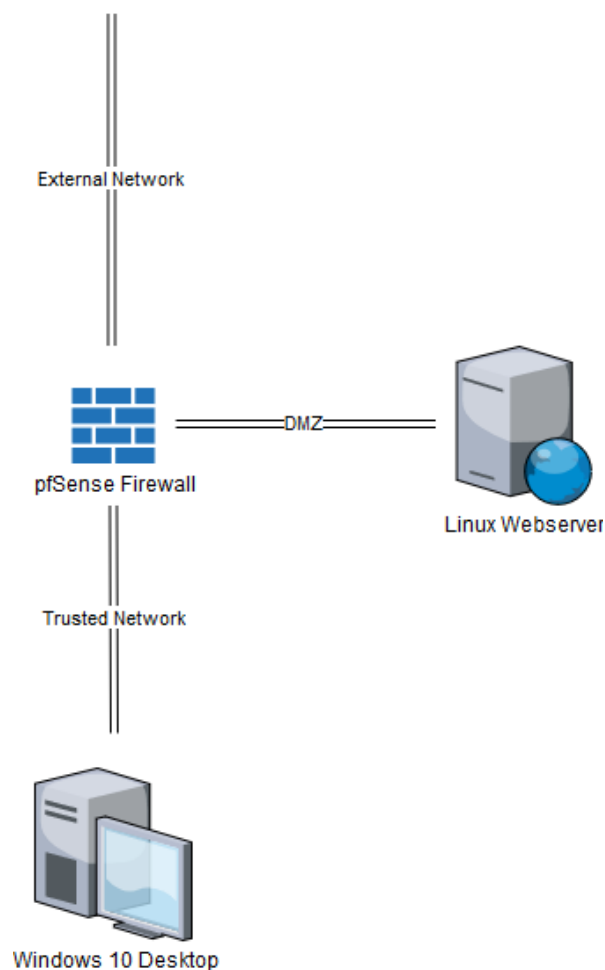


## LAB 3: Network Virtualisation

### DOCUMENTATION

*Author: Manish Subedi*

This lab helped us learn and experiment the configuration of network in a virtualised environment using pfSense, which is a free open-source software. First, we installed pfSense as a new VM for our hypervisor. We need it to route traffic between different networks and it also acts as a Firewall. We implemented port forwarding and bandwidth limiting as few of the features of pfSense in this lab.



*Figure: Network Design for lab 3*

IP address configurations:

Networks / Devices	IP address	Default gateway
Trusted Network / LAN (192.168.132.254)	Offered range 192.168.132.100 – 192.168.132.150	192.168.132.254
Desktop (Win 10 VM)	192.168.132.100	
DMZ (10.10.10.254)	Offered range 10.10.10.100 - 10.10.10.150	10.10.10.254
Server (Linux VM)	10.10.10.100	
pfSense VM	192.168.228.130	192.168.228.2

### pfSense VM Configuration:

HOSTNAME: PFSense\_LAB3

DOMAIN: MYLAB3.ARPA

MEMORY: 1 GB

PROCESSORS: 1

HARD DISK (SCSI): 20GB

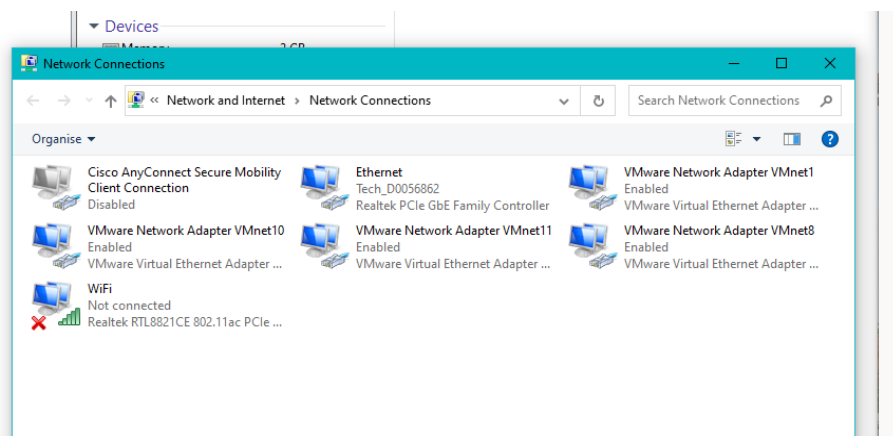
INTERFACES: WAN (192.168.228.130/24), LAN (192.168.132.254/24) &

DMZ (10.10.10.254/24)

GATEWAY: WAN\_DHCP @ 192.168.228.2

## Configuration of Network Adapters in pfSense

Three different Network Adapters were needed for the environment, one for external network, second for the Linux server (DMZ) and the third for the trusted network. This is because we will have three interfaces, one for pfSense to connect to the Internet, another - pfSense to Linux and the last – pfSense to trusted network. A quick view is shown below.



