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.

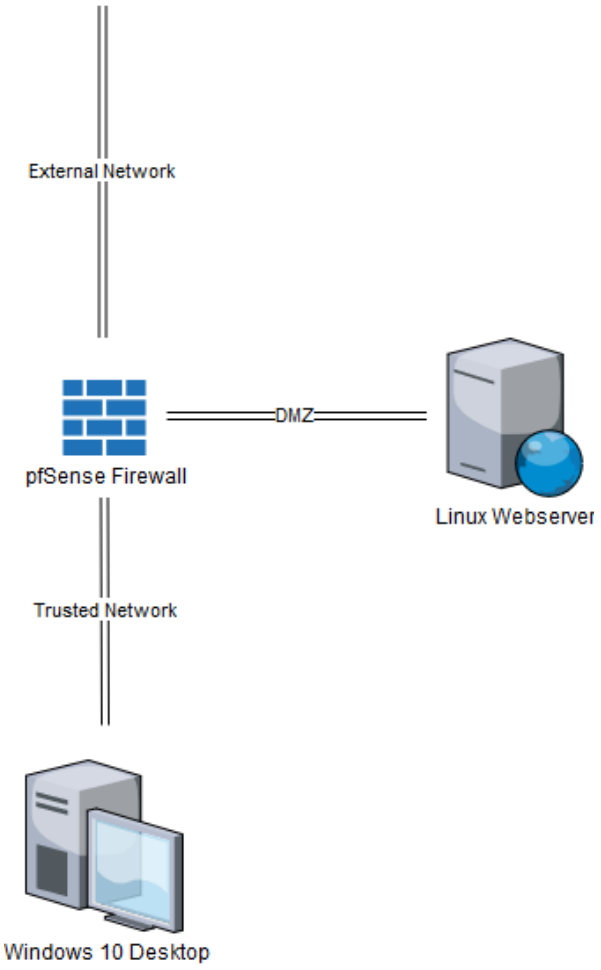


Figure: Network Design for lab 3

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.

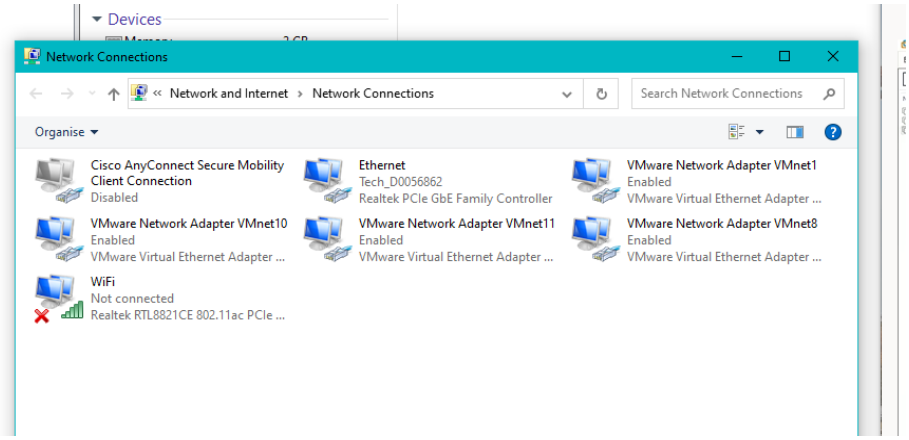


Figure: Network Adapters

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.

