Networking Lab

Jayviar Laporte

Github link- <https://github.com/Jay00324466>

Table of Contents:

**Making the Script File pg 1**

**Adding the Commands: pgs 2-10**

* Ip addr
* Nslookup
* Netstat
* Arp
* Date
* Append

**Adding Permissions pgs 10-11**

**Making The Script File**

To make our script, we need to make a file. To make the file, we use the command:

Nano networkingscript

For any script, we need to make sure we have a schbang at the top of the file. This tells the kernel what interpreters are needed to run the commands. The schbang for this file looks like this:

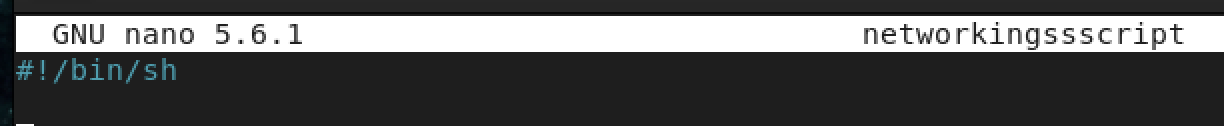
#!/bin/sh

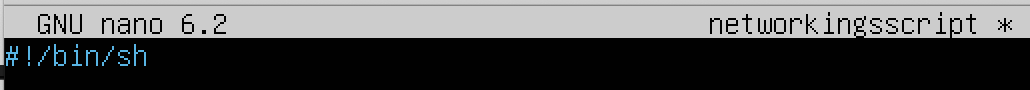
*Note: syntax may look different depending on what you plan on doing with the file. For this case we just want to execute everything within the file.*

*The reason we need this at the top of our script is because it specifies what commands are needed to run the code. /bin/sh is the path that takes us to our systems shell which is ultimately the systems default shell which is where our file will be.*

-Once both tasks are done our screen should look like this:

Centos

Ubuntu



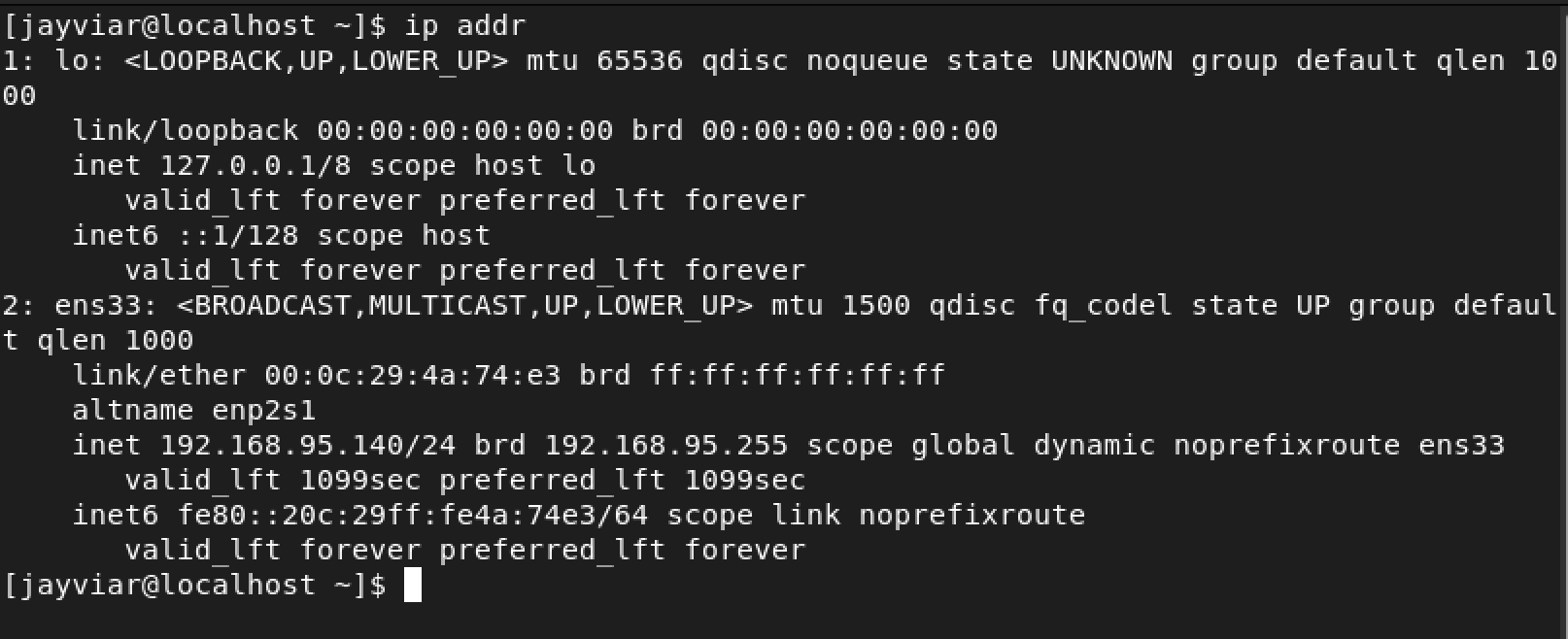
**Adding the commands**

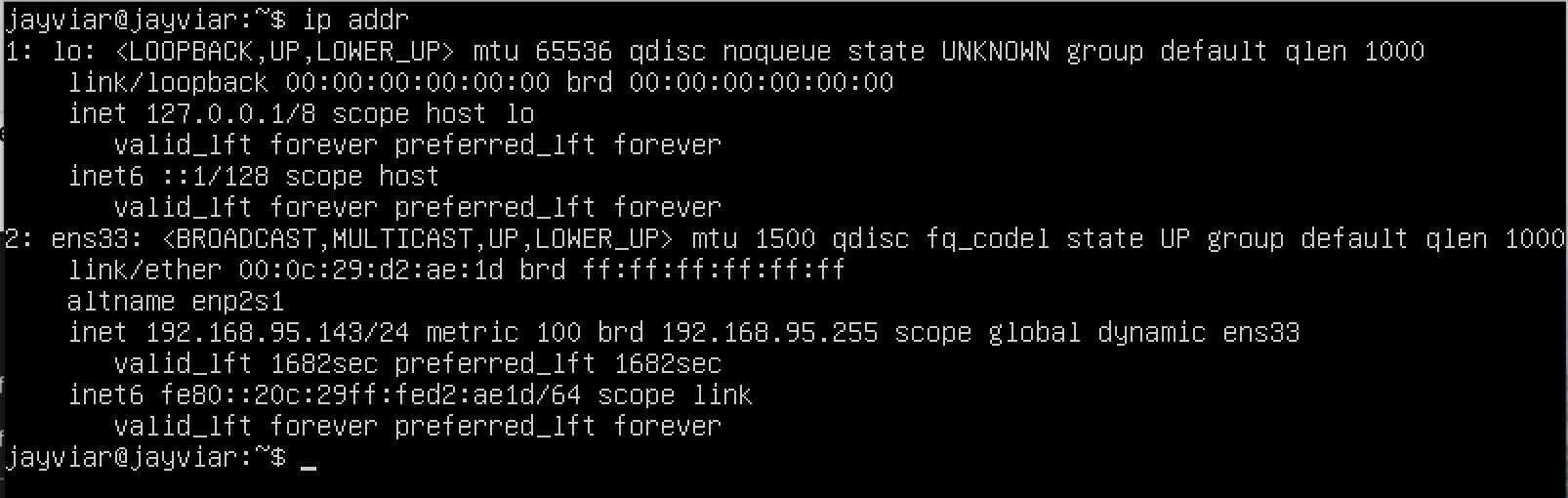
Now to add commands to the script!

**IP addr**

* To Start. We want to add the ip addr command. This command can prove to be useful as someone who may not know the ip address of either server or others on the network will be able to see it.

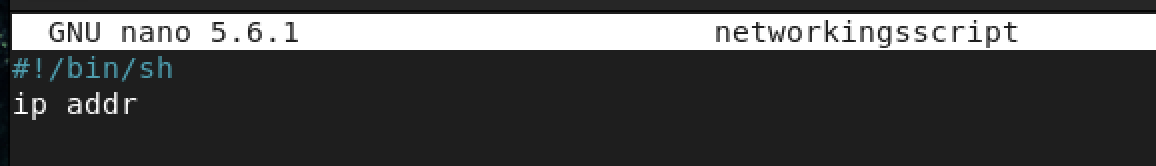
Centos

Ubuntu

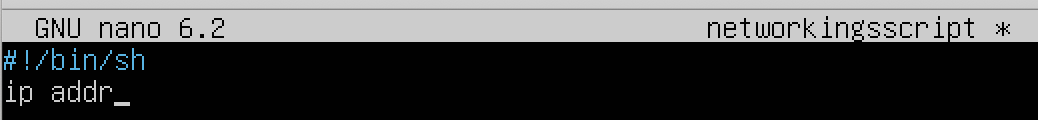


-What our script looks like now

Centos



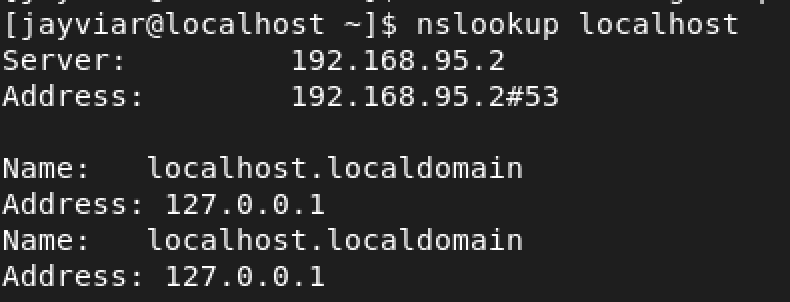
Ubuntu



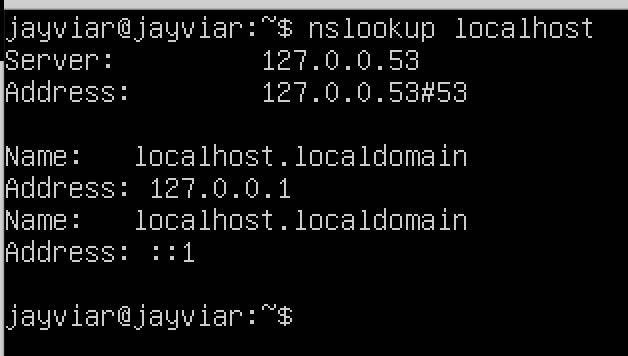
**Nslookup**

* For our next command we are going to add nslookup localhost. This command will get information from the dns server and obtain its domain name and or ip address or both.

Centos

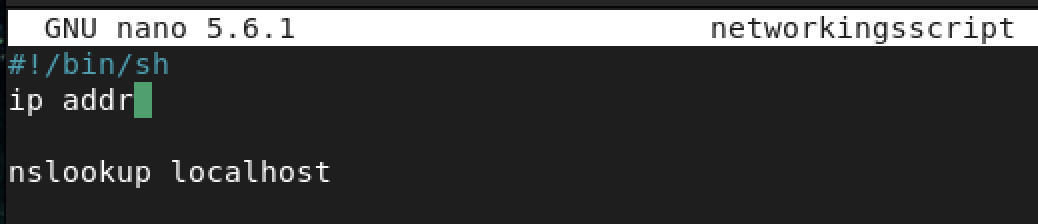


Ubuntu

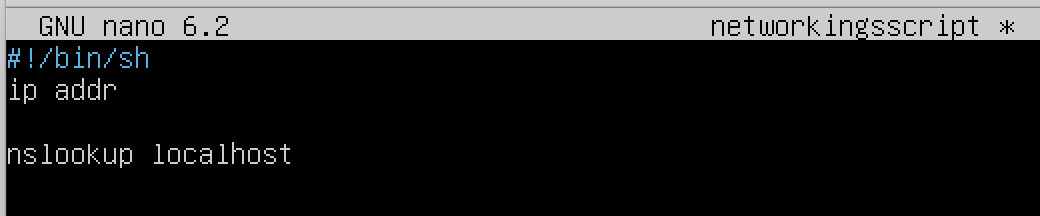


Our script should now look this:

Centos



Ubuntu



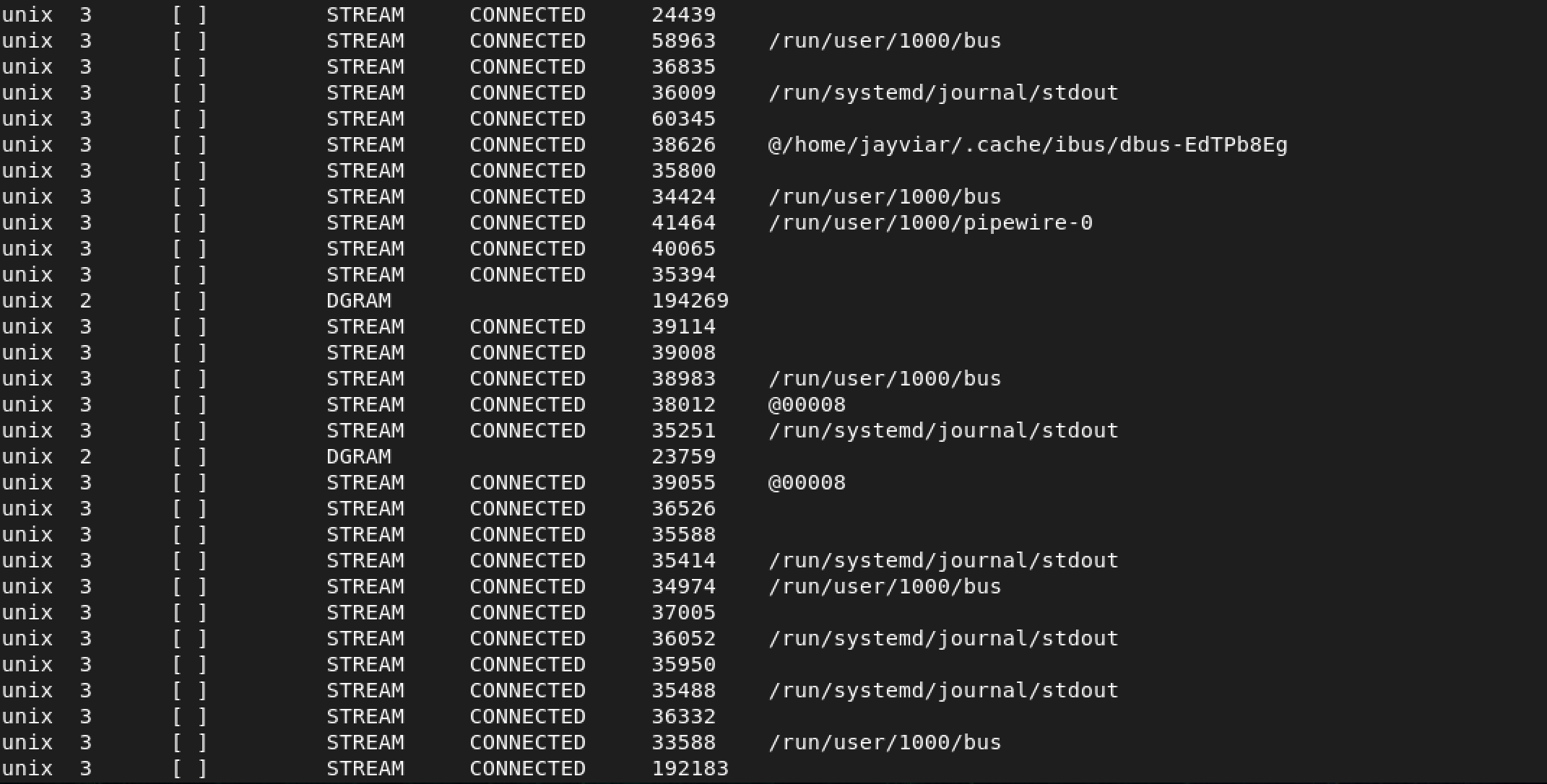
**netstat**

We are now going to be adding in the netstat -ano command. What this will do is display all open ports to our server. This may prove to be helpful as it will let troubleshooting become easier. By seeing what ports are open we can see what connections are up and even identify is something Is not up when it should be

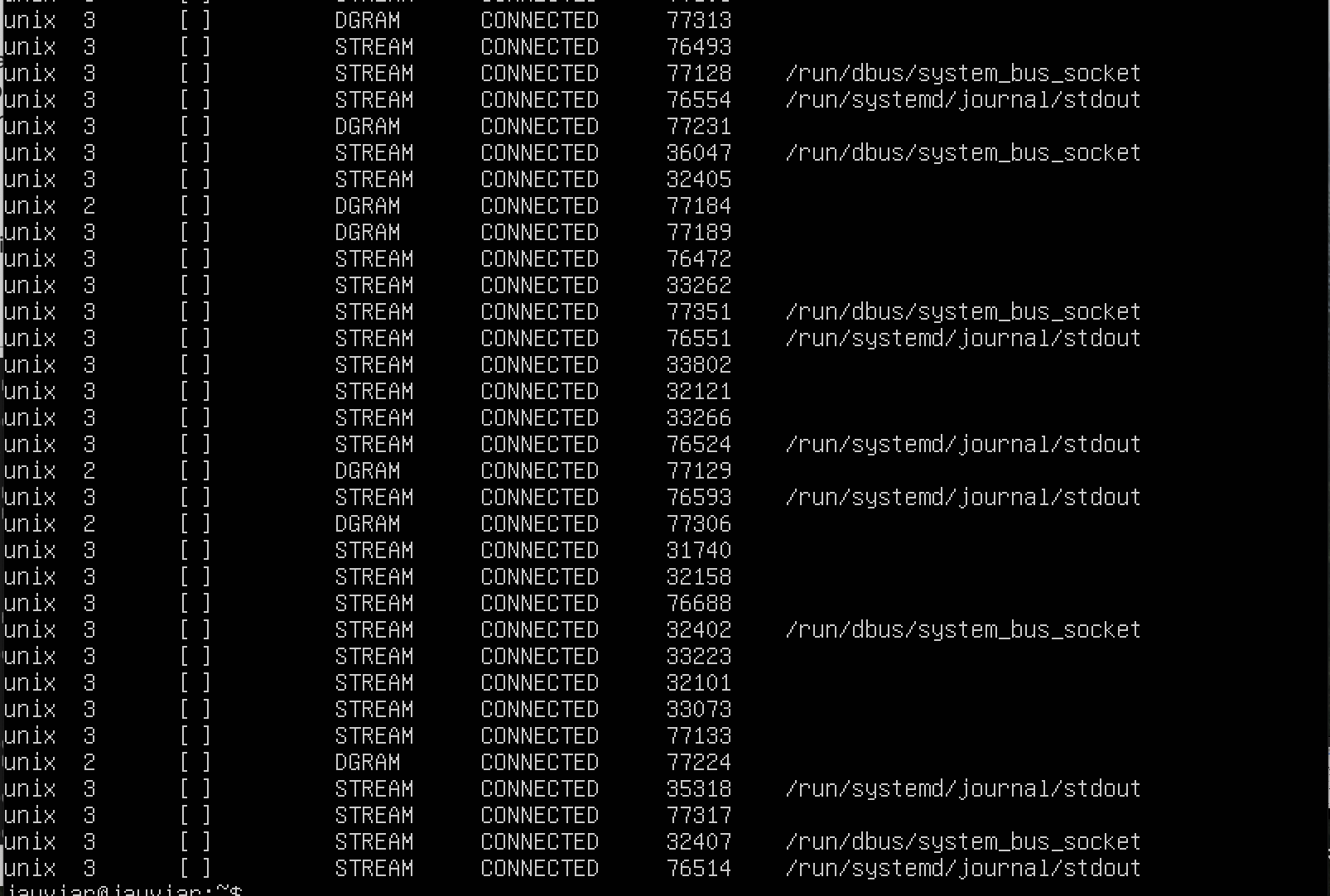
The output of the command will look like this:

{screenshot cut short as there are many lines of code. However this is a snippet of what it should look like}

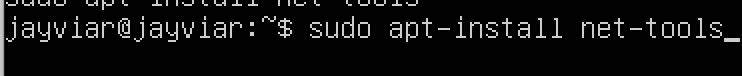
Centos



Ubuntu

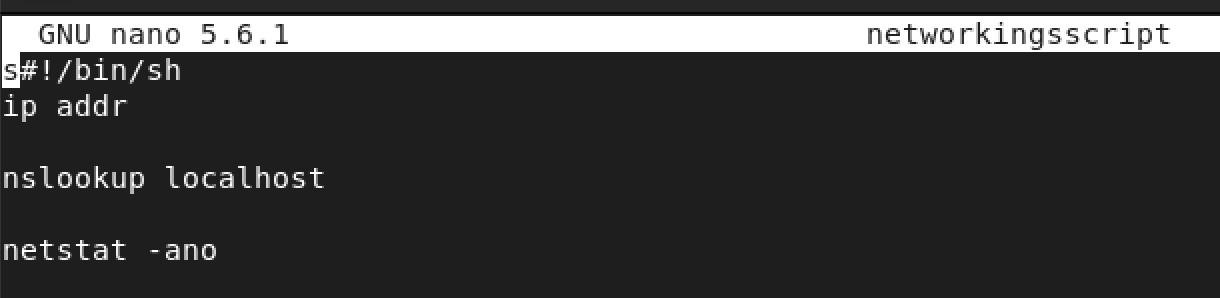


***Note****: We need to install netstat on to our ubuntu server for the script to work on both centos and ubuntu with identical code: Here is what that looks like:*

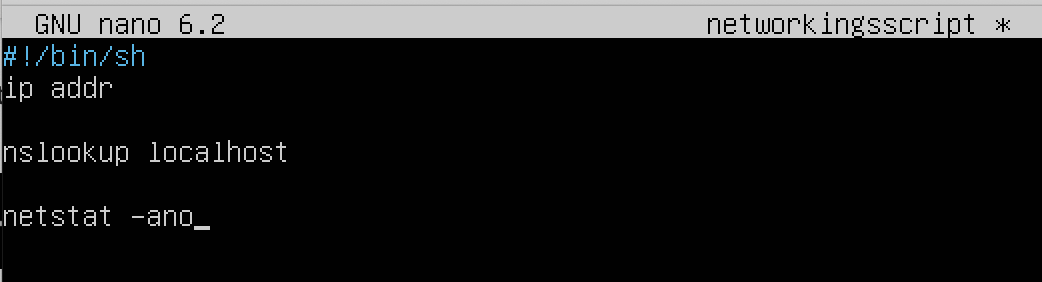


How it will look in our script:

Centos



Ubuntu

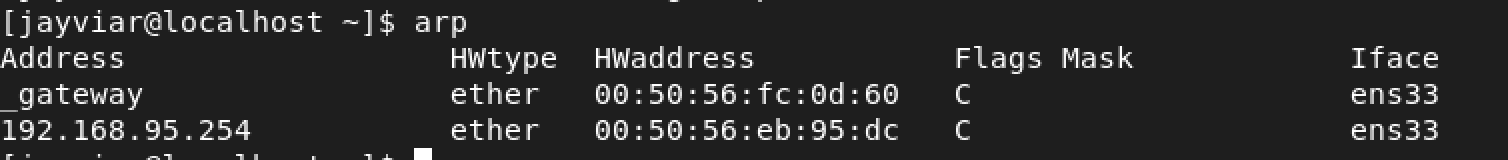


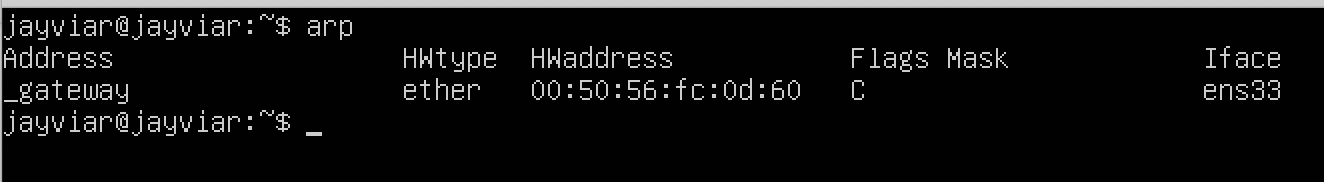
**Arp**

For one of our last commands, we are going to add one that will show some information regarding the mac and ip addresses. This command is called Arp. in short, it will basically display the MAC and ip address of the devices on the server if it at all has them. Somewhat pairing them.

