

# Firewall and NAT Assignment

- 1 . Create a virtual machine having the os centos
  - a. Install firewall in the vm(centos might have firewall installed in default).(firewalld or iptables)

Ans:

Firewalld is already installed in centos by default but if we need to install it we can use the command

```
~ sudo yum install firewalld
```

We can check the status using the command

```
~ sudo firewall-cmd --state
```

```
[root@localhost ~]# firewall-cmd --state
running
```

OR

```
~ sudo systemctl status firwalld
```

```
[root@localhost ~]# systemctl status firewall-cmd
Unit firewall-cmd.service could not be found.
[root@localhost ~]# systemctl status firewalld
■ firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-11-02 23:13:40 +0545; 2min 57s ago
     Docs: man:firewalld(1)
   Main PID: 1472 (firewalld)
   CGroup: /system.slice/firewalld.service
           └─1472 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid

Nov 02 23:13:40 localhost.localdomain systemd[1]: Stopped firewalld - dynamic firewall daemon.
Nov 02 23:13:40 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Nov 02 23:13:40 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
Nov 02 23:13:40 localhost.localdomain firewalld[1472]: WARNING: AllowZoneDrifting is enabled. T...w.
Hint: Some lines were ellipsized, use -l to show in full.
```

b. Block certain ip range/subnet using firewallld.

Ans:

To block certain Ip range/subnet we can use the command , here the ip range of 192.168.2.0 - 192.168.2.255 is blocked

```
~ sudo firewall-cmd --permanent --add-rich-rule="rule
family='ipv4' source address='192.168.2.0/24' reject"
```

```
[root@localhost ~]# firewall-cmd --permanent --add-rich-rule="rule family='ipv4' source address='192
.168.2.0/24' reject"
success
```

Now the new rule can be seen using the command

```
~ sudo firewall-cmd --list-all
```

```
[root@localhost ~]# sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8
  sources:
  services: dhcpv6-client http https ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
    rule family="ipv4" source address="192.168.2.0/24" reject
[root@localhost ~]# _
```

- c. Allow http, https and ssh connection using firewall.

Ans:

Http, https and ssh are services which can be allowed or blocked, to allow them we can use the following command

```
~ sudo firewall-cmd --permanent --zone=public --add-service=<service name>
```

```
[root@localhost ~]# sudo firewall-cmd --permanent --zone=public --add-service=http
success
[root@localhost ~]# sudo firewall-cmd --permanent --zone=public --add-service=https
success
[root@localhost ~]# sudo firewall-cmd --permanent --zone=public --add-service=ssh
success
```

We can check the services allowed with the command

```
~ sudo firewall-cmd --permanent --zone=public --list-services
```

```
[root@localhost ~]# sudo firewall-cmd --zone=public --list-services
dhcpv6-client http https ssh
[root@localhost ~]#
```

d. You can add other rules as well as you prefer.

Note: The firewall rules should be saved permanently

Ans:

We can as well open ports or port range to the zone using the command

```
~ sudo firewall-cmd --permanent --zone=public --add-port=4040/tcp
```

```
[root@localhost ~]# sudo firewall-cmd --permanent --zone=public --add-port=4040/tcp
success
[root@localhost ~]# _
```

I almost all of the above commands **--permanent** flags were used to set the options permanently, but these options are not effective immediately, only after service restart/reload.

So, we need to use restart the service after these so we need to use the command

```
~ sudo firewall-cmd --reload
```

