Cyberhawks Meeting 3/16/2021

Linux Static IP, DHCP, and SAMBA setup (Ubuntu Server)

Prerequisite material:

Ubuntu Server 20.whatever https://ubuntu.com/download/server (select option 2 then download)

Ubuntu Desktop (or any GUI Linux probably) https://ubuntu.com/download/desktop/thank-you?version=20.04.2.0&architecture=amd64

You will need to install all three of these in VirtualBox. You can start and update them, just don't do anything too weird that will change the network configuration! Also, they will be running simultaneously so make sure you didn't give any of them too many resources or you are going to have some problems.

Note: To install VirtualBox extensions in Ubuntu Desktop you will need to run the command:

sudo apt update -y

Followed by:

sudo apt install make gcc perl -y

This will allow you to then go to the Devices menu in VirtualBox and select the "Insert Guest Additions CD Image". Follow the prompt to enter password and run software. Restart and now you can dynamically adjust the screen size and allow copy/paste functionality between host and VM.

Disclaimer: I am not a genius, I learned this stuff somewhere, here are the reference links:

Static IP: https://www.howtoforge.com/linux-basics-set-a-static-ip-on-ubuntu

DHCP: https://vitux.com/how-to-setup-dhcp-server-on-ubuntu/

SAMBA: https://ubuntu.com/tutorials/install-and-configure-samba#1-overview

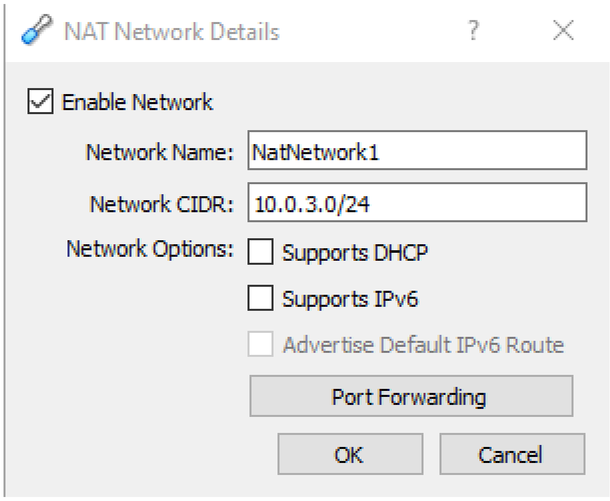
Extra link if you want to connect to SAMBA from Windows 10 (it works and I tried it, I will not be going over this in tutorial, however): https://ubuntu.com/tutorials/install-and-configure-samba#1-overview

Setup:

First, we need to create and configure a NAT network without DHCP enabled already in VirtualBox:

Go to File → Preferences and select “Network” on the left panel.

Create a network with the following specifications (you can name it whatever):



☑ Enable Network

Network Name: NatNetwork1

Network CIDR: 10.0.3.0/24

Network Options: □ Supports DHCP

□ Supports IPv6

□ Advertise Default IPv6 Route

This network will use Network Address Translation to create an internal network for our host devices to communicate on. This network has an implied default gateway of 10.0.3.1 and the /24 means our subnet is 255.255.255.0 (important later). DHCP is turned off because we will manage this with an instance of Ubuntu Server.

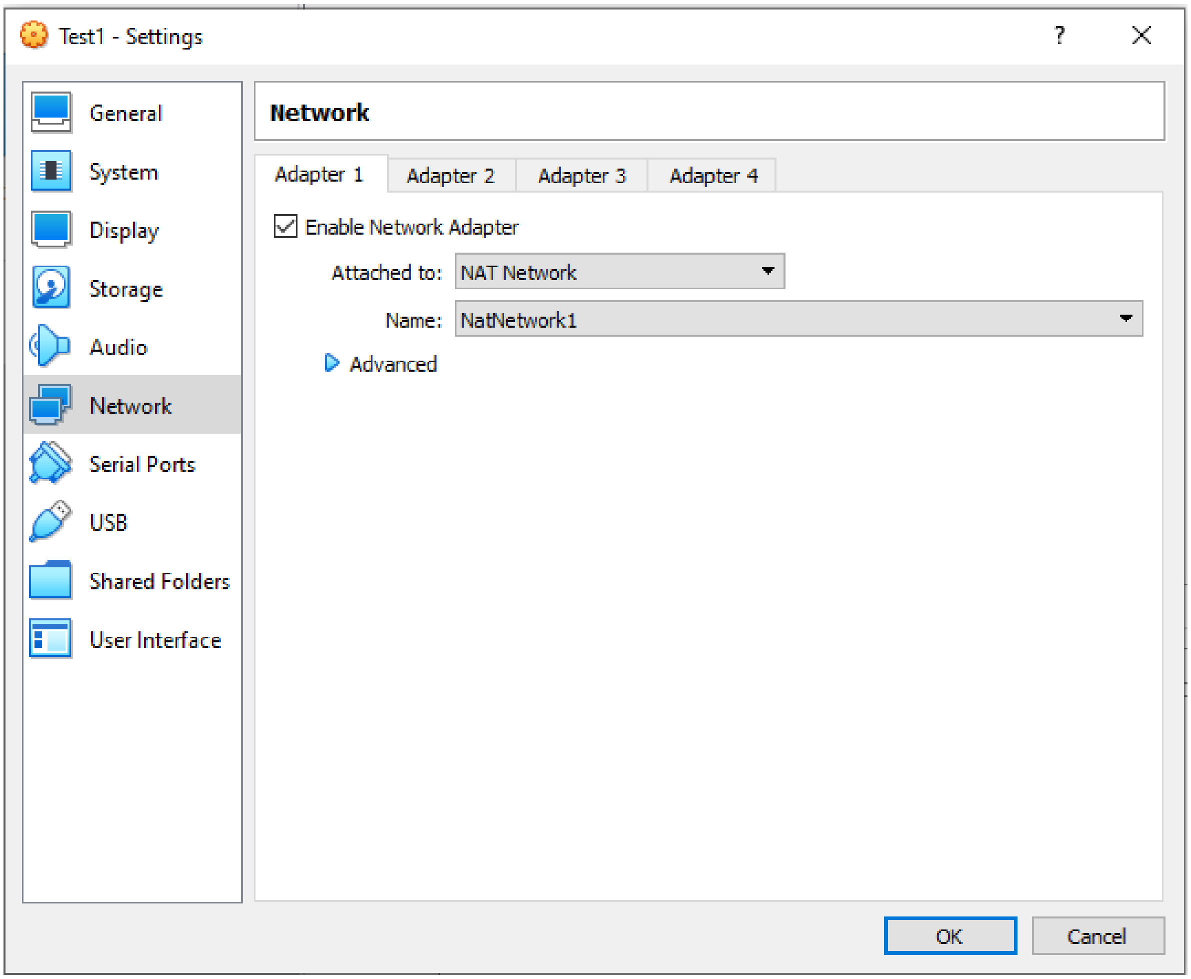
Static IP:

We turned DHCP off on this network. As a result, we need to give our DHCP server a static IP address that is valid on the network. Since the IP of the default gateway is 10.0.3.1, we will make our DHCP server address 10.0.3.2.

To configure the static IP, we will need to make sure that in the settings of our first Ubuntu Server machine (mine is named Test1).

Select the "Network" option on the left pane.

Ensure that you select "NAT network" and the name of the network we just created:



Test1 - Settings

Network

Adapter 1

☑ Enable Network Adapter

Attached to: NAT Network

Name: NatNetwork1

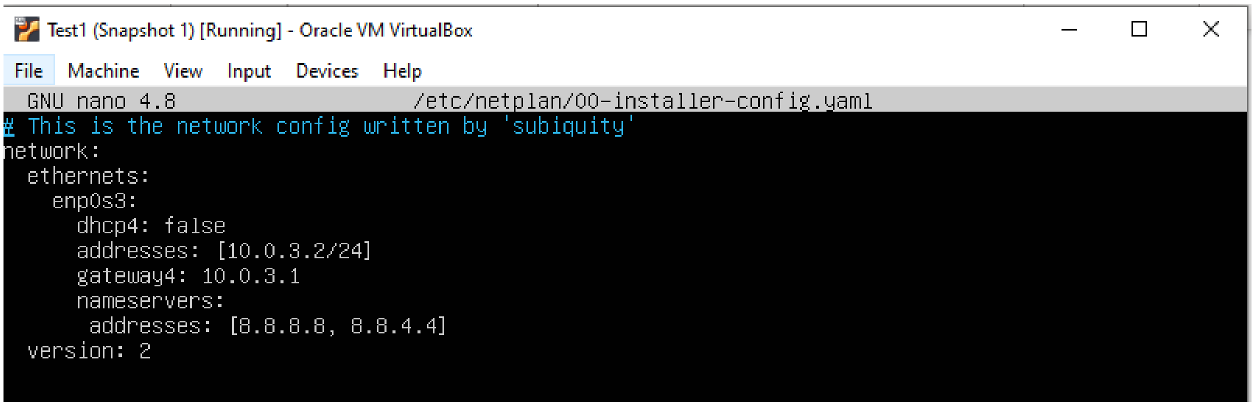
Start your machine, it will take two full minutes because it can't find the network. We will fix that.

Once logged in type:

sudo nano /etc/netplan/00*

You will be prompted for your password.

Edit the file and make it look like this:



```
  GNU nano 4.8                    /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [10.0.3.2/24]
      gateway4: 10.0.3.1
      nameservers:
       addresses: [8.8.8.8, 8.8.4.4]
  version: 2
```

Press ctrl+x and then press "y" and then hit the enter key to save your file.

What we did here was:

a. Told the machine to not expect a DHCP server with "dhcp4: false"
b. Assigned the address of 10.0.3.2 to the machine with the first "addresses" line
c. Tell the machine to use 10.0.3.1 as our default gateway (our virtual router)
d. Assign Google's free DNS servers to use for DNS queries in the addresses under "nameservers"

