

Scotti-BYTE Enterprise Consulting Services

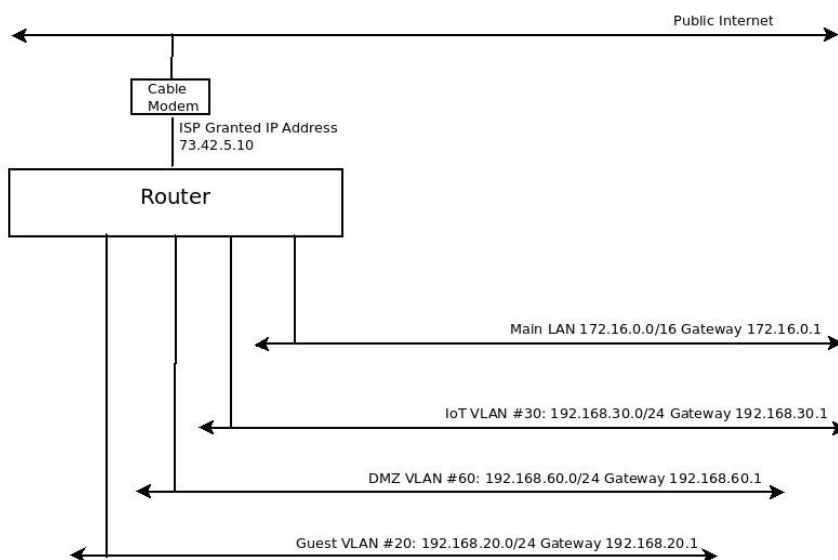
Networking Tutorial: Accessing a VLAN

In the previous network tutorial we discussed some of the advantages of a managed switch. A managed switch goes beyond layer 2 bridging and supports routing and VLANs which is layer 3 routing. Recall that a VLAN is a method of creating a separate network from your main network.

On an unmanaged network, we generally have one LAN and all the devices on that LAN share a common address space and can communicate with all other devices. Even in a managed switch the initial main LAN performs similarly and is usually referred to as VLAN 1, however a number is normally not associated with it. That one single VLAN is referred to as an untagged VLAN which is all that most home networks have.

When you add VLANs to a managed switch, each VLAN receives a number also referred to as a tag. Each VLAN normally has its own address range and gateway and also has its own DHCP address scope. VLANs may or may not route traffic between them. A router that is enabled to manage Layer 3 can establish and manage rules for routing or blocking traffic between VLANs.

As discussed in the initial network tutorial, a practical use of a VLAN might be placing all of your internet cameras on a separate VLAN so those cameras do not share the network with your



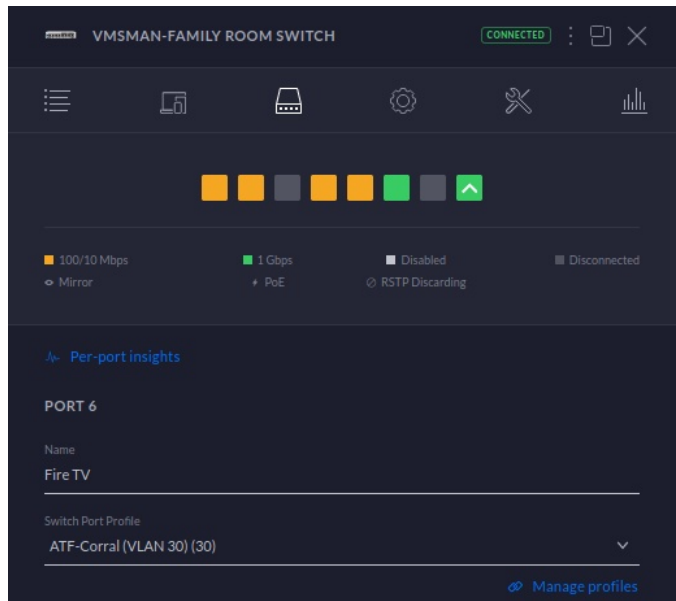
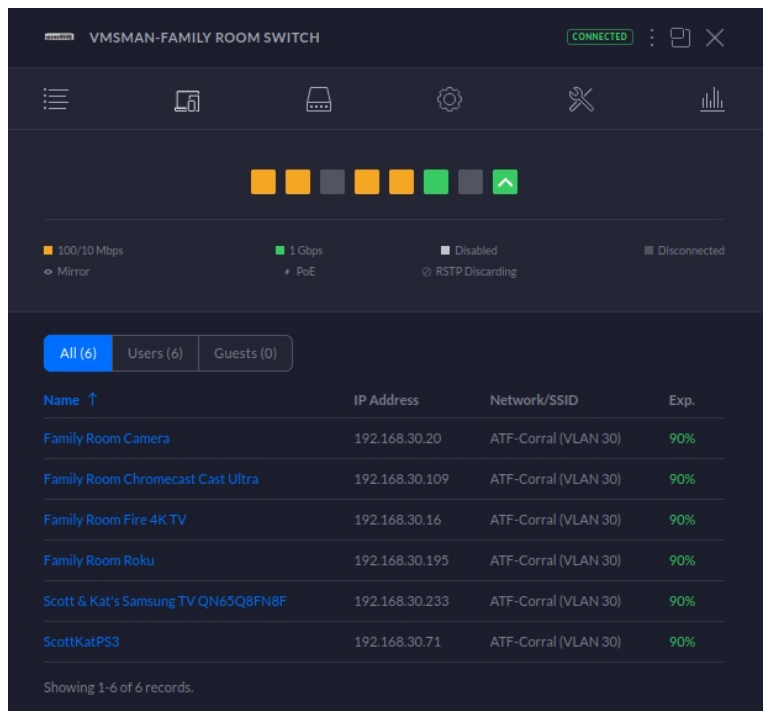
computers. This may provide an advantage related to security and traffic management. In the example network above, there is a MAIN LAN and three VLANs. The gateway address for