*Figure: Network Adapters*

## Configuration of IP addresses and DHCP Servers

(WAN) was assigned a static IP address and DHCP server was enabled as well. The other two interfaces (LAN and DMZ) were also assigned with IP addresses as shown below, but without DHCP server.

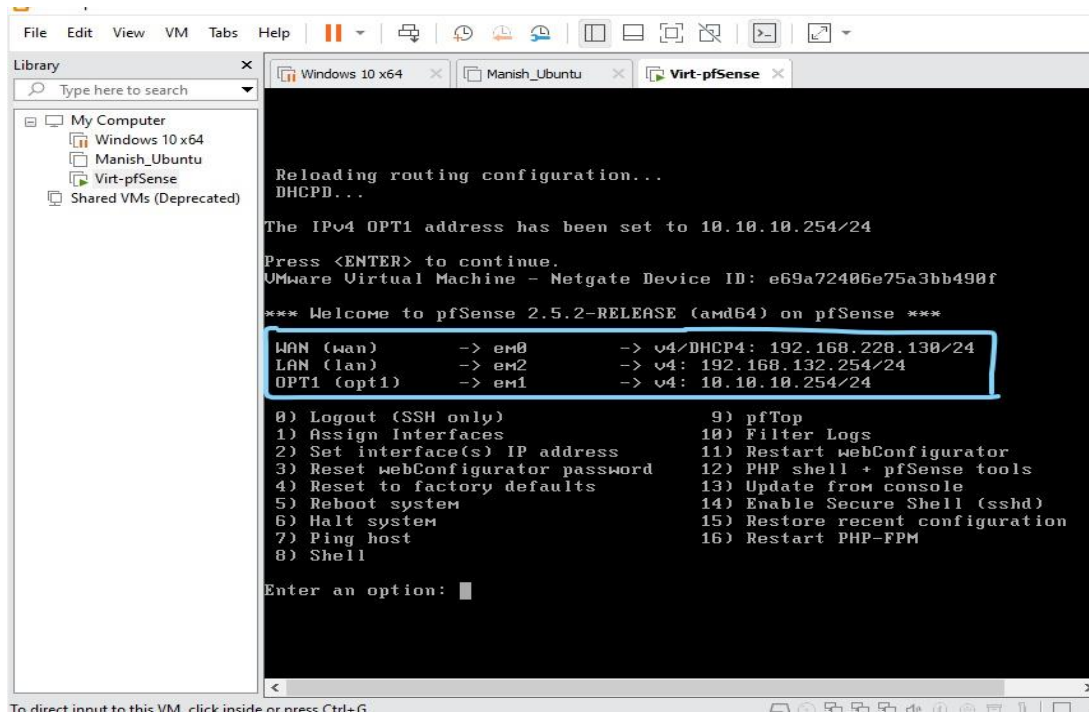


Figure: IP address assigned to the interfaces

HTTP was enabled on LAN interface. When the configuration was ready, pfSense could be accessed through its IP address from Windows VM and even from external servers. Later, all external requests were forwarded to the web server. The pfSense GUI configurator could only be accessed with machines on pfSense's local network. The network configurations for all VMs were correct and functioning properly. Few of the instances during the process are attached below.