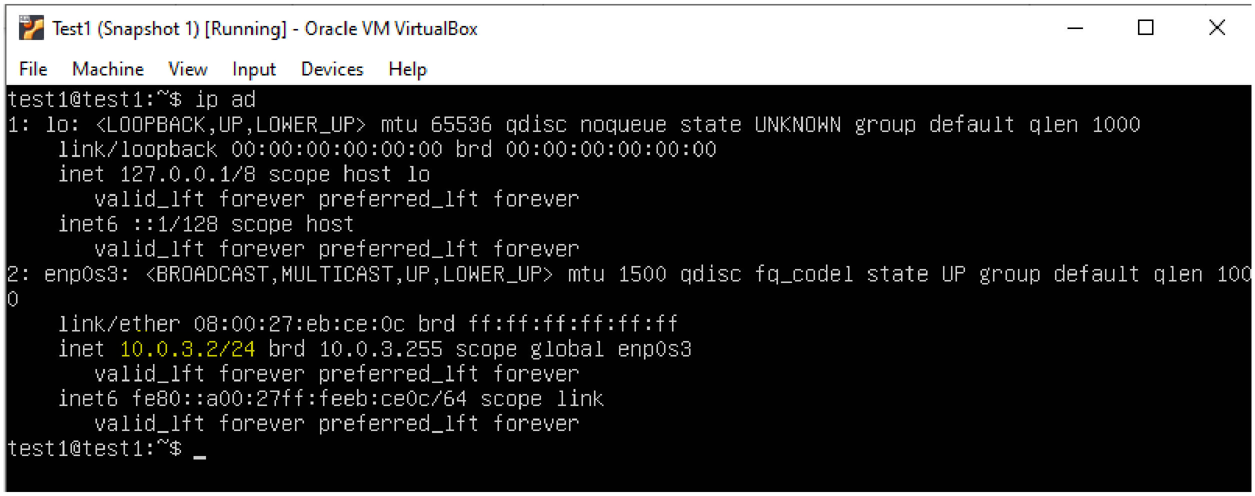
Enter the command:

sudo netplan apply (if you have errors your netplan .yaml file from above has a typo.

Enter the command:

ip ad

Your results should look like this:


Test1 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
test1@test1:~$ ip ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:eb:ce:0c brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.2/24 brd 10.0.3.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feeb:ce0c/64 scope link
       valid_lft forever preferred_lft forever
test1@test1:~$ _
```

We now have the static address of 10.0.3.2 assigned to our machine. We also provided a default gateway and DNS servers, so you should have full access to the internet.

Note: See the name of the adapter, “enp0s3”? We will need this in the next step.

Test with:

sudo apt update

If this works, you did it correctly.

We will need to remember to exclude this address and the default gateway from our DHCP scope in the next step.

DHCP

We will need to install a package for DHCP (Still in our Ubuntu Server machine with the static address):

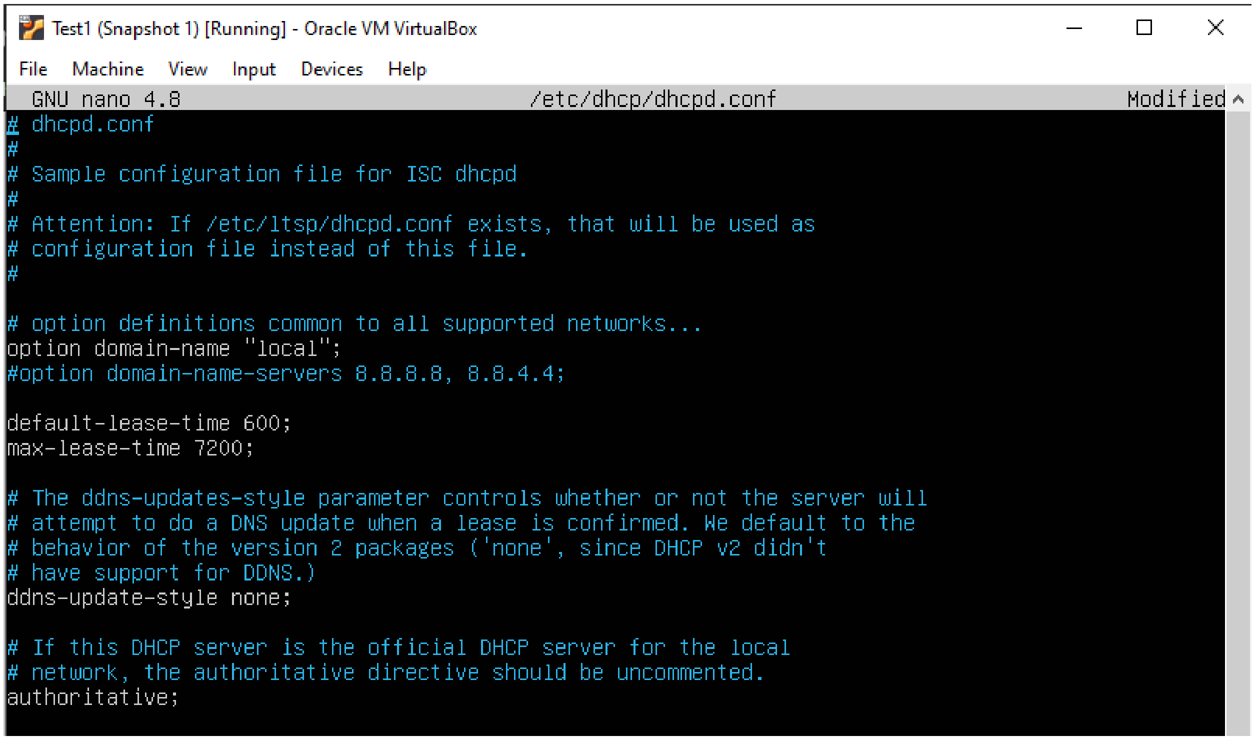
sudo apt install isc-dhcp-server -y

The goal here is to make our dhcpd.conf file automatically assign any computer that connects to the network with an IP address in our network’s range in the scope that we tell it to select from. This is what makes DHCP “dynamic”.

To edit your .conf file enter the command:

sudo nano /etc/dhcp/dhcp.conf

Your top portion of your file should look like this:

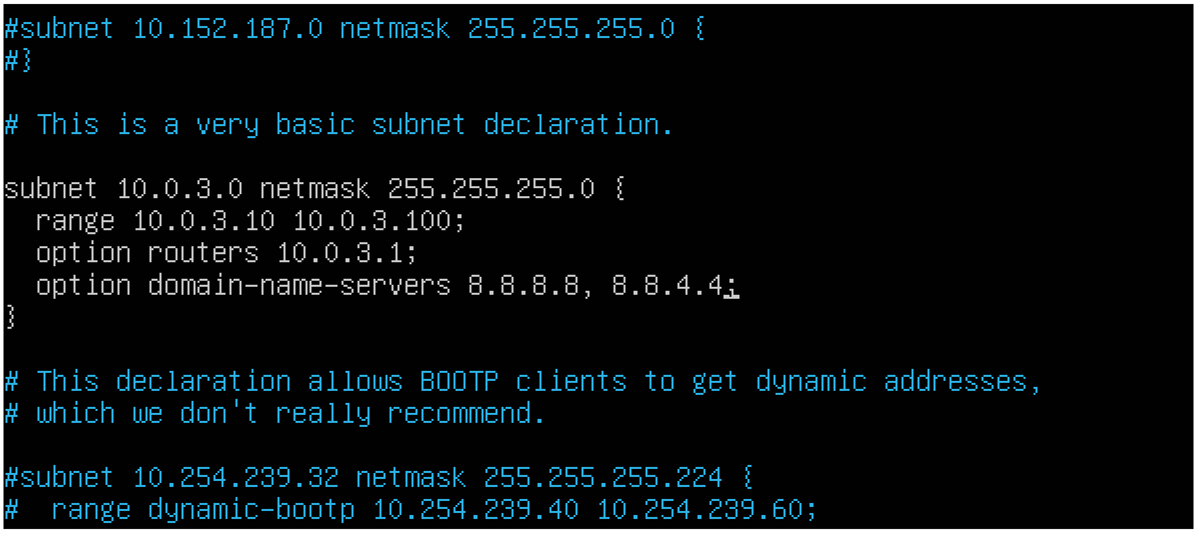


Test1 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
  GNU nano 4.8                         /etc/dhcp/dhcpd.conf                         Modified
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#

# option definitions common to all supported networks...
option domain-name "local";
#option domain-name-servers 8.8.8.8, 8.8.4.4;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
```

