**Ubuntu Networking**

Networking

Here are some basic things a system administrator will need to know about the networking environment for their Ubuntu Server. This could vary depending on what the use for the server is. This Ubuntu Server is networked on my home router through DHCP, which means that the IP Address, subnet, gateway, and DNS settings are automatically given to the server from my home router. In a larger environment, I think it is good practice to set the IP to static, which means it will remain consistent and never change. This helps with scripting and automation when the IP Address is needed.

**Default Settings**

A screenshot of a computer

Description automatically generated

The default network configuration on this server is DHCP, which means the server will automatically receive an IP Address from the modem as shown above. The command **ifconfig** shows network information such as IP Address, subnet, broadcast address, and MAC Address of each interface. When the system boots, these settings are loaded from a configuration text file located in this path (**/etc/netplan$/00-installer-config.yaml**). Run the command **cat /etc/netplan$/00-installer-config.yaml** to view the default network settings for the server. It is important to note that Ubuntu 20.04, or later, uses something called the Subiquity Installer. This tool simplifies the install process and generates the default settings, you will see below a line in the screenshot stating, “This is the network config written by ‘subiquity’.

A screenshot of a computer

Description automatically generated

Here we can see the ethernet interface for this server is using DHCP, as well as which version of Netplan. The settings within this file can be edited using a text editing tool like emacs. For example, to change the IP Address from DHCP to static, use emacs to change the DHCP setting from *true* to *false*, and save the file. See below:

A black screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

Remember to press CTRL + X then CTRL + C to save the file. And then reload the netplan using command **sudo netplan apply**.

A screenshot of a computer program

Description automatically generated

Apply and reload / apply the new netplan configuration.

A screenshot of a computer screen

Description automatically generated

Set the IP Address to something within the correct subnet. The **dev** option declares which interface this IP Address is being assigned to, in this case it is the ethernet adapter. Last, the **brd** is the option used to set the broadcast IP, which is used to communicate to all devices on the local network.

1. **Command – “apt list –installed | less”**

A screenshot of a computer screen

Description automatically generated

* One of the first things I would check on my server is what programs are installed or updated. The command shows the list of installed programs, and **| less** gives us the ability to arrow up and down through the output, which is not available by default in the terminal.

1. **Command – “cat /var/log/apt/history.log | less”**

A screenshot of a computer

Description automatically generated

* Along with knowing what is installed, it is critical to know when these programs were installed / updated. This **history.log** file keeps track of package installs, upgrades and removals. Optional - use **| less** at the end of the command to have the option to arrow up/down through the output.

1. **Command – “cat /etc/netplan/\*.yaml”**

A screen shot of a computer

Description automatically generated

* This command shows which network interfaces are enabled, and if the IP Address is set statically or through DHCP. If the server is experiencing network connectivity issues, this is a good place to start troubleshooting.

1. **Command – “ip addr show” & “ip route show”**

A screenshot of a computer

Description automatically generated

* To ensure the server has the correct IP settings, use **ip addr show** to see the IP Address Subnet. Then **ip route show** to confirm the routing / gateway settings. If there are issues with communication on the server, this is a fundamental area to check and make sure the settings are correct.

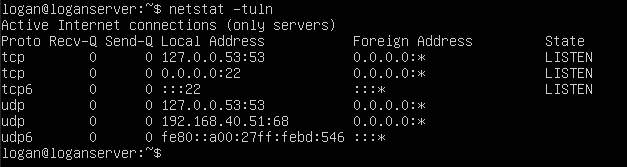
1. **Command – “cat /etc/resolve.conf”**

A screenshot of a computer screen

Description automatically generated

* To show the DNS settings for the Ubuntu server, run **cat /etc/resolv.conf.** DNS is crucial when tying host names to IP Addresses and incorrect settings can cause connection issues. Verify the settings are correct to ensure proper connectivity.

1. **Command – “netstat -tuln”**



* The **netstat** command is part of the **net-tools** package we installed previously. It can be used to check the open ports and ‘listening’ services on the server. The system admin should regularly check the open ports to find potential security vulnerabilities on the server. Make sure only necessary ports are open to keep out malicious activity on the server. The **-tuln** option with **netstat** shows TCP ports (**t**), UDP ports (**u**), listening sockets (**l**), and **n** will display the IP Address rather than the host name. It is good to note that some companies have very strict security compliance policies regarding which ports can be open. Be sure to document which ports are open and why. For instance, some apps and services may need to have specific ports opened to function correctly.

1. **Command – “sudo ufw status”**

A screenshot of a computer

Description automatically generated

* Knowing the status of the Firewall is another thing admins will need to check and configure to ensure connectivity. This command shows the active firewall rules on the server and which ports and services are allowed.

1. **Script**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

The purpose of this script is to prepare a report containing every command I covered. This type of report can be used to audit the server and confirm the settings are consistent, and no undocumented changes have been made. After running the script, use **ls** to confirm the output of the script is there. And use **cat *filename*** to view the report.

First, I set a variable called ‘date’ which just stores today’s date in a variable, I did this so I can include it in the name of the report. I like the idea of users opening this report and knowing how current this data is. Without current and reliable data, it is very difficult to troubleshoot issues and keep up with server maintenance. I then created one more variable named “file” which stores the name of the file I am sending this report to. This made it more efficient for me so I could avoid typing out the long file name after each command.

With these commands, my goal was to cover the essential network settings needed for troubleshooting issues. Typically, when I run into a networking issue at work, the first thing I check is the network configuration of the client system that is trying to talk to the server. It is important to confirm that the IP Address, subnet, and gateway are set correctly. This covers the Network Layer (layer 3) of the OSI Model we learned in Networks class.

Next, it is vital that we confirm the state of the firewall and open ports. The system admin should regularly check the open ports to find potential security vulnerabilities on the server. If there are any unnecessary firewall rules in place that could present a security risk to the system with unwanted access to the server. Getting consistent updates and reports on both the firewall and ports is very important for keeping the server secure.

**Sources**

* <https://www.aholdengouveia.name/LinuxAdmin/networking.html>
* <https://ubuntu.com/server/docs/configuring-networks>
* <https://linuxconfig.org/ubuntu-22-04-network-configuration>
* <https://unix.stackexchange.com/questions/724273/how-to-configure-a-network-interface-on-ubuntu-server-22-04>