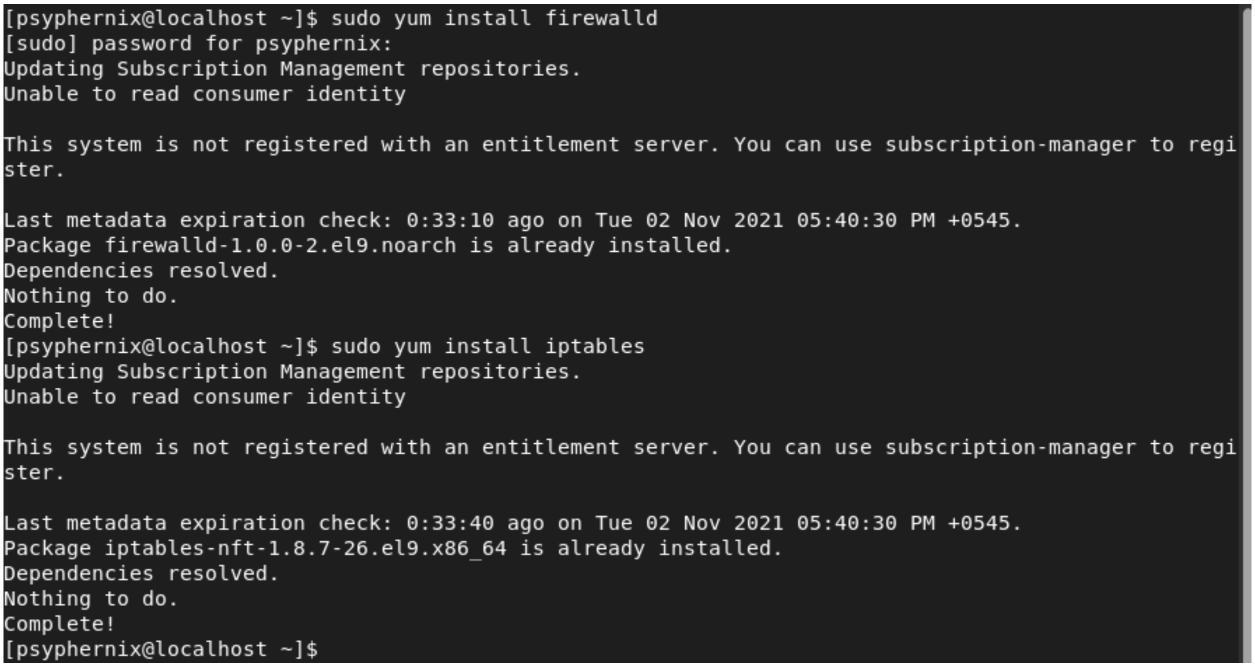
1. The latest CentOS Stream 9 has been installed in VMware Workstation Pro 16 with the following specs:
   - Processor: 4
   - RAM: 4 GB
   - Hard Disk: 50GB
   - Network Adapter: Host-only

a. Firewalld and iptables can be installed with the following commands on CentOS though it was preinstalled:

**$ sudo yum install firewalld**
**$ sudo yum install iptables**



```
[psyphernix@localhost ~]$ sudo yum install firewalld
[sudo] password for psyphernix:
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to regi
ster.

Last metadata expiration check: 0:33:10 ago on Tue 02 Nov 2021 05:40:30 PM +0545.
Package firewalld-1.0.0-2.el9.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[psyphernix@localhost ~]$ sudo yum install iptables
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to regi
ster.

Last metadata expiration check: 0:33:40 ago on Tue 02 Nov 2021 05:40:30 PM +0545.
Package iptables-nft-1.8.7-26.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[psyphernix@localhost ~]$
```

*Figure 1 Installing firewalld and iptables in CentOS*

b. To block certain IP ranges, rich rules from firewalld package can be used. 192.168.200.0/24 subnet is being blocked in the following command:

**$ sudo firewall-cmd --permanent --add-rich-rule="rulefamily='ipv4' source address='192.168.200.0/24' reject'**

- --permanent makes rules permanent even after system reboot.
- A rule can be made either to block incoming or outgoing traffic using source and destination respectively. In the above command, all inbound traffic of IP version 4 is blocked.