The domain-name was changed to “local” and we told the server to be the authoritative DHCP server on the network (the main one).

Next, modify the DHCP topology further down in the file as follows:



```
#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

subnet 10.0.3.0 netmask 255.255.255.0 {
  range 10.0.3.10 10.0.3.100;
  option routers 10.0.3.1;
  option domain-name-servers 8.8.8.8, 8.8.4.4;
}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
```

(make sure to close the squiggly brackets)

Press ctrl+x, then “Y”, then hit the Enter key to save this file.

What we did here was provide the network IP and subnet mask (10.0.3.0 255.255.255.0, which means the same as 10.0.3.0/24)

Next, we assigned our range on the network from .10 to 100, giving 101 available addresses to choose from for the network.

Finally, we assigned Google’s DNS servers to every machine that connects to the network and makes a request to the DHCP server.

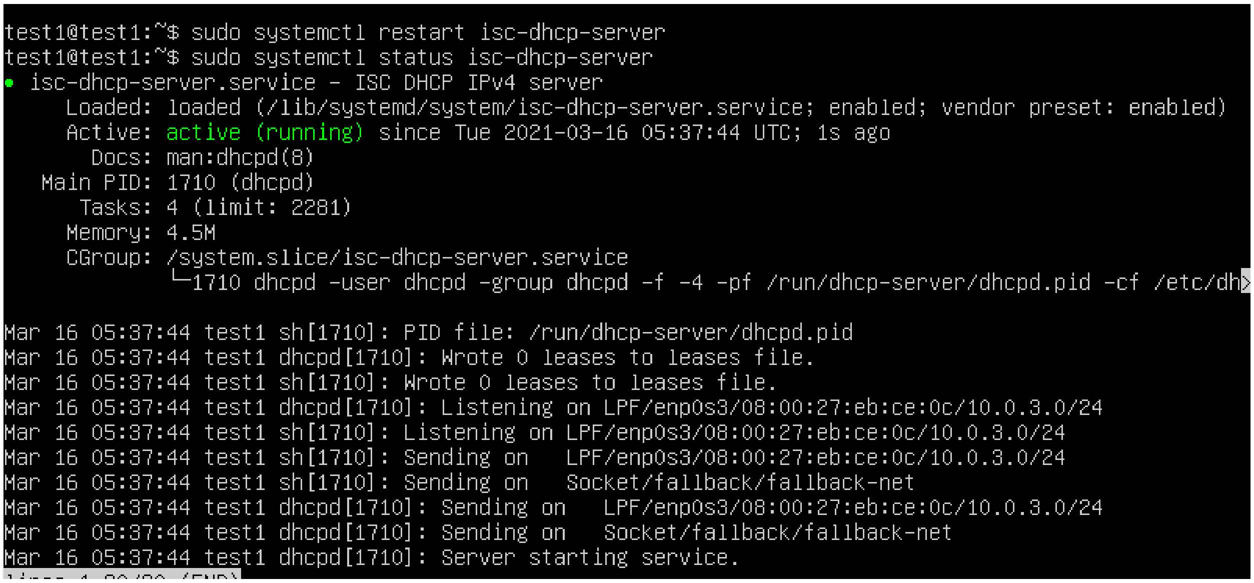
Type the command:

sudo systemctl restart isc-dhcp-server

Then:

sudo systemctl status isc-dhcp-server

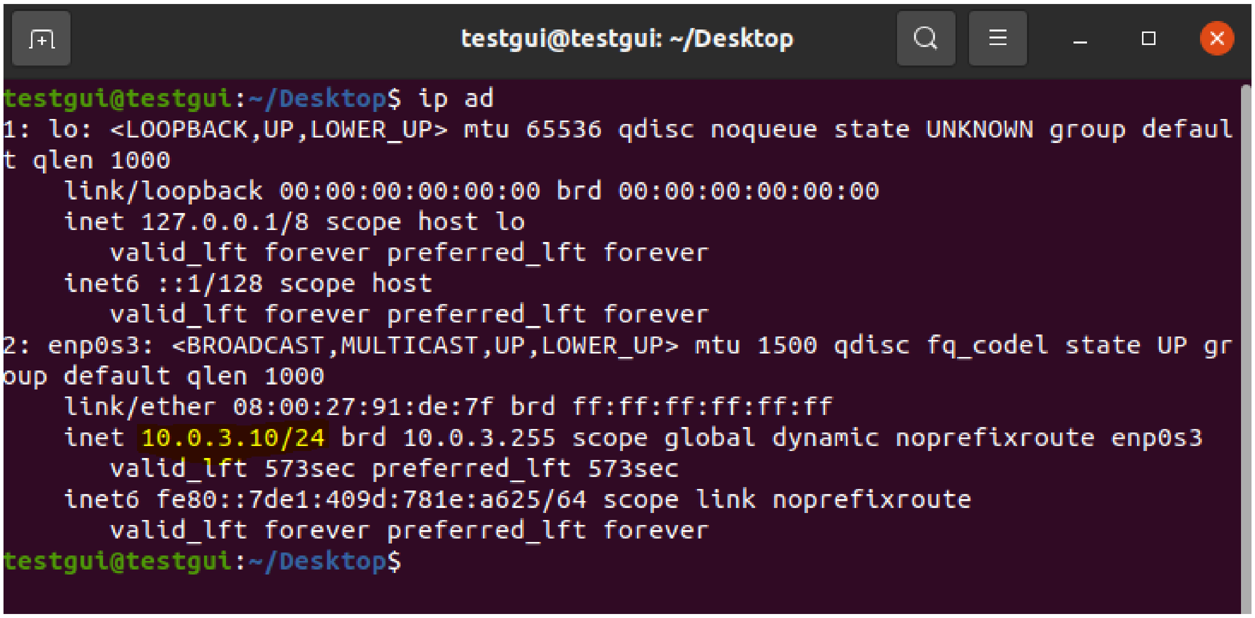
Your status output should look like this:



```
test1@test1:~$ sudo systemctl restart isc-dhcp-server
test1@test1:~$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2021-03-16 05:37:44 UTC; 1s ago
       Docs: man:dhcpd(8)
   Main PID: 1710 (dhcpd)
      Tasks: 4 (limit: 2281)
     Memory: 4.5M
     CGroup: /system.slice/isc-dhcp-server.service
             └─1710 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dh>

Mar 16 05:37:44 test1 sh[1710]: PID file: /run/dhcp-server/dhcpd.pid
Mar 16 05:37:44 test1 dhcpd[1710]: Wrote 0 leases to leases file.
Mar 16 05:37:44 test1 sh[1710]: Wrote 0 leases to leases file.
Mar 16 05:37:44 test1 dhcpd[1710]: Listening on LPF/enp0s3/08:00:27:eb:ce:0c/10.0.3.0/24
Mar 16 05:37:44 test1 sh[1710]: Listening on LPF/enp0s3/08:00:27:eb:ce:0c/10.0.3.0/24
Mar 16 05:37:44 test1 sh[1710]: Sending on   LPF/enp0s3/08:00:27:eb:ce:0c/10.0.3.0/24
Mar 16 05:37:44 test1 sh[1710]: Sending on   Socket/fallback/fallback-net
Mar 16 05:37:44 test1 dhcpd[1710]: Sending on   LPF/enp0s3/08:00:27:eb:ce:0c/10.0.3.0/24
Mar 16 05:37:44 test1 dhcpd[1710]: Sending on   Socket/fallback/fallback-net
Mar 16 05:37:44 test1 dhcpd[1710]: Server starting service.
```

If it says “failed” you made a typo in your dhcpd.conf file.

Moment of truth! Edit the network on your Ubuntu Desktop machine to be on the same NAT network as your DHCP server and start it up. We should pull the first IP address on the scope we provided:



```
testgui@testgui:~/Desktop$ ip ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:91:de:7f brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.10/24 brd 10.0.3.255 scope global dynamic noprefixroute enp0s3
       valid_lft 573sec preferred_lft 573sec
    inet6 fe80::7de1:409d:781e:a625/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
testgui@testgui:~/Desktop$
```