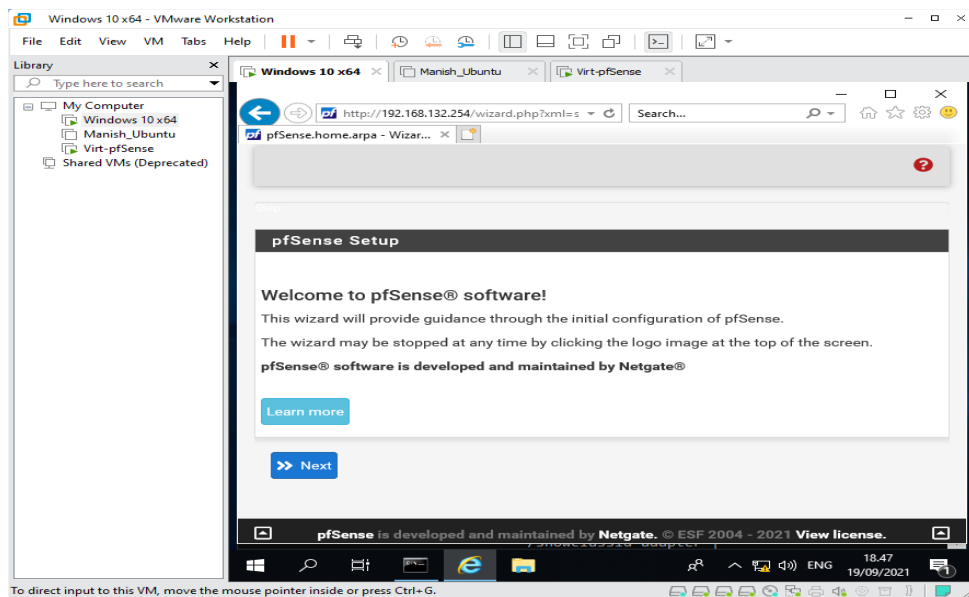


Figure: PfSense up and running

Status / DHCP Leases

Search

Search term

All

Search

Clear

Enter a search string or \*nix regular expression to filter entries.

Leases

IP address	MAC address	Client Id	Hostname	Description	Start	End	Online	Lease Type	Actions
<div>✓</div> 192.168.132.100	00:0c:29:c4:05:dc		DESKTOP-453PL6I		2021/09/26 13:07:59	2021/09/26 15:07:59	online	active	<div><div>+</div><div>+</div></div>
<div>✓</div> 10.10.10.100	00:0c:29:42:01:96		manish-virtUbuntu		2021/09/26 13:07:45	2021/09/26 15:07:45	online	active	<div><div>+</div><div>+</div></div>

Leases in Use

Interface	Pool Start	Pool End	# of leases in use
LAN	192.168.132.10	192.168.132.245	1
DMZ	10.10.10.100	10.10.10.150	1

+

Show all configured leases

Clear all DHCP leases

Figure: DHCP on pfSense

## Configuring firewall service, NAT port forwarding and SSH

The pfSense was configured to act as a firewall such that hosts from trusted network (LAN net - in this scenario) can access the webserver, but the webserver can not access any of the hosts in the trusted network. This

was done with creating a Firewall rule for DMZ that would allow inbound IPv4 and IPv6 traffic but not towards the trusted network interface.