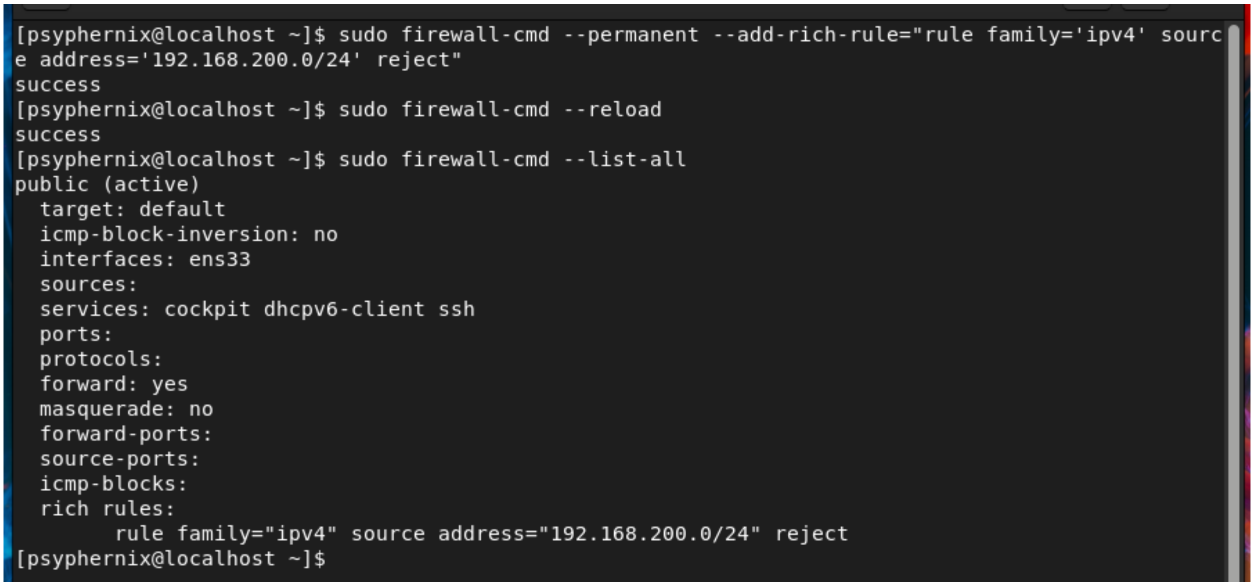
- Reject at the end of command means the device will be notified with their request being rejected.

For changes to come into effect, reload firewalld service using the command:

**$ sudo firewall-cmd –reload**

To check changes, the following command should be used:

**$ sudo firewall-cmd –list-all**



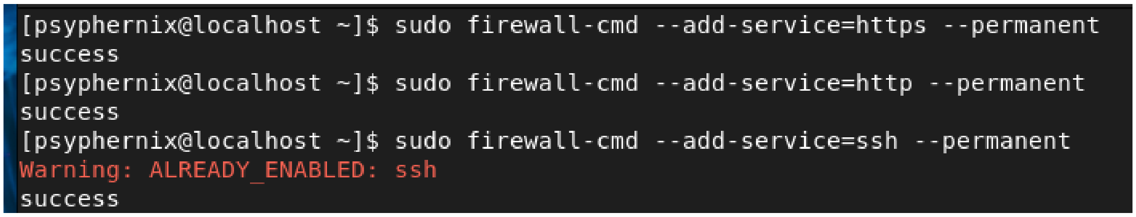
```
[psyphernix@localhost ~]$ sudo firewall-cmd --permanent --add-rich-rule="rule family='ipv4' sourc
e address='192.168.200.0/24' reject"
success
[psyphernix@localhost ~]$ sudo firewall-cmd --reload
success
[psyphernix@localhost ~]$ sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule family="ipv4" source address="192.168.200.0/24" reject
[psyphernix@localhost ~]$
```

*Figure 2 Blocking subnet using firewalld*

c. To allow HTTP, HTTPS, and ssh; and to make changes permanent following commands should be used:

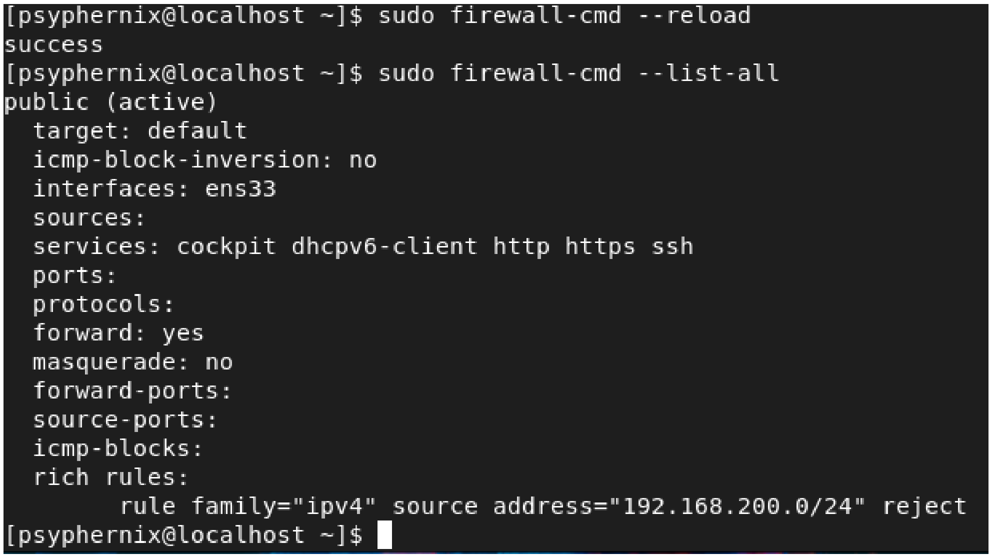
**$ sudo firewall-cmd –add-service=https –permanent**
**$ sudo firewall-cmd –add-service=http --permanent**
**$ sudo firewall-cmd –add-service=ssh –permanent**

Note: --add-service will allow using the default port of the service, if changes has been made to port –add-port or –remove-port should be used.



```
[psyphernix@localhost ~]$ sudo firewall-cmd --add-service=https --permanent
success
[psyphernix@localhost ~]$ sudo firewall-cmd --add-service=http --permanent
success
[psyphernix@localhost ~]$ sudo firewall-cmd --add-service=ssh --permanent
Warning: ALREADY_ENABLED: ssh
success
```

*Figure 3 Blocking a service using firewalld*



```
[psyphernix@localhost ~]$ sudo firewall-cmd --reload
success
[psyphernix@localhost ~]$ sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: cockpit dhcpv6-client http https ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule family="ipv4" source address="192.168.200.0/24" reject
[psyphernix@localhost ~]$
```

*Figure 4 Reloading firewalld and checking changes.*

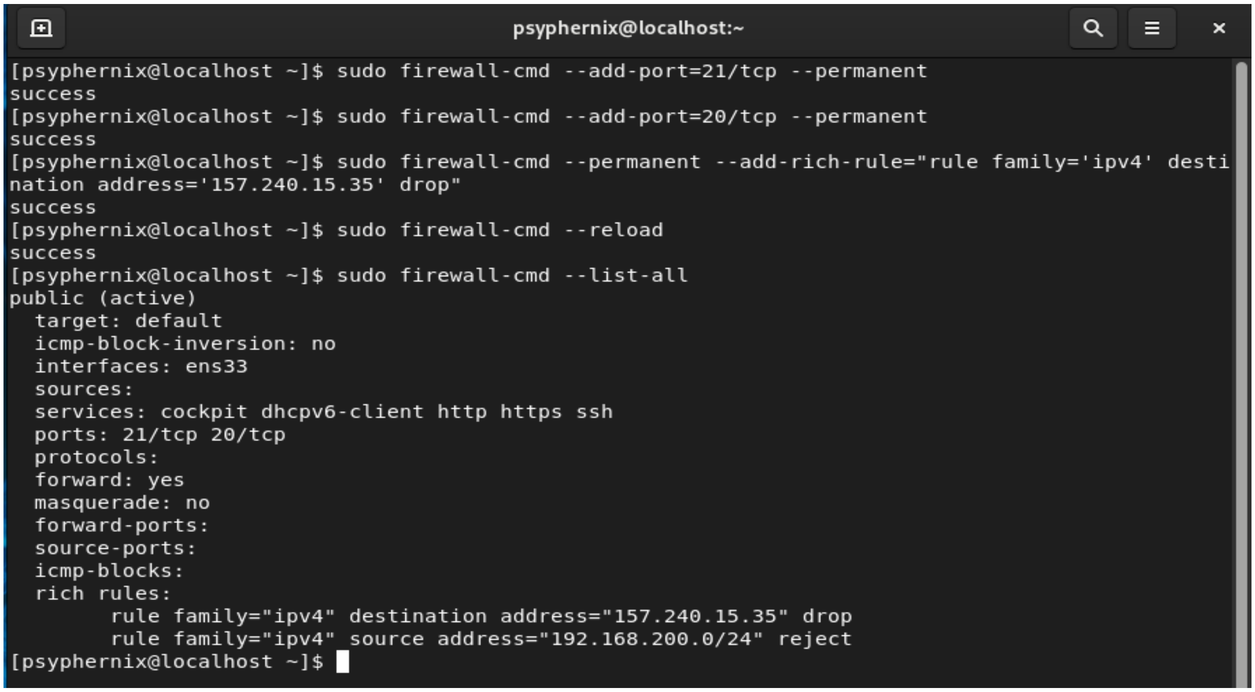
d. To allow or block any port, --add-port or –remove-port should be used respectively. To allow FTP service permanently, which uses two ports – 20 for data and 21 for control – both using TCP, the following command should be used:

**$ sudo firewall-cmd –add-port=20/tcp –permanent**
**$ sudo firewall-cmd –add-port=21/tcp –permanent**

To block certain IP addresses using rich and make it permanent, the following command should be used:

**$ sudo firewall-cmd –permanent –add-rich-rule=”rule family=’ipv4’ destination address=’157.240.15.35’ drop”**

In this command, 157.240.15.35 will be blocked, which is one of the IPs of facebook.com, for anyone accessing this site, without notifying.

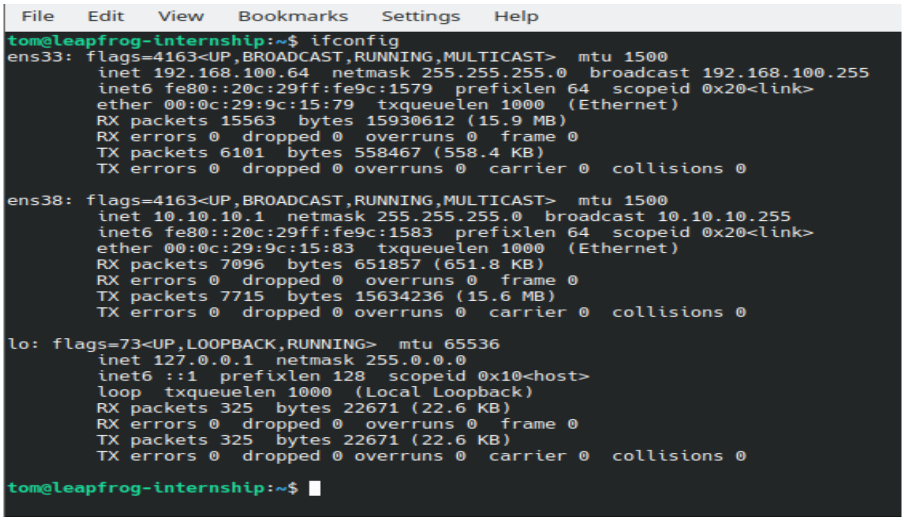


```
[psyphernix@localhost ~]$ sudo firewall-cmd --add-port=21/tcp --permanent
success
[psyphernix@localhost ~]$ sudo firewall-cmd --add-port=20/tcp --permanent
success
[psyphernix@localhost ~]$ sudo firewall-cmd --permanent --add-rich-rule="rule family='ipv4' desti
nation address='157.240.15.35' drop"
success
[psyphernix@localhost ~]$ sudo firewall-cmd --reload
success
[psyphernix@localhost ~]$ sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: cockpit dhcpv6-client http https ssh
  ports: 21/tcp 20/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule family="ipv4" destination address="157.240.15.35" drop
        rule family="ipv4" source address="192.168.200.0/24" reject
[psyphernix@localhost ~]$ 
```

*Figure 5 Allowing FTP using port number and blocking certain IP.*

2. Ubuntu has been installed in VM with the following specification:
   - Processor: 4
   - RAM: 4 GB
   - Hard Disk: 50 GB
   - Network Adaptor 1: Bridged mode (using an inbuilt wireless card of the laptop)
   - Network Adaptor 2: Host-only

   To use Ubuntu as a router, the first thing is to do is to configure the WAN and LAN interface.



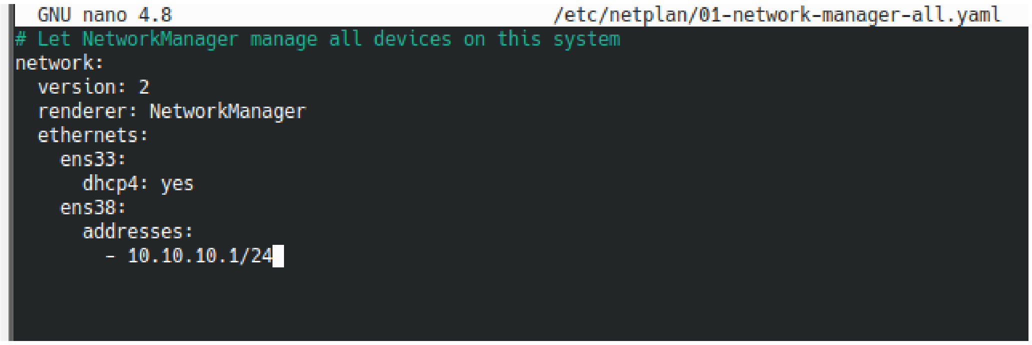
```
tom@leapfrog-internship:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.64  netmask 255.255.255.0  broadcast 192.168.100.255
        inet6 fe80::20c:29ff:fe9c:1579  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:9c:15:79  txqueuelen 1000  (Ethernet)
        RX packets 15563  bytes 15930612 (15.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6101  bytes 558467 (558.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens38: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.1  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::20c:29ff:fe9c:1583  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:9c:15:83  txqueuelen 1000  (Ethernet)
        RX packets 7096  bytes 651857 (651.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7715  bytes 15634236 (15.6 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 325  bytes 22671 (22.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 325  bytes 22671 (22.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tom@leapfrog-internship:~$ 
```