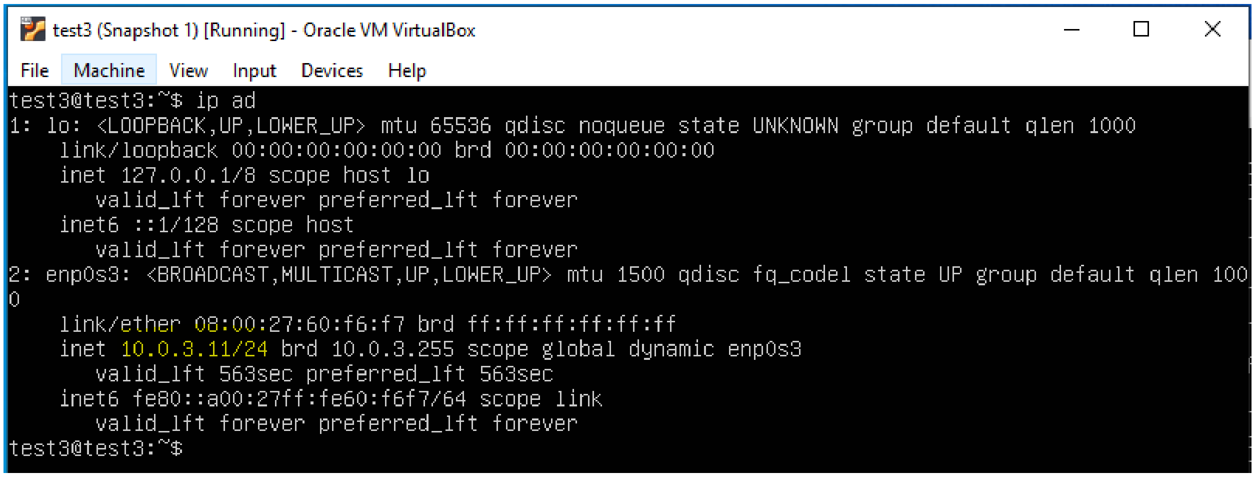
If not, something is wrong. Are you on the right NAT Network? Is your DHCP Server status active (running) on the other machine?

SAMBA (SMB or Server Message Block)

Samba allows computers to share files with one another and should be accessible by Windows networks (it isn't when we do it this way, see the links at the top of the file to figure out how to though). Also, you would need to be on the same network as your host computer to share files between devices using SAMBA.

To begin, configure your last server to connect to the same NAT network as the other machines and start it up.

You should pull the next available IP address in your scope:



```
test3@test3:~$ ip ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:60:f6:f7 brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.11/24 brd 10.0.3.255 scope global dynamic enp0s3
       valid_lft 563sec preferred_lft 563sec
    inet6 fe80::a00:27ff:fe60:f6f7/64 scope link
       valid_lft forever preferred_lft forever
test3@test3:~$
```

10.0.3.11, good.

Next run the following commands:

sudo apt update -y

sudo apt install samba -y

Once the install finishes, we need to create a directory to share to the network.

Enter the commands

cd ~

(The squiggly symbol is the tilda key, it's right above the left tab)

The makes sure you are in your home directory.

Then:

mkdir NetworkShare

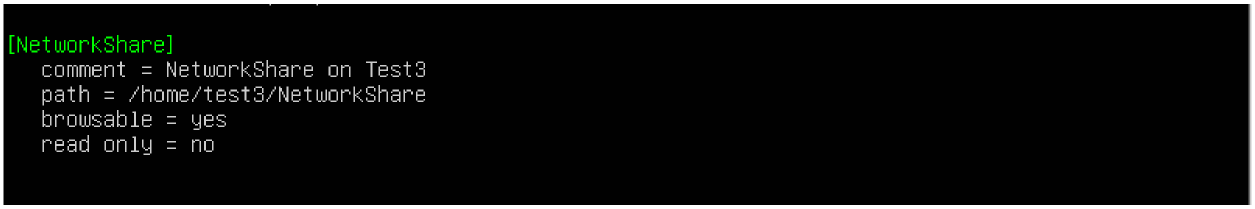
This creates a directory in our home folder at /home/Test3/NetworkShare.

Now we need to edit the Samba configuration file.

Type:

sudo nano /etc/samba/smb.conf

scroll all the way down to the bottom of this and insert the following:



```
[NetworkShare]
   comment = NetworkShare on Test3
   path = /home/test3/NetworkShare
   browsable = yes
   read only = no
```

Press ctrl+x then “Y” then hit the Enter key to save this.