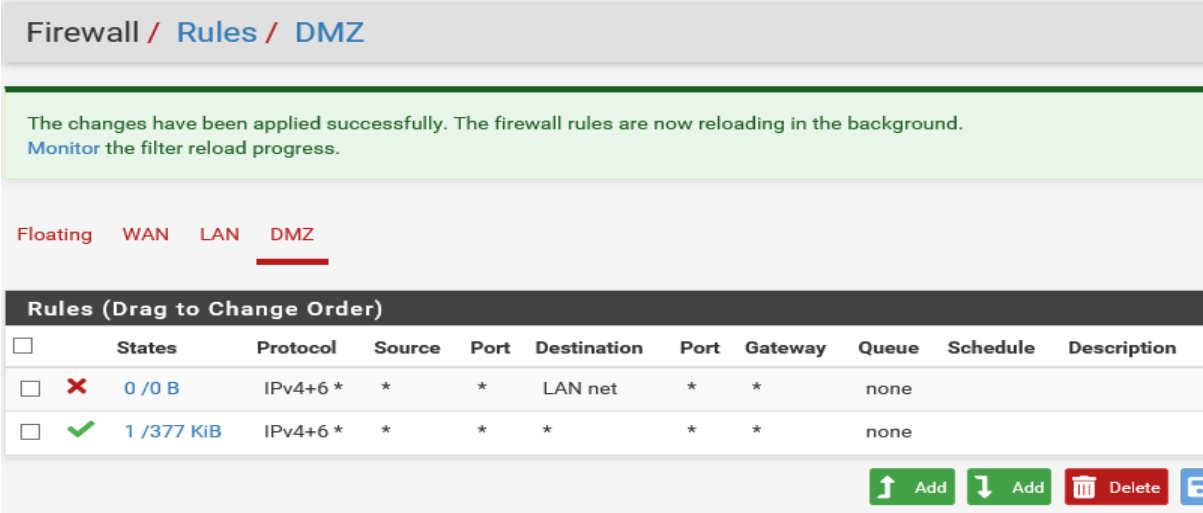


Figure: pfSense as a firewall

The NAT port forwarding was also configured so that webserver was accessible from external networks. All http requests on port 80 would be redirected to the webserver.

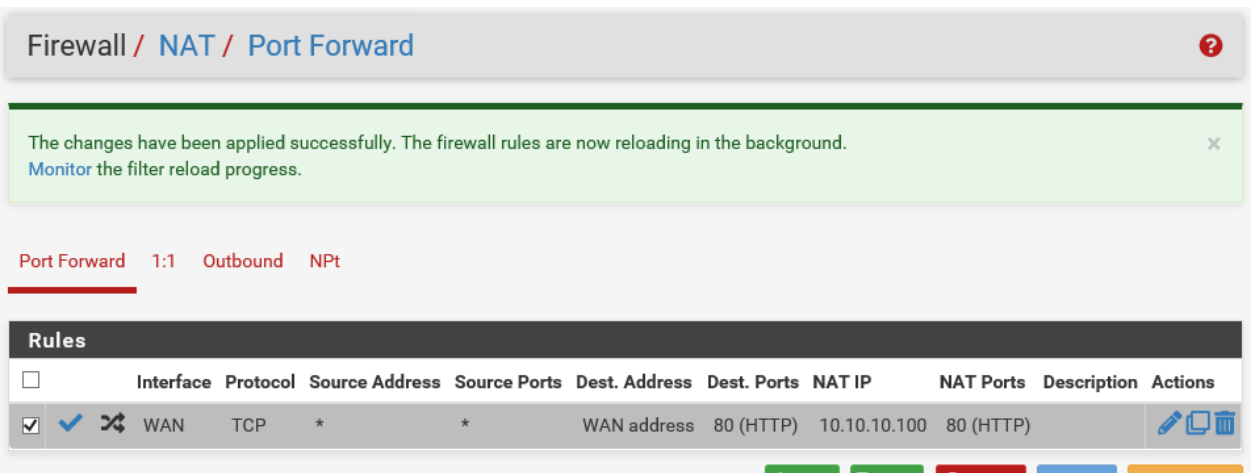
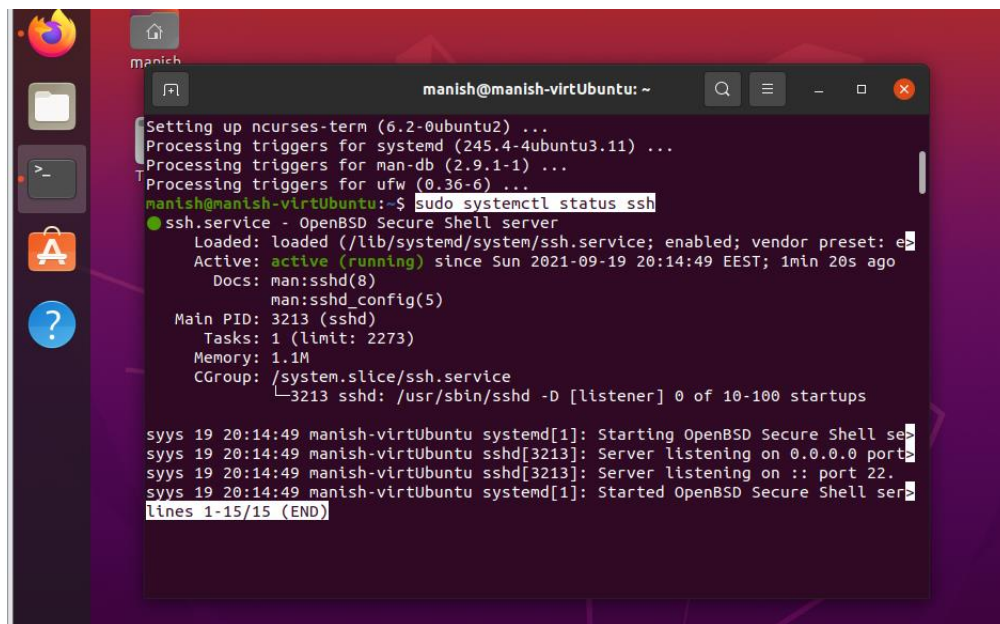