each VLAN is the address of the router for that VLAN. The gateway is the route to take in order to get to the the router or to get to the Internet. The layer 3 routing rules on the router describe what traffic is allowed to go where. In my network, things like cameras, smart speakers, and smart hubs are on the Internet of Things (IoT) network. Those devices all get a DHCP address of 192.168.30.x as you can see from the diagram above. My router rules say that my main LAN can communicate with those devices and receive responses from them. However, none of the devices on the IoT VLAN can initiate traffic to my main LAN. This protects my computers from devices that might not provide the best security or that may make outbound connections to unsafe locations.

On a managed network, it is possible to define a wireless SSID that is on a particular VLAN number. That way, wifi devices will be placed on that VLAN when the specified devices connect to that wifi network. In the case of a port on a managed switch, we can define the switch port with a profile that dictates that anything connected to that port will be on a particular VLAN. In my case, I have switch ports in my family room entertainment center that are defined for the IoT VLAN and

they host devices like a Roku, a PS3, Amazon Fire Stick 4K and a camera. This provides a very secure way of hosting devices that may not be completely secure. The image at the left shows these devices connected to the family room switch. This is a managed switch and so it allows the use of the tagged VLANs. Note that these devices all have addresses in the 192.168.30.x network range, whereas my main LAN has addresses in the network range 172.16.x.y.

My routing rules prevent these devices from ever accessing devices on my Main LAN.

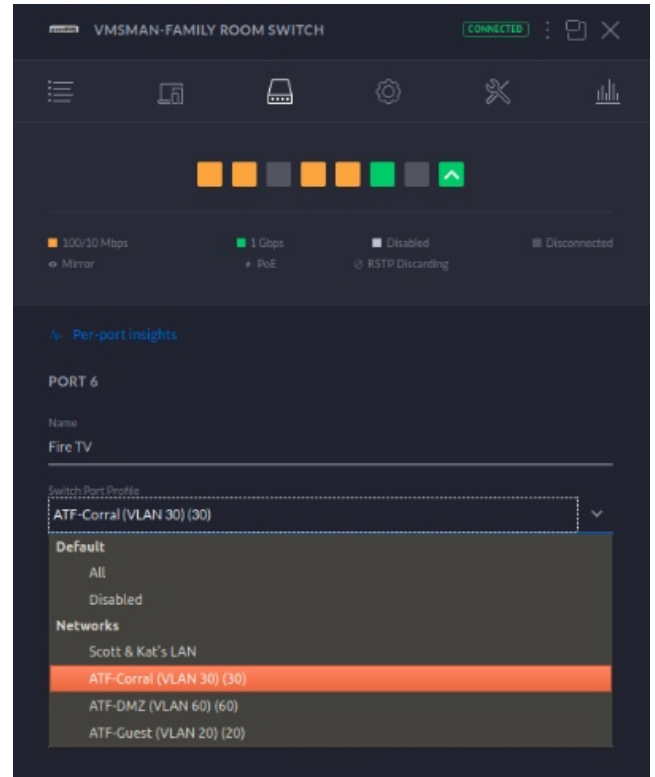


In the image at the left, you can see the switch port profile for port number six on the switch that hosts the Fire TV device and that it is defined for VLAN 30, which is the IoT network, named ATF-Corral on my network.

