Lab 2: IPSec VPN

Introduction

The cloud paradigm changes the way applications are developed, architected, and operated. Improved agility, speed of development, and increased automation in operations allow enterprises to truly benefit from cloud infrastructure.

Usually the enterprise would want to setup encrypted connection from on-prem to Oracle cloud for performing POC's and do some small migrations. Later enterprises can later setup Fastconncet, out dedicated private conncetion from on-premises data center to Oracle cloud.

Objective

This lab walks you to the steps needed to create a VPN connection from a Linux VM to an OCI VPN Connect. On the public VM from the previous lab, we will install an opensource vpn software. In another region, we will create a VPN connection to this VM.

Pre-requisites

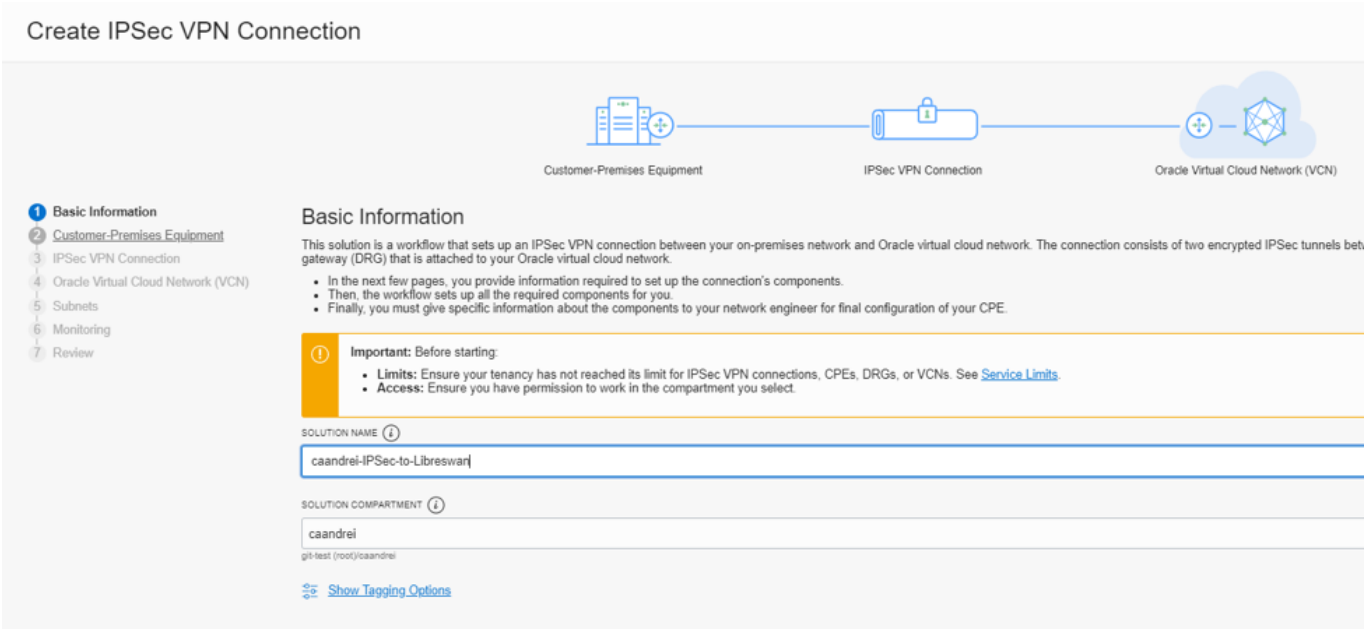
You need the setup from Lab1 to continue working on this Lab

Process Overview

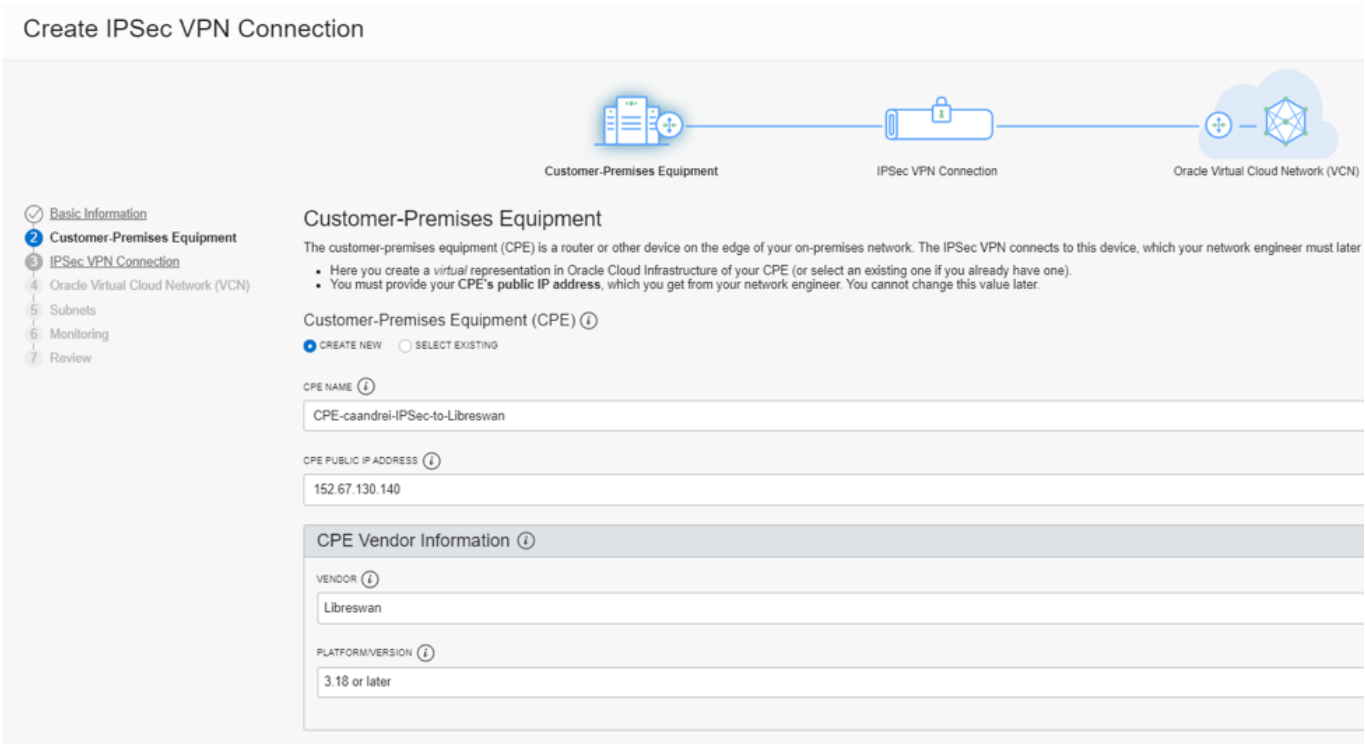
Create an OCI IPSec Connection

Navigate to Networking > Overview and click the “Start VPN Wizard”.

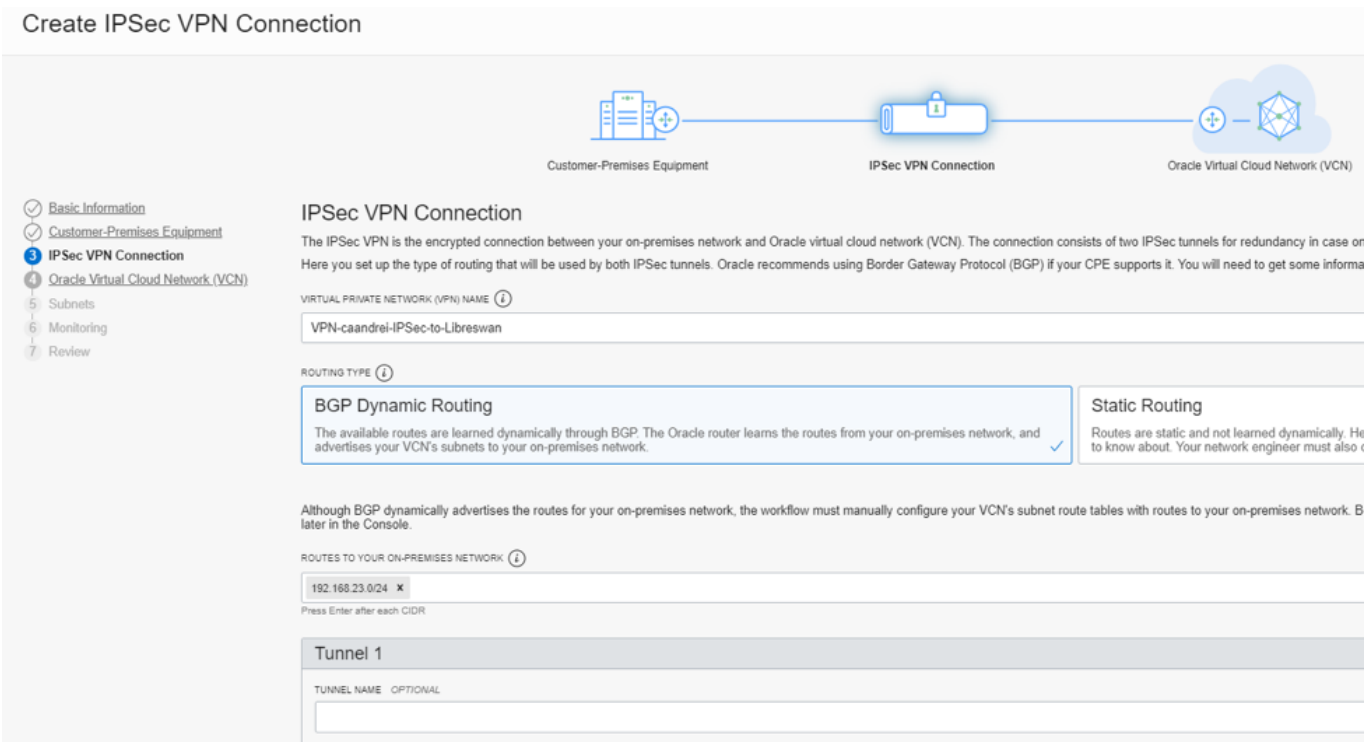
Fill in the Name of the IPSec Connection



Fill in the public IP address of the public VM created in Lab1 and select Libreswan from the Vendor drop-down. Select the Platform/Version



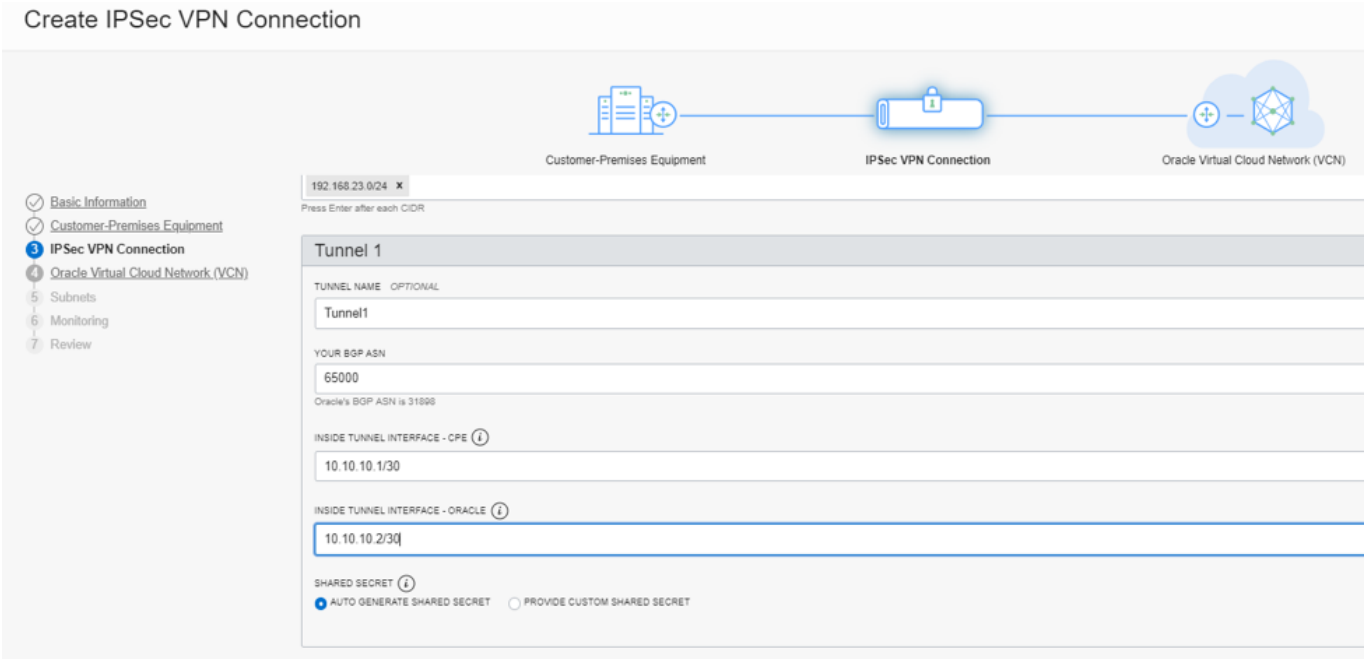
Fill in the CIDR space of the networks that are behind the Libreswan (192.168.23.0/24)



Fill in the name of the Tunnel1, and the BGP information.

We will use ASN65000 and a /30 for connectivity between BGP peers: 10.10.10.0/30.

