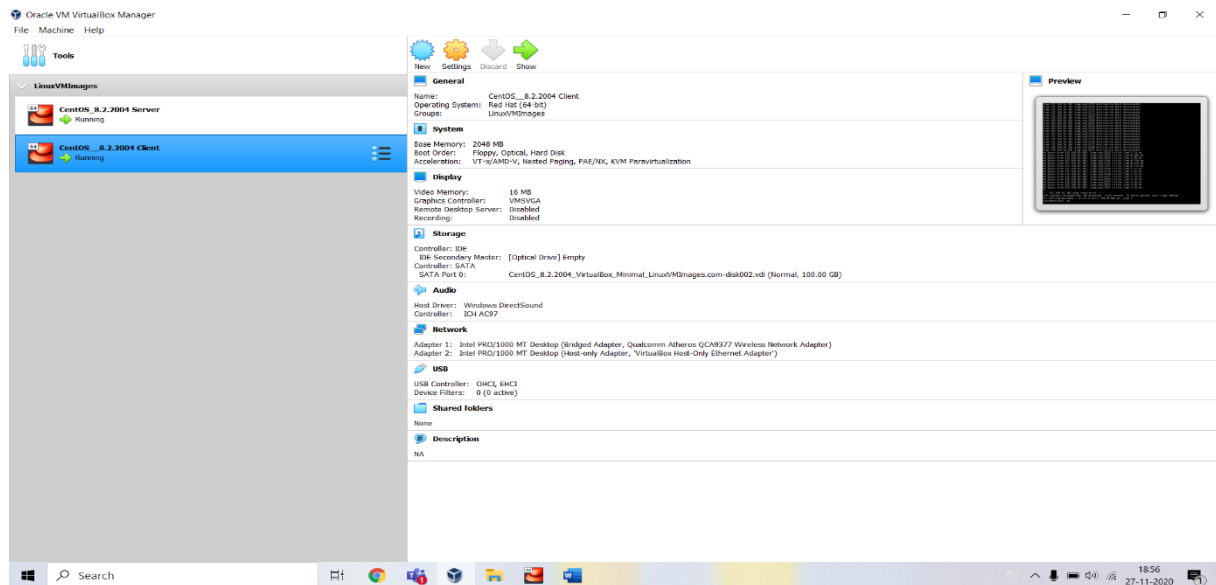# Report on VPN (Task 1)