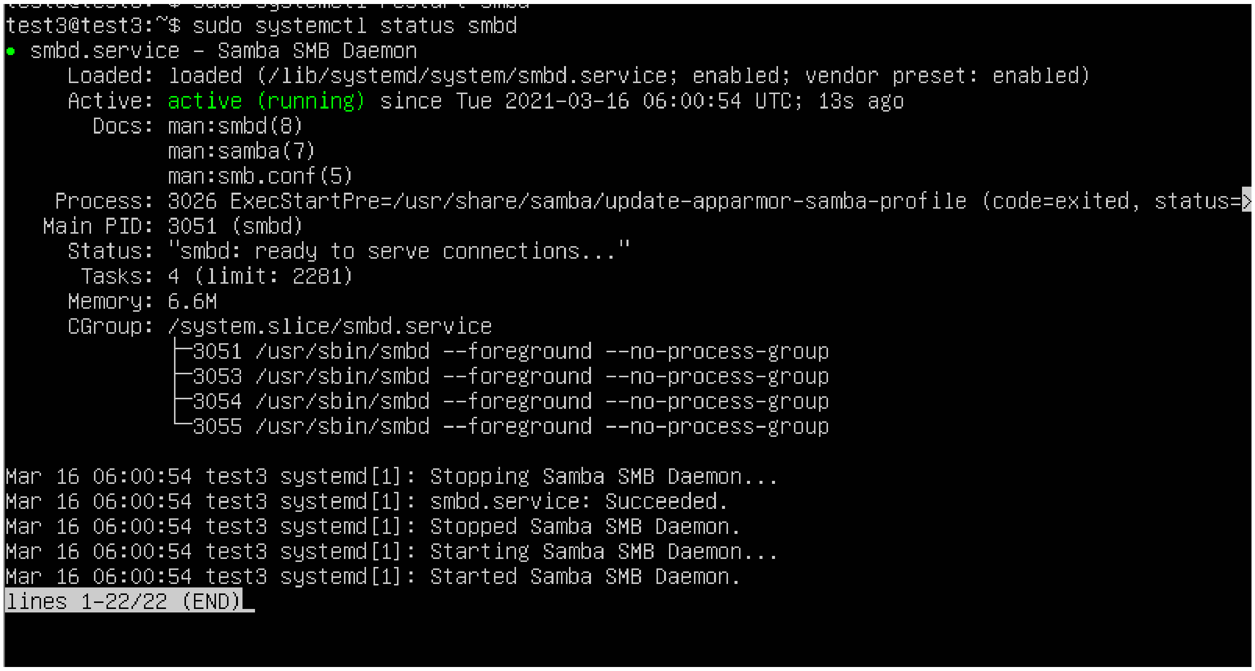
This provides the SAMBA service with the path to our shared folder and some details about access.

Now enter the commands:

sudo systemctl enable smbd

sudo systemctl restart smbd

sudo systemctl status smbd



```
test3@test3:~$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
     Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2021-03-16 06:00:54 UTC; 13s ago
       Docs: man:smbd(8)
             man:samba(7)
             man:smb.conf(5)
    Process: 3026 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=>
   Main PID: 3051 (smbd)
     Status: "smbd: ready to serve connections..."
      Tasks: 4 (limit: 2281)
     Memory: 6.6M
     CGroup: /system.slice/smbd.service
             ├─3051 /usr/sbin/smbd --foreground --no-process-group
             ├─3053 /usr/sbin/smbd --foreground --no-process-group
             ├─3054 /usr/sbin/smbd --foreground --no-process-group
             └─3055 /usr/sbin/smbd --foreground --no-process-group

Mar 16 06:00:54 test3 systemd[1]: Stopping Samba SMB Daemon...
Mar 16 06:00:54 test3 systemd[1]: smbd.service: Succeeded.
Mar 16 06:00:54 test3 systemd[1]: Stopped Samba SMB Daemon.
Mar 16 06:00:54 test3 systemd[1]: Starting Samba SMB Daemon...
Mar 16 06:00:54 test3 systemd[1]: Started Samba SMB Daemon.
lines 1-22/22 (END)
```

If this isn’t running, there is an error in your smb.conf file.

Now we need to create a password for our user account on the SAMBA service.

Enter the command:

sudo smbpasswd -a test3

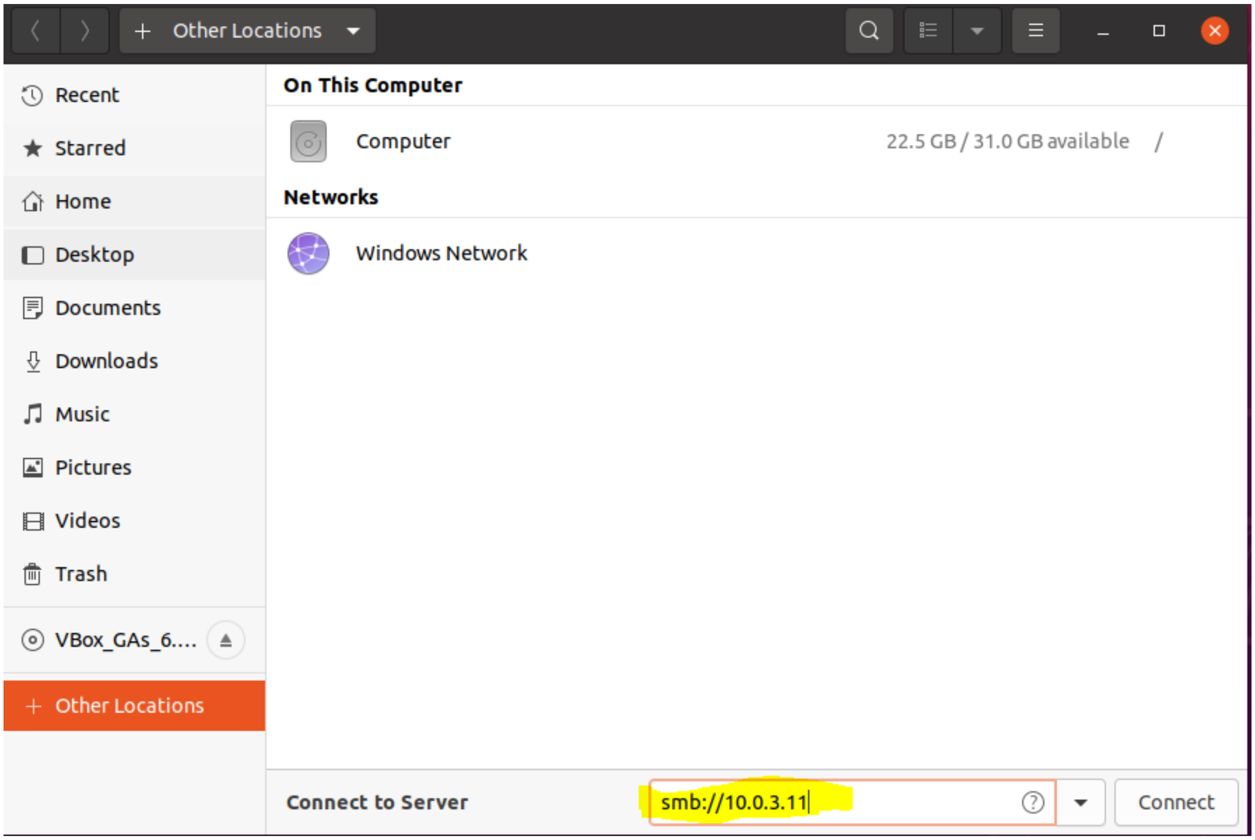
Note: In my example, “test3” is the name of my user on the computer. You need to use the name of a user on the computer for this to work.