The output will look like this:

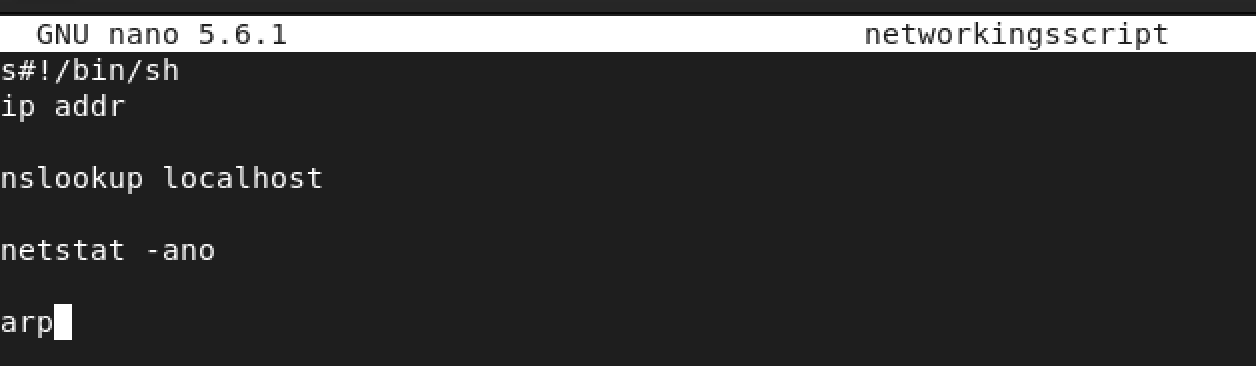
Centos



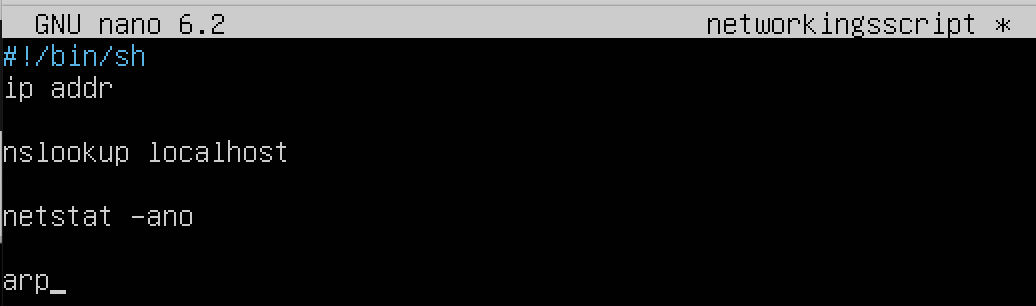
Ubuntu 

The script will look like this now:

Centos



Ubuntu

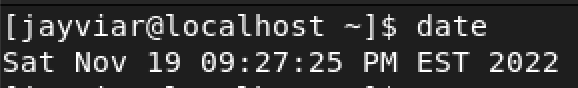


**Date**

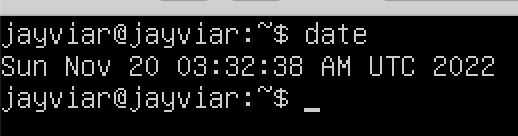
Finally, we are going to add a very simple command, date. This command very literally displays the current date and time of the server. The reason for this, is because we want to make sure that this information is stored in the file at the very bottom so when it is open and or run, it will be the first thing a user sees.

The command output will look like this:

Centos

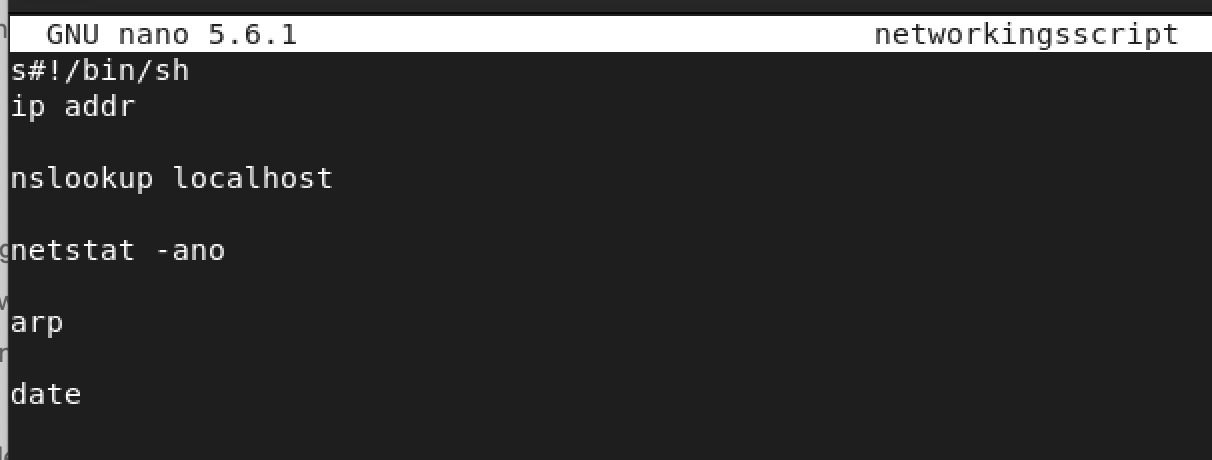


Ubuntu

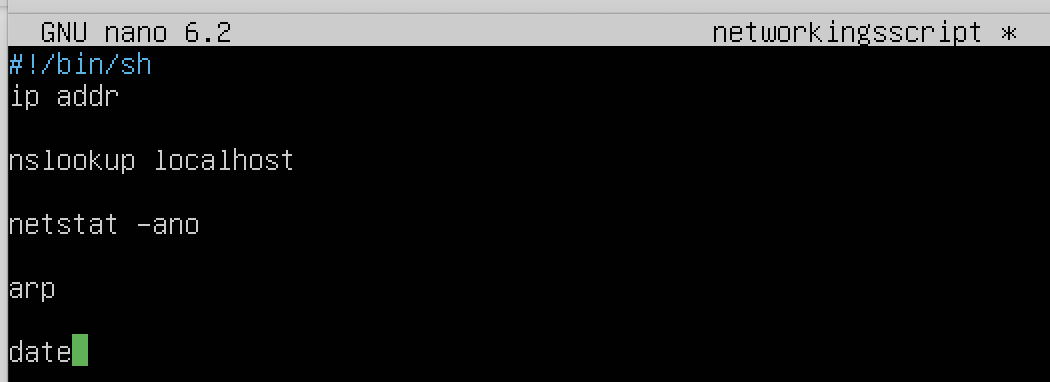


The script will now look like this:

Centos



Ubuntu



**Append**

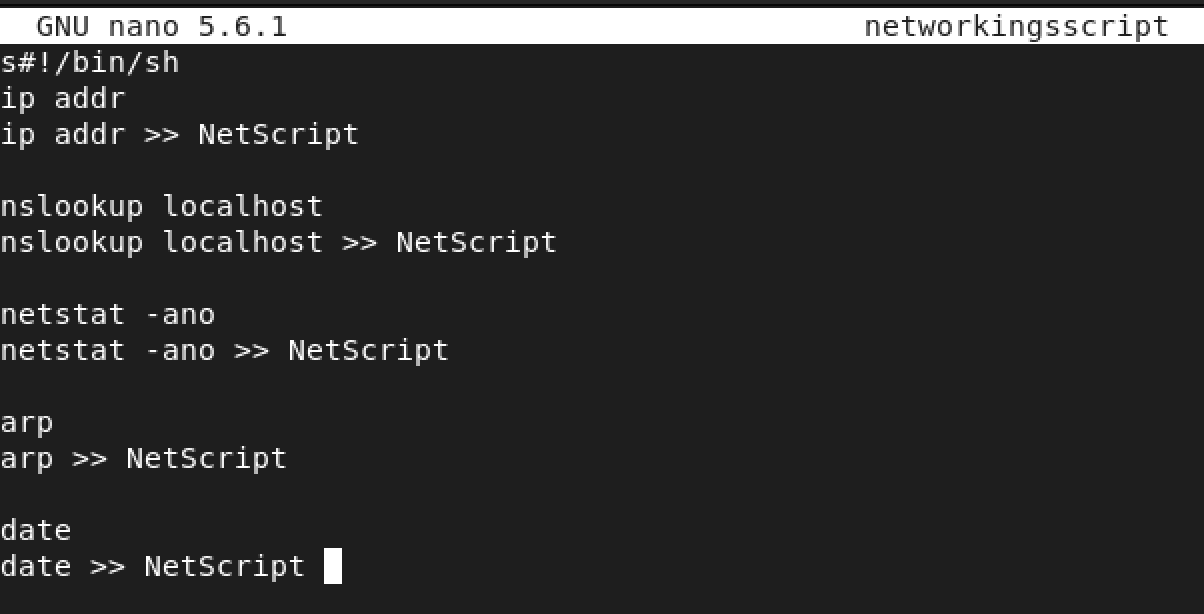
Notice how our file just has commands and that’s it. This is good, but we also want to send it to a file. What we want to do is take all the previous commands and add them again, but this time with a direction pointing to a certain file. We do this by using the >> command. The syntax for this looks like:

command >> filename

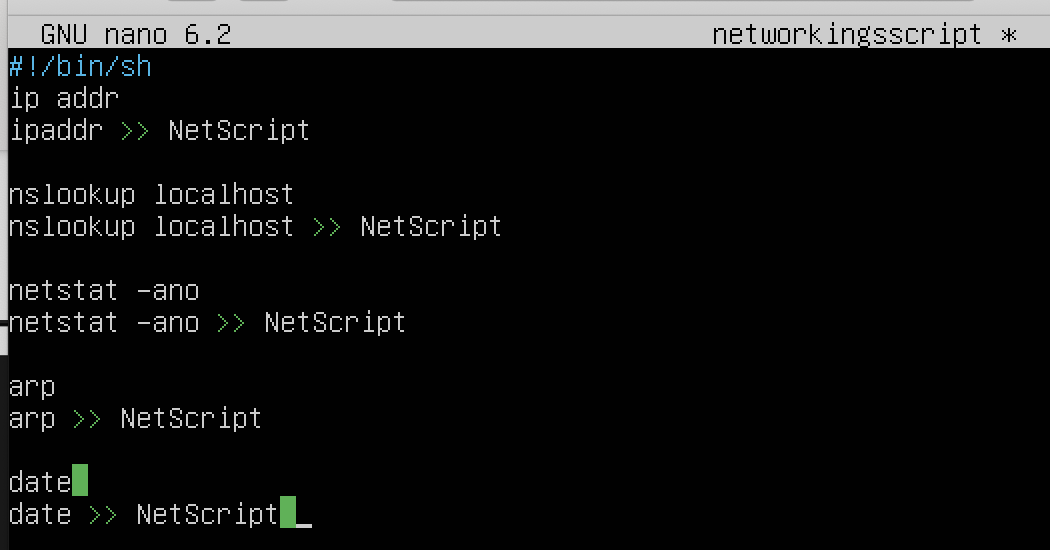
Notice how we need to specify a filename after the >> command. This can be any name we want. Either a existing file, or a new filename. The cool thing about this is that if it is a new file it will automatically create it and put all of this inside it.

Below is a screenshot of what the finished product with every command in our script repeated with itself being sent to the filename NetScript:

Centos



Ubuntu



**Adding Permissions**

Finally, since this is a script, we need to give it chmod permissions. For simplicity's sake. We are just giving It all permissions to just run the code easily.

