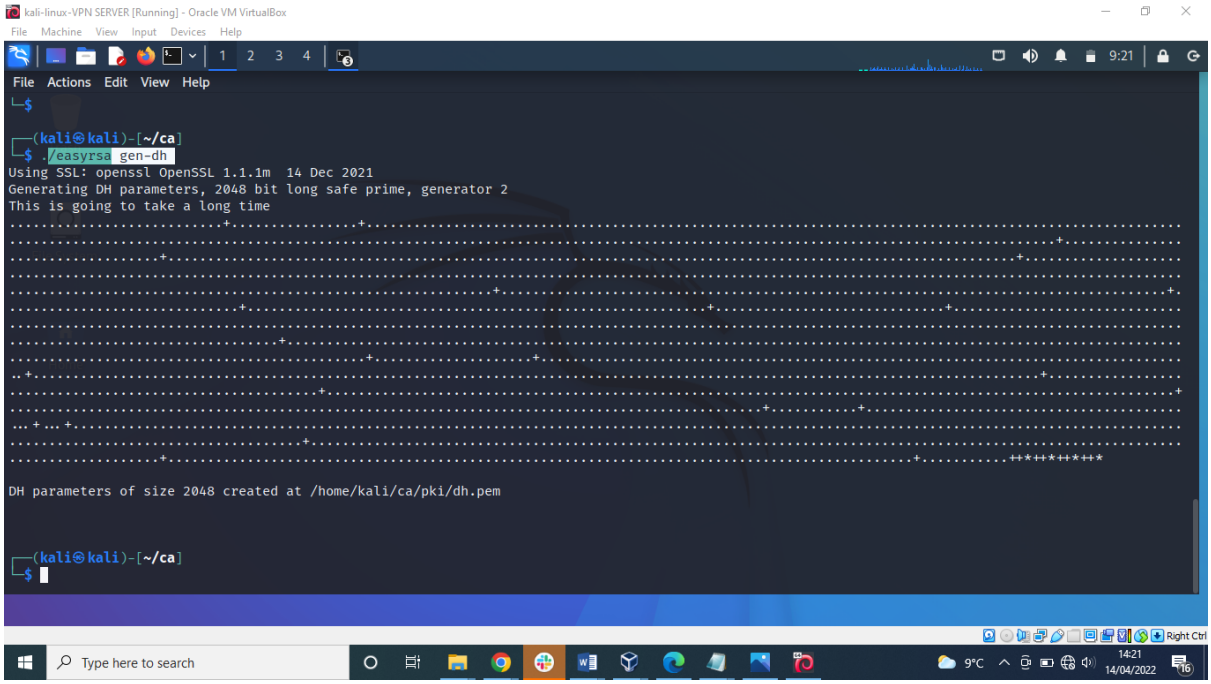
5. Generate certificate and private –key for client2

$ ./easyrsa build-client-full client2



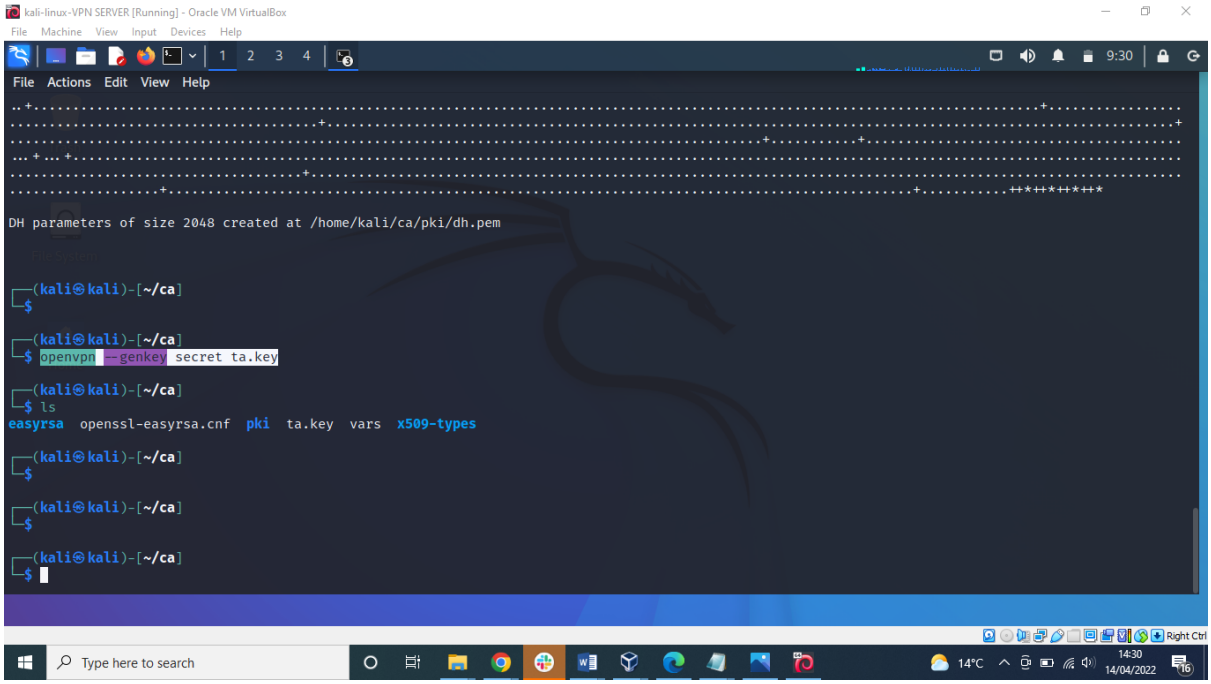
6. Generate DH (Diffie Hellman) parameters