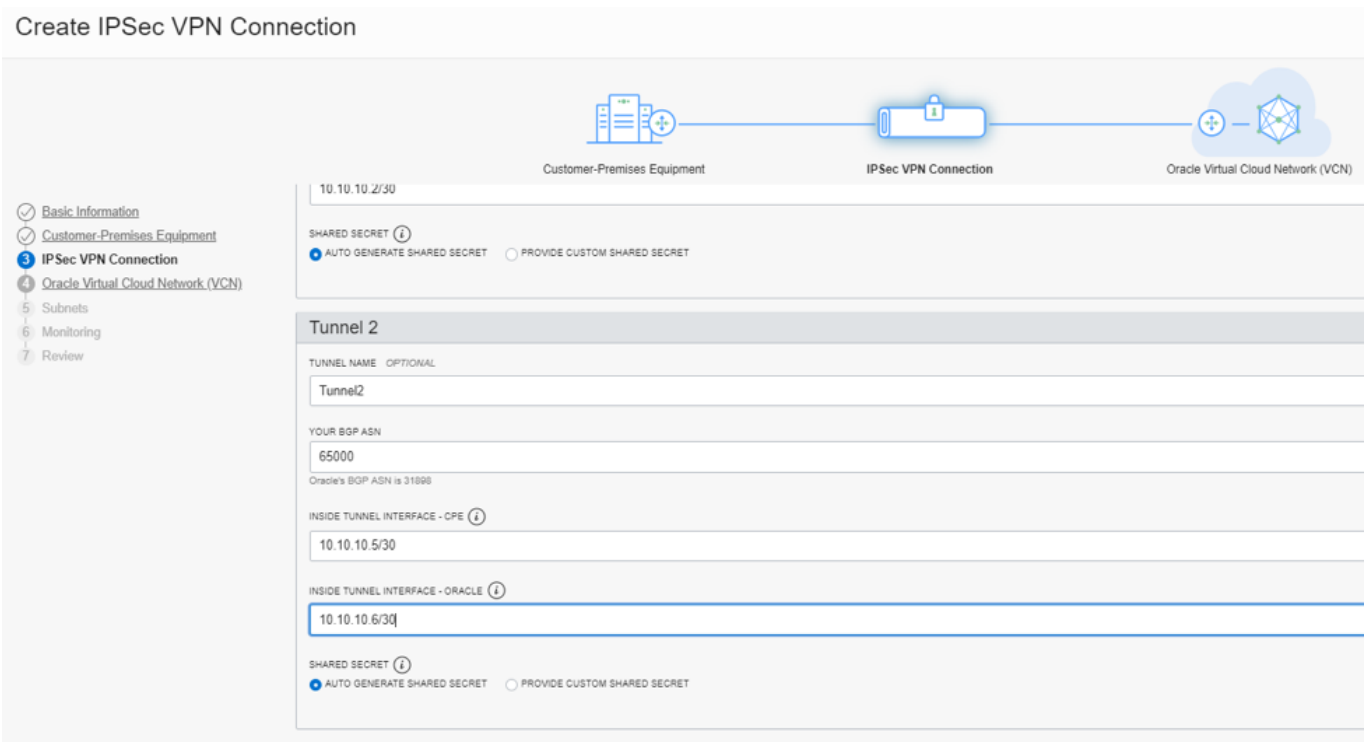
The CPE will use 10.10.10.1/30 and the oracle side will use 10.10.10.2/30



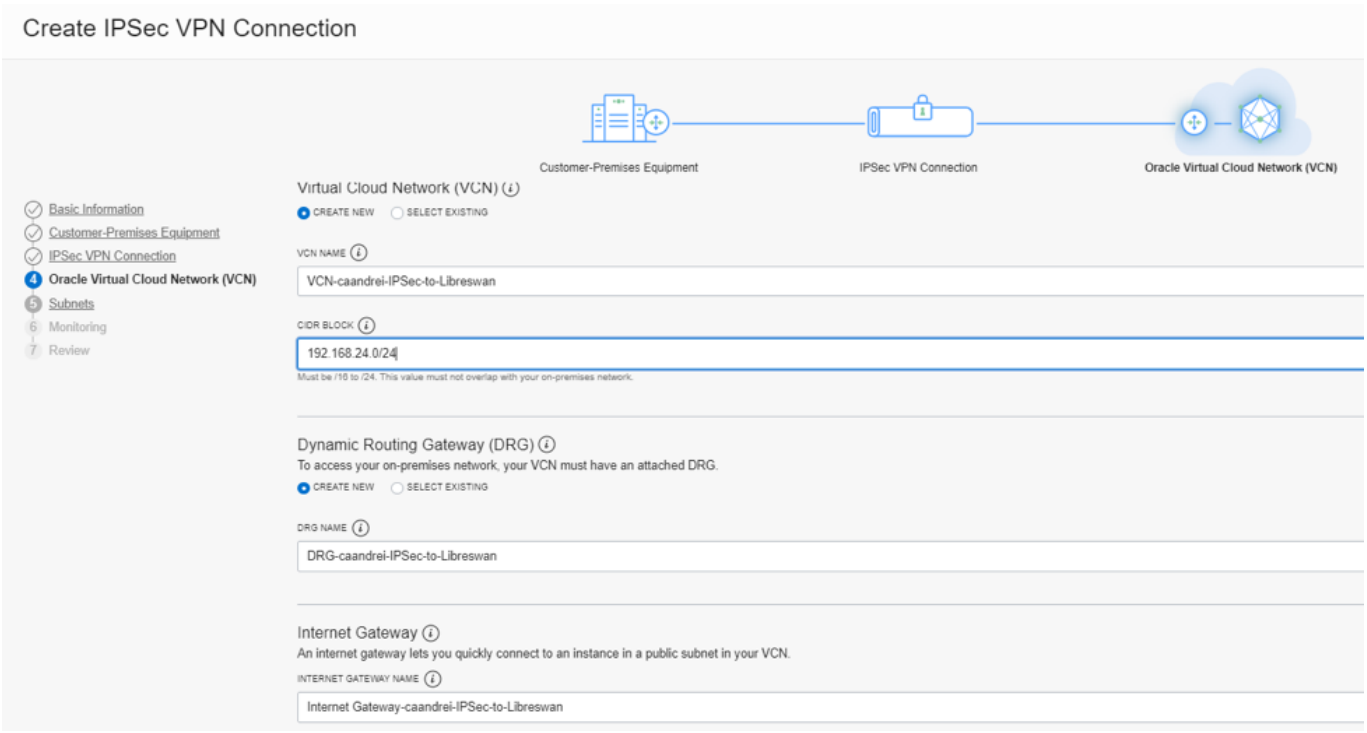
Fill in the name of the Tunnel2, and the BGP information.

We will use ASN65000 and a /30 for connectivity between BGP peers: 10.10.10.4/30.

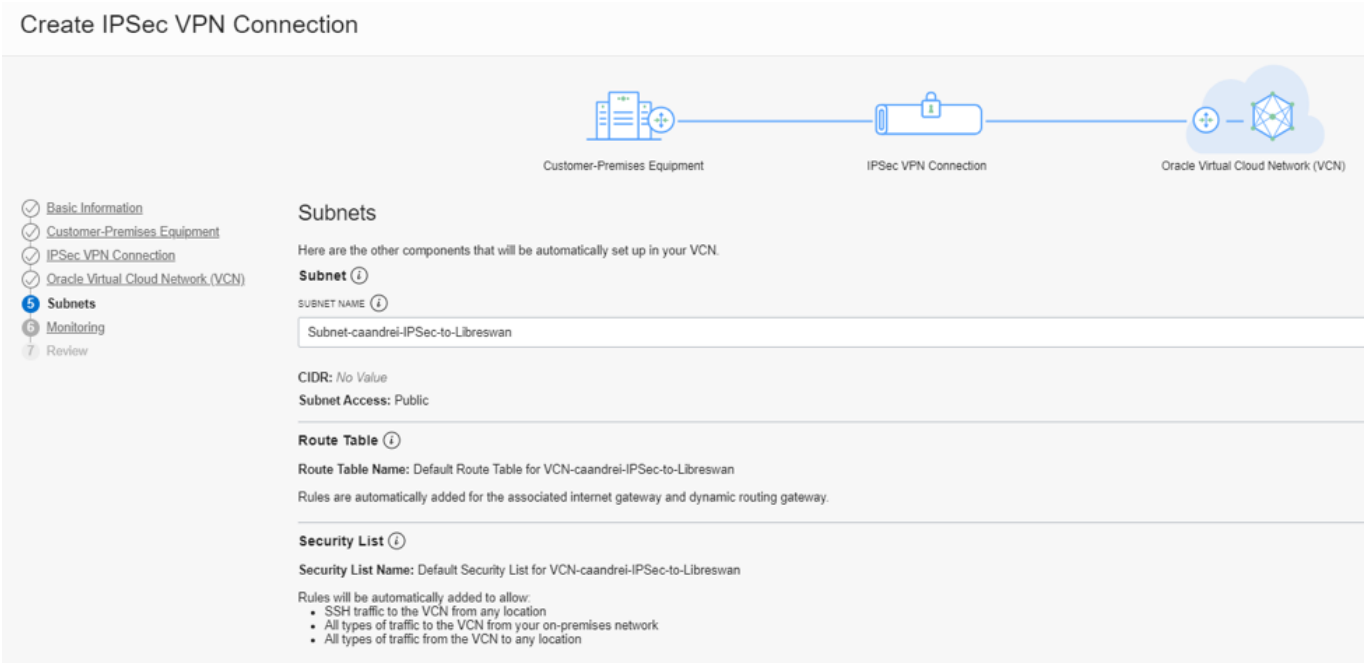
The CPE will use 10.10.10.5/30 and the oracle side will use 10.10.10.6/30



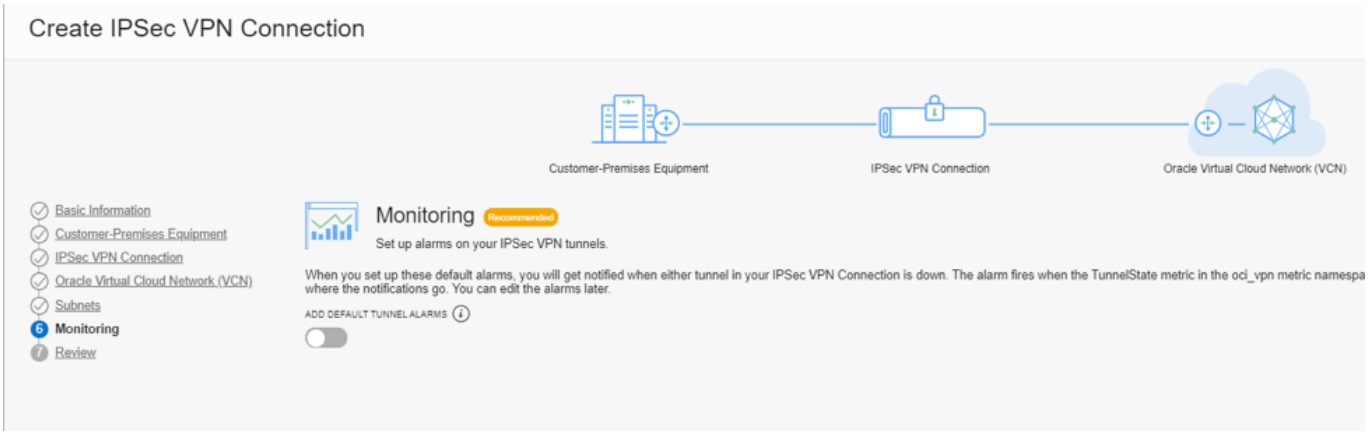
Specify the CIDR space for the VCN from OCI: 192.168.24.0/24



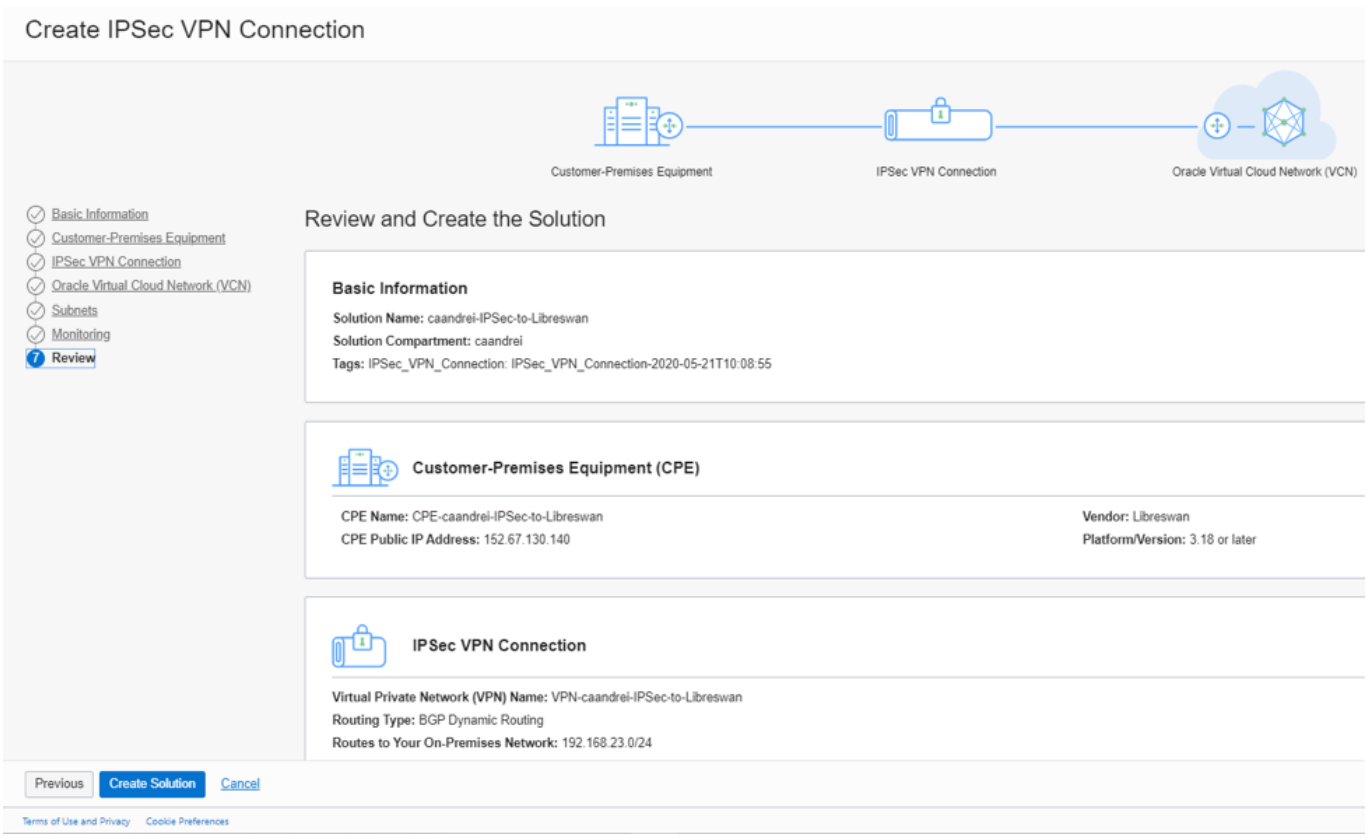
On the next screen, you will have the details of the networking artifacts that will be created.