So finally, the rules would be

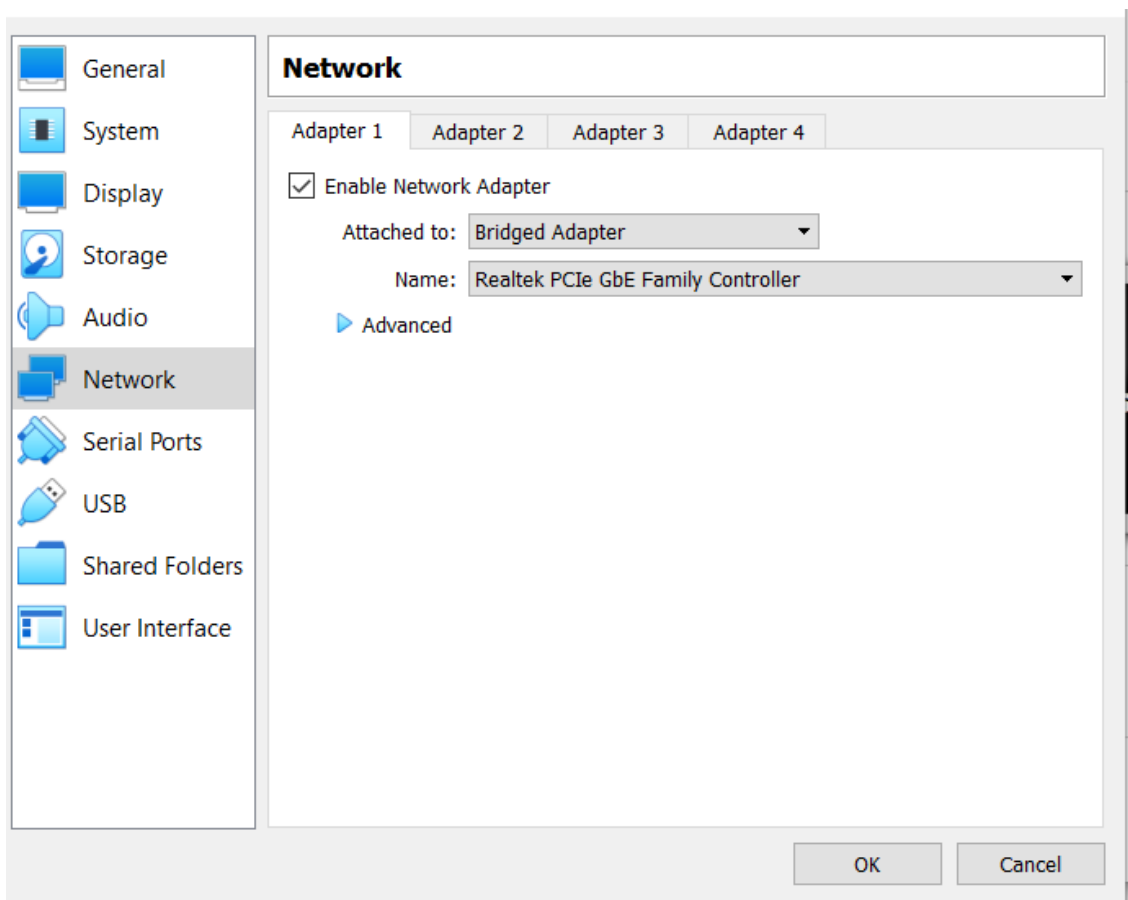
```
[root@localhost ~]# sudo firewall-cmd --list-all
public (active)
target: default
icmp-block-inversion: no
interfaces: enp0s3 enp0s8
sources:
services: dhcpv6-client http https ssh
ports: 4040/tcp
protocols:
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
    rule family="ipv4" source address="192.168.2.0/24" reject
[root@localhost ~]# _
```