$ ./easyrsa gen-dh



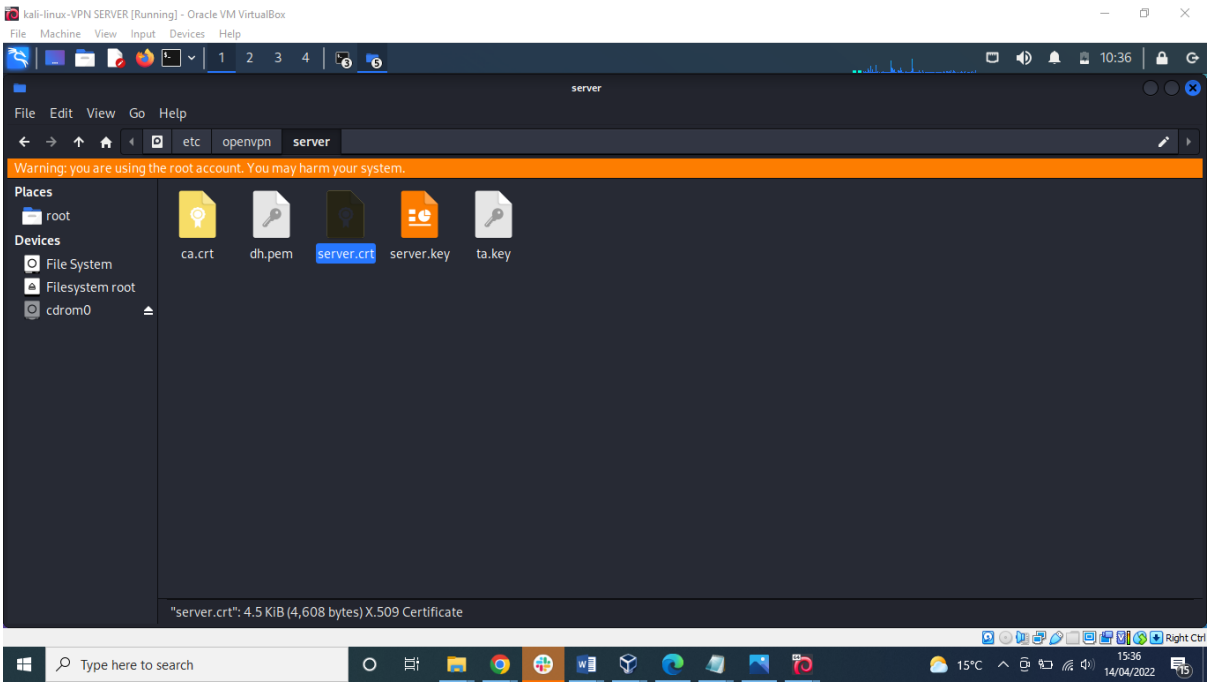
7. Generate TLS authentication key

$ openvpn –genkey secret ta.key



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



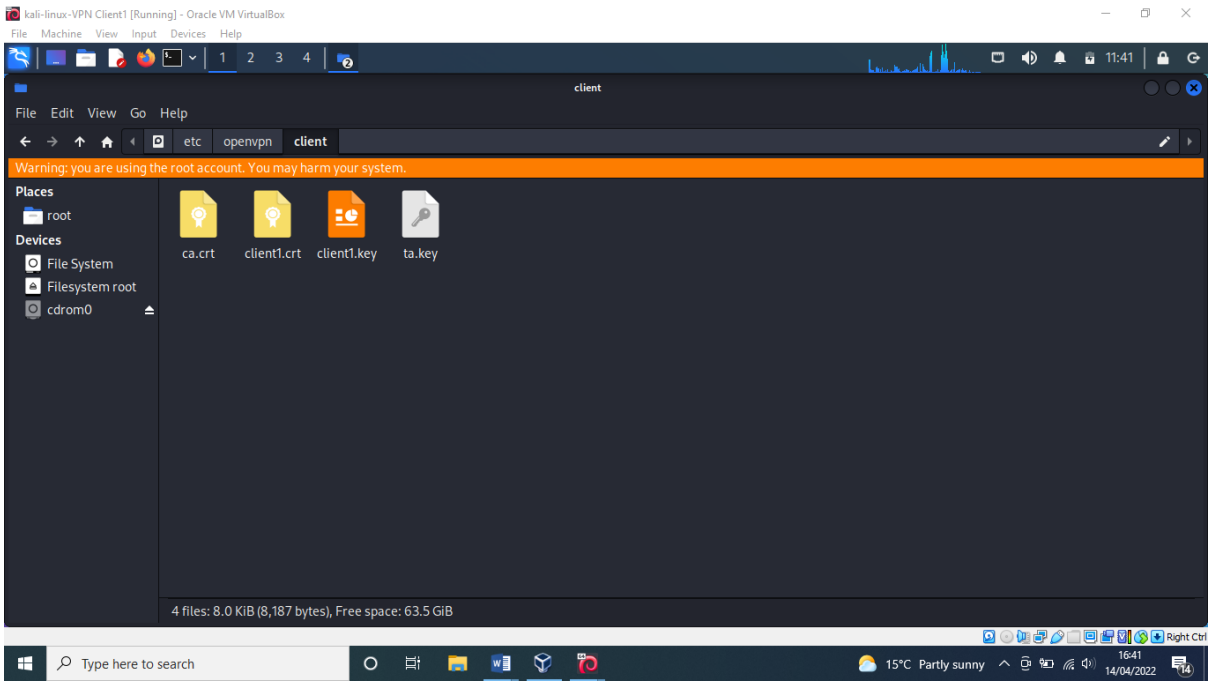
2. Copy the files below from ca to your /etc/openvpn/client on the client1 VM