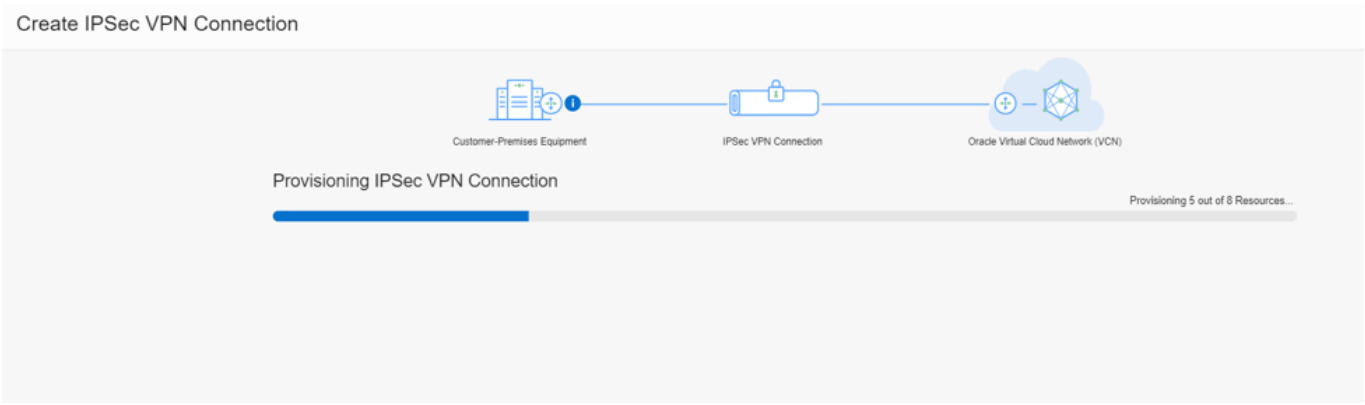
On the next screen you will have the Monitoring options

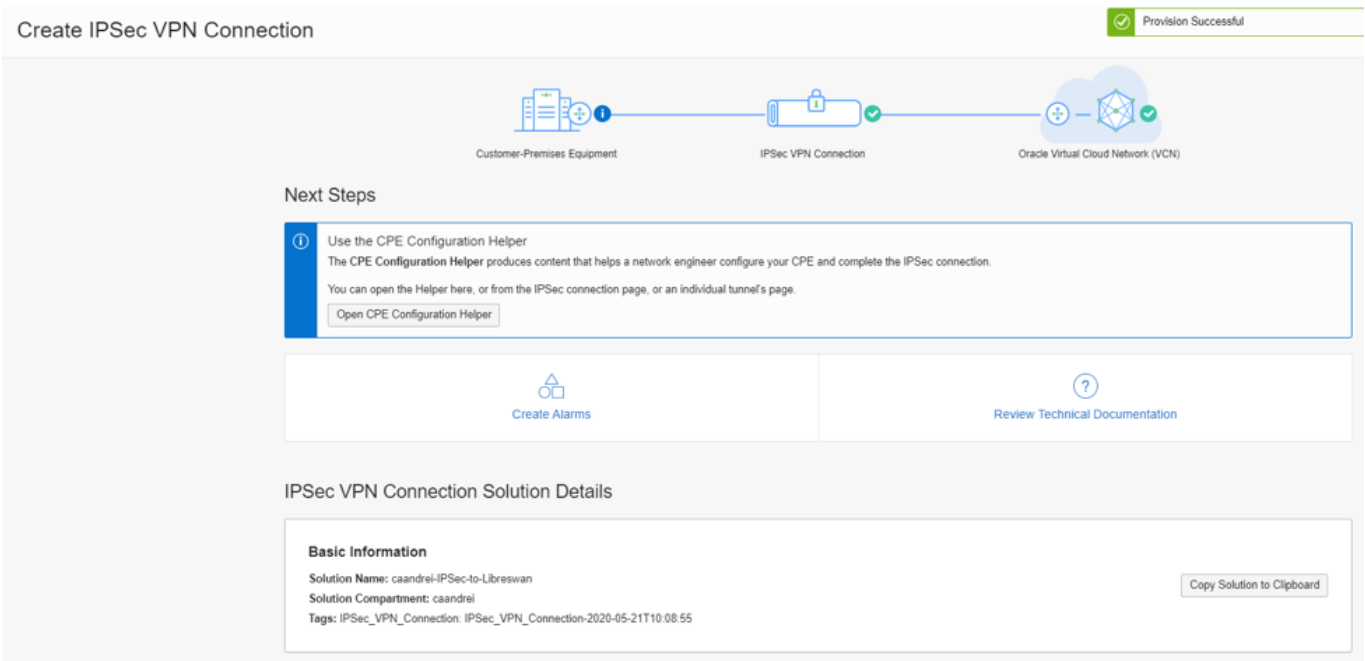


The last screen will present the overall configuration that will be created.



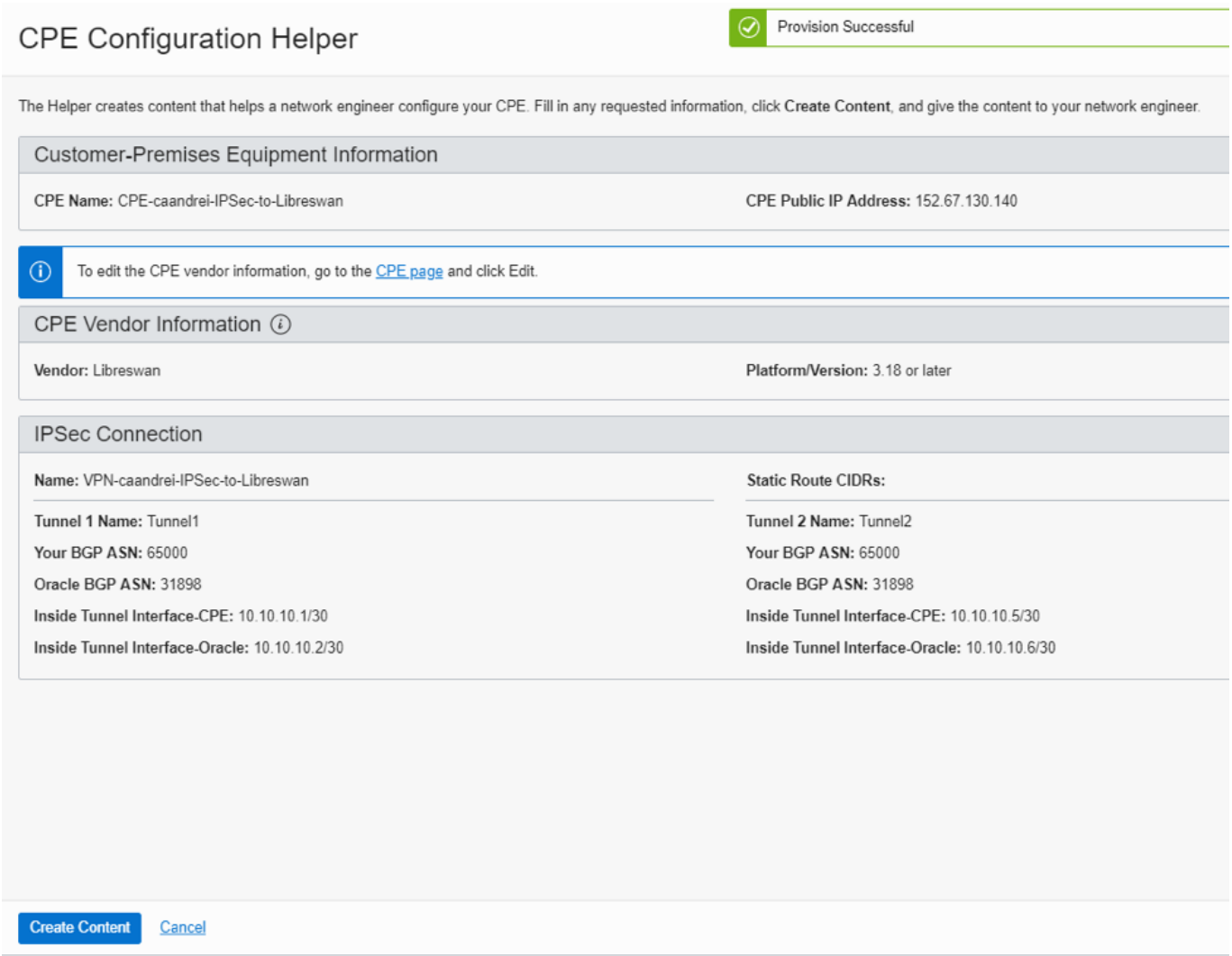
Push the “Create Solution” button. You will have the progress of the provisioning:



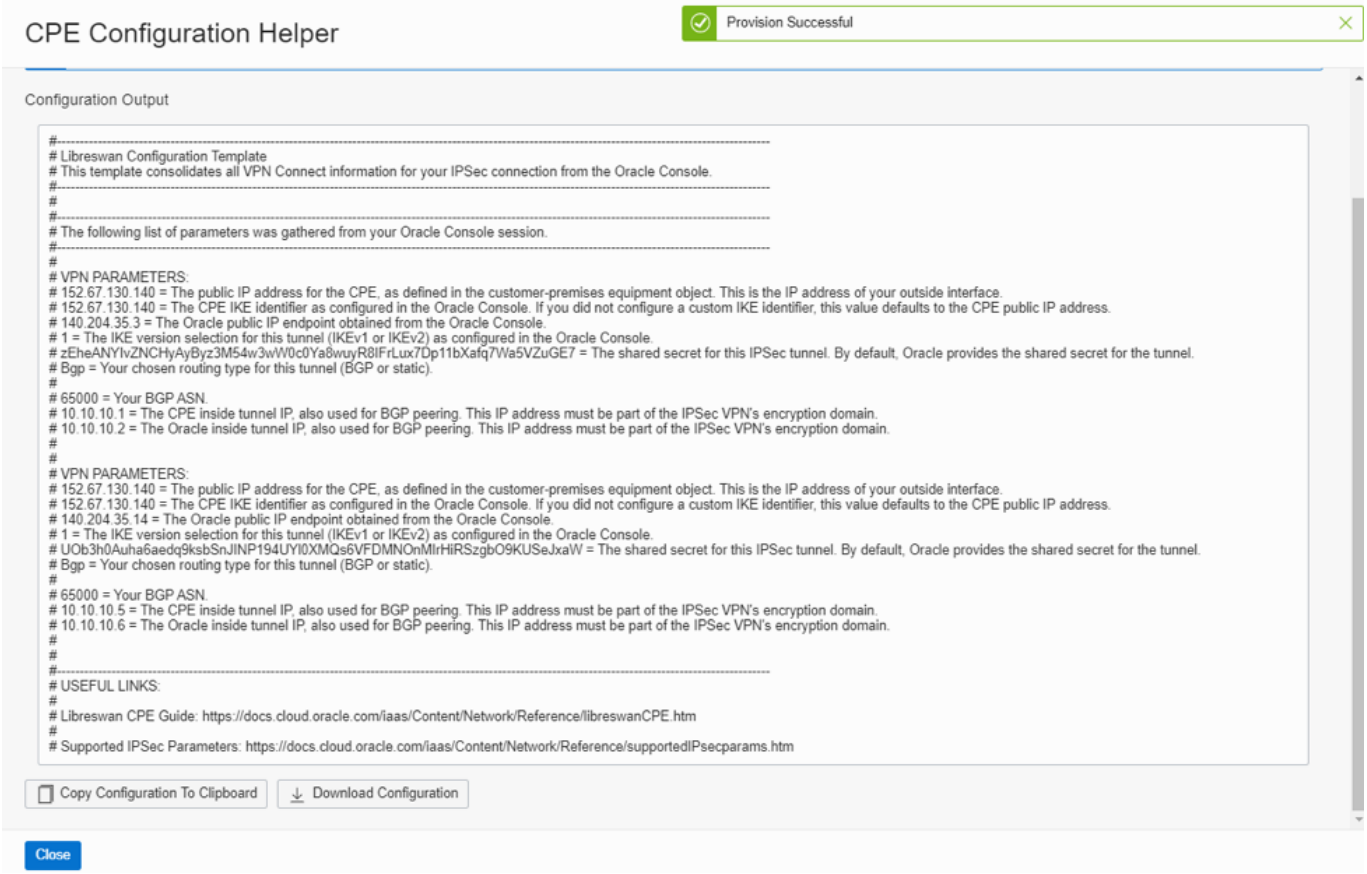


Wait a few minutes and click on the “Open CPE Configuration Helper”.

You will see the summary of what was configured on the OCI side.



Push the “Create Content” button



Download the configuration

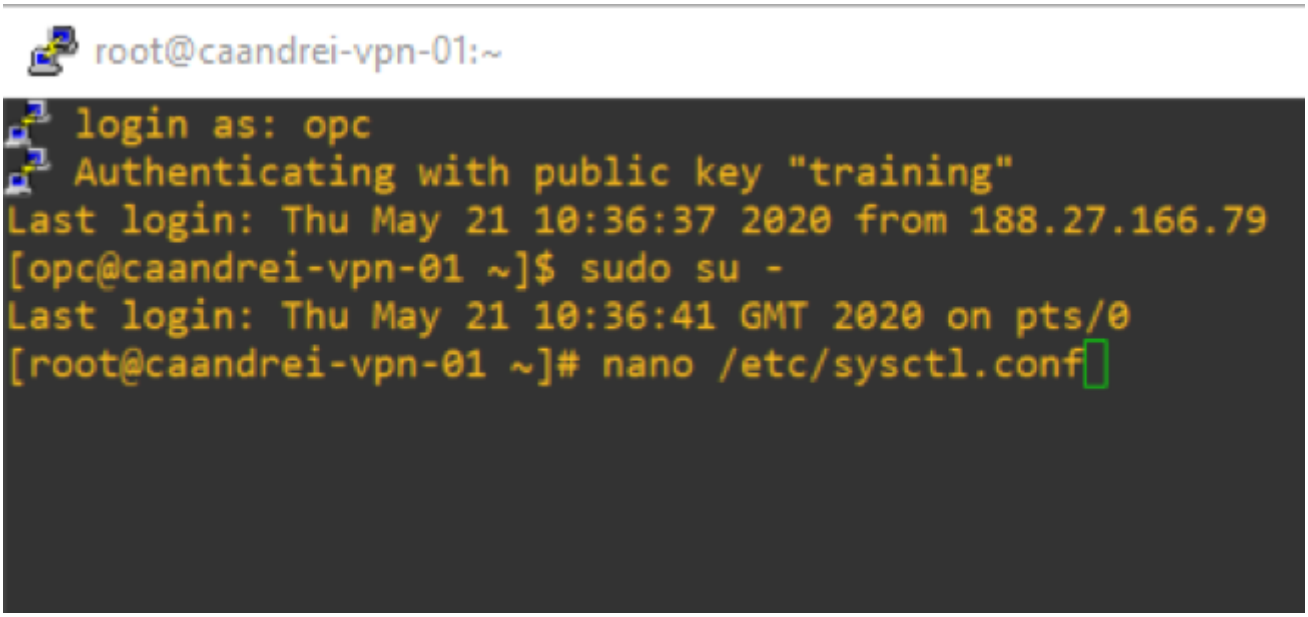
Install and Configure Libreswan

Follow the official documentation for Libreswan here.

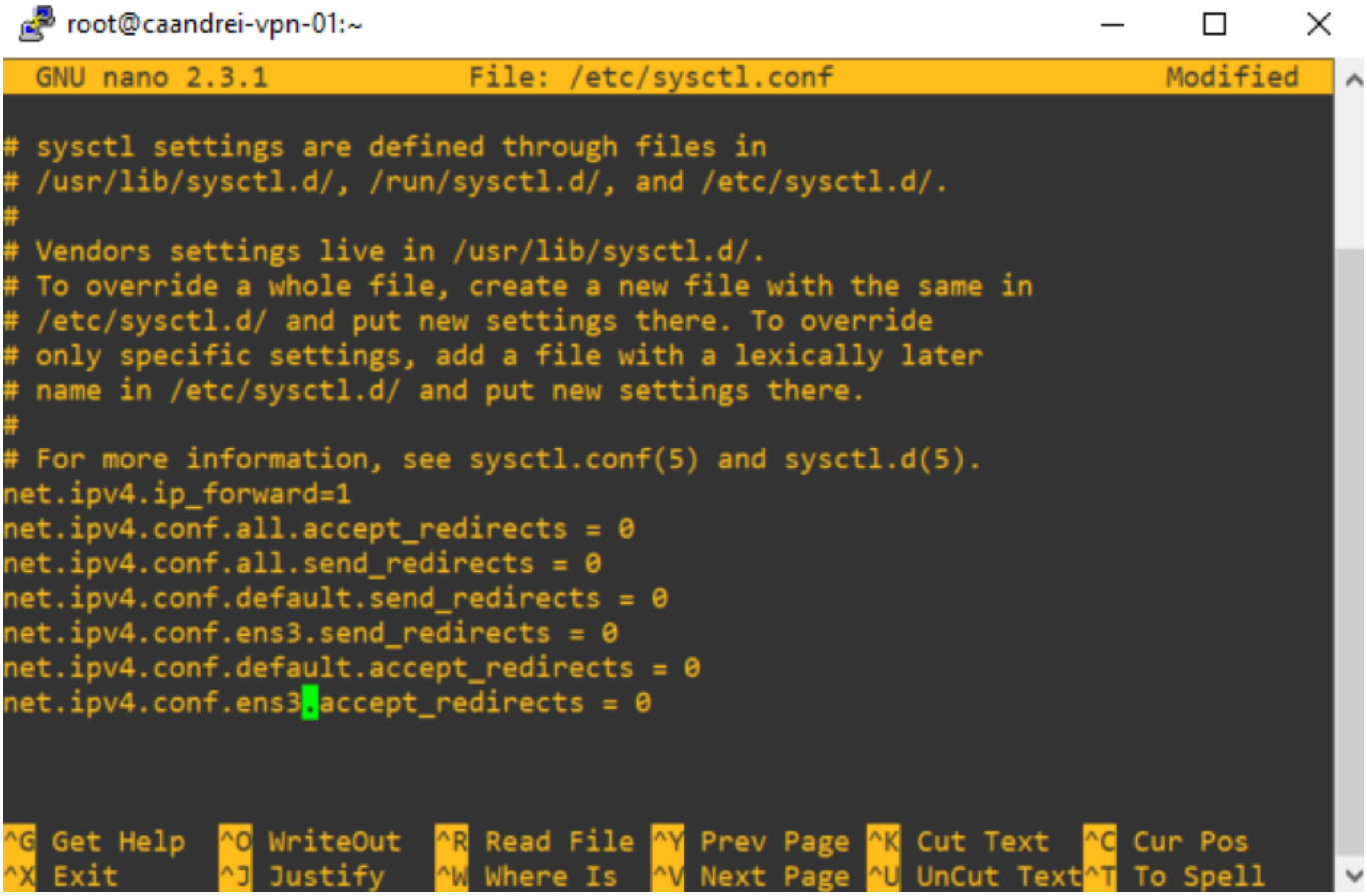
Scroll to the “Configuration Process” and start with Task 1. This will enable the routing on the Linux instance.