The selection of the port profile can be seen in the drop down menu at the right. You can see all of the VLAN Networks listed with the ATF-Corral (IoT) VLAN highlighted.

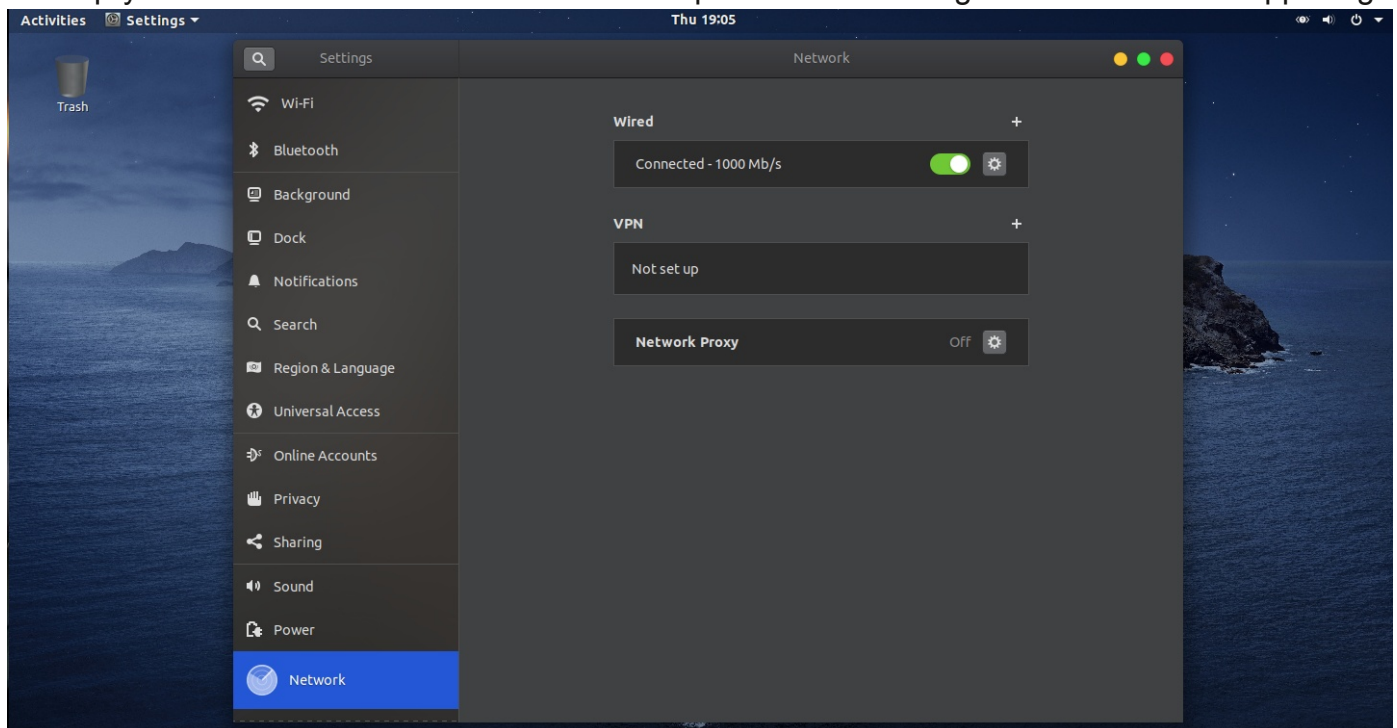
In addition, note that you can disable the particular port, or set the port to the "All" profile. "All" means the port can potentially use VLAN tagging if the connected client ethernet controller and driver software for that controller understands VLAN tagging which is defined in the IEEE 802.1Q specification which supports VLAN tagging of ethernet frames.

Generally, the main LAN ports are set to the "All" profile unless there is a security reason why client nodes connected to the Main LAN should not have the ability to ever connect to another VLAN.



The rest of this document will be devoted to seeing how to configure our Ubuntu guest to connect to one particular tagged VLAN only. You should understand that if you configure a network controller, by default it will connect to the default untagged main LAN and get a DHCP address from it. In order to connect to a VLAN, a new network profile on the client machine must be created or the port profile must be changed to that VLAN.

Start up your Ubuntu VM which we created in a prior exercise. Login and click on the upper right



corner of the top panel. Select the screwdriver/wrench icon to open the "settings" window and select "network" from the left side of that window. Your screen should look like above.

```
scott@scott-VirtualUbuntu: ~  
File Edit View Search Terminal Help  
scott@scott-VirtualUbuntu:~$ ifconfig  
  
Command 'ifconfig' not found, but can be installed with:  
  
sudo apt install net-tools  
  
scott@scott-VirtualUbuntu:~$
```

Note that you see only one "Wired" connection. That is the connection to your Main LAN which is untagged and so it is the default.