Ca.crt

Client1.crt

Client1.key

Ta.key



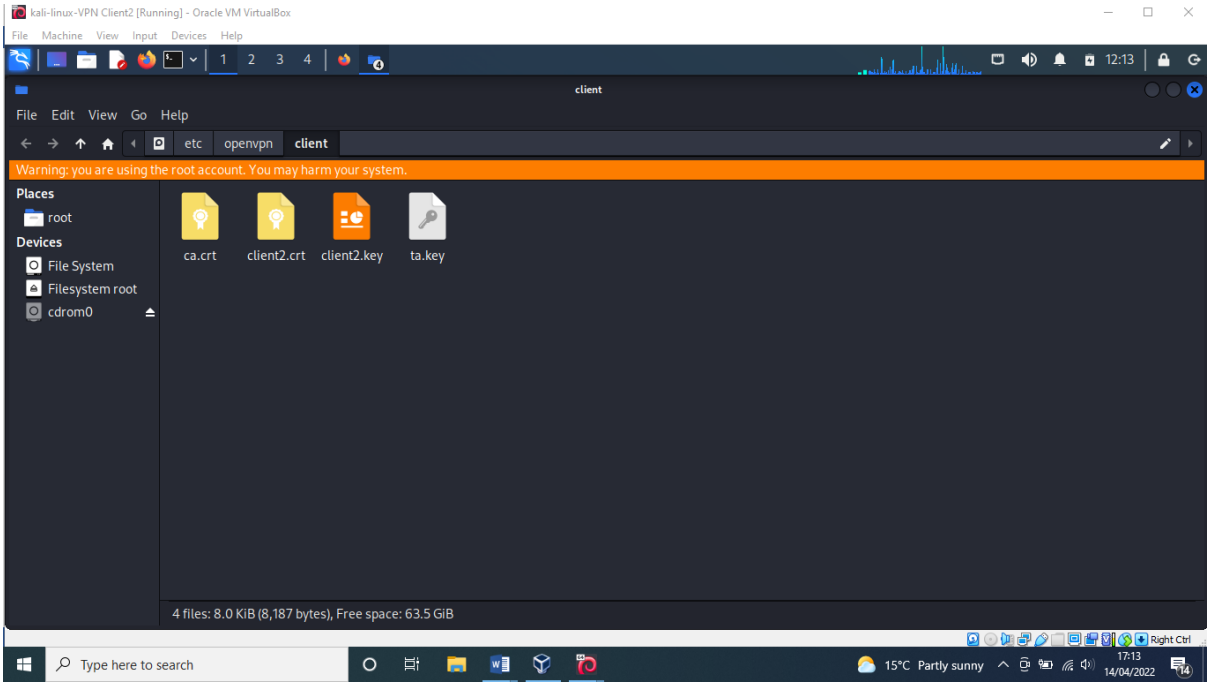
3. Copy files below from ca to your /etc/openvpn/client on the client2 VM