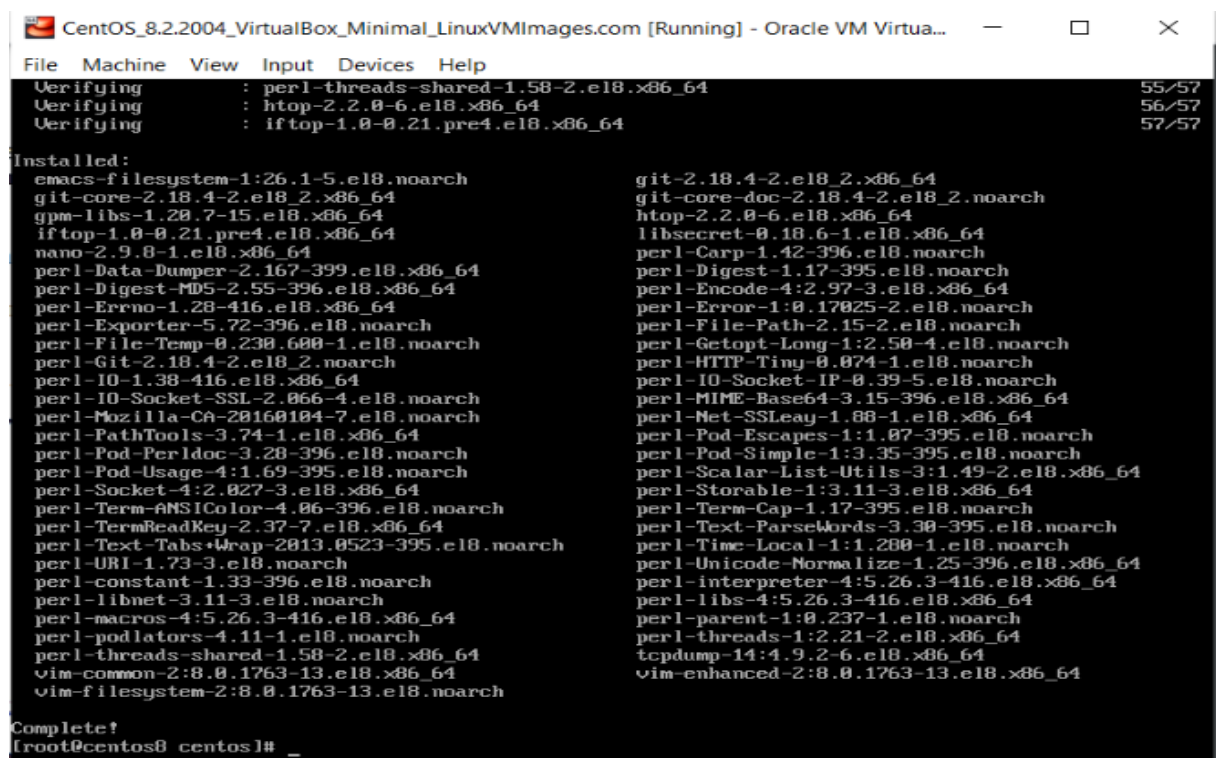**Name**: Aravindh Jawahar

Course – 1 Introduction

Virtual Machines:

VM Server and VM Client

Installing **EPEL** release packages and **TCPDUMP** for monitoring the **TCP** packets transmission and receiving.
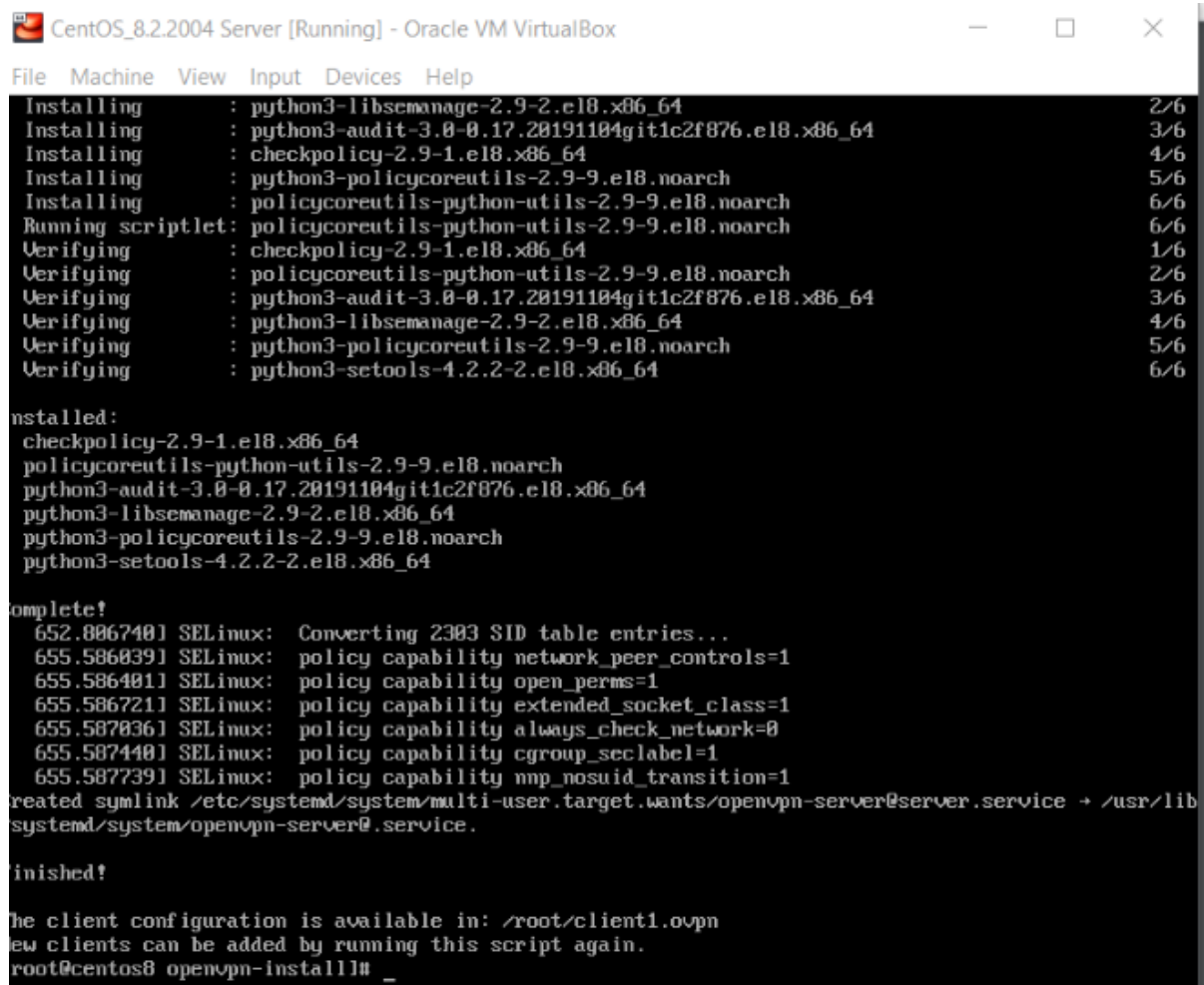
Cloning VPN installation script:

```
ash cd openvpn-install: No such file or directory
[root@centos8 ~]# git clone https://github.com/Nyr/openvpn-install.git
Cloning into 'openvpn-install'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 490 (delta 0), reused 1 (delta 0), pack-reused 487
Receiving objects: 100% (490/490), 157.79 KiB | 1.20 MiB/s, done.
Resolving deltas: 100% (254/254), done.
```