Figure: Port forwarding with pfSense

SSH server was installed in the Linux server using the command 'sudo apt install openssh-server' and then enabled with the command 'sudo systemctl enable --now ssh.service'. The status was confirmed with 'sudo systemctl status ssh' command as shown below.

A screenshot of a Linux terminal window. The window title is 'manish@manish-virtUbuntu: ~'. The terminal shows the output of the command 'sudo systemctl status ssh'. The output indicates that the 'ssh.service' is 'active (running)'. It also shows system logs for the service starting and listening on port 22. The terminal text is as follows:

```
Setting up ncurses-term (6.2-0ubuntu2) ...
Processing triggers for systemd (245.4-4ubuntu3.11) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for ufw (0.36-6) ...
manish@manish-virtUbuntu:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: ena
   Active: active (running) since Sun 2021-09-19 20:14:49 EEST; 1min 20s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 3213 (sshd)
      Tasks: 1 (limit: 2273)
     Memory: 1.1M
    CGroup: /system.slice/ssh.service
            └─3213 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

syys 19 20:14:49 manish-virtUbuntu systemd[1]: Starting OpenBSD Secure Shell se
syys 19 20:14:49 manish-virtUbuntu sshd[3213]: Server listening on 0.0.0.0 port
syys 19 20:14:49 manish-virtUbuntu sshd[3213]: Server listening on :: port 22.
syys 19 20:14:49 manish-virtUbuntu systemd[1]: Started OpenBSD Secure Shell ser
lines 1-15/15 (END)
```

Figure: Configuring SSH server in Linux

The configuration for port forwarding available at *Firewall >> NAT >> Port Forward* was used again to create new rules to forward SSH server access requests for the Webserver and desktop. The SSH server on the webserver can now be accessed at TCP port 2222 and the desktop at port 4422 as given in the lab configuration requirements. The figure below shows all the port forwarding decisions that would be made by the pfSense.

pfSense.home.arpa - Firewa... x Apache2 Ubuntu Default Page...

Port Forward 1:1 Outbound NPT

Rules

<input type="checkbox"/>			Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports	Description
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	4422	192.168.132.100	22 (SSH)	
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	2222	10.10.10.100	22 (SSH)	
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP	*	*	WAN address	80 (HTTP)	10.10.10.100	80 (HTTP)	

Figure: Port forwarding by pfSense

The webserver can now be accessed using SSH as shown below.

```

C:\Users\Manis>ssh manish@192.168.228.130 -p 2222
manish@192.168.228.130's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-34-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 updates can be applied immediately.
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sun Sep 19 20:26:40 2021 from 192.168.228.1
manish@manish-virtUbuntu:~$ connection established as the port was configured
  
```

Figure: Successfully accessed webserver using SSH

Also, the desktop can now be accessed using SSH after successful installation of SSH server as shown below.



```
Administrator: Windows PowerShell

PS C:\> Get-Service sshd
Name
-----
State : Installed
Name : OpenSSH.Server~~~~0.0.1.0
State : NotPresent

PS C:\Windows\system32> Add-WindowsCapability -Online - Name OpenSSH.Server~~~~0.0.1.0
Add-WindowsCapability : A positional parameter cannot be found that accepts argument '-'.
At line:1 char:1
+ Add-WindowsCapability -Online - Name OpenSSH.Server~~~~0.0.1.0
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (:) [Add-WindowsCapability], ParameterBindingException
+ FullyQualifiedErrorId : PositionalParameterNotFound,Microsoft.Dism.Commands.AddWindowsCapabilityCommand

PS C:\Windows\system32> Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

Path :
Online : True
RestartNeeded : False

PS C:\Windows\system32> Start-Service sshd
PS C:\Windows\system32> Get-Service sshd

Status Name DisplayName
-----
Running sshd OpenSSH SSH Server

PS C:\Windows\system32> Set-Service -Name sshd -StartupType 'Automatic'
Set-Service : The term 'Set-service' is not recognized as the name of a cmdlet, function, script file, or
operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and
try again.
At line:1 char:1
+ Set-Service -Name sshd -StartupType 'Automatic'
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Set-service:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Windows\system32> Set-Service -Name sshd -StartupType 'Automatic'
PS C:\Windows\system32>
```