To verify this, open a terminal in the VM with a CTRL-ALT-T. When

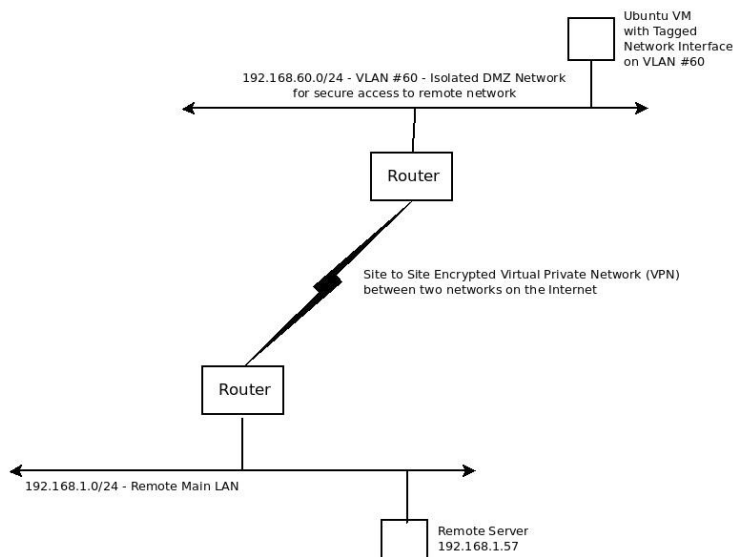
the terminal opens, type "ifconfig". You will get an error that ifconfig is not found. Follow the directions and install "net-tools". Once it is installed, try the ifconfig command again.

```
scott@scott-VirtualUbuntu: ~  
File Edit View Search Terminal Help  
scott@scott-VirtualUbuntu:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 172.16.1.74 netmask 255.255.0.0 broadcast 172.16.255.255  
    inet6 fe80::83a3:34cd:e3e4:a210 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:53:2e:bb txqueuelen 1000 (Ethernet)  
    RX packets 11407 bytes 10695997 (10.6 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 3530 bytes 256177 (256.1 KB)  
    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 122 bytes 10743 (10.7 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 122 bytes 10743 (10.7 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
scott@scott-VirtualUbuntu:~$
```

See on the left where the "enp0s3 interface has been given address 172.16.1.74. That is on the main LAN.

One of the VLANs that I have created on my network is VLAN #60 which I refer to as my demilitarized zone (DMZ). The DMZ is designed to temporarily host a computer that requires special access. In this example, there is a site to site Encrypted Virtual Private Network (VPN) between my network and a network elsewhere on the Internet. The VPN assures that the link over the Internet is secure and the DMZ VLAN prevents a system from the remote VLAN from gaining access to my main LAN which is at a different address range from the 192.168.60.0/24 network range of the DMZ. The goal is for the Ubuntu VM to connect to VLAN 60 through the same

ethernet controller and switch port, but to get an address on the DMZ network and then to use the VPN on that network to access a remote server at 192.168.1.57 on the other side of the VPN.

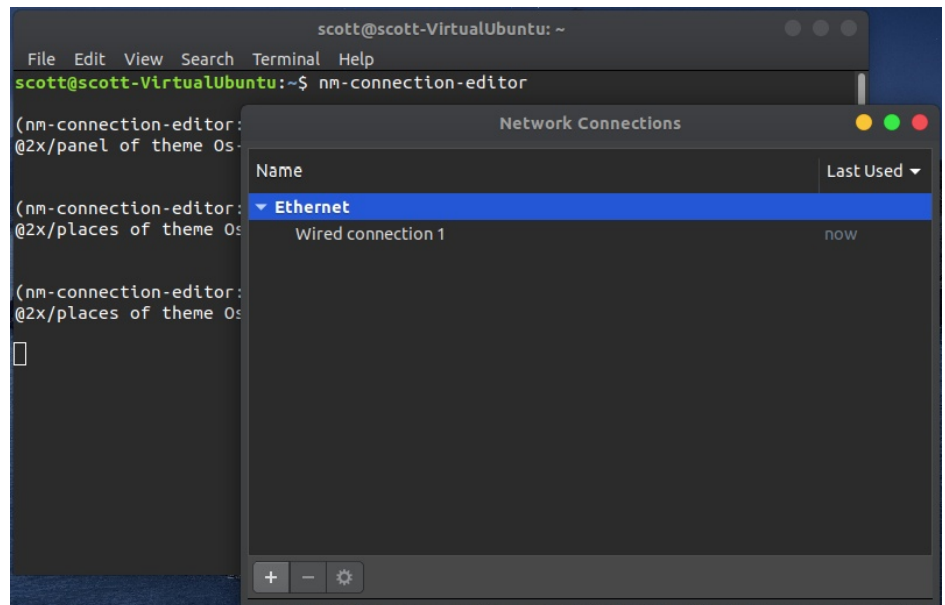


This is possible because the switch port to which the host PC for the VM is connected has its switch port profile set to "All". This will allow the VM to create a network connection which will be tagged to VLAN #60 thus granting the VM an address on 192.168.60.x.