*Figure 6 Interfaces in Ubuntu (Router)*

In this case, network adaptor 1 (ens33) is being configured as a WAN port to connect to the internet and network adaptor 2 is being configured as a LAN port to provide internet access using NAT to other VMs. 10.10.10.0/24 subnet will be used for LAN. 10.10.10.1 will be used as an interface IP of the LAN port, which will be the gateway for other VMs. DHCP4 will be used to distribute IPs to VMs automatically. The whole process is given step by step below:

I. Netplan is used to configure interface using the command:

**$ sudo nano /etc/netplan/01-network-manager-all.yaml**
**$ sudo netplan apply**



```
  GNU nano 4.8                                  /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
      dhcp4: yes
    ens38:
      addresses:
        - 10.10.10.1/24
```

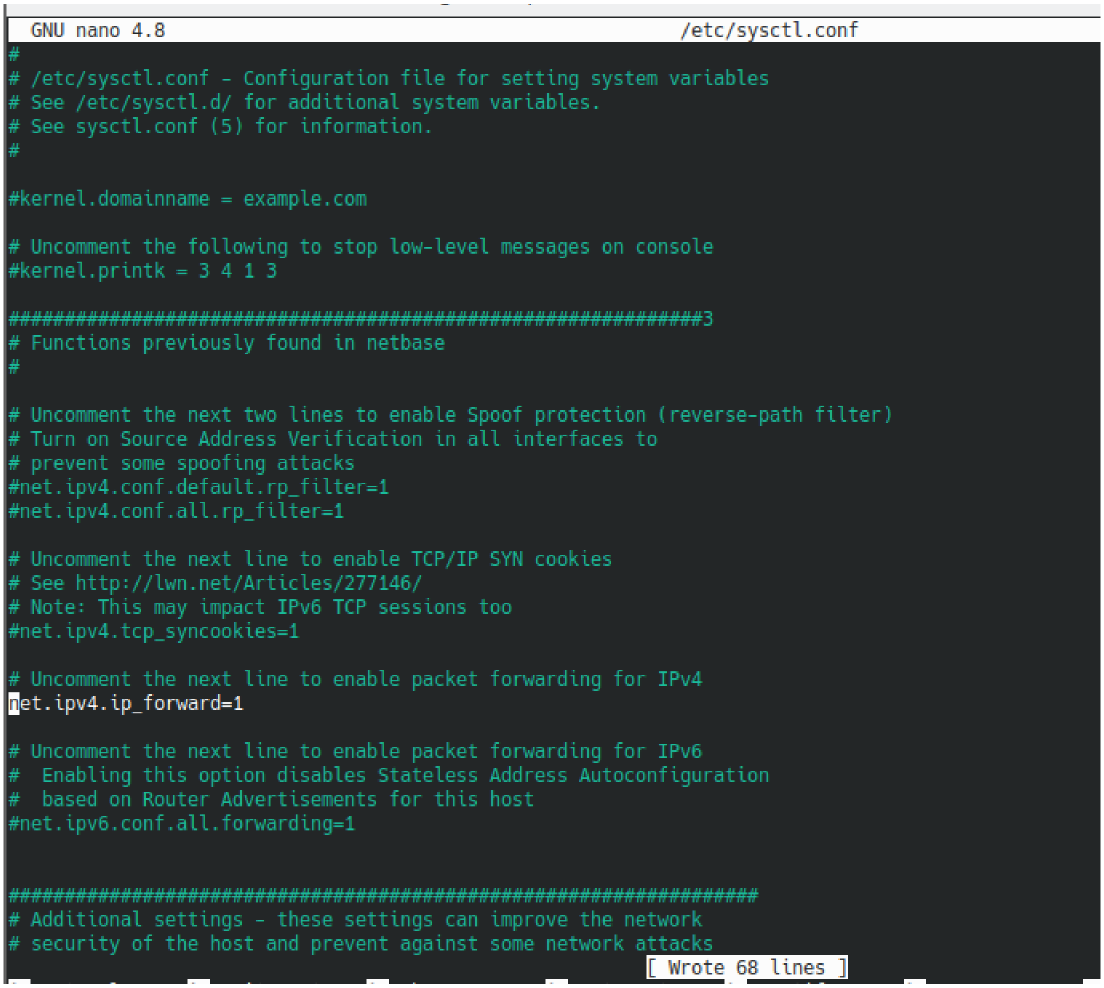
*Figure 7 Netplan configuration*

II. Sysctl is configured to forward ipv4 packets

**$ sudo nano /etc/sysctl.conf**

**$ sudo sysctl -p**

**$ sudo sh -c echo 1 /proc/sys/net/ipv4/ip forward**



```
  GNU nano 4.8                                            /etc/sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

##############################################################3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1