Login to the public VM from Lab1. Switch user to root and edit the /etc/sysctl.conf file

```
nano /etc/syscrl.conf
```



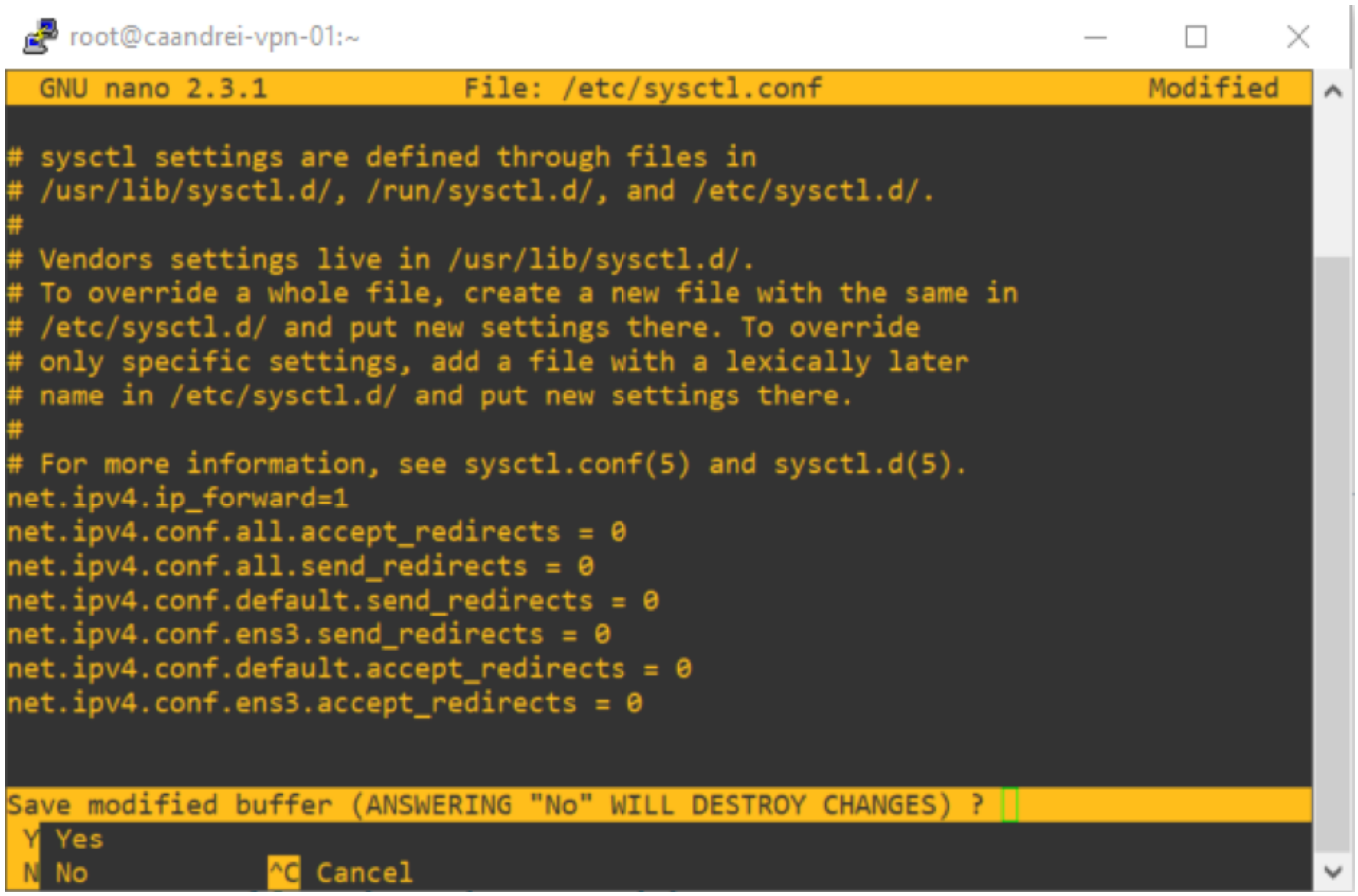
Add the routing information



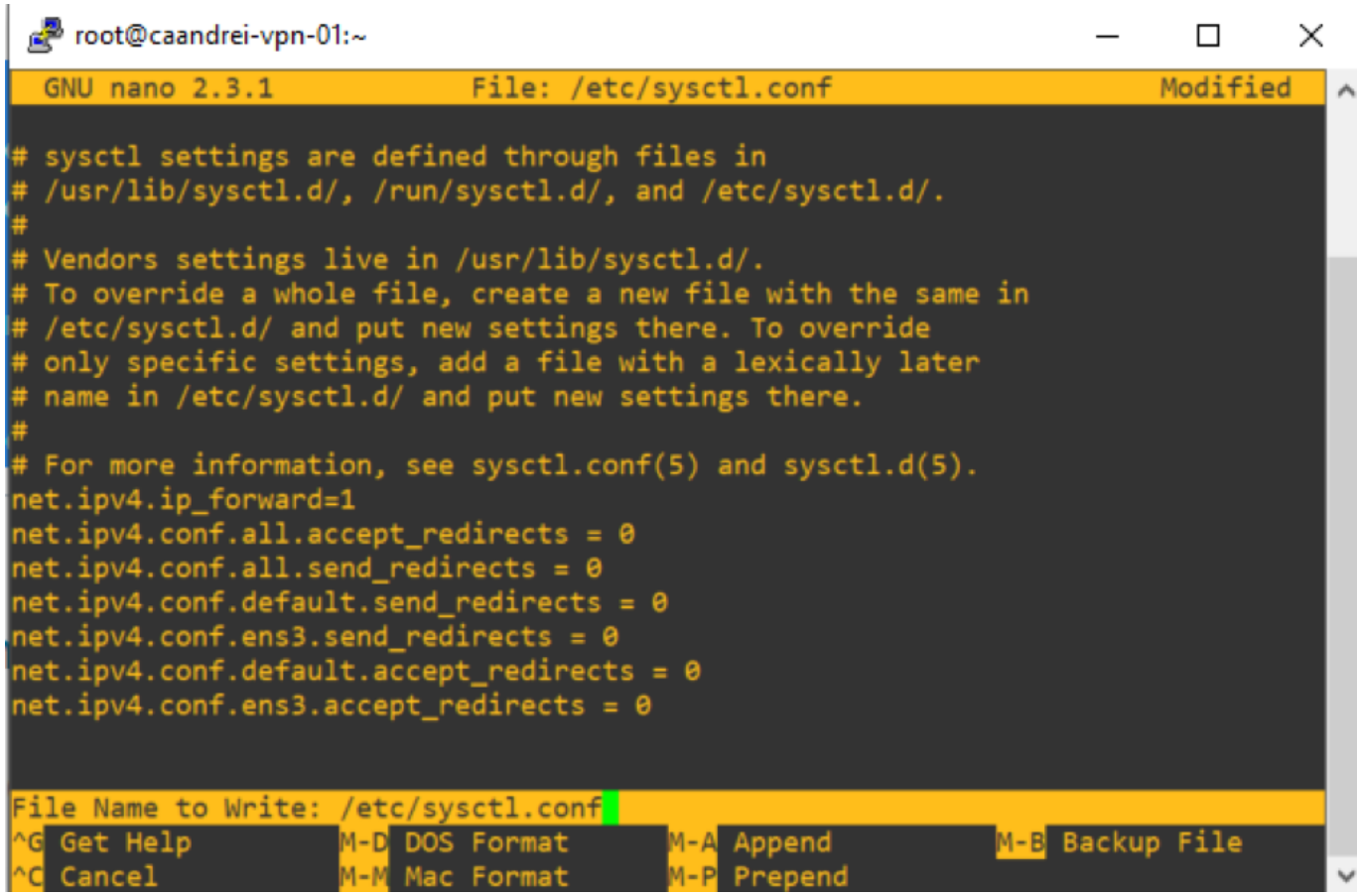
Please be aware that in the documentation, the interface name of the VM is eth0. You can check the name of your interface with ifconfig command.

In my screenshot, the interface name is ens3.

Save the config with ctrl+X and type “Y” to save



Hit “Enter” and confirm the filename.