Ca.crt

Client2.crt

Client2.key

Ta.key

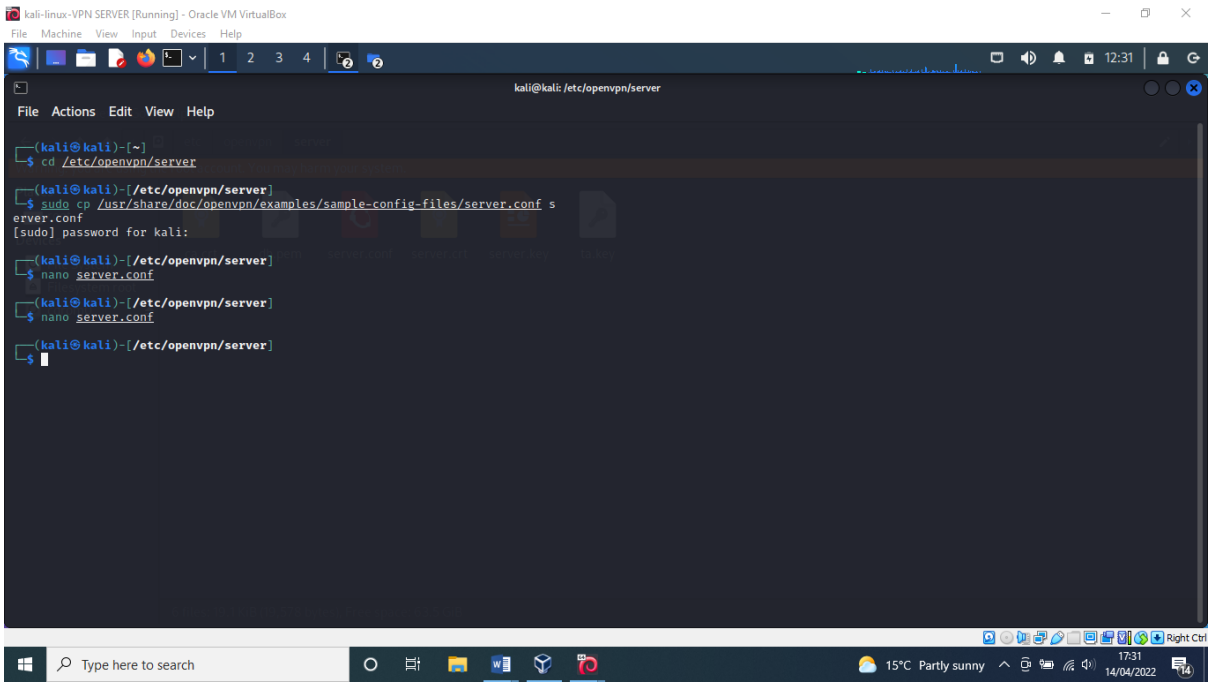


Step 7: Edit server config

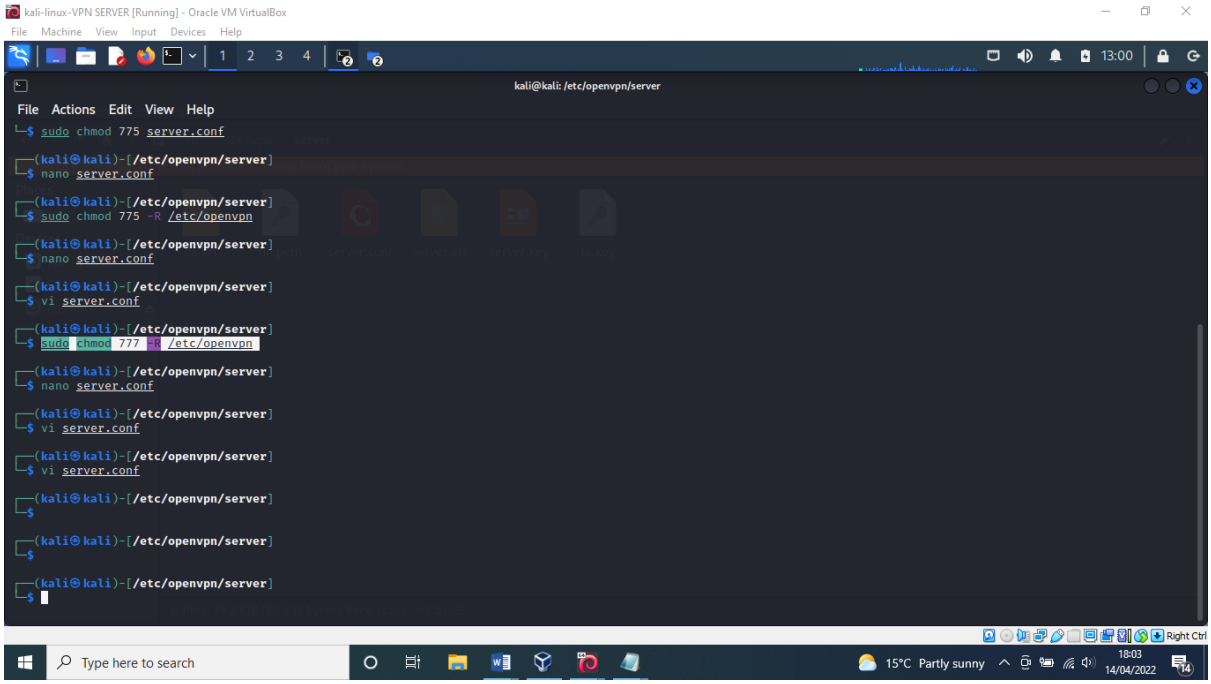
Type in the commands as shown on screenshot