First note that it is not possible to ping the remote server initially because the VM still has an address on the main LAN.

```
scott@scott-VirtualUbuntu: ~  
File Edit View Search Terminal Help  
scott@scott-VirtualUbuntu:~$ ping 192.168.1.57  
PING 192.168.1.57 (192.168.1.57) 56(84) bytes of data.  
^C  
--- 192.168.1.57 ping statistics ---  
71 packets transmitted, 0 received, 100% packet loss, time 71747ms  
scott@scott-VirtualUbuntu:~$
```

In the terminal, launch "nm-connection-editor". You can save this program to the dock if you like.

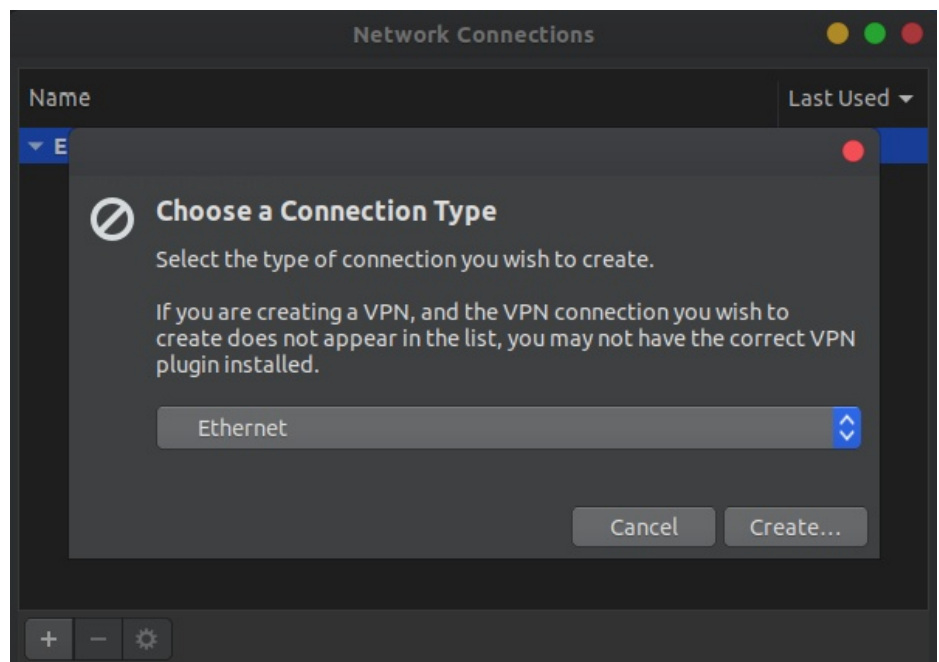


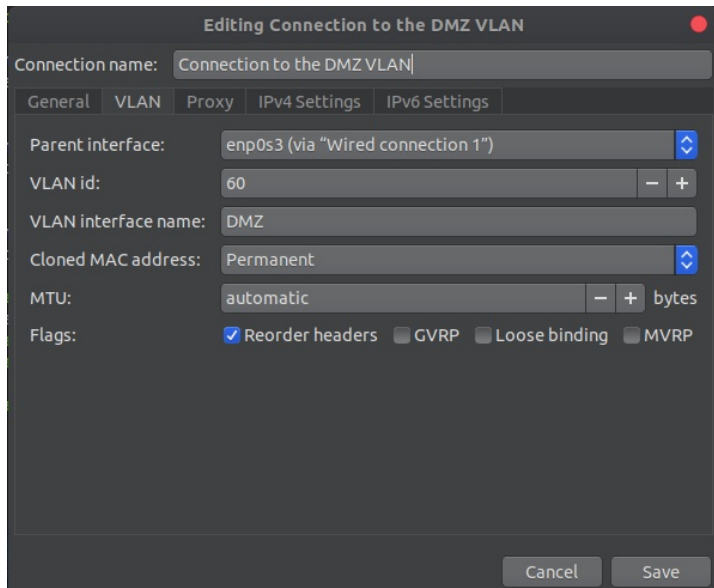
Note there is only one "wired connection" now and it is the untagged connection to the main LAN.