For this, run the command:

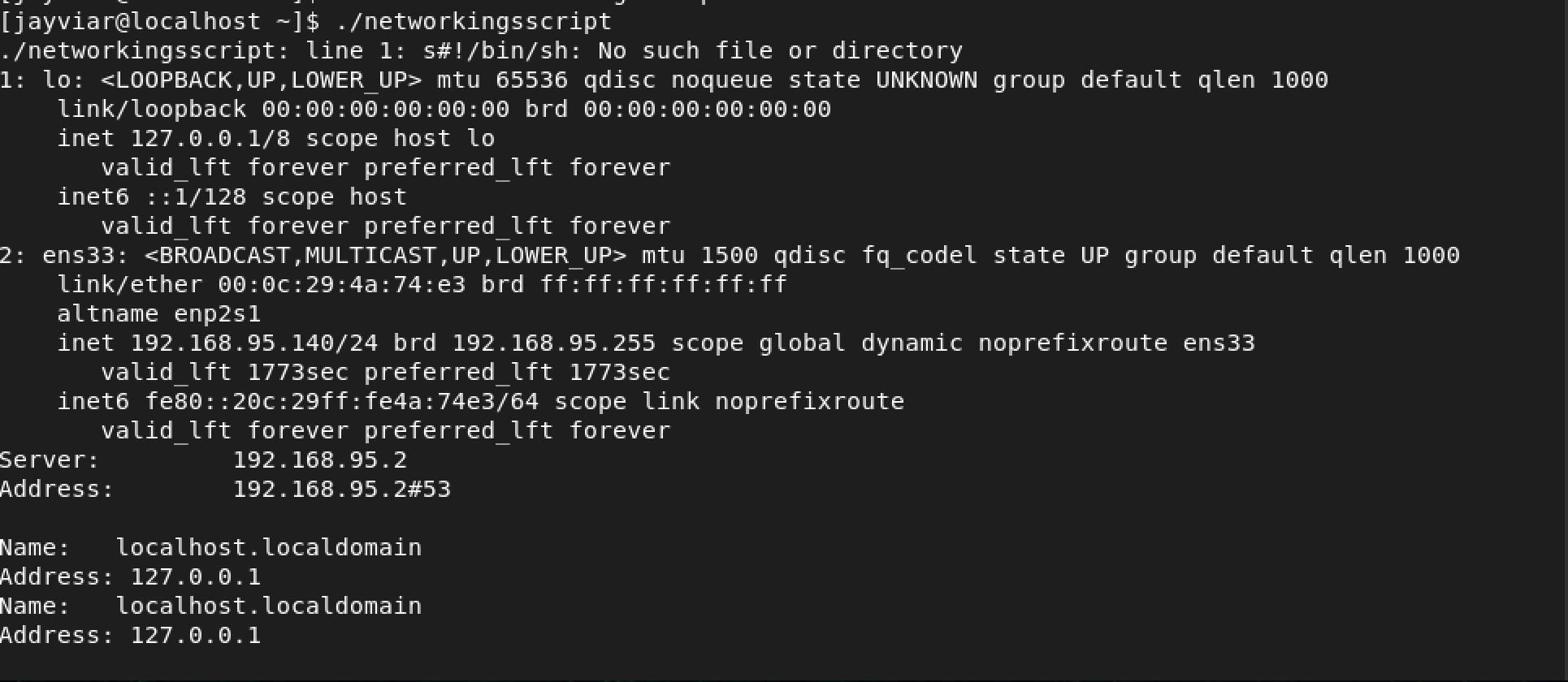
Chmod 777 {networkingsscript}

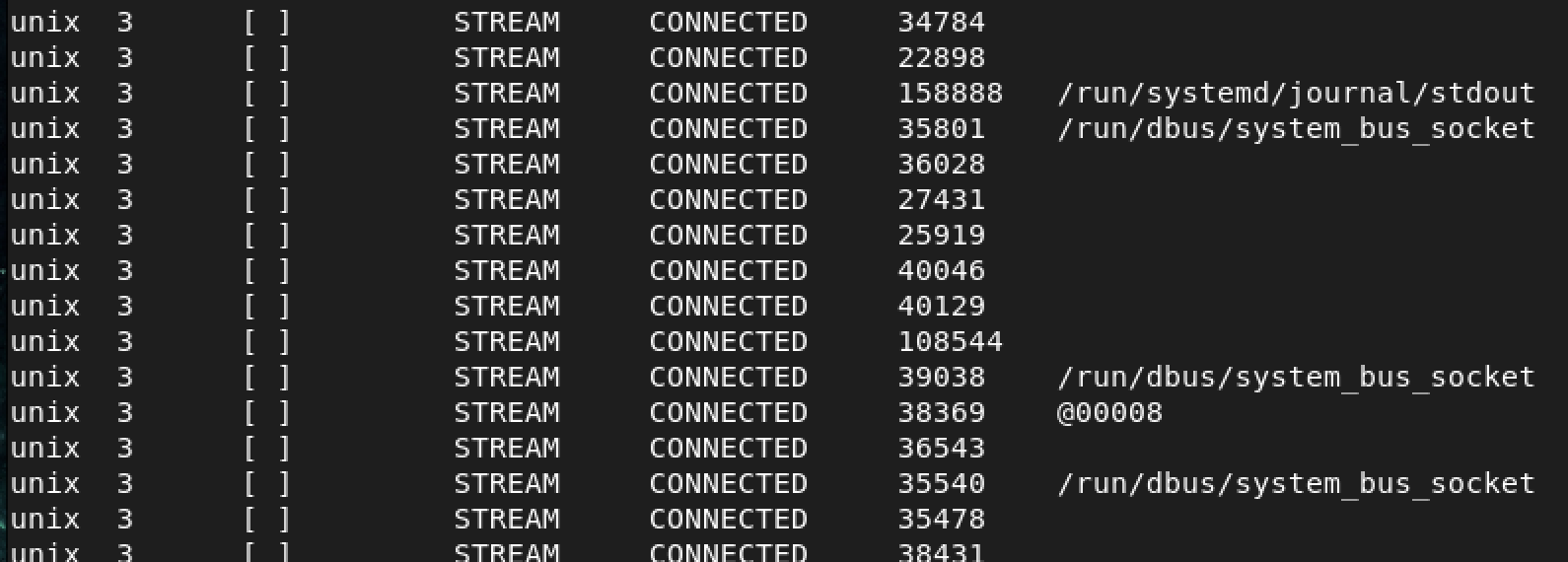
Run the script using

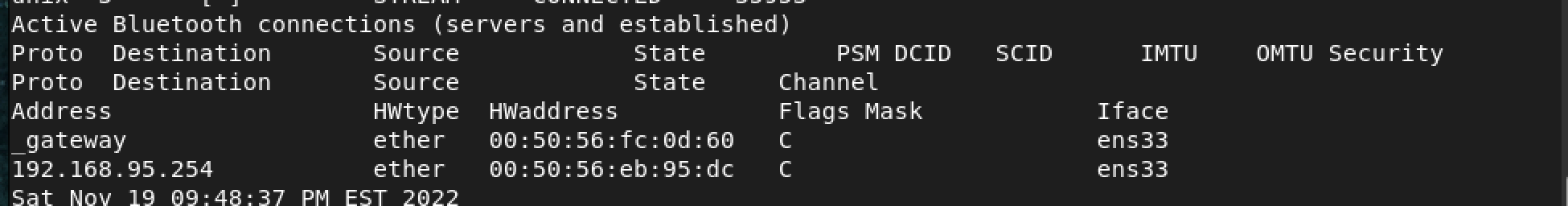
./networkingsscript

The following screenshots are the start, part of the middle, and end. The output is very long so if one wishes to see the entirety of it. If one wishes to view the entire output. The GitHub will have the script attached so anyone can run this on their own.

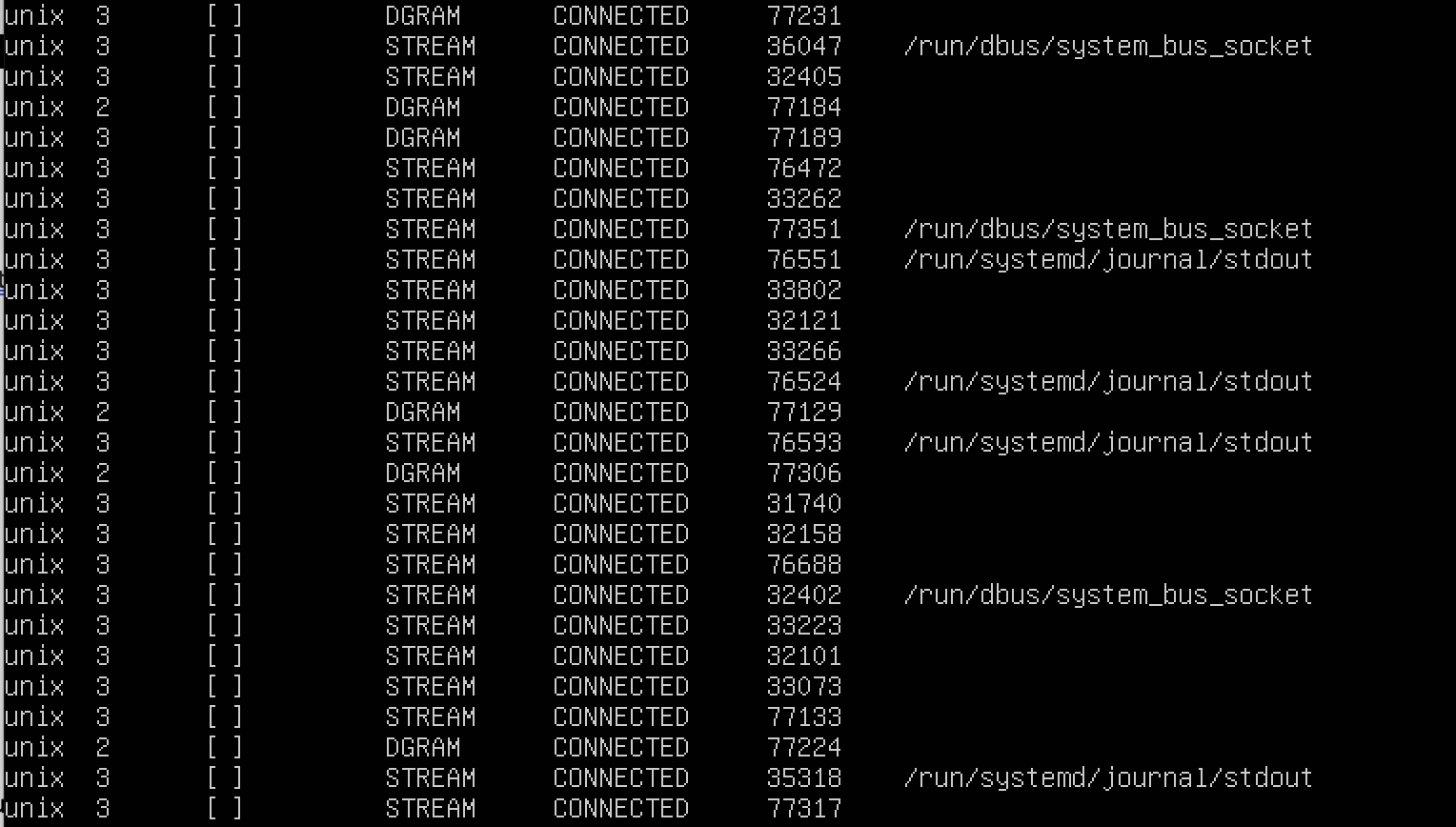
**Centos**

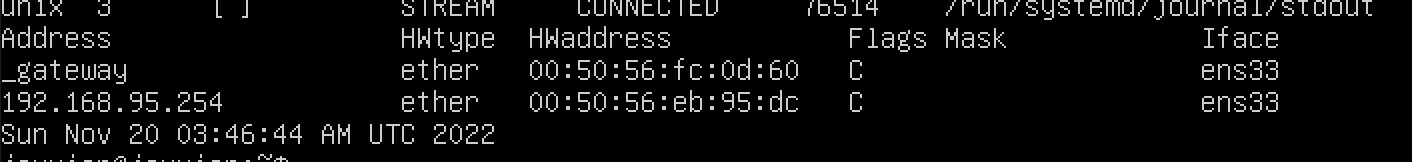






**Ubuntu**





References:

* <https://access.redhat.com/sites/default/files/attachments/rh_ip_command_cheatsheet_1214_jcs_print.pdf>
* <https://www.ibm.com/docs/en/aix/7.2?topic=arp-command>

[ARP Explained - Address Resolution Protocol](https://www.youtube.com/watch?v=cn8Zxh9bPio&t=15s)

[](https://www.youtube.com/watch?v=cn8Zxh9bPio&t=15s)

<https://www.alphr.com/how-to-check-which-ports-open-windows-10-pc/>