###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
                                                         [ Wrote 68 lines ]
```

*Figure 8 sysctl configuration*

III. Iptables is used to enable NAT and IP masquerading

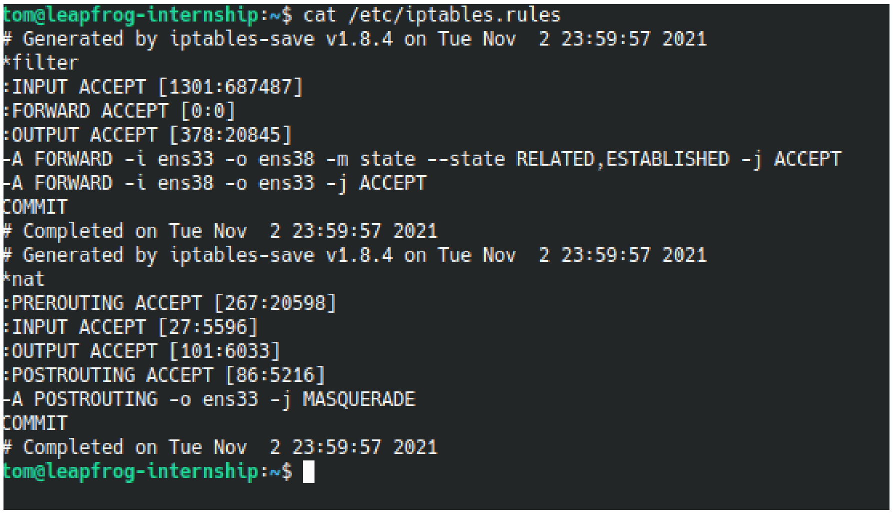
**$ sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE**

**$ sudo iptables -A FORWARD -i ens33 -o ens38 -m state --state RELATED,ESTABLISHED -j ACCEPT**

**$ sudo iptables -A FORWARD -i ens38 -o ens33 -j ACCEPT**

**$ sudo -i**

**$ iptables-save > /etc/iptables.rules**

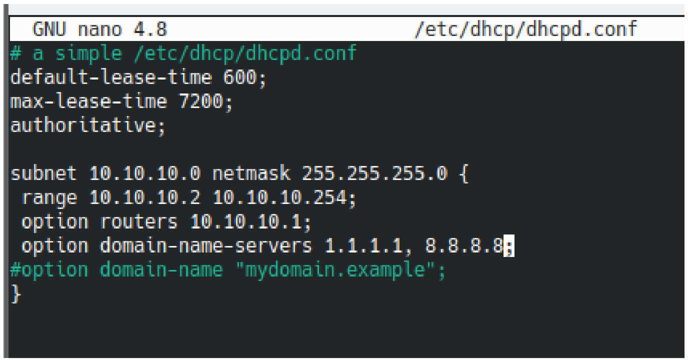


```
tom@leapfrog-internship:~$ cat /etc/iptables.rules
# Generated by iptables-save v1.8.4 on Tue Nov  2 23:59:57 2021
*filter
:INPUT ACCEPT [1301:687487]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [378:20845]
-A FORWARD -i ens33 -o ens38 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i ens38 -o ens33 -j ACCEPT
COMMIT
# Completed on Tue Nov  2 23:59:57 2021
# Generated by iptables-save v1.8.4 on Tue Nov  2 23:59:57 2021
*nat
:PREROUTING ACCEPT [267:20598]
:INPUT ACCEPT [27:5596]
:OUTPUT ACCEPT [101:6033]
:POSTROUTING ACCEPT [86:5216]
-A POSTROUTING -o ens33 -j MASQUERADE
COMMIT
# Completed on Tue Nov  2 23:59:57 2021
tom@leapfrog-internship:~$
```