$ cd /etc/openvpn/server

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



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



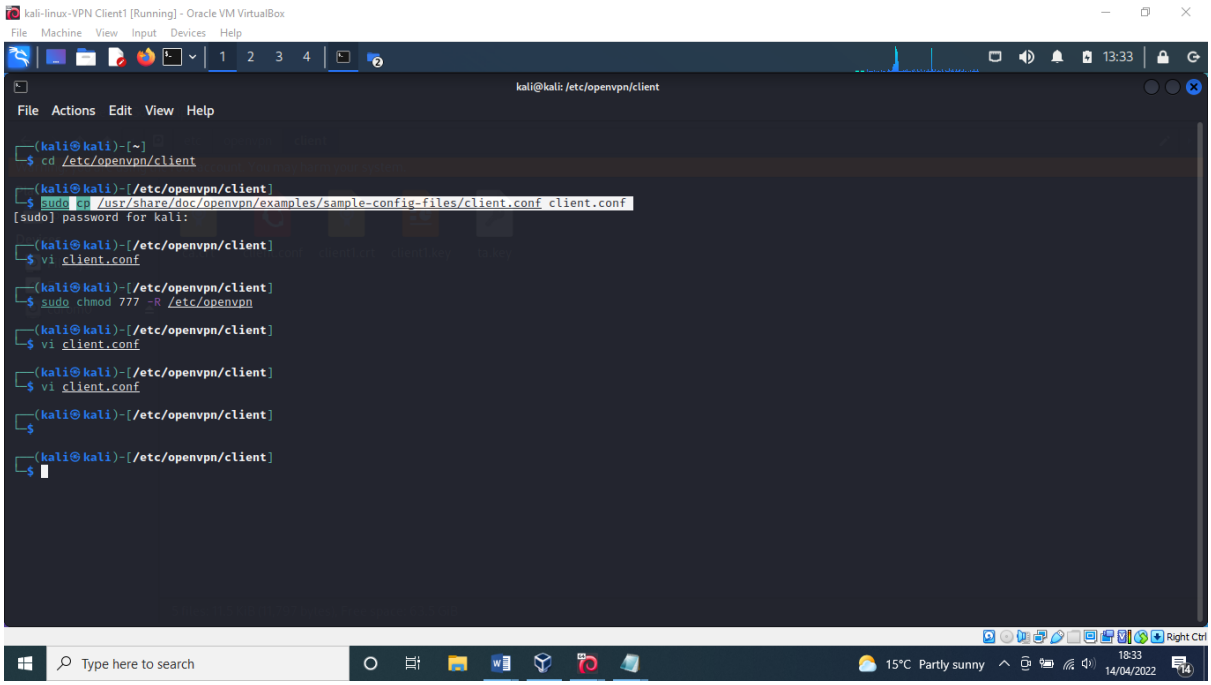
Now you can edit the server.conf

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



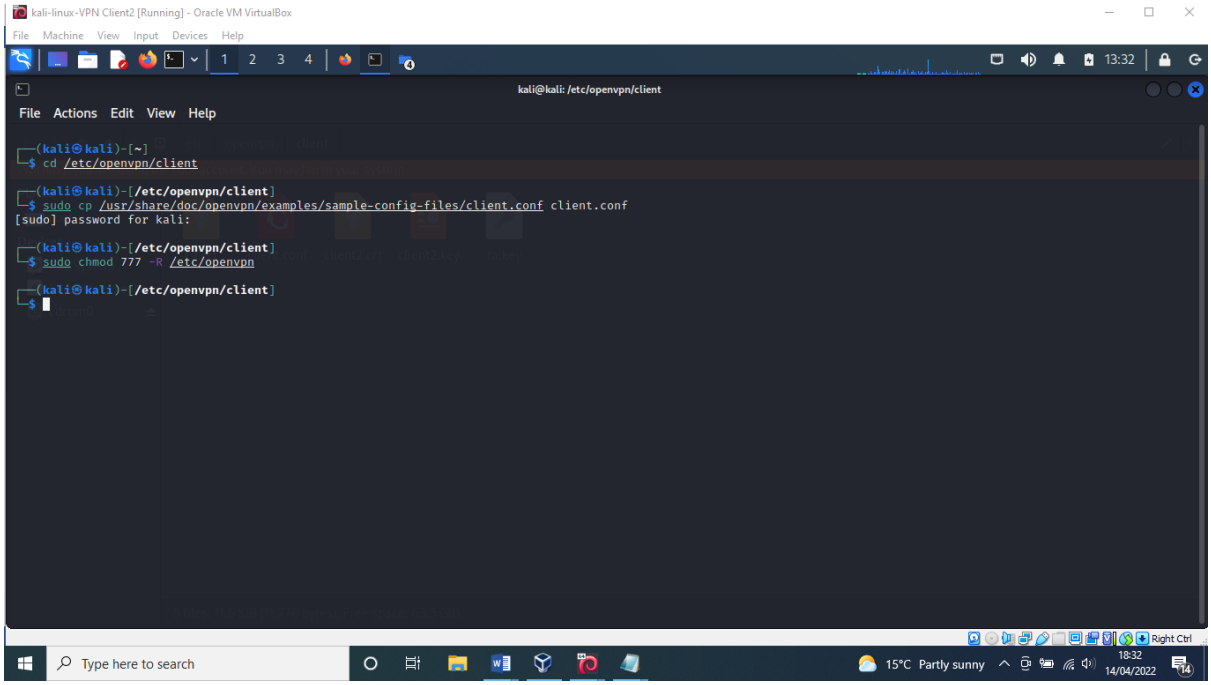
Now edit the rules

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)