Installing **VPNServer** on Server machine:

```
[root@centos8 ~]# cd openvpn-install
[root@centos8 openvpn-install]# chmod +x openvpn-install.sh
[root@centos8 openvpn-install]#
```

```
CentOS_8.2.2004 Server [Running] - Oracle VM VirtualBox          —    □    ×

File  Machine  View  Input  Devices  Help
Installing        : python3-libsemanage-2.9-2.el8.x86_64                    2/6
Installing        : python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64    3/6
Installing        : checkpolicy-2.9-1.el8.x86_64                            4/6
Installing        : python3-policycoreutils-2.9-9.el8.noarch               5/6
Installing        : policycoreutils-python-utils-2.9-9.el8.noarch          6/6
Running scriptlet: policycoreutils-python-utils-2.9-9.el8.noarch          6/6
Verifying         : checkpolicy-2.9-1.el8.x86_64                            1/6
Verifying         : policycoreutils-python-utils-2.9-9.el8.noarch          2/6
Verifying         : python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64    3/6
Verifying         : python3-libsemanage-2.9-2.el8.x86_64                    4/6
Verifying         : python3-policycoreutils-2.9-9.el8.noarch               5/6
Verifying         : python3-setools-4.2.2-2.el8.x86_64                      6/6

Installed:
  checkpolicy-2.9-1.el8.x86_64
  policycoreutils-python-utils-2.9-9.el8.noarch
  python3-audit-3.0-0.17.20191104git1c2f876.el8.x86_64
  python3-libsemanage-2.9-2.el8.x86_64
  python3-policycoreutils-2.9-9.el8.noarch
  python3-setools-4.2.2-2.el8.x86_64

Complete!
  652.806740] SELinux:  Converting 2303 SID table entries...
  655.586039] SELinux:  policy capability network_peer_controls=1
  655.586401] SELinux:  policy capability open_perms=1
  655.586721] SELinux:  policy capability extended_socket_class=1
  655.587036] SELinux:  policy capability always_check_network=0
  655.587440] SELinux:  policy capability cgroup_seclabel=1
  655.587739] SELinux:  policy capability nnp_nosuid_transition=1
Created symlink /etc/systemd/system/multi-user.target.wants/openvpn-server@server.service → /usr/lib
systemd/system/openvpn-server@.service.

Finished!

The client configuration is available in: /root/client1.ovpn
New clients can be added by running this script again.
[root@centos8 openvpn-install]# _
```
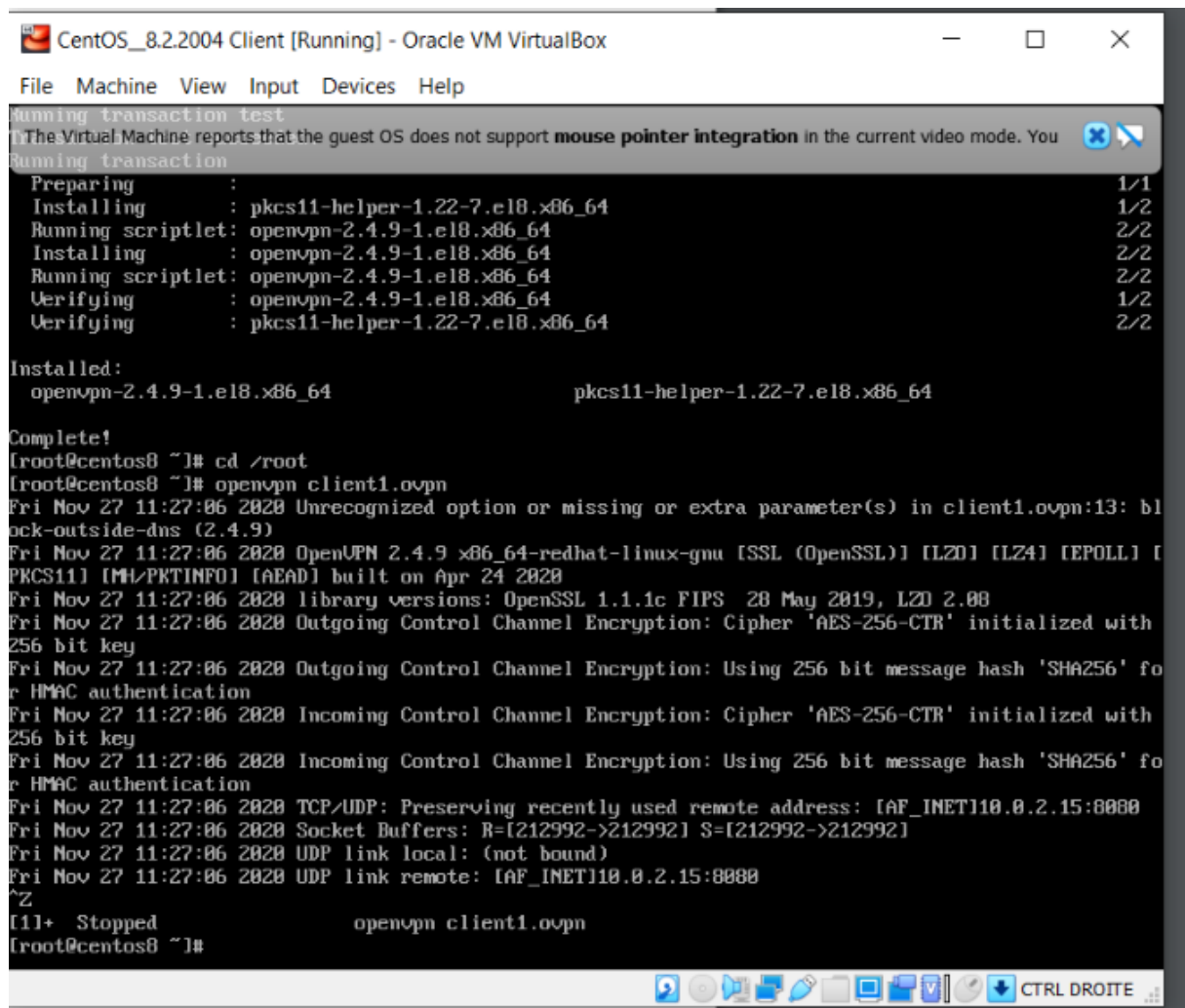
scp command is used to get the client1.ovpn from server system.

```
[root@centos8 ~]# scp root@192.168.56.102:/root/client1.ovpn .
root@192.168.56.102's password:
client1.ovpn                                100%  4989    528.8KB/s    00:00
[root@centos8 ~]# _
```

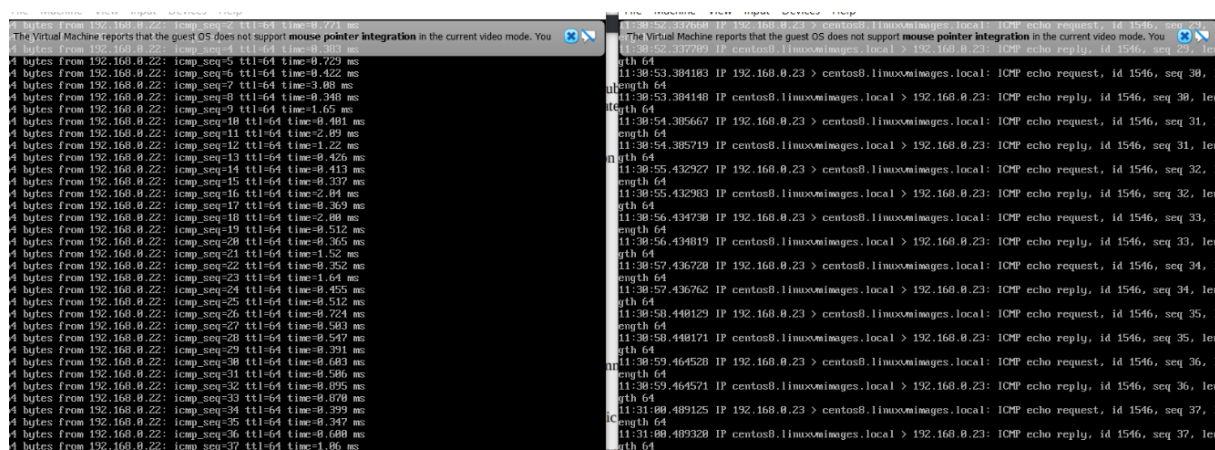Transferring client1.ovpn file from Server to client using **SCP** command and **credentials**:

Ping on server physical IP address and it correspondent traffic on server

Ping on tunnel IP

(11) We cannot see the traffic on server for tunnel IP because tunnel is in virtual private network and it is not known by the client .

(12) In server machine we installed VPN server (open VPN) and started it by setting the transfer methods(UDP) along with the physical IP address of the server. Now the tunnel is created on the server machine with its corresponding IP address makes the virtual network. OVPN file which act as a certificate to communicate the server from client , this can be achieved by scp command. A ping on the server with its physical interface and its traffic can be seen on server using the tcpdump command. While ping on tunnel interface address , ping gets successful and we cannot see the traffic of ICMP on server because of its encryption over the virtual private network.