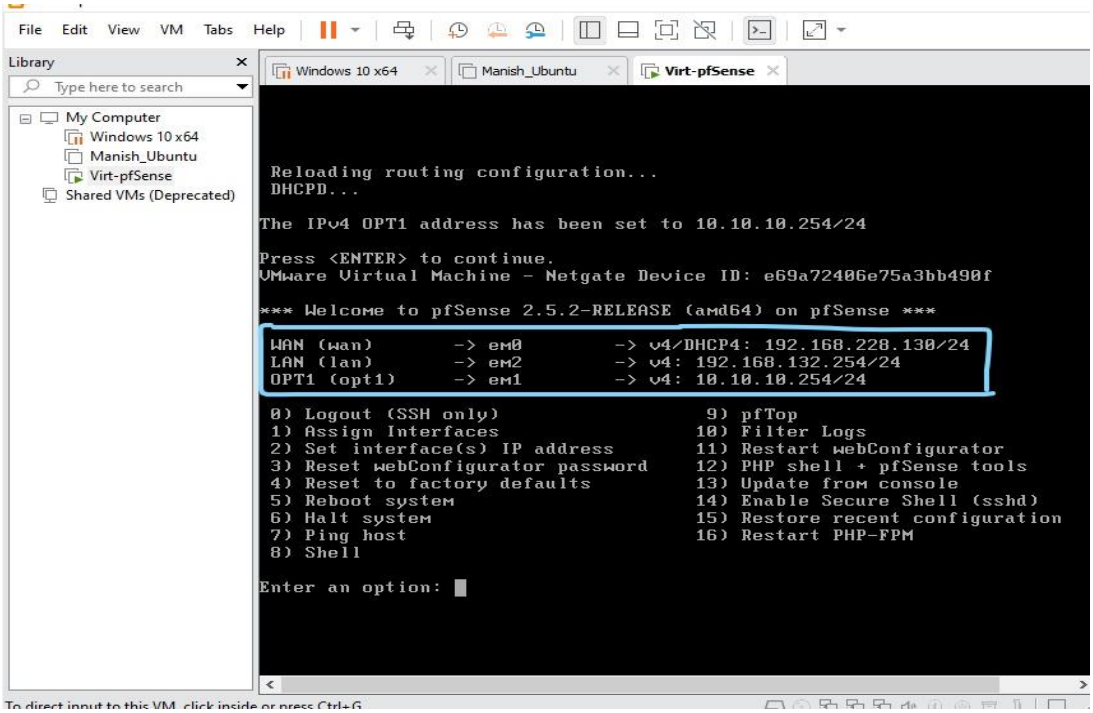


Figure: IP address assigned to the interfaces

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.