*Figure 9 iptables.rules*

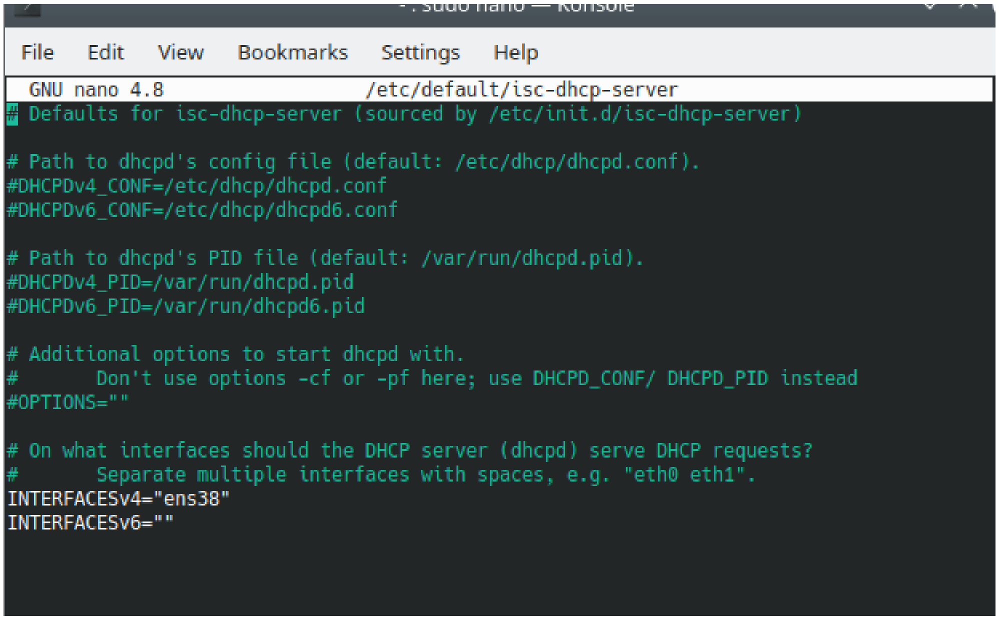
IV. Installing and configuring DHCP server

**$ sudo apt install isc-dhcp-server**
**$ sudo nano /etc/dhcp/dhcpd.conf**
**$ sudo nano /etc/default/isc-dhcp-server**
**$ sudo systemctl restart isc-dhcp-server.service**
**$ sudo systemctl status isc-dhcp-server.service**



```
  GNU nano 4.8                          /etc/dhcp/dhcpd.conf
# a simple /etc/dhcp/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 10.10.10.0 netmask 255.255.255.0 {
 range 10.10.10.2 10.10.10.254;
 option routers 10.10.10.1;
 option domain-name-servers 1.1.1.1, 8.8.8.8;
#option domain-name "mydomain.example";
}
```