Click the "+" at the bottom of the Network Connections screen.

Choose the drop down menu at right that says "Ethernet", change it to "VLAN" and click "Create".

This will access the screen to create your new tagged VLAN connection.

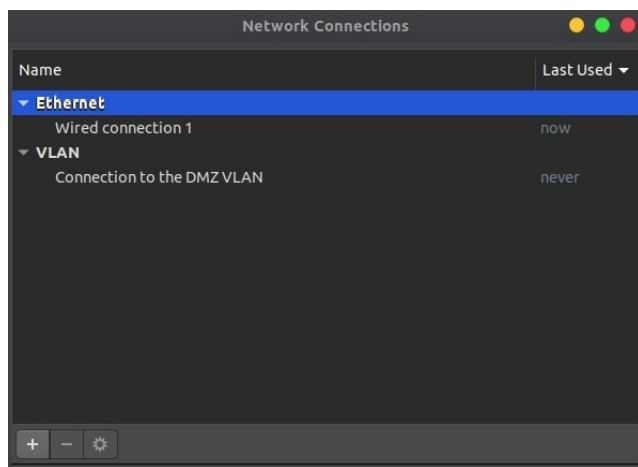




Enter the information as indicated on the screen image at left. Note the the "parent interface will be whatever your VM Ethernet device is.

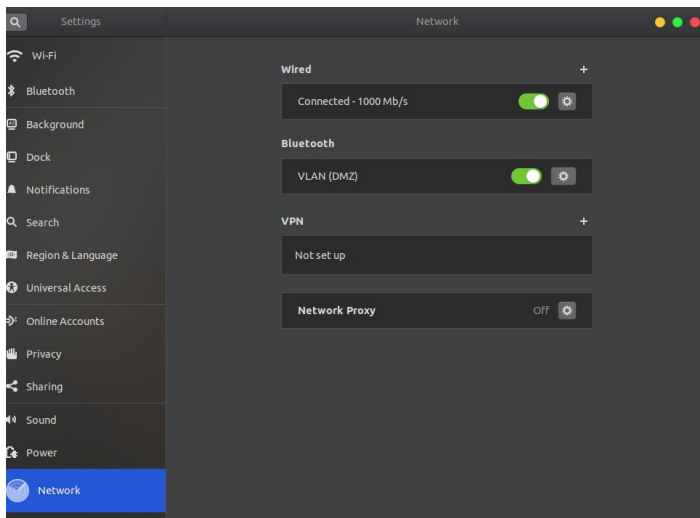
This connection will effectively provide a network connection only to the tagged VLAN #60 which is the DMZ network.

Click "Save".



Your network connections screen should now appear as on the left with your new VLAN connection. This is interesting because we have one ethernet controller and two connections which go to separate networks. This is the beauty of VLANs.

You can close the network settings screen now.



Go to the upper right hand corner of the top panel, click and then click on the wrench to access settings. Select "Network" in the settings window.

The Gnome network settings has an interesting bug where it lists our new VLAN connection as a Bluetooth connection. This does not affect the operation though.

Note that both the Wired connection (main LAN) and the DMZ VLAN are both turned on.

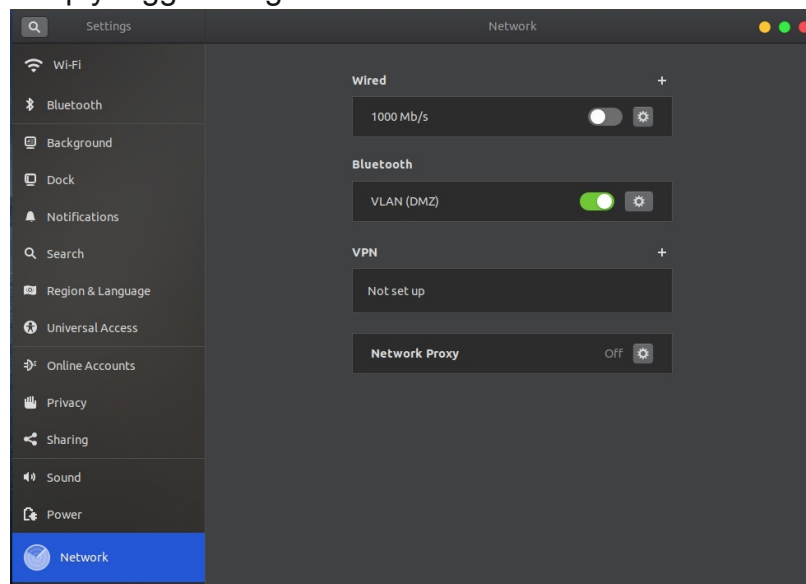
In the terminal, execute "ifconfig" to see the current network connections and you will see both the main LAN at 172.16.1.74 and the DMZ at 192.168.60.36 in the example at right.