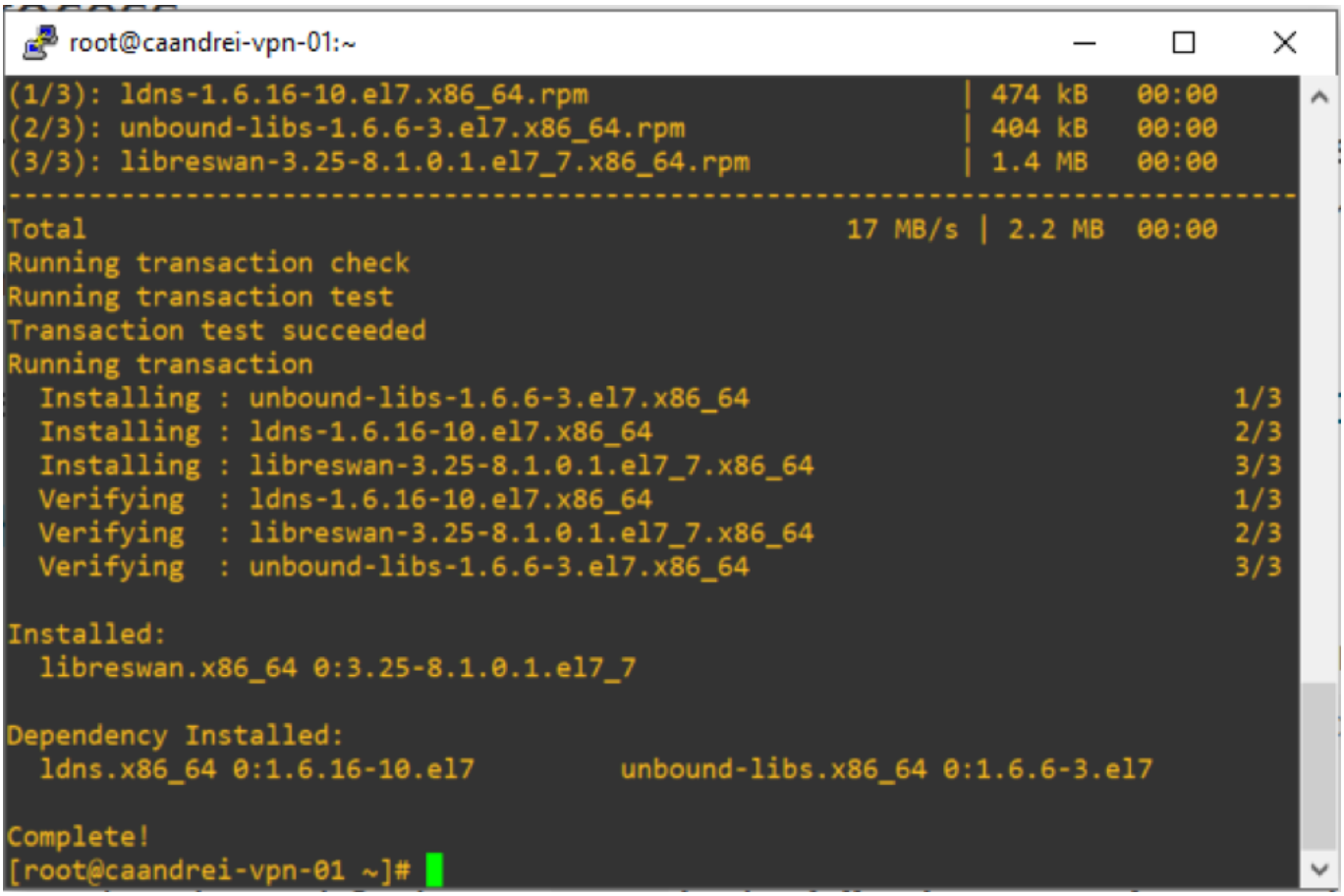


Reload the routing configuration with

```
sudo sysctl -p
```

Now install the Libreswan software

```
sudo yum install libreswan -y
```

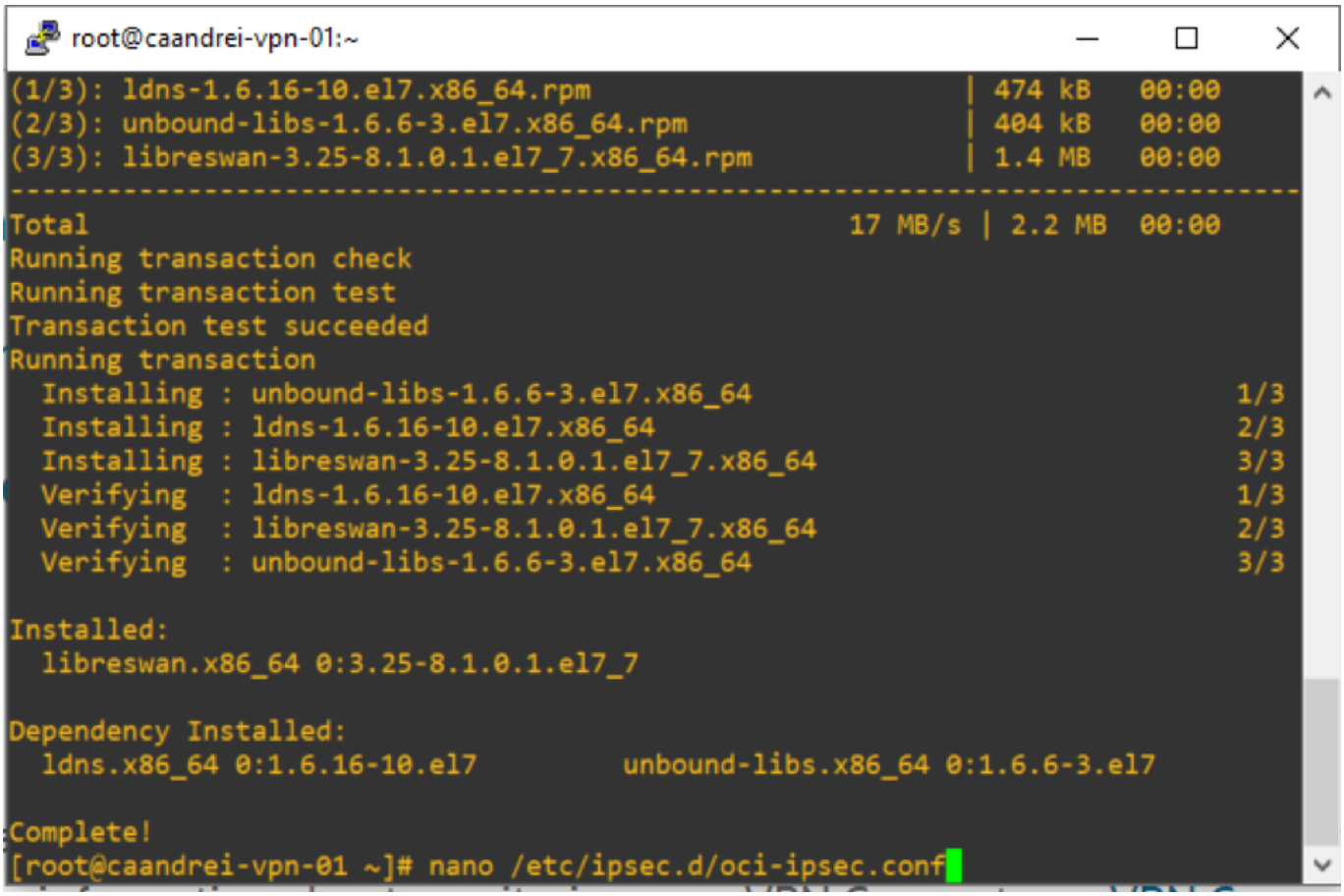


Proceed to Task 3 from the documentation and create the config file. Copy-paste the template in an editor (Notepad) and replace the variables with the information that we already have.

```
conn oracle-tunnel-1
      left=152.67.130.140
      # leftid=${cpePublicIpAddress} # See preceding note about 1-1 NAT device
      right=140.204.35.3
      authby=secret
      leftsubnet=0.0.0.0/0
      rightsubnet=0.0.0.0/0
      auto=start
      mark=5/0xffffffff # Needs to be unique across all tunnels
      vti-interface=vti1
      leftvti=10.10.10.1/30
      vti-routing=no
      ikev2=no # To use IKEv2, change to ikev2=insist
      ike=aes_cbc256-sha2_384;modp1536
      phase2alg=aes_gcm256;modp1536
      encapsulation=yes
      ikelifetime=28800s
      salifetime=3600s
conn oracle-tunnel-2
      left=152.67.130.140
      # leftid=${cpePublicIpAddress} # See preceding note about 1-1 NAT device
      right=140.204.35.14
      authby=secret
      leftsubnet=0.0.0.0/0
      rightsubnet=0.0.0.0/0
      auto=start
      mark=6/0xffffffff # Needs to be unique across all tunnels
      vti-interface=vti2
      leftvti=10.10.10.5/30
      vti-routing=no
      ikev2=no # To use IKEv2, change to ikev2=insist
      ike=aes_cbc256-sha2_384;modp1536
      phase2alg=aes_gcm256;modp1536
      encapsulation=yes
      ikelifetime=28800s
      salifetime=3600s
```

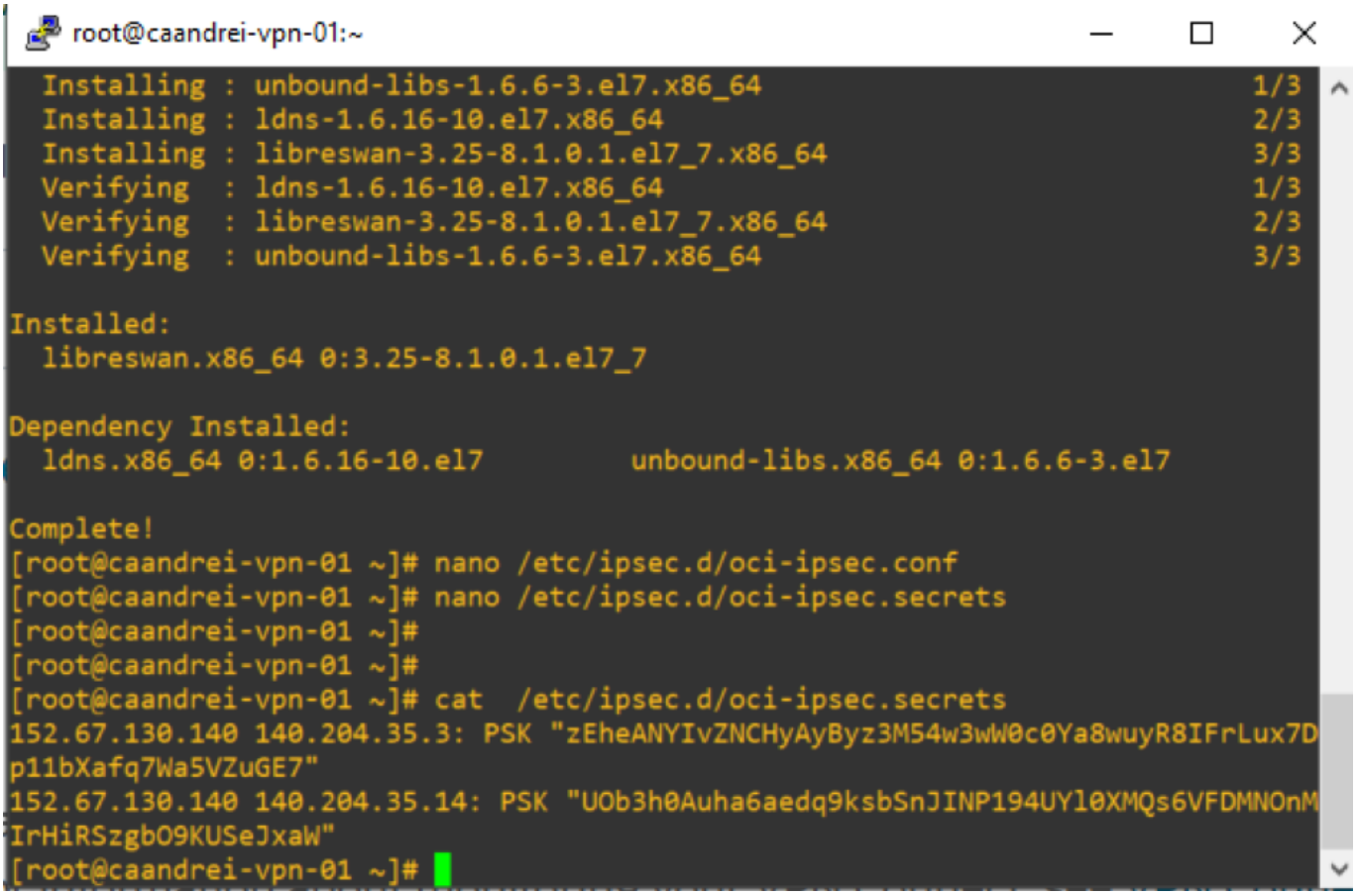
Notice that two things are modified from the template: the *vti* interface (I used *vti1* and *vti2*) and the ip addresses assigned to the *vti* interface (*"leftvti="*)

Take the config from the editor and create the config file (*/etc/ipsec.d/oci-ipsec.conf*)



Follow Task 4 and create the secrets file.

The file will look like this:



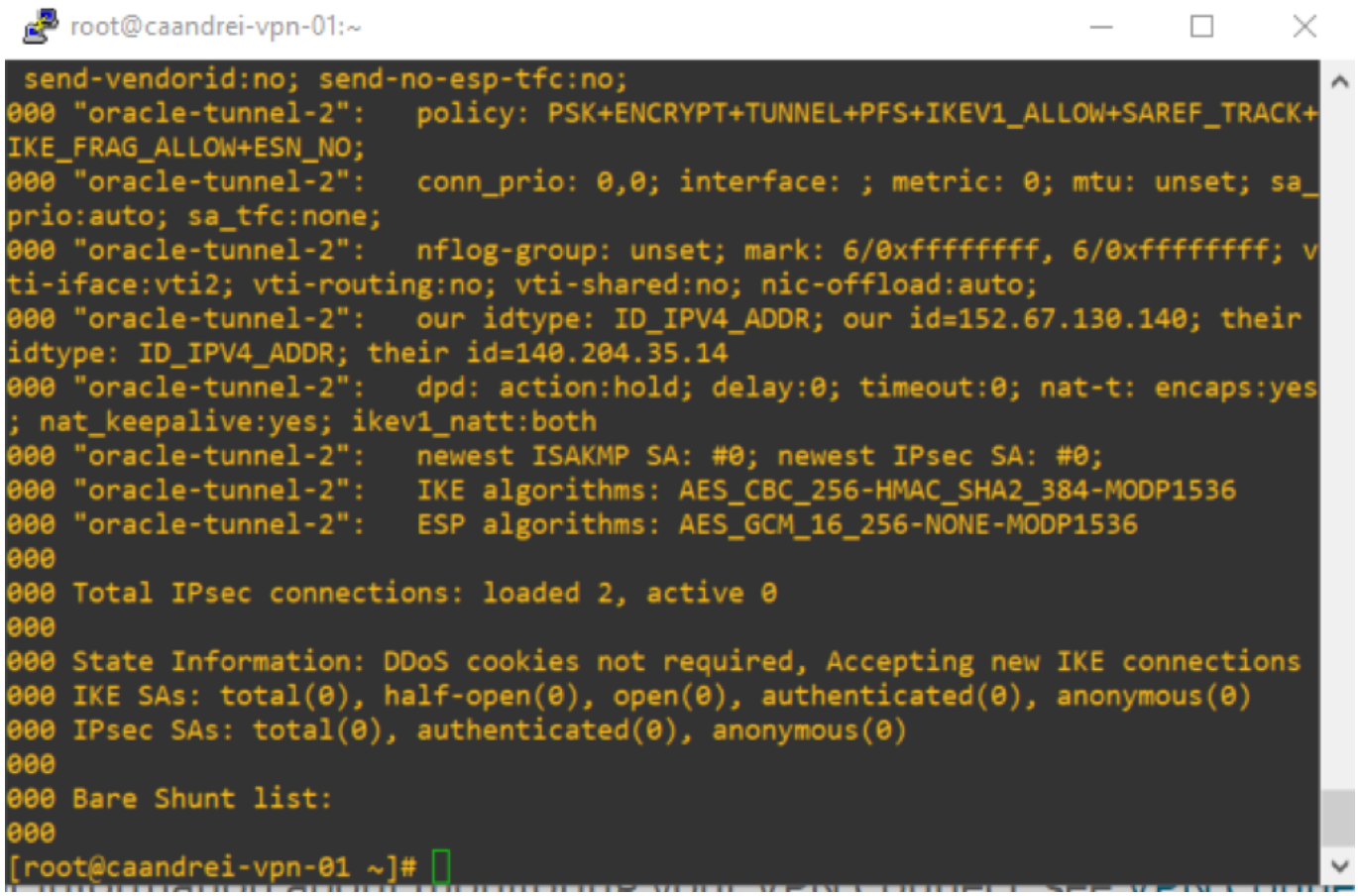
Start the ipsec service

```
sudo systemctl start ipsec
```

Check the status of the connection

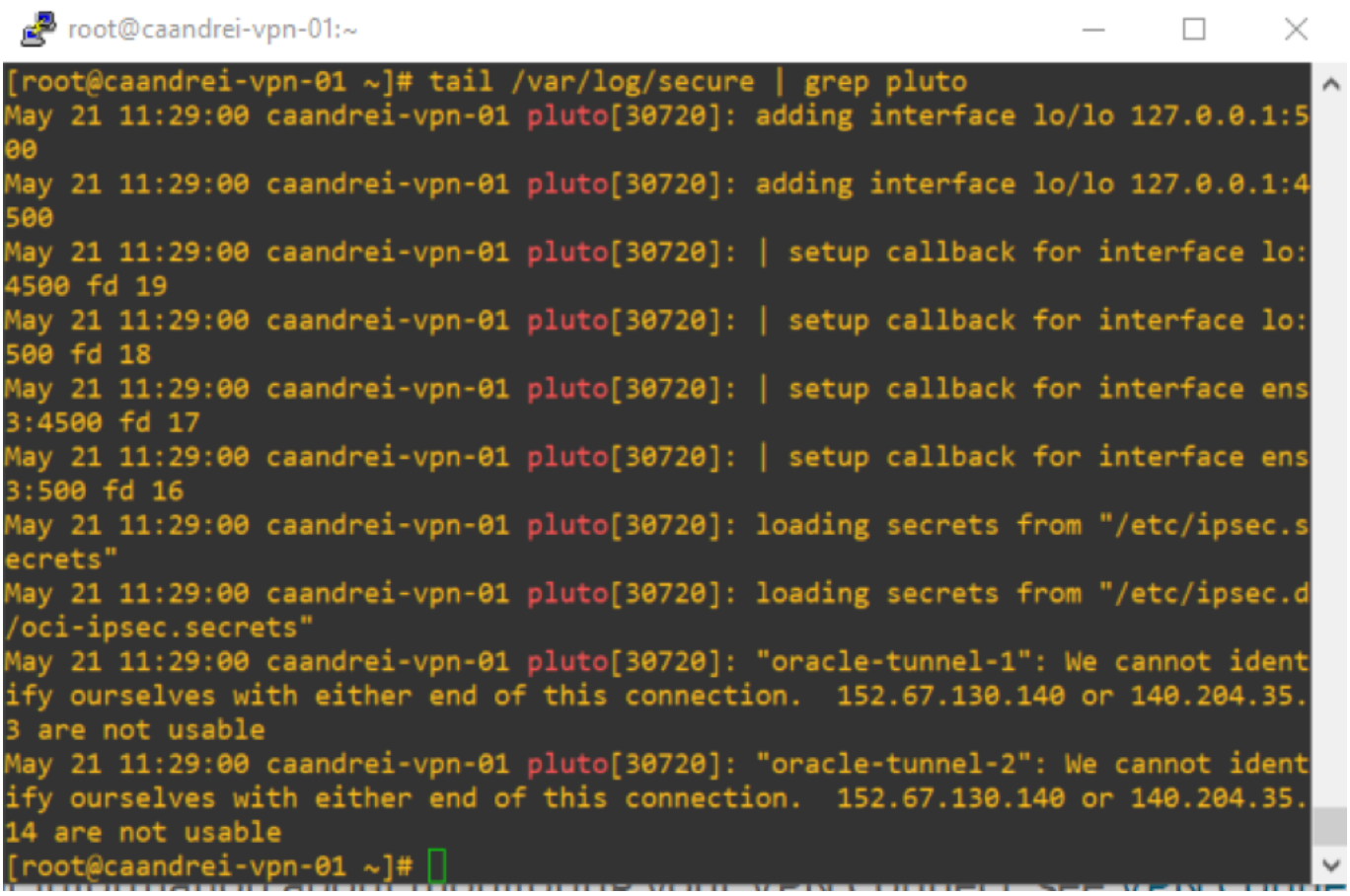
```
ipsec status
```

You will notice that the tunnels are not up.