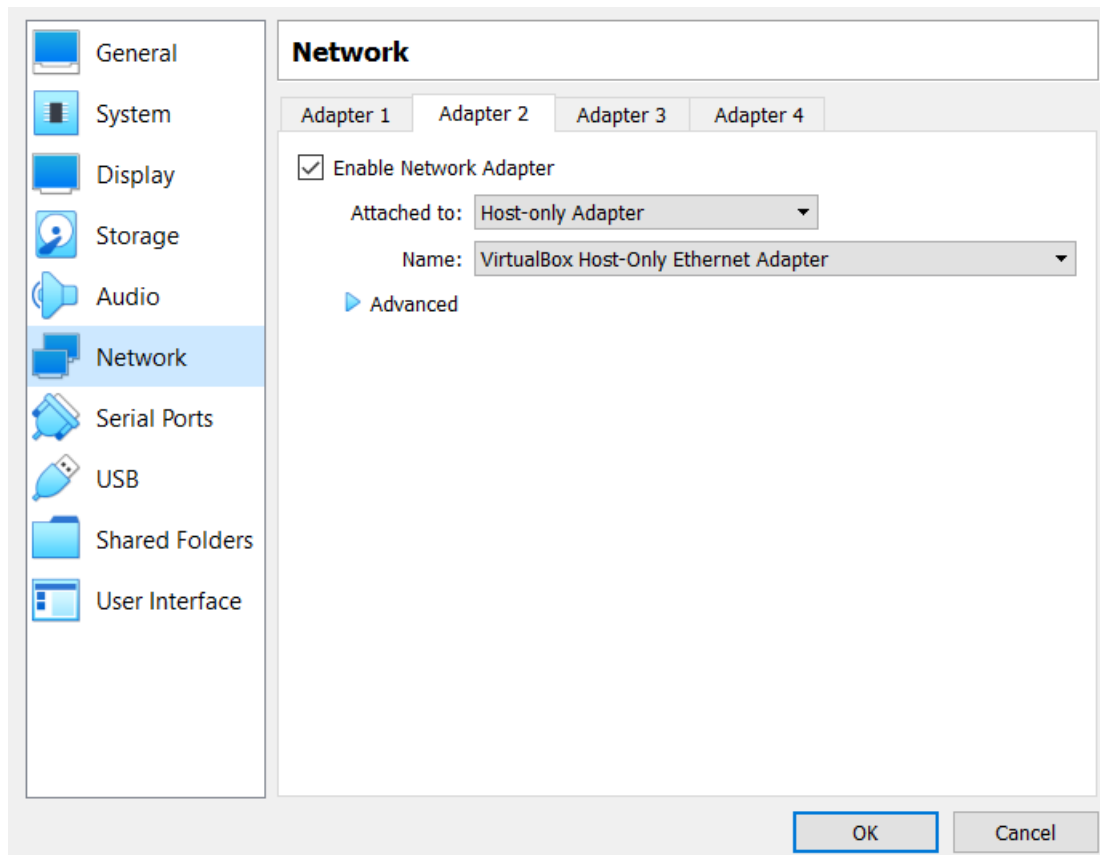
1. Create one vm with 2 network interfaces one should behave as WAN and another as LAN. Create another vm attaching the previously created LAN interface to it.
  - a. Implement NAT in the first vm, so that the second vm can access the internet.

Note: Configure the first vm as a router, so make the LAN interfaces in the first vm as gateway to the LAN network. And in the second vm configure the gateway to the ip of the first vm LAN ip.

Ans:

**First VM** needs to have two network adapters set as:





**Second VM** needs only one adapter, i.e the Host -only adapter (Name must be same)

### To configure the First VM

Firstly we set IP forward as 1

```
~ echo 1 > /proc/sys/net/ipv4/ip_forward
```

Run these commands to set the IPTABLE Routing rules for NATing

```
~ modprobe iptable_nat
~ iptables -F
~ iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
~ iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
```

## To configure the Second VM

We need to set the default gateway as the private IP of the first VM (Server), which can be done with the command

```
~ route add default gw 192.168.56.1
```

In this way Linux system can be used as NAT server and so can be used to access private ip

Client Routes (VM2):

```
tom@tom-VirtualBox: ~  
tom@tom-VirtualBox:~$ route  
Kernel IP routing table  
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface  
default          192.168.56.107  0.0.0.0          UG    0      0      0 enp0s3  
192.168.56.0     0.0.0.0          255.255.255.0    U     100    0      0 enp0s3
```

Server Routes (VM1):

```
[root@localhost ~]# route -n  
Kernel IP routing table  
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface  
0.0.0.0          192.168.1.1     0.0.0.0          UG    100    0      0 enp0s3  
192.168.1.0     0.0.0.0          255.255.255.0    U     100    0      0 enp0s3  
192.168.56.0     0.0.0.0          255.255.255.0    U     101    0      0 enp0s8  
[root@localhost ~]#
```

Pinging to google.com

```
tom@tom-VirtualBox:~$ ping google.com  
PING google.com (142.250.182.174) 56(84) bytes of data.  
64 bytes from google.com (142.250.182.174): icmp_seq=1 ttl=56 time=30.3 ms  
64 bytes from google.com (142.250.182.174): icmp_seq=2 ttl=56 time=31.0 ms  
64 bytes from google.com (142.250.182.174): icmp_seq=3 ttl=56 time=31.1 ms  
^C  
--- google.com ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2002ms  
rtt min/avg/max/mdev = 30.330/30.795/31.060/0.330 ms  
tom@tom-VirtualBox:~$
```

```
tom@tom-VirtualBox:~$ traceroute google.com  
traceroute to google.com (142.250.182.174), 30 hops max, 60 byte packets  
1 192.168.56.107 (192.168.56.107) 5.123 ms 4.896 ms 4.683 ms  
2 Broadcom.Home (192.168.1.1) 2.177 ms 1.994 ms 1.803 ms
```