Figure: Installation of SSH server in desktop

```
CA Select Administrator: c:\windows\system32\cmd.exe

C:\Users\Manis>ssh manish@192.168.228.130
C:\Users\Manis>
C:\Users\Manis>ssh manish@192.168.228.130 -p 2222
manish@192.168.228.130's password:

C:\Users\Manis>ssh manish@192.168.228.130 -p 4422
The authenticity of host '[192.168.228.130]:4422 ([192.168.228.130]:4422)' can't b
e established.
ECDSA key fingerprint is SHA256:n1aRLuk1iYwtQXZfV3CTbZres0BeqDuhXhHCssPV4Dg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.228.130]:4422' (ECDSA) to the list of known h
osts.
manish@192.168.228.130's password:
Microsoft Windows [Version 10.0.17763.2183]
(c) 2018 Microsoft Corporation. All rights reserved.

manish@DESKTOP-453PL6I C:\Users\Manish>SSH access successful on windows VM too
```

Figure: Successfully accessed the desktop using SSH

Finally, the limiters were created under Traffic Shaper / Limiters to limit the max bandwidth on trusted network to 10 Mbit/s.

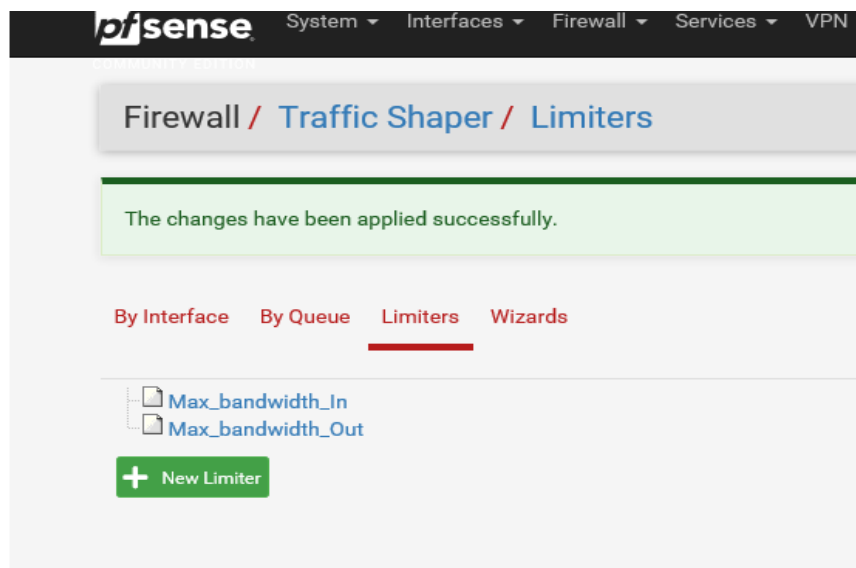


Figure: Bandwidth limiters for trusted network

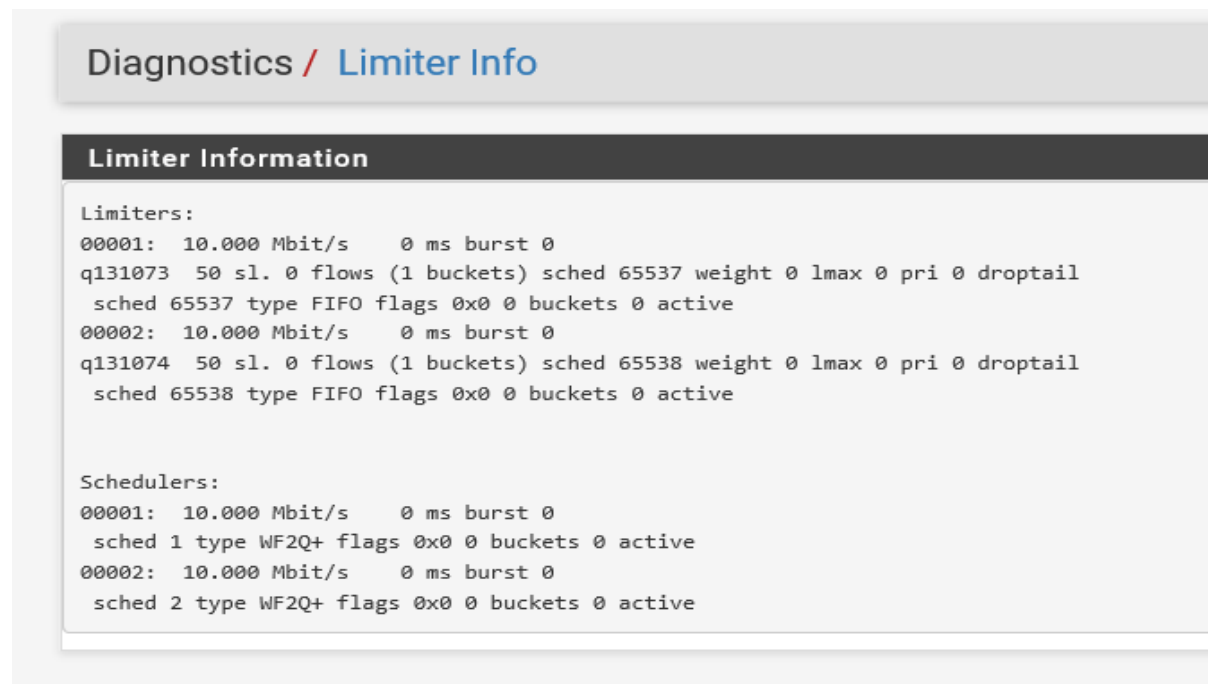
and these limiters could be assigned to the LAN net under the advanced setting at Firewall / Rules / Edit for LAN net or the trusted network. '**In / Out pipe**' was the exact field to assign the limiters.

Choose out ip priority to apply.

Schedule	<input type="text" value="none"/>	▼
Leave as 'none' to leave the rule enabled all the time.		
Gateway	<input type="text"/>	
Leave as 'default' to use the system routing table. Or choose a gateway to utilize policy based routing. Gateway selection is not valid for "IPV4+IPV6" address family.		
In / Out pipe	<input type="text" value="Max_bandwidth_In"/>	<input type="text" value="Max_bandwidth_Out"/>