*Figure 10 DHCP server configuration*



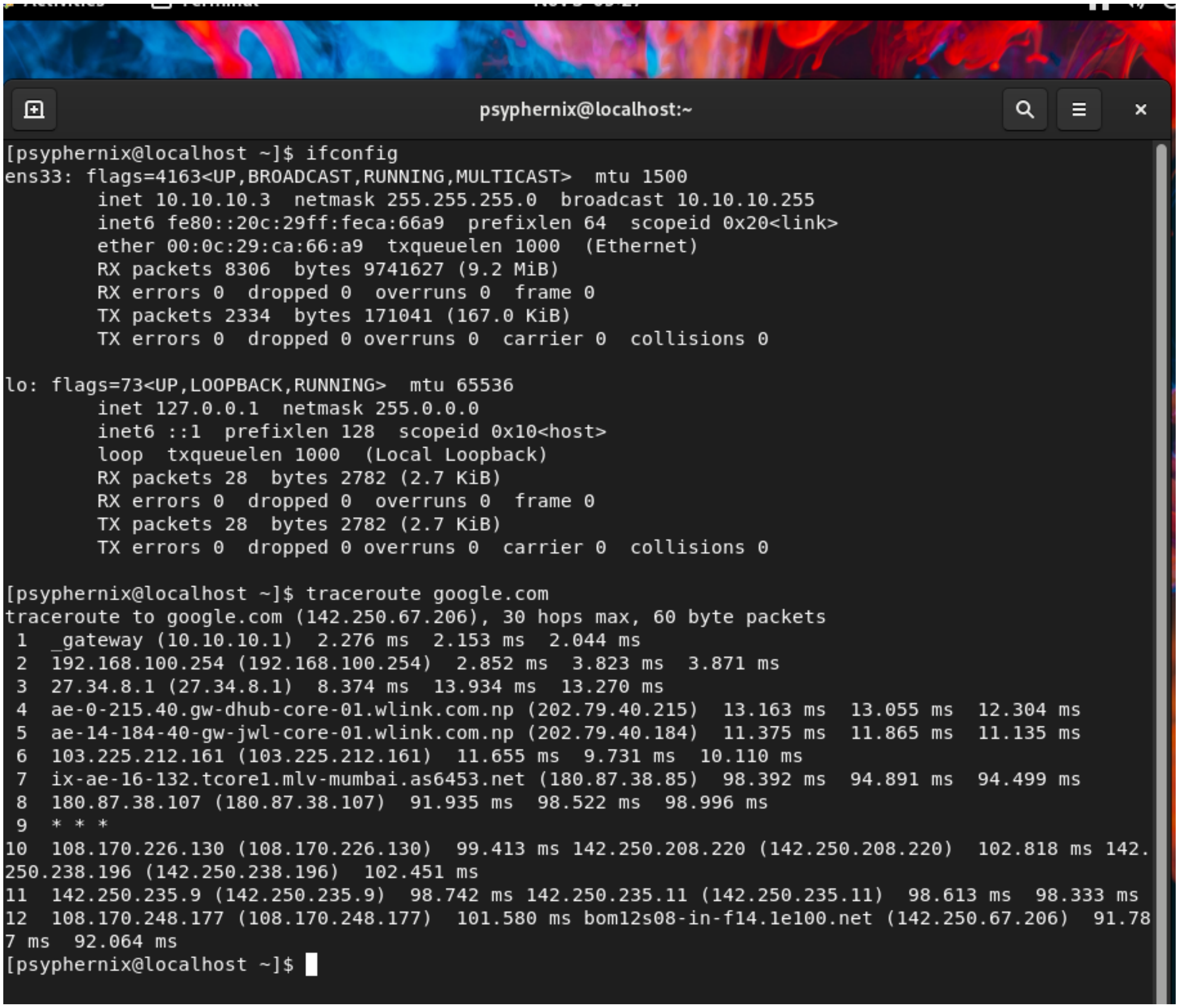
```
  GNU nano 4.8                      /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="ens38"
INTERFACESv6=""
```

*Figure 11 ISC DHCP Server configuration*

Now we can check whether DHCP worked and another VM in LAN is getting an internet connection.

**$ ifconfig**

**$ traceroute google.com**



```
[psyphernix@localhost ~]$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.3  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::20c:29ff:feca:66a9  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:ca:66:a9  txqueuelen 1000  (Ethernet)
        RX packets 8306  bytes 9741627 (9.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2334  bytes 171041 (167.0 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 28  bytes 2782 (2.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28  bytes 2782 (2.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[psyphernix@localhost ~]$ traceroute google.com
traceroute to google.com (142.250.67.206), 30 hops max, 60 byte packets
 1  _gateway (10.10.10.1)  2.276 ms  2.153 ms  2.044 ms
 2  192.168.100.254 (192.168.100.254)  2.852 ms  3.823 ms  3.871 ms
 3  27.34.8.1 (27.34.8.1)  8.374 ms  13.934 ms  13.270 ms
 4  ae-0-215.40.gw-dhub-core-01.wlink.com.np (202.79.40.215)  13.163 ms  13.055 ms  12.304 ms
 5  ae-14-184-40-gw-jwl-core-01.wlink.com.np (202.79.40.184)  11.375 ms  11.865 ms  11.135 ms
 6  103.225.212.161 (103.225.212.161)  11.655 ms  9.731 ms  10.110 ms
 7  ix-ae-16-132.tcore1.mlv-mumbai.as6453.net (180.87.38.85)  98.392 ms  94.891 ms  94.499 ms
 8  180.87.38.107 (180.87.38.107)  91.935 ms  98.522 ms  98.996 ms
 9  * * *
10  108.170.226.130 (108.170.226.130)  99.413 ms 142.250.208.220 (142.250.208.220)  102.818 ms 142.
250.238.196 (142.250.238.196)  102.451 ms
11  142.250.235.9 (142.250.235.9)  98.742 ms 142.250.235.11 (142.250.235.11)  98.613 ms  98.333 ms
12  108.170.248.177 (108.170.248.177)  101.580 ms bom12s08-in-f14.1e100.net (142.250.67.206)  91.78
7 ms  92.064 ms
[psyphernix@localhost ~]$ 
```

*Figure 12 Internet connectivity checking LAN using CentOS VM*