The exciting part is that both connections are hosted on the same ethernet controller and the same cable. A router and switch that can do layer 3 routing provides this powerful capability.

Without specific static routes, having both connections turned on might be confusing.

For our purposes, we want to turn off the main LAN connection from the network settings screen. Simply toggle the green indicator to off next to the wired connection.

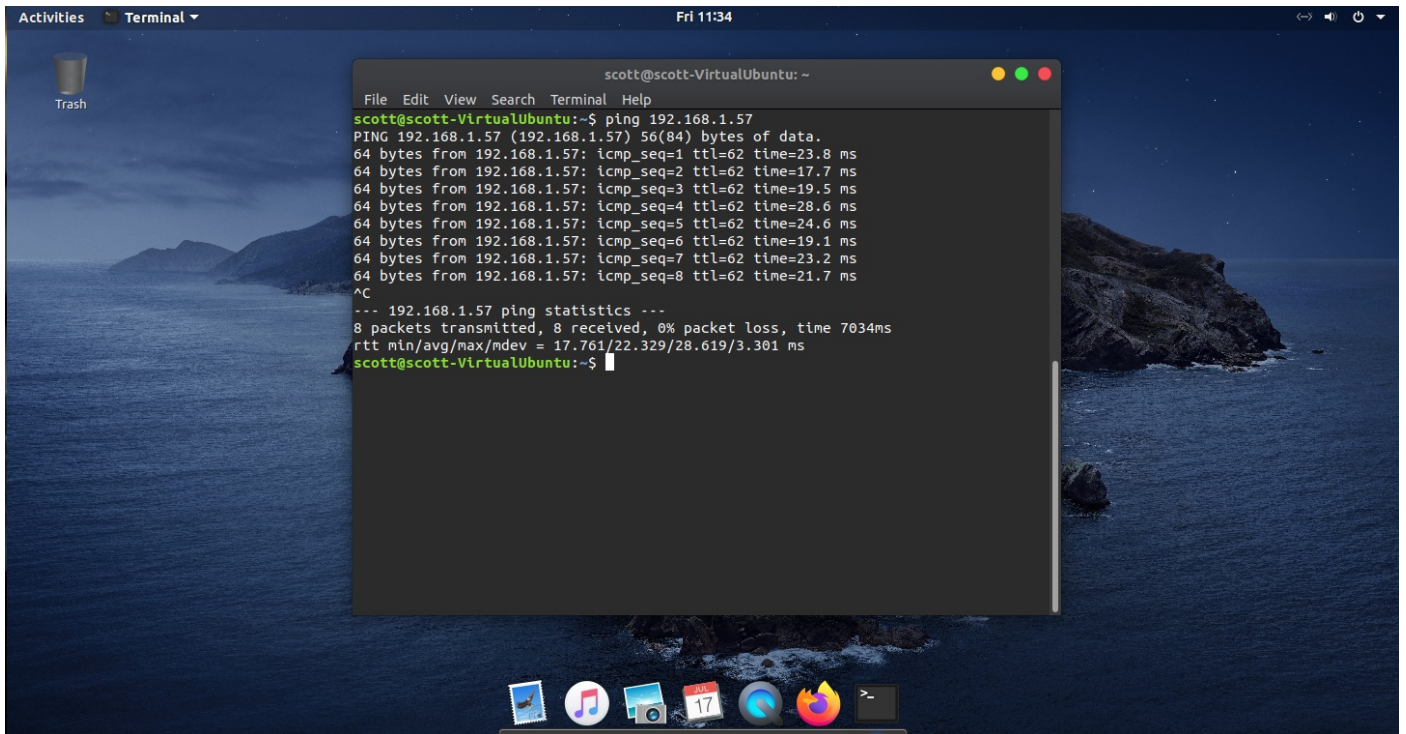
```
scott@scott-VirtualUbuntu: ~  
File Edit View Search Terminal Help  
scott@scott-VirtualUbuntu:~$ ifconfig  
DMZ: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.60.36 netmask 255.255.255.0 broadcast 192.168.60.255  
    inet6 fe80::cd07:aa27:e327:ea6f prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:53:2e:bb txqueuelen 1000 (Ethernet)  
    RX packets 319 bytes 55728 (55.7 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 111 bytes 13504 (13.5 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 172.16.1.74 netmask 255.255.0.0 broadcast 172.16.255.255  
    inet6 fe80::83a3:34cd:e3e4:a210 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:53:2e:bb txqueuelen 1000 (Ethernet)  
    RX packets 42988 bytes 19962736 (19.9 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 3567 bytes 304676 (304.6 KB)  
    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 268 bytes 23508 (23.5 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 268 bytes 23508 (23.5 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```



You can see in the screen image at the left that only the DMZ VLAN is turned on. Exit the network settings screen now.