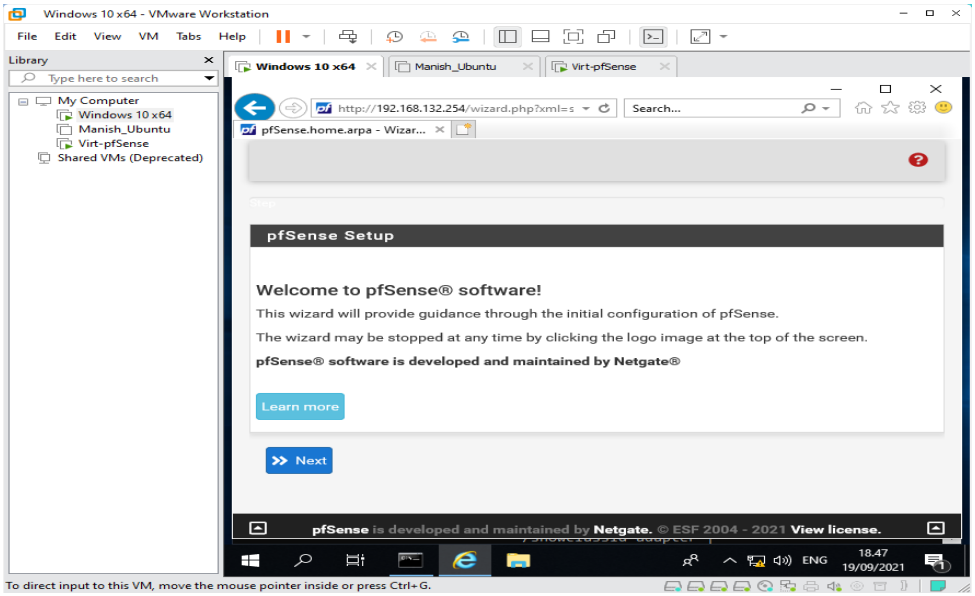


Figure: PfSense up and running

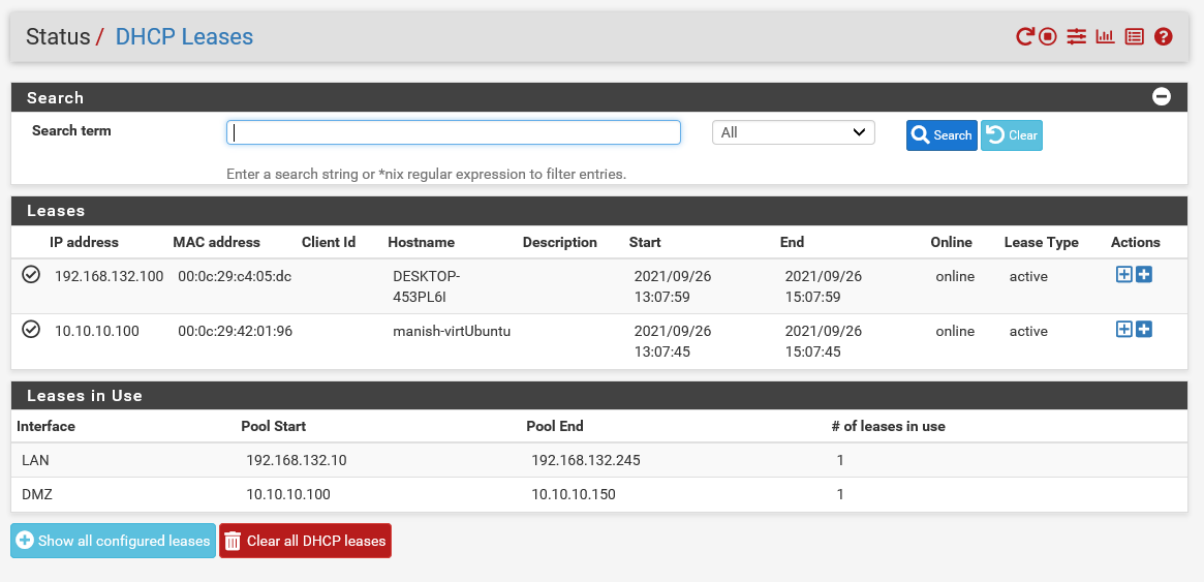


Figure: DHCP on pfSense