Check the logs from the Libreswan

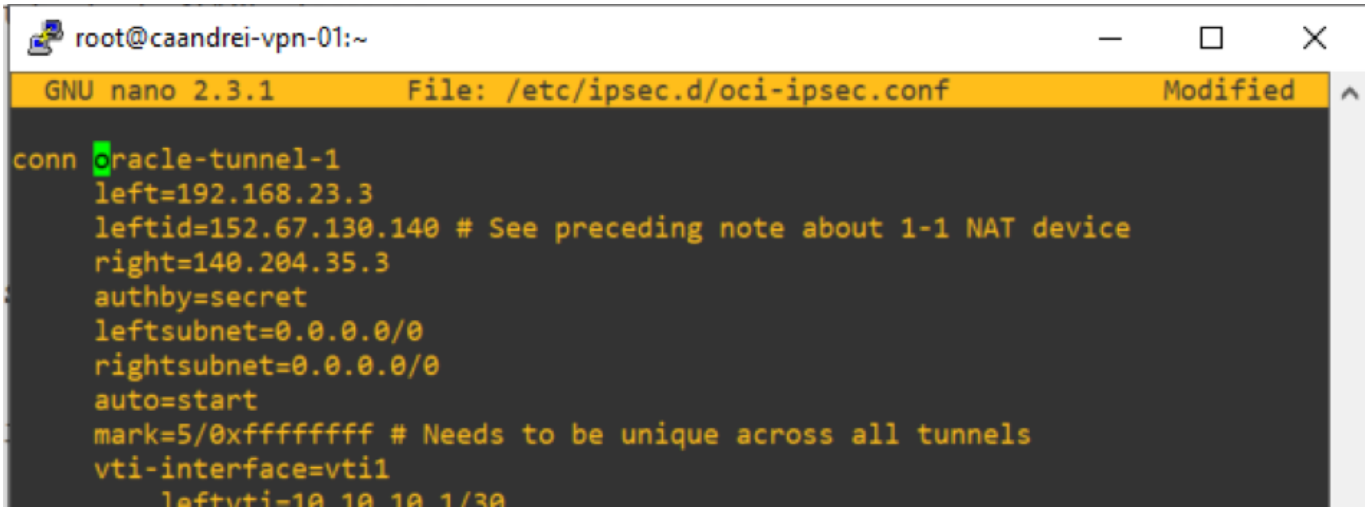
```
tail /var/log/secure | grep pluto
```



The following line in the log will help identify the problem

```
oracle-tunnel-1": We cannot identify ourselves with either end of this
connection.  152.67.130.140 or 140.204.35.3 are not usable
```

This means that we did not fully configure the libreswan. On ens3 interface we do not have a public IP address, so we need to adjust the config and use as left the private ip address configured on the interface and for the leftid the public ip address. Please make sure you adjust both of the tunnels.

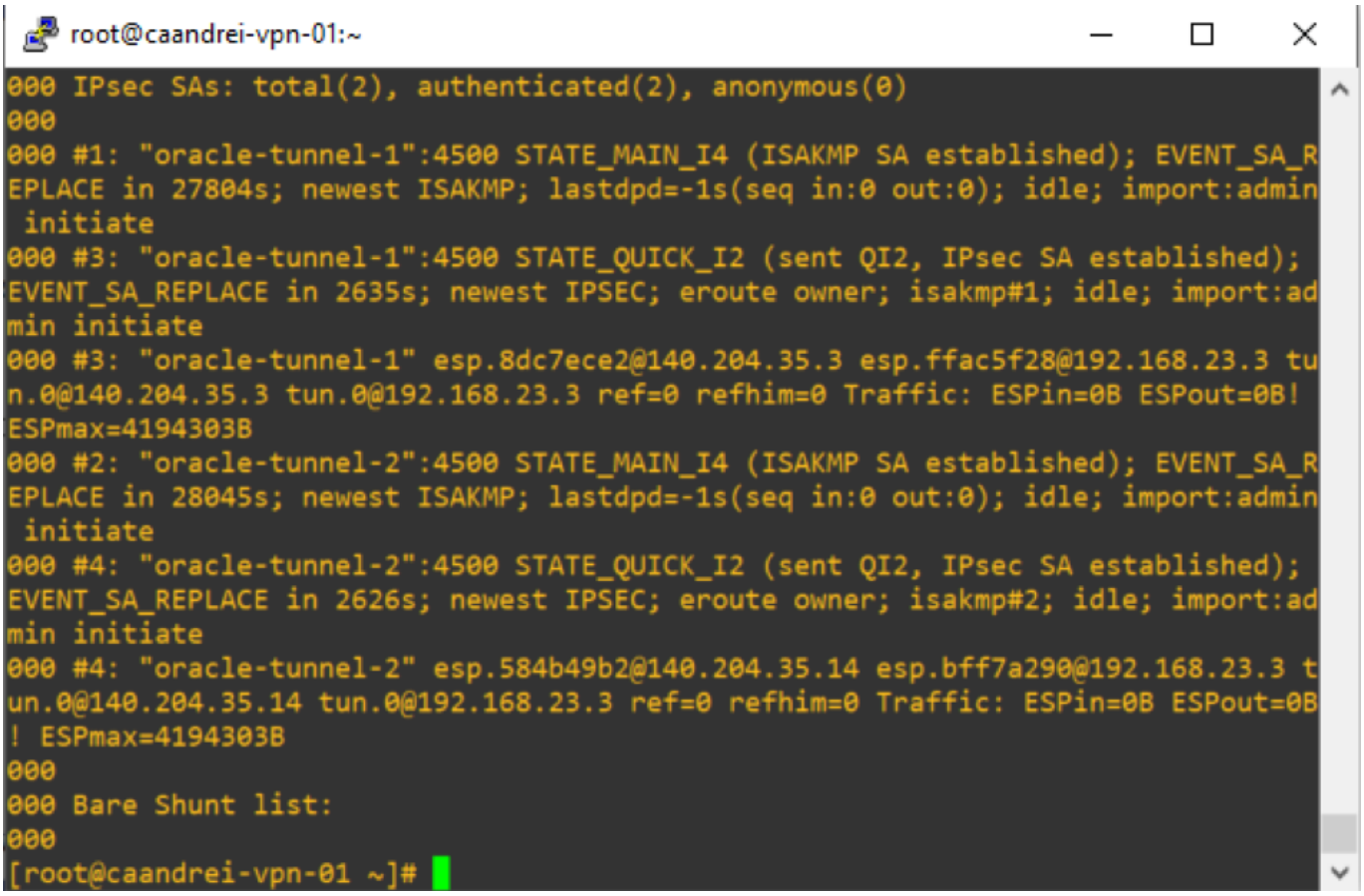


Restart the ipsec service

```
sudo systemctl restart ipsec
```

Check the status of the service

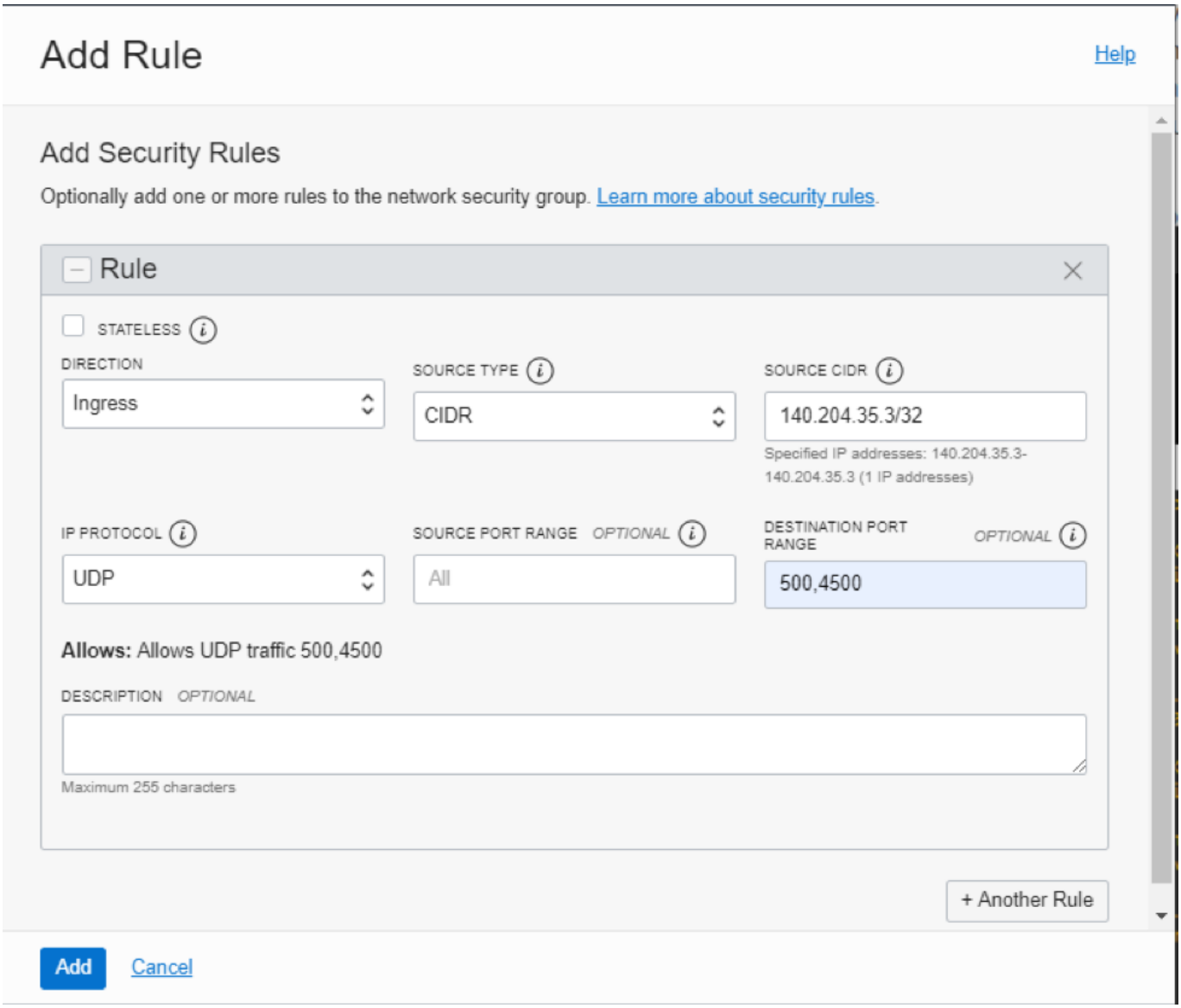
```
ipsec status
```



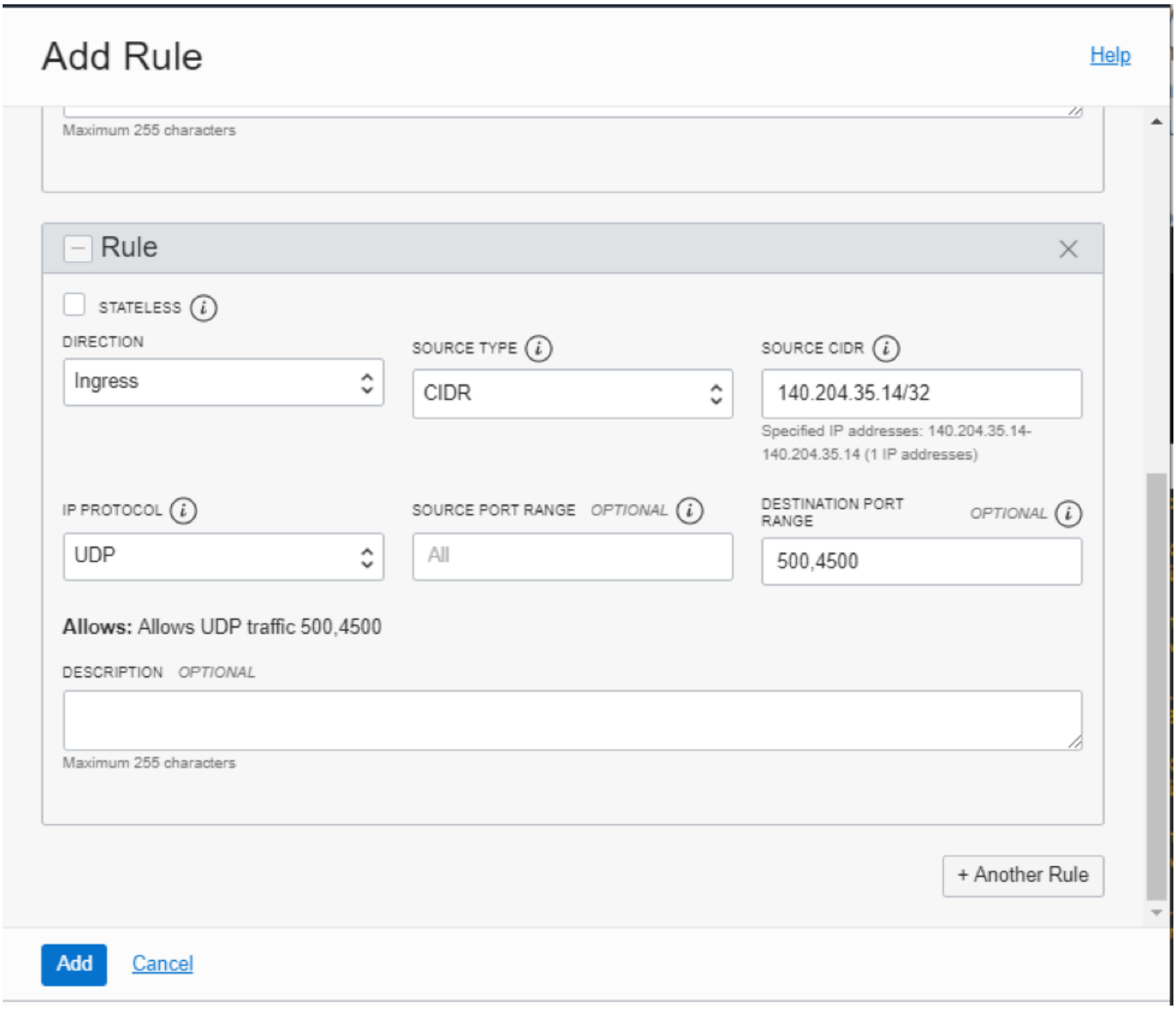
You can observe that the tunnels are up (IPsec SA established).

Setup Firewall Rules

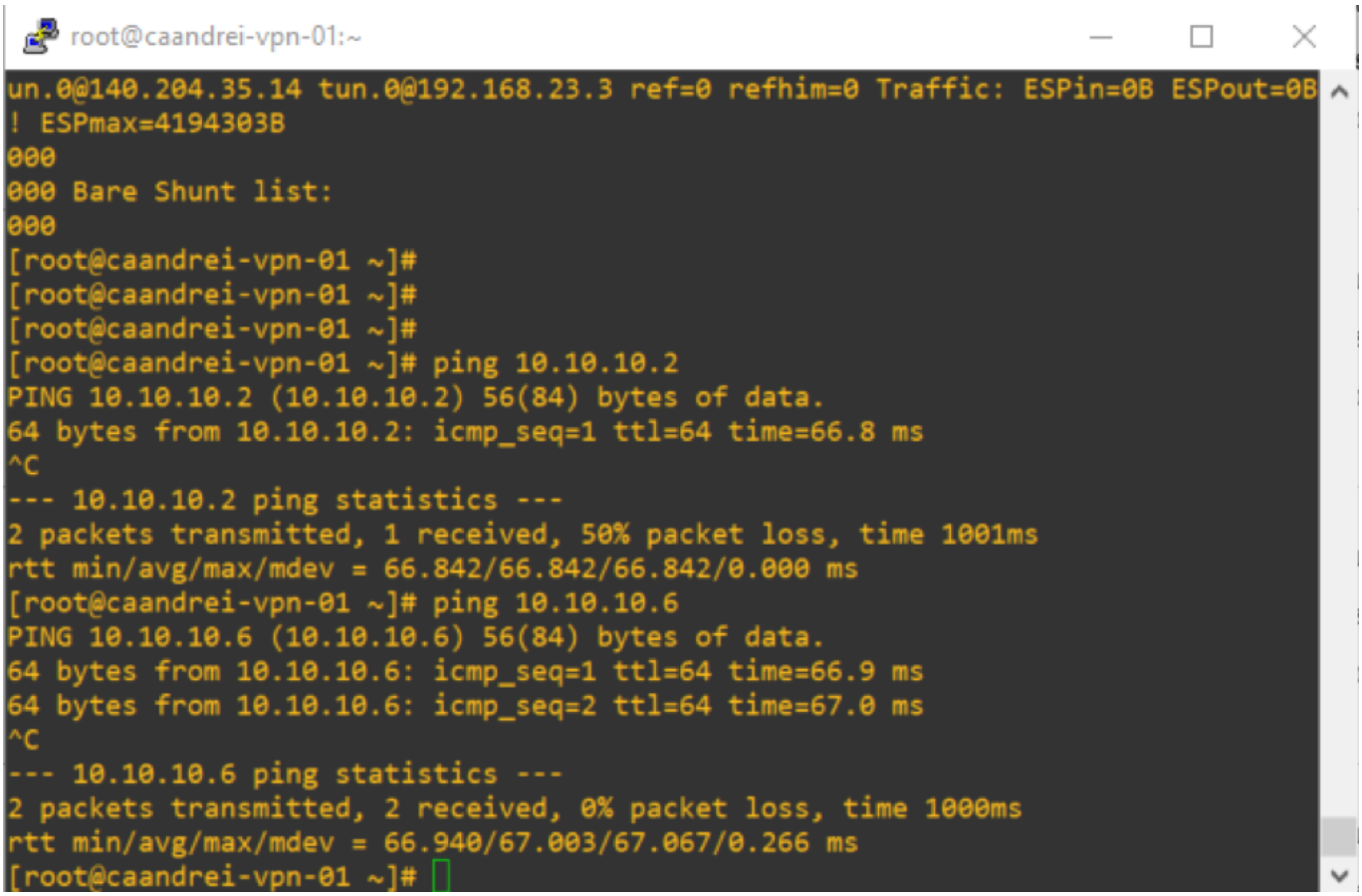
Adjust the NSG of the public VM:



Add a another rule for the second ipsec peer.



Push the “Add” button. Test the connectivity over the ipsec tunnel by pinging the other side tunnel interface



Install Quagga

Check the Ipsec connection the on DRG. You will notice that the IPsec tunnel is up but the bgp connection is down.

We need to install a bgp service on the public VM.

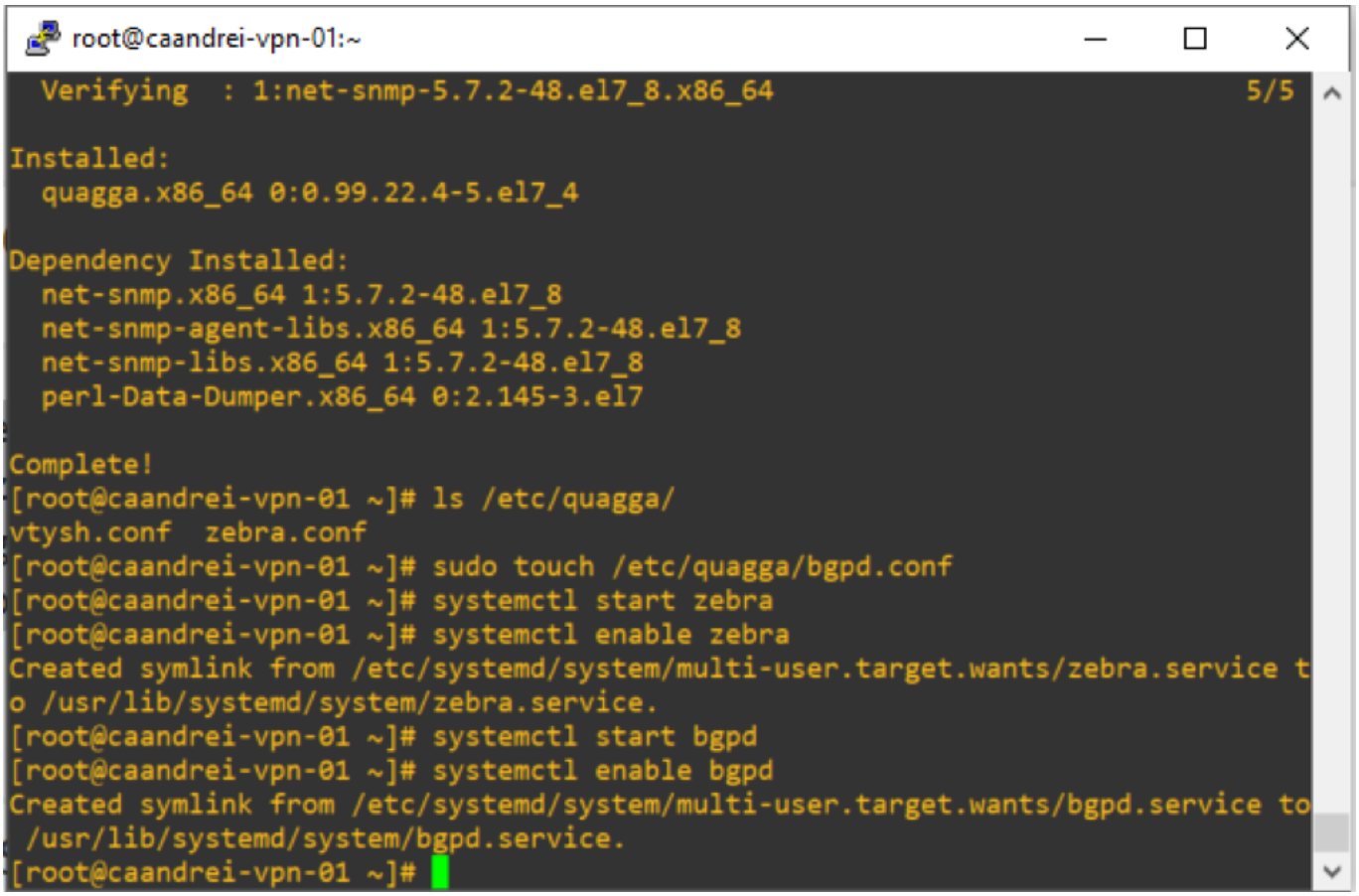
```
sudo yum install quagga –y
```

Create the bgpd config file:

```
sudo touch /etc/quagga/bgpd.conf
```

Enable quagga daemons

```
systemctl start zebra
systemctl enable zebra
systemctl start bgpd
systemctl enable bgpd
```



Enter in the quagga console

```
vtysh
```

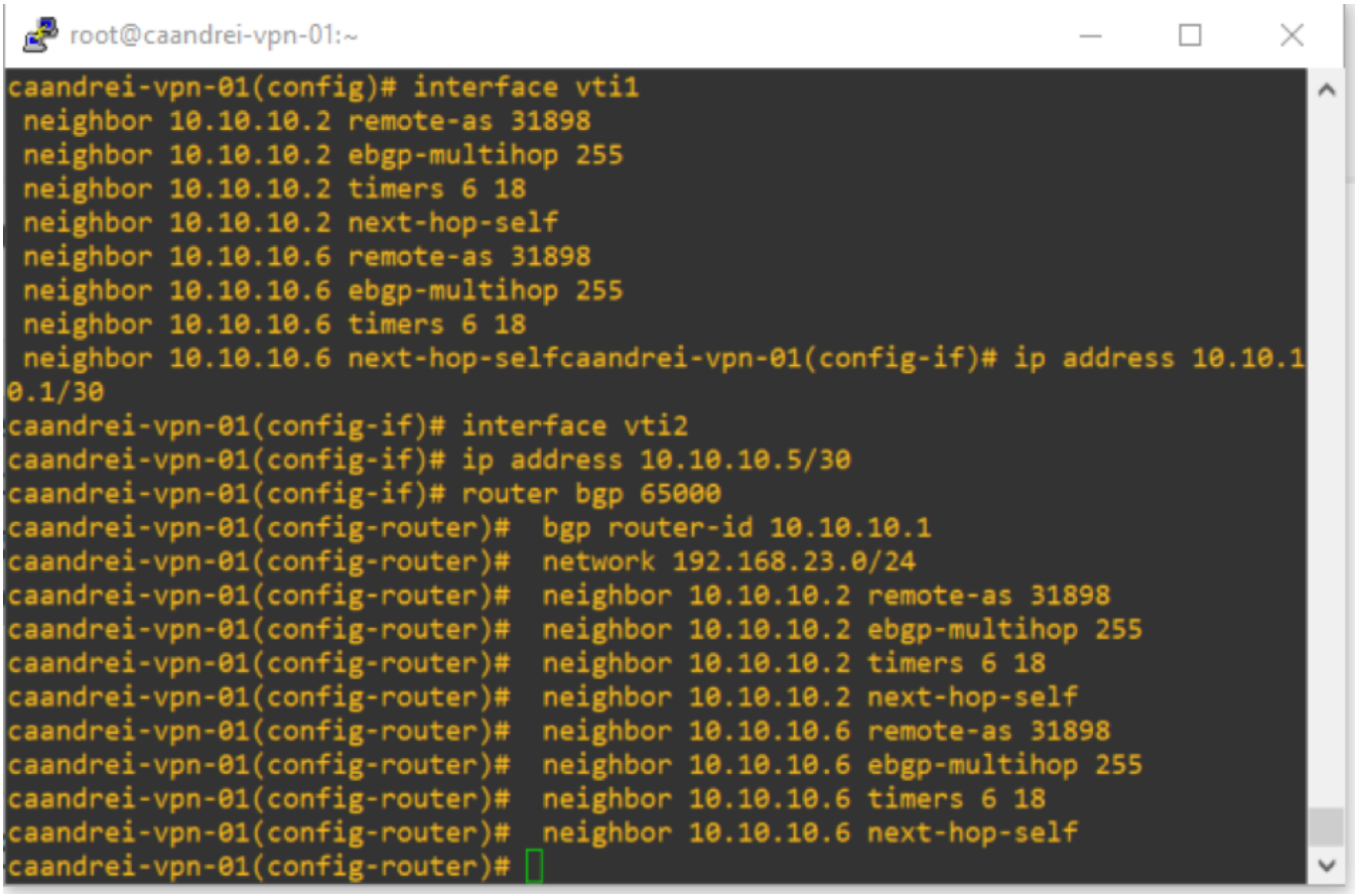
Enter in the configuration mode

```
conf t
```

Paste the following config:

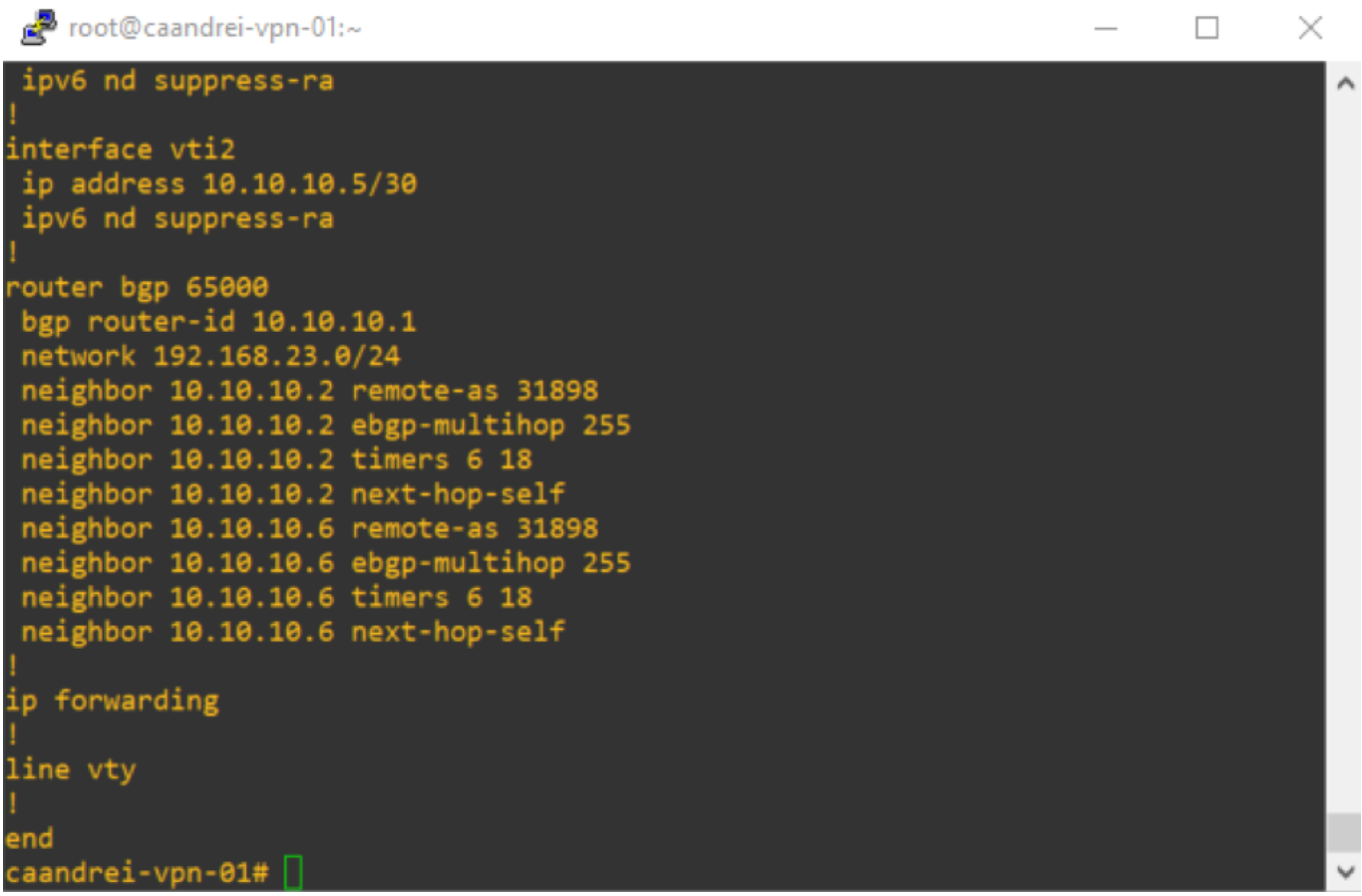
```
interface vti1
ip address 10.10.10.1/30
interface vti2
ip address 10.10.10.5/30
router bgp 65000
 bgp router-id 10.10.10.1
 network 192.168.23.0/24
 neighbor 10.10.10.2 remote-as 31898
 neighbor 10.10.10.2 ebgp-multihop 255
 neighbor 10.10.10.2 timers 6 18
 neighbor 10.10.10.2 next-hop-self
 neighbor 10.10.10.6 remote-as 31898
 neighbor 10.10.10.6 ebgp-multihop 255
```

```
   neighbor 10.10.10.6 timers 6 18
   neighbor 10.10.10.6 next-hop-self
```

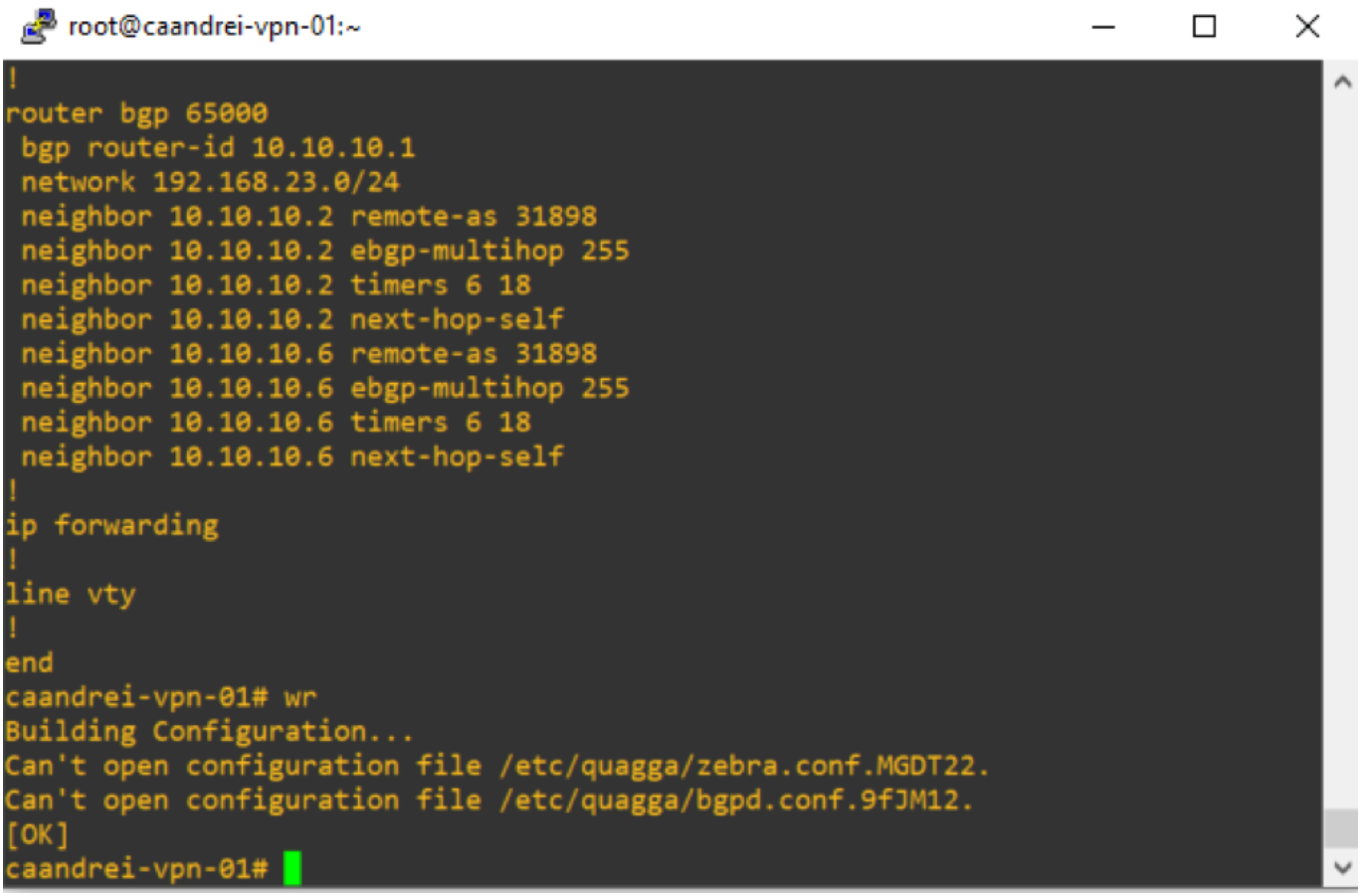


Issue CTRL+Z to exit the config mode. Check the configuration that you just pasted with

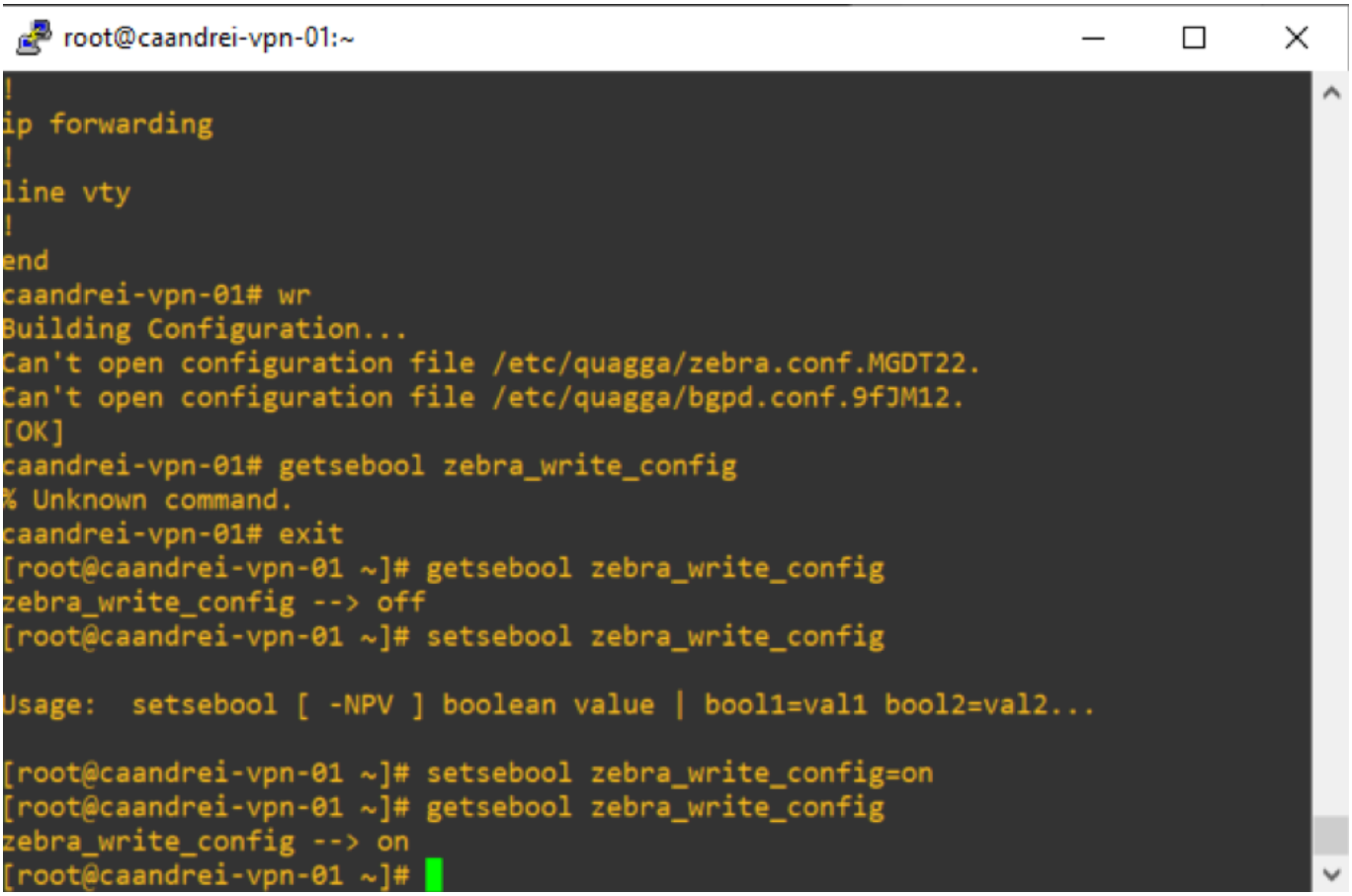
```
sh run
```



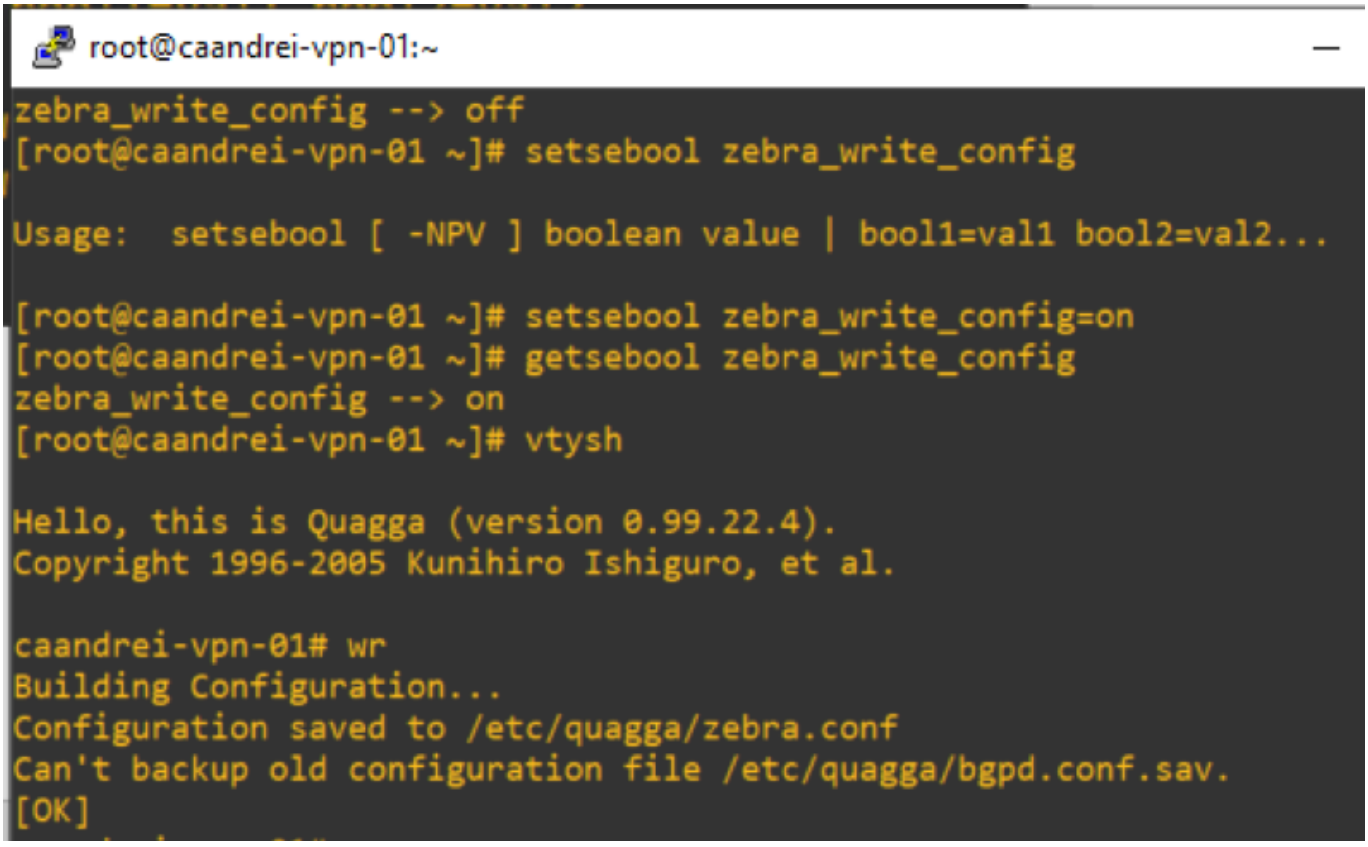
Issue *wr* command to save the config



You will notice that the config file can't be written. This can be fix by following the instruction here



Now return to *vtysh* and save the config



Tests

Check the BGP status

```
show ip bgp sum
```

```
caandrei-vpn-01# sh ip bgp summ
BGP router identifier 10.10.10.1, local AS number 65000
RIB entries 3, using 336 bytes of memory
Peers 2, using 9120 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.10.10.2      4 31898      72      72        0    0    0 00:06:38        1
10.10.10.6      4 31898      73      71        0    0    0 00:06:39        1

Total number of neighbors 2
```

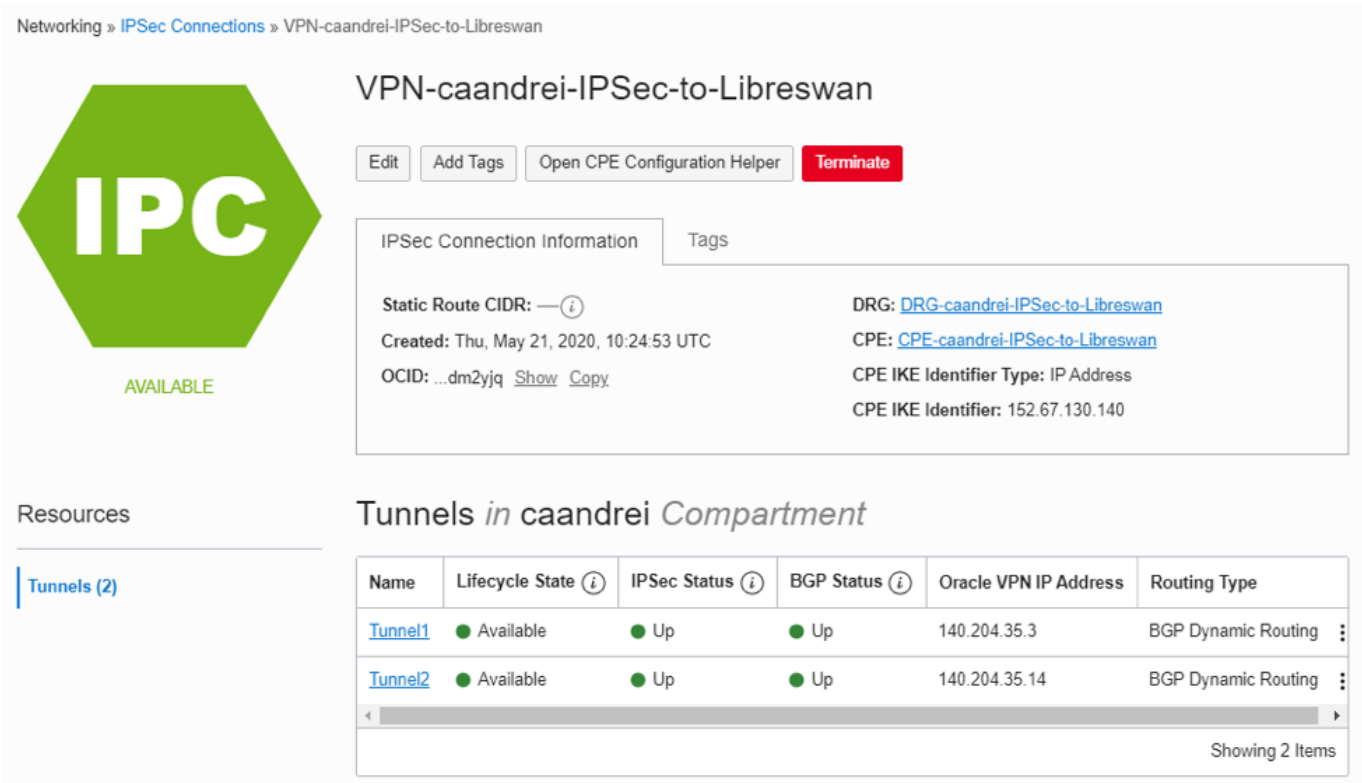
Display the routes learn by bgp

```
sh hip route bgp
```

```
caandrei-vpn-01# sh ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

B>* 192.168.24.0/24 [20/0] via 10.10.10.6, vti2, 00:07:46
```

Check the status of the IPsec connection on the DRG



Both bgp sessions are UP!