README:

The following readme documents will detail how to set up the DNS server, DHCP server, NFS server, FTP server and a Router.

**DNS Server**

This tutorial will give you the basis on how to create a DNS server with a forwarder and reverse lookup

Firstly, you should run the IP address to see the IP address of your virtual machine. Your IP address will be used in place of the sample IP address in the images below



Next, run the command “**sudo apt install bind9**” This will install the packages that will help you set up the DNS server

After installation, check that it is installed by running:

**named -v**

run cd /etc/bind to enter your directory, run **ls** to see all the files in your directory, it should look like this

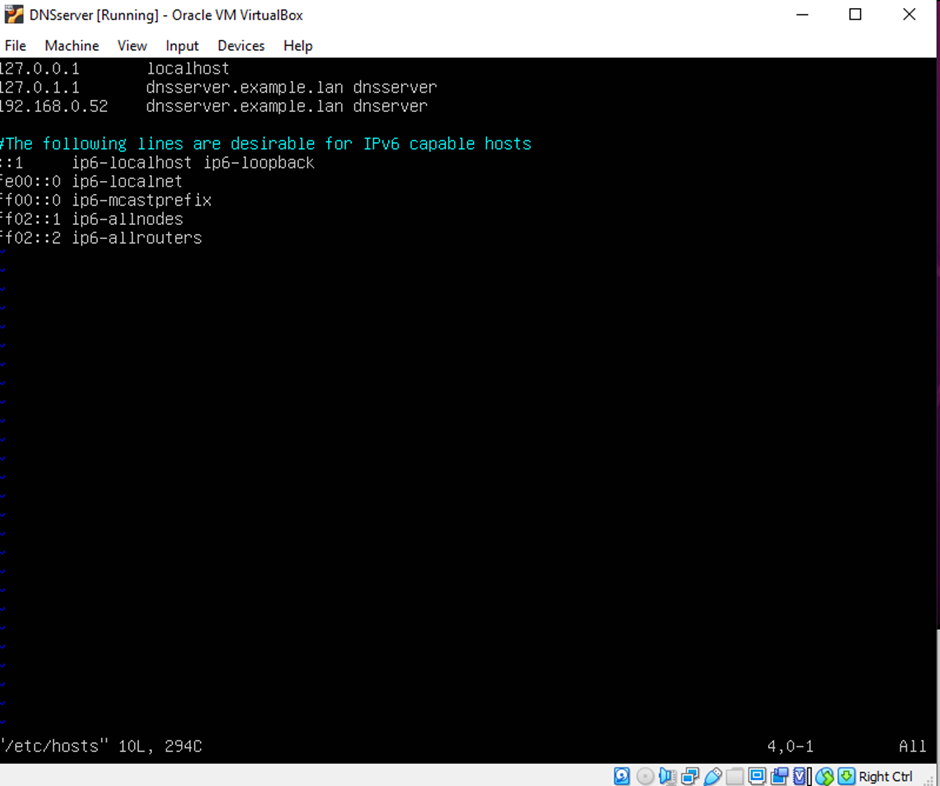
Text

Description automatically generated

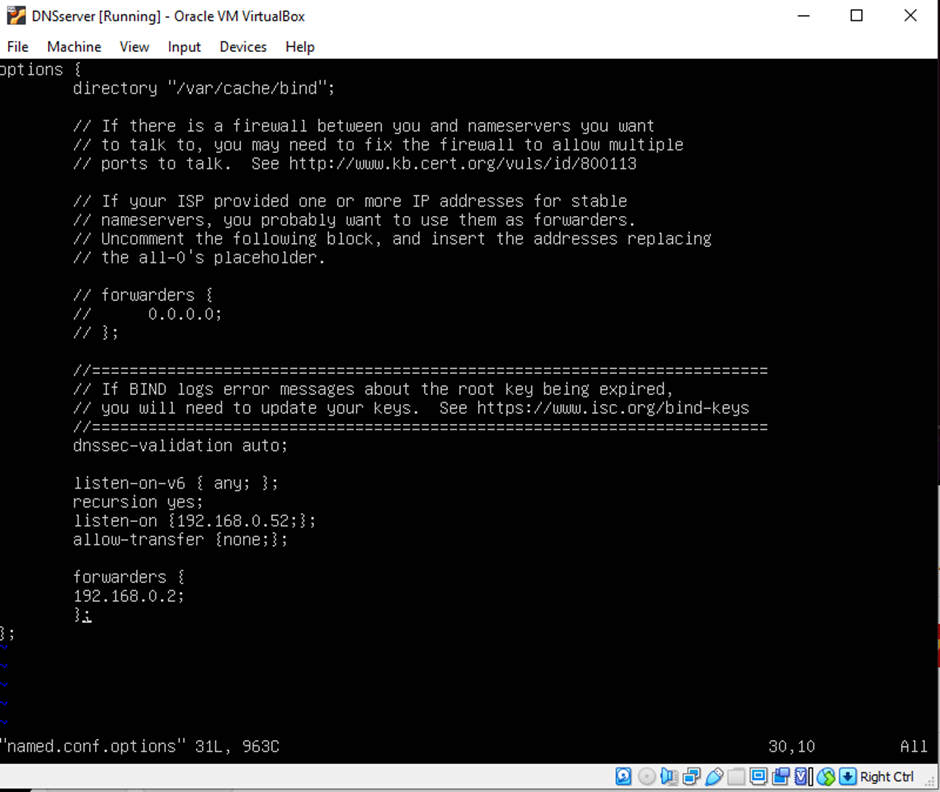
Run the command **Sudo vim /etc/hosts.** Insert the following into the file

Run the command “hostname” to check your hostname and run “hostname –fqdn” to check your dns server name.

Then run the command “**sudo cp named.conf.options named.conf.options.orig”**



Save and exit the file, now Run the command **“sudo vim named.conf.options”** and enter the following into the file. This will set the configurations for the DNS server (make sure to replace the IP address in the image with your VM’s IP address.)



After entering the details, run the command **“named-checkconf”** to check for any errors

Exit the file. Now, write the command **“sudo cp named.conf.local named.conf.local.orig”** to make a copy of the file

then enter “**sudo nano named.conf.local**”, in this file, the forward lookup zone and reverse lookup zone will be created .now enter the following that is shown in the image below:

Text

Description automatically generated

(Make sure to enter your machines IP address rather than the one shown in the images. rather than the one shown in the images.)

After entering the details, run the command **“named-checkconf”** to check for any errors

To get started on creating a records file, you’re going to copy the db.local file into a new file called db.example.lan by entering the command **“sudo cp db.local db.example.lan”**

Now run the command **“sudo nano db.example.lan”** and enter the following into the file.

Text

Description automatically generated

After editing the file, save and exit the file. enter the command “**named-checkzone example.lan db.example.lan**” the response to this command should be an “OK” if the syntax is correct.

Now you must copy the file “db.127” using the command **sudo cp db.127 db.<0.(the first two numbers of your IP address reversed )>**

Then write the command **sudo nano** **db.<0.(the first two numbers of your IP address reversed )>** and enter the following into the file

Text

Description automatically generated

The following details in this file are pointer records for each domain name as seen in the image of the file

Save and exit the file, now enter the command **named-checkzone 0.168.192.in-addr.arpa db.0.168.192**

No run the command **sudo service bind9 restart** to restart the DNS server. Then run **sudo service bind9 status** to boot it up and check its status. The