```
scott@scott-VirtualUbuntu: ~  
File Edit View Search Terminal Help  
scott@scott-VirtualUbuntu:~$ ifconfig  
DMZ: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.60.36 netmask 255.255.255.0 broadcast 192.168.60.255  
    inet6 fe80::cd07:aa27:e327:ea6f prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:53:2e:bb txqueuelen 1000 (Ethernet)  
    RX packets 576 bytes 103806 (103.8 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 146 bytes 16781 (16.7 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    ether 08:00:27:53:2e:bb txqueuelen 1000 (Ethernet)  
    RX packets 44433 bytes 20148465 (20.1 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 3650 bytes 312113 (312.1 KB)  
    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 299 bytes 25992 (25.9 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 299 bytes 25992 (25.9 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
scott@scott-VirtualUbuntu:~$
```

If you perform another "ifconfig" you will see that the enp0s3 interface is listed, but there is no longer an IP address listed for it.

A screenshot of a Linux desktop environment. The background is a scenic image of a coastline with mountains and water. In the top-left corner, there is a 'Trash' icon. The top bar shows 'Activities', 'Terminal', and the time 'Fri 11:34'. A terminal window is open in the center, displaying the output of a ping command to 192.168.1.57. The terminal shows 8 successful ping packets with varying round-trip times (ranging from 17.7 ms to 28.6 ms) and a summary indicating 0% packet loss and an average time of 22.329 ms. The desktop has a dock at the bottom with icons for a file manager, music, camera, calendar, and other applications.

```
scott@scott-VirtualUbuntu: ~  
File Edit View Search Terminal Help  
scott@scott-VirtualUbuntu:~$ ping 192.168.1.57  
PING 192.168.1.57 (192.168.1.57) 56(84) bytes of data.  
64 bytes from 192.168.1.57: icmp_seq=1 ttl=62 time=23.8 ms  
64 bytes from 192.168.1.57: icmp_seq=2 ttl=62 time=17.7 ms  
64 bytes from 192.168.1.57: icmp_seq=3 ttl=62 time=19.5 ms  
64 bytes from 192.168.1.57: icmp_seq=4 ttl=62 time=28.6 ms  
64 bytes from 192.168.1.57: icmp_seq=5 ttl=62 time=24.6 ms  
64 bytes from 192.168.1.57: icmp_seq=6 ttl=62 time=19.1 ms  
64 bytes from 192.168.1.57: icmp_seq=7 ttl=62 time=23.2 ms  
64 bytes from 192.168.1.57: icmp_seq=8 ttl=62 time=21.7 ms  
^C  
--- 192.168.1.57 ping statistics ---  
8 packets transmitted, 8 received, 0% packet loss, time 7034ms  
rtt min/avg/max/mdev = 17.761/22.329/28.619/3.301 ms  
scott@scott-VirtualUbuntu:~$
```

If I now ping the remote server on the other side of the VPN at 192.168.1.57, you can see from the screen above that we have connectivity. This works because the site to site VPN was established and connected to our DMZ VLAN #60 and our tagged connection to VLAN #60 allows us to have an address of 192.168.60.36 which can route to 192.168.1.57 which is on the other side of the VPN on the remote network.

The use of switch port profile configurations is very powerful. Besides setting a switch port to the "All" profile, for even greater security, I could have created a custom profile for the switch port that would allow connection to the MAIN LAN and the DMZ VLAN, but no other VLANs.

If you work for a company, chances are they have you configure a client based VPN tunnel on your computer to the network at your company. This assures that all traffic over the public internet is secure in an encrypted tunnel.

In our example, we used a site to site VPN which can provide all the computers on one network access to all the computers on a remote network securely. In our example, we did not want the remote network to be able to access all of the systems on our main LAN and so I had previously created a tagged VLAN #60 which I named DMZ on my router.

I leveraged the "All" switch port profile to allow me to create a new network connection on my VM which specifically tagged VLAN #60. This allowed my guest VM to have an address on the DMZ VLAN and to access resources connected to it.

While all this was taking place, the host computer was still communicating on the main LAN. You can see from this exercise that VMs are very flexible, Linux VMs are very easy to configure, and VLANs provide enhanced security and flexibility for your network.