Go ahead and pick a password.

To test this out, bring back up your Ubuntu Desktop or Linux with a GUI. Open your file browser (folder on left quick launch pane).

Next, select the “Other Locations” option at the bottom of the left pane.

Type the following to get access to your shared drive in the space:



Connect to Server: smb://10.0.3.11

Now hit “Connect”

Double click the “Network Share” folder.

You will be prompted for credentials. Select “Registered user” and type in your username and and the password you selected in the last step. If this doesn’t work, you may have made a typo in the smb.conf or in the name of your shared network folder.

You can now add files to this folder and access them from any other machine on the network.

To access the shared folder from your DHCP server we will need a package.

Enter the command:

sudo apt install smbclient -y

Quickly create a file in your home directory to transfer to the shared drive:

echo “hello samba” > stuff.txt

Now we can connect to the share with the command:

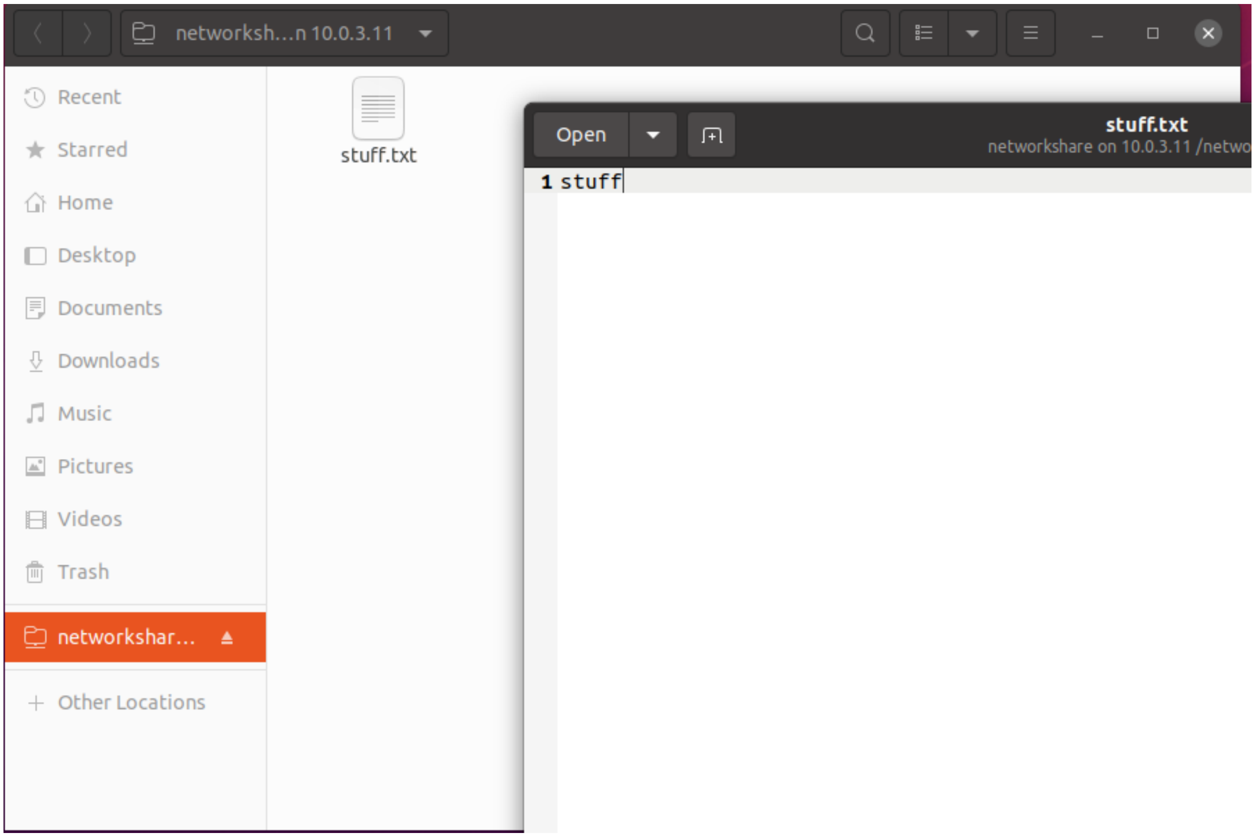
smbclient \\\\10.0.3.11\\NetworkShare –user=test3

This prompt us for the password and then connect us.

To transfer your file:

put stuff.txt

Once this finishes, go back to your Ubuntu Desktop and look in the file share:



```
stuff
```

Success!