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.

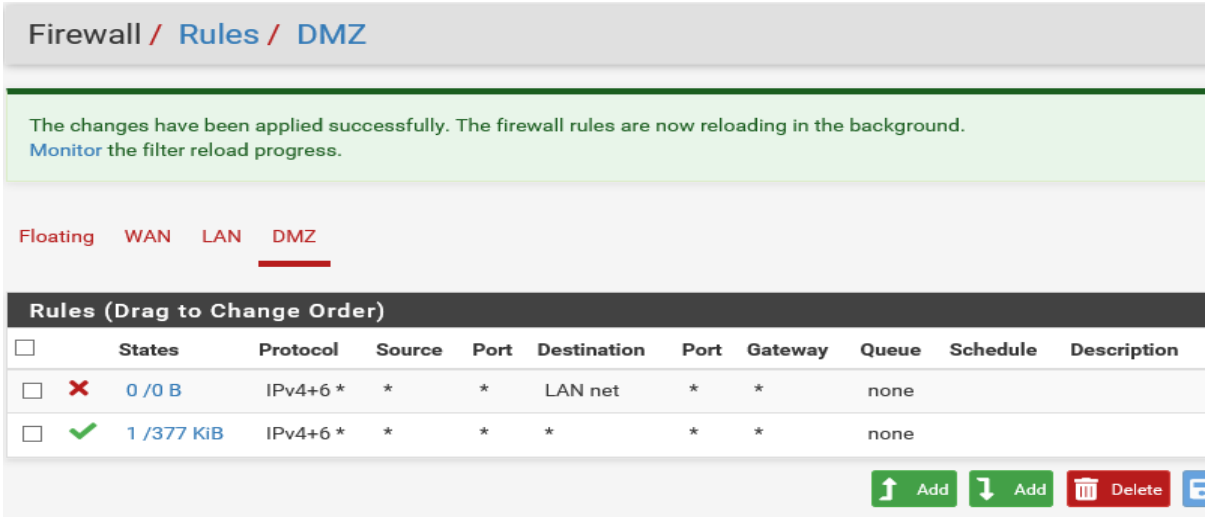


Figure: pfSense as a firewall

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.