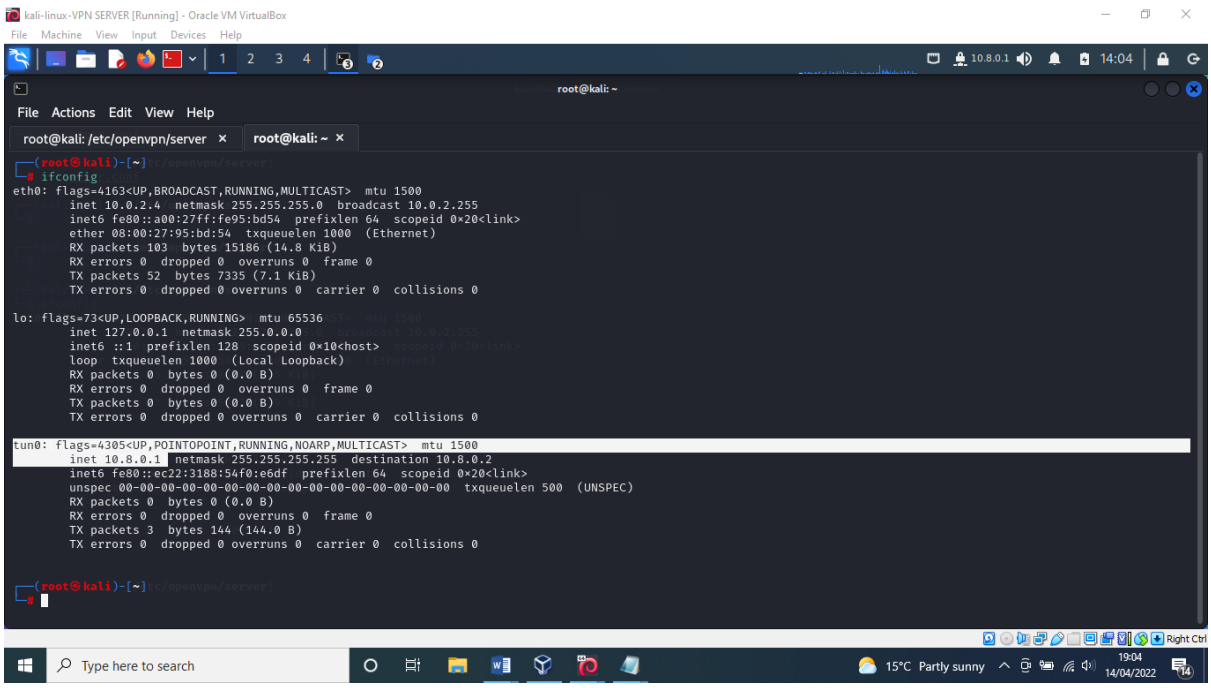
$ cd /etc/openvpn/client

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



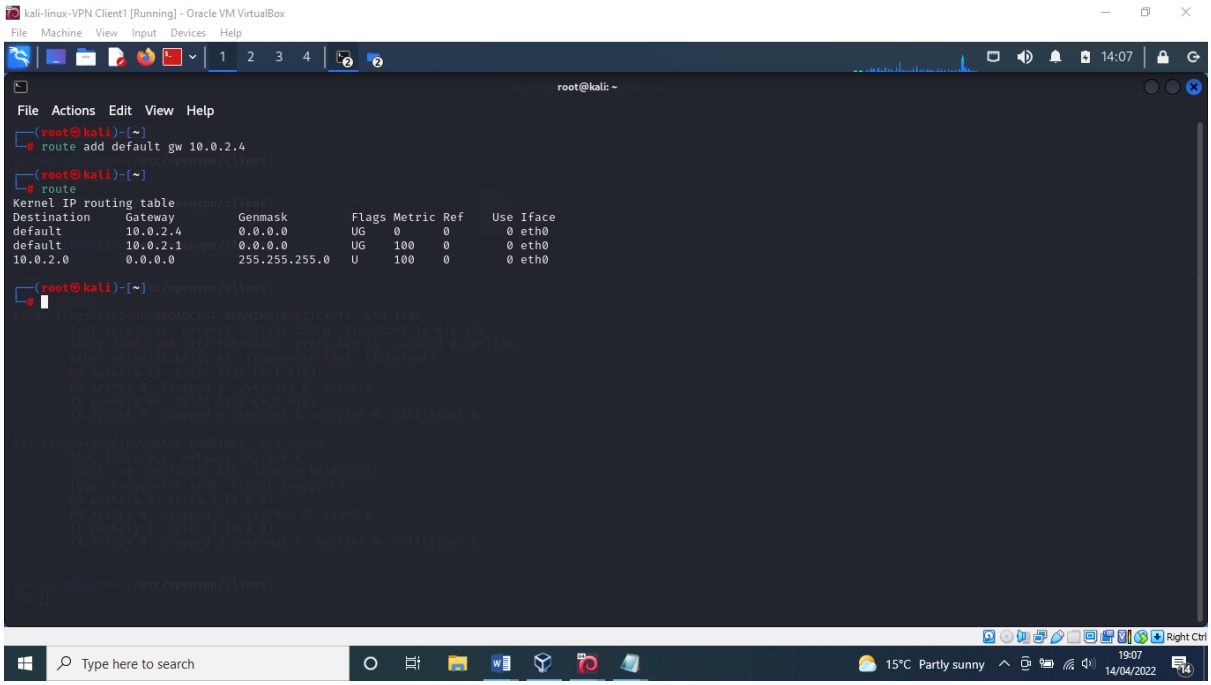
Step 10: Start VPN server and clients

For Linux VPN server

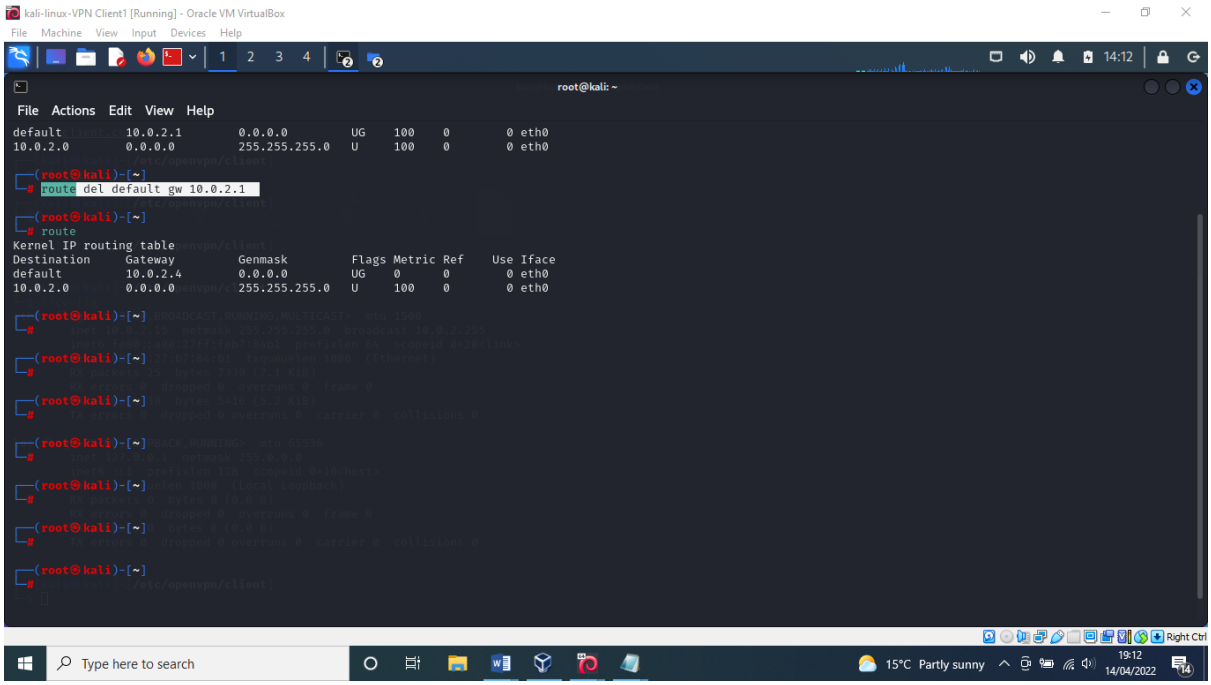