Choose the Out queue/Virtual interface only if In is also selected. The Out selection is applied to traffic leaving the interface where the rule is created, the In selection is applied to traffic coming into the chosen interface. If creating a floating rule, if the direction is In then the same rules apply, if the direction is Out the selections are reversed, Out is for incoming and In is for outgoing.		
Queue / Queue	<input type="text" value="none"/>	<input type="text" value="none"/>

Figure: Limiting the bandwidth

The diagnostic for the Limiter Info reads as below. I learned that limiters are currently the only way to achieve per-IP address or per-network bandwidth rate limiting using pfSense software.



```
Limiters:
00001: 10.000 Mbit/s    0 ms burst 0
q131073  50 sl. 0 flows (1 buckets) sched 65537 weight 0 lmax 0 pri 0 droptail
  sched 65537 type FIFO flags 0x0 0 buckets 0 active
00002: 10.000 Mbit/s    0 ms burst 0
q131074  50 sl. 0 flows (1 buckets) sched 65538 weight 0 lmax 0 pri 0 droptail
  sched 65538 type FIFO flags 0x0 0 buckets 0 active

Schedulers:
00001: 10.000 Mbit/s    0 ms burst 0
  sched 1 type WF2Q+ flags 0x0 0 buckets 0 active
00002: 10.000 Mbit/s    0 ms burst 0
  sched 2 type WF2Q+ flags 0x0 0 buckets 0 active
```

## Conclusion

Network virtualisation makes configuration and management of existing and new networks easy and cost-effective. The changes in the network environment would be instantly implemented. The resources are also highly available. As virtualisation wanders around kind of same benefits with each of the labs that we have performed till now, this lab too demonstrated the processes involved which went through quite easily.

With tools such as pfSense, and its GUI configurator, it was very easy to configure port forwarding decisions and the limiters in addition to enabling the firewall feature of pfSense. The SSH server access was done instantly with just defining a set of rules in GUI configurator and with such ease.