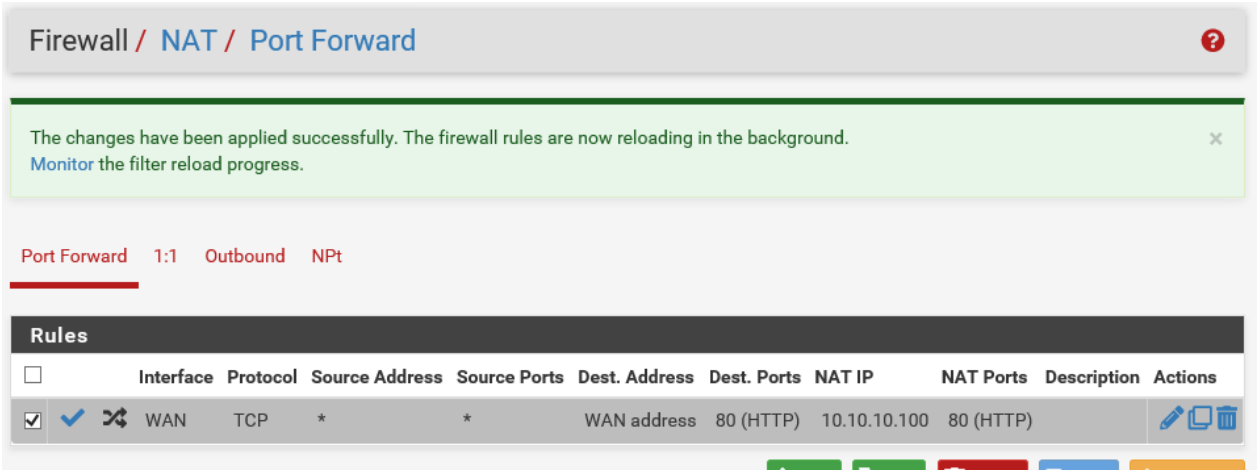


Figure: Port forwarding with pfSense

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.

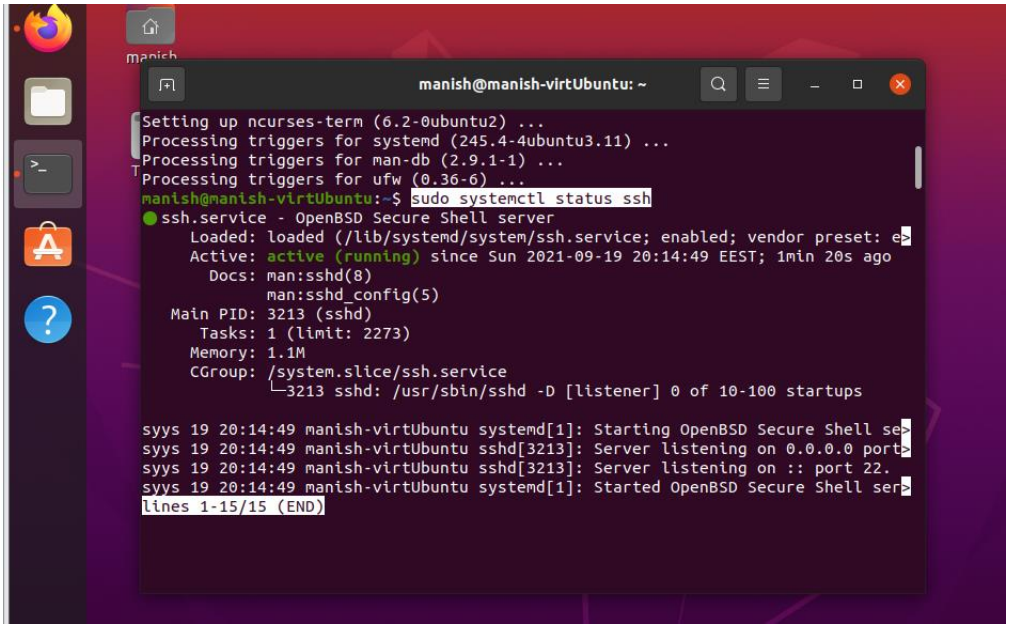


Figure: Configuring SSH server in Linux

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.