For the Clients; see screenshot below showing that the VPN has started

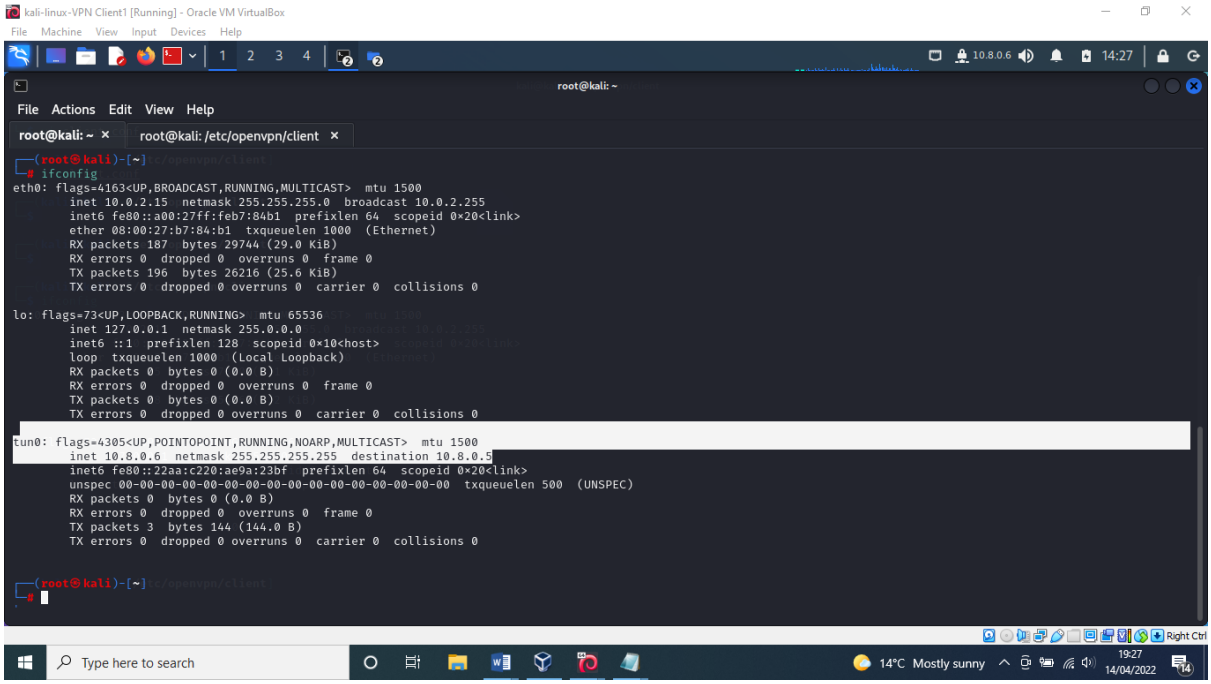
For Client 1 , Add the default route to ensure that it uses the server IP 10.0.2.4 as gateway using the command below