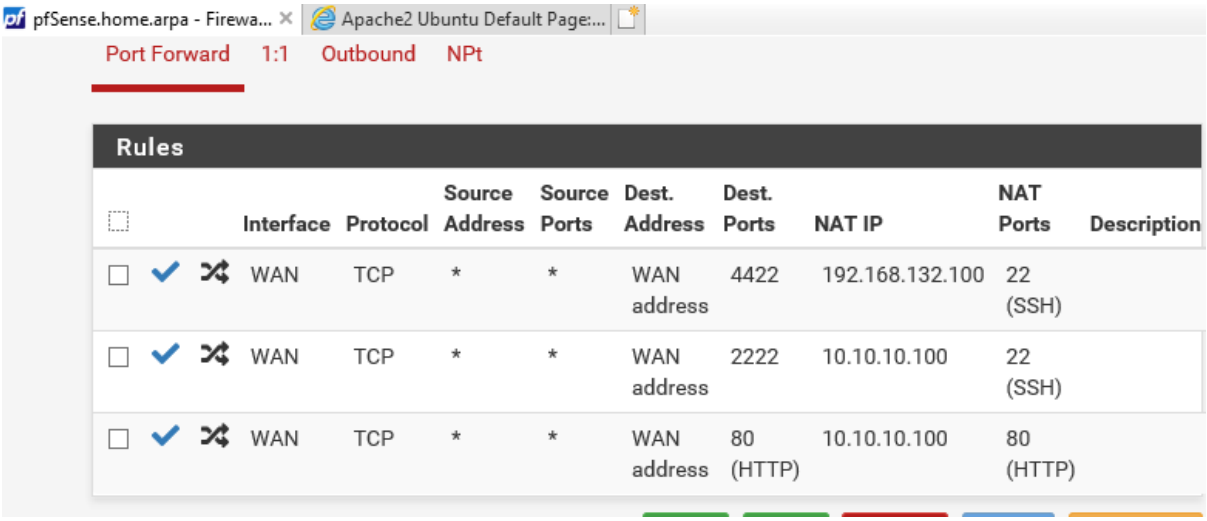


Figure: Port forwarding by pfSense

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

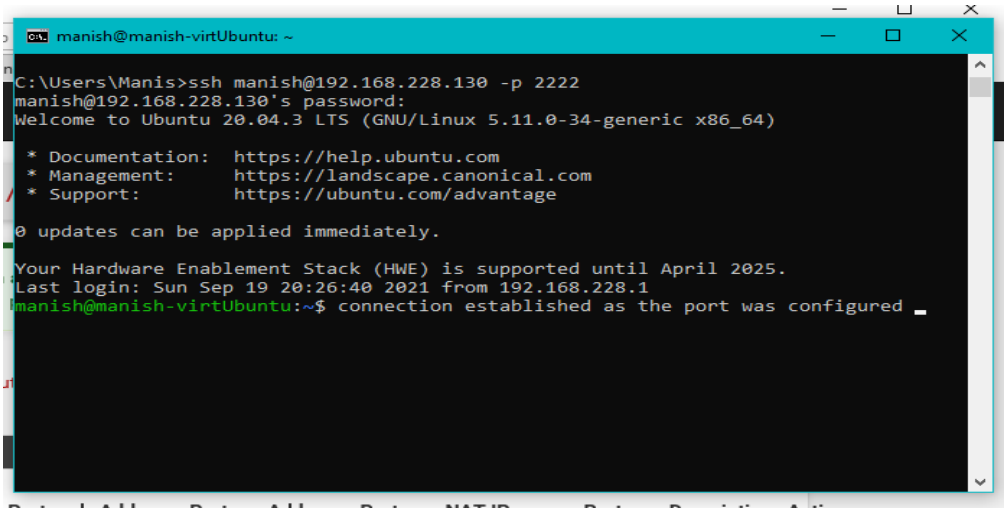


Figure: Successfully accessed webserver using SSH

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

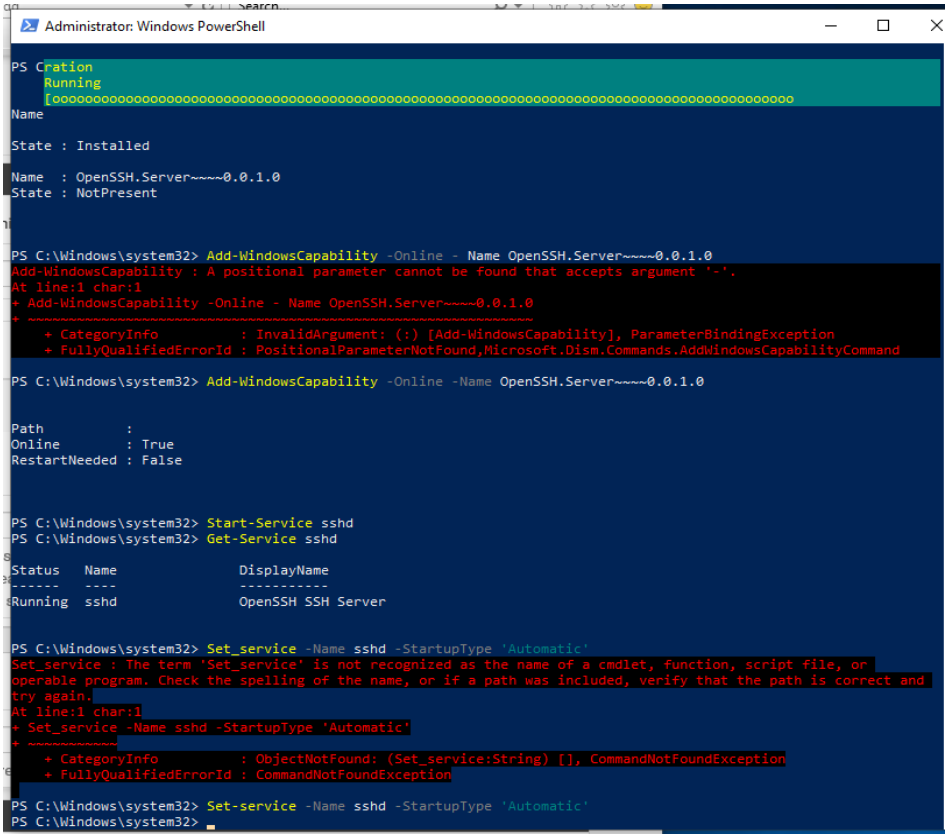


Figure: Installation of SSH server in desktop

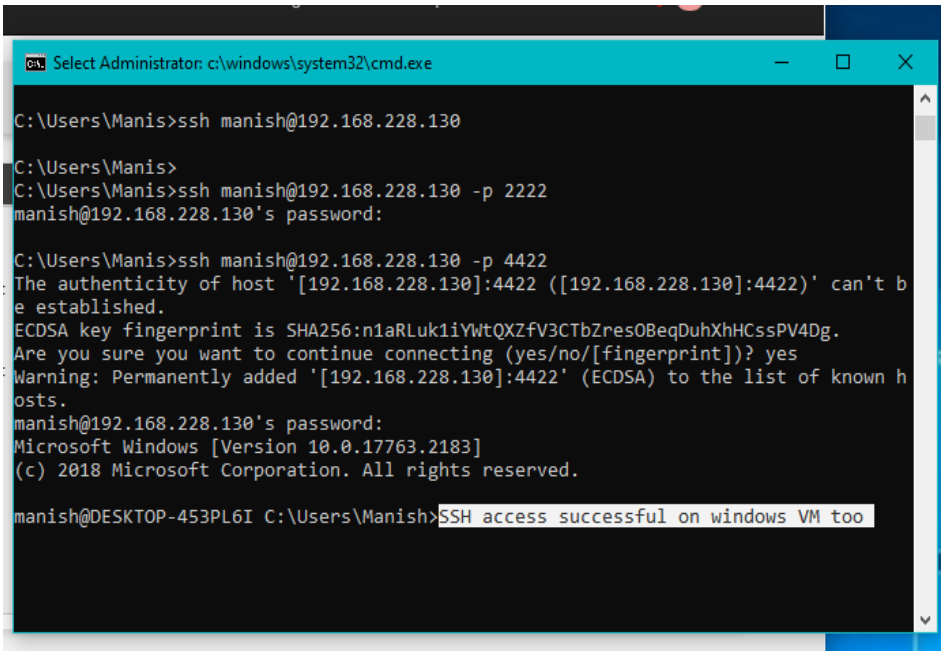


Figure: Successfully accessed the desktop using SSH

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

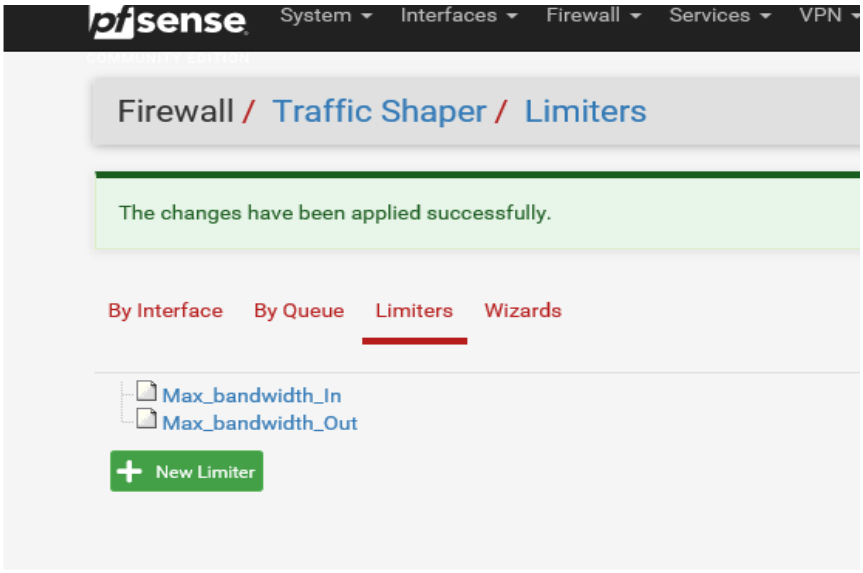


Figure: Bandwidth limiters for trusted network

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.

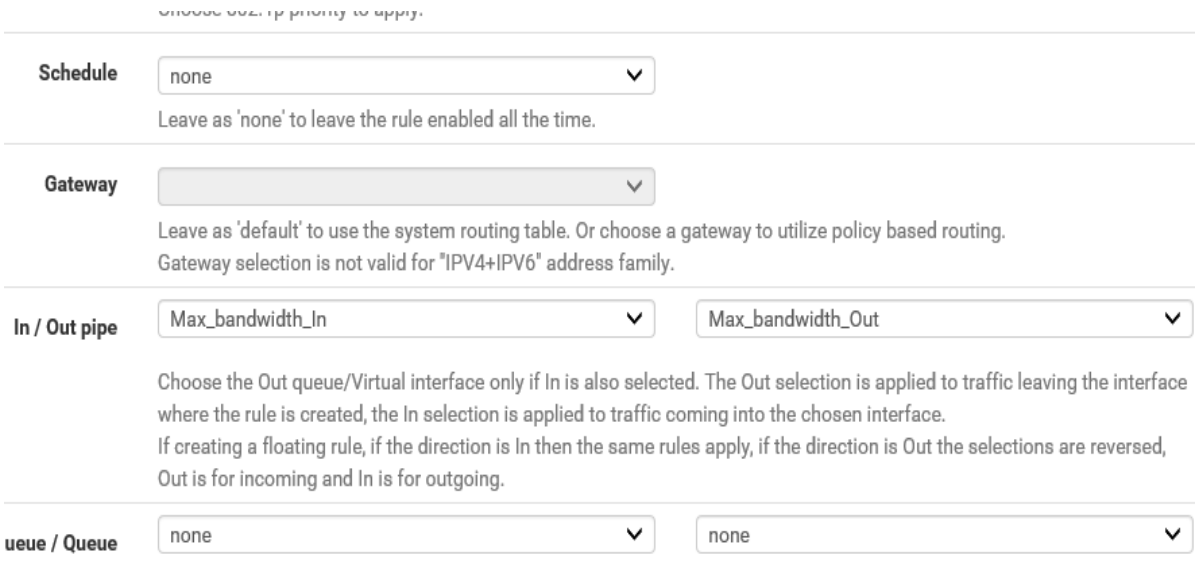
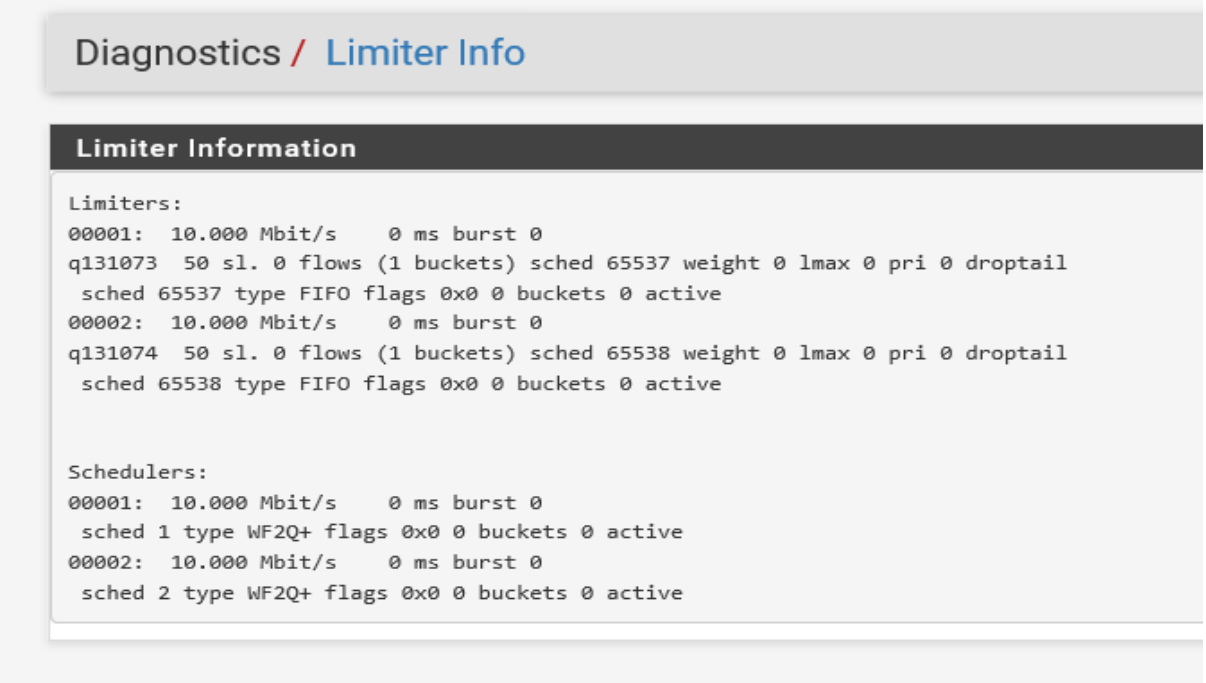


Figure: Limiting the bandwidth

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



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.