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

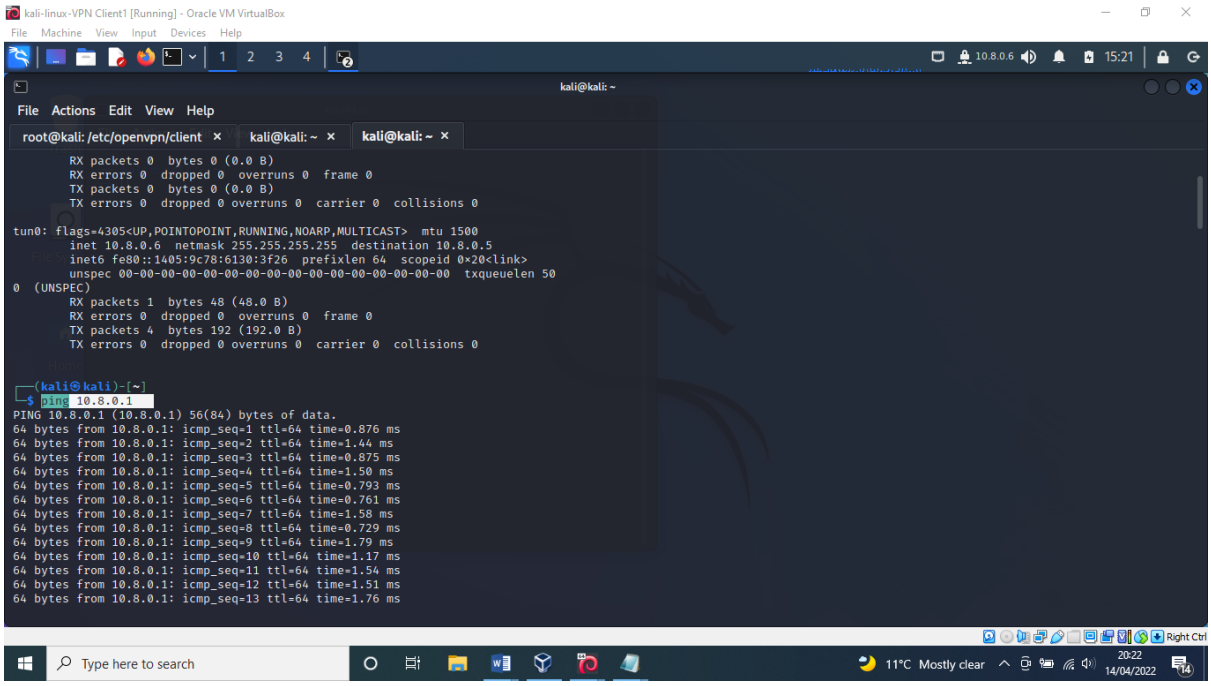


Now start the VPN client on Client1



Step 11: Test VPN connection

Ping 10.8.0.1



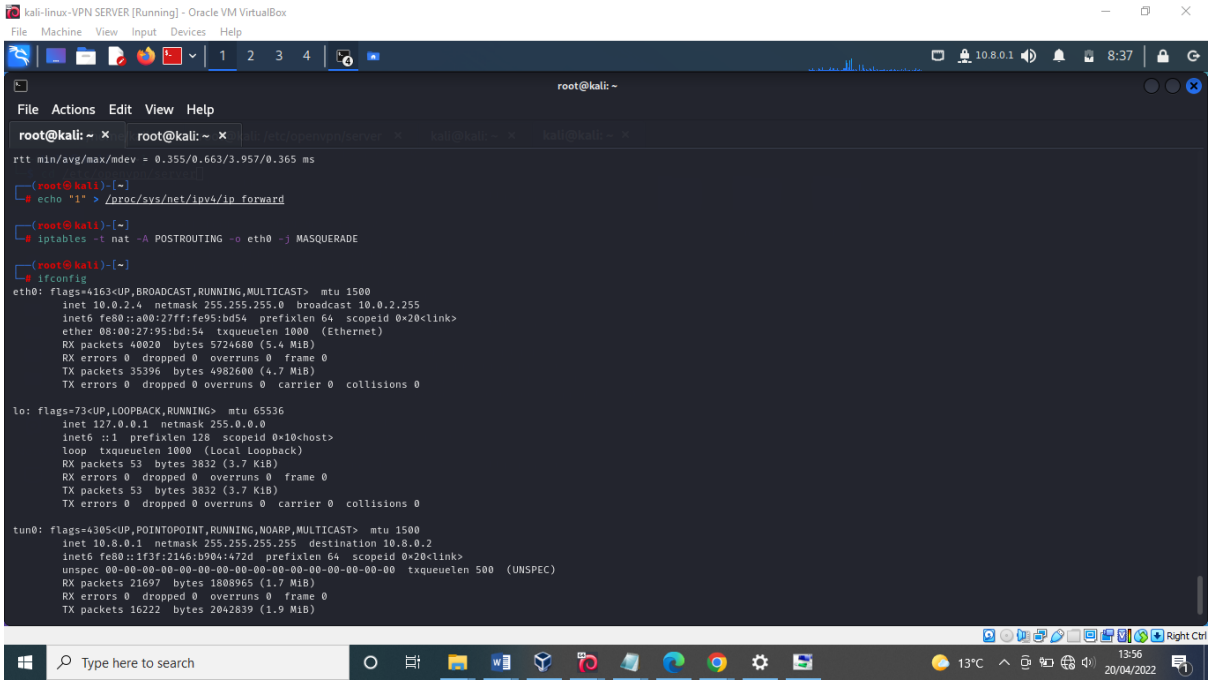
Step 12: Set IP forwarding and masquerading (On Linux VPN server) – These commands show below are important as it enables access to the internet from the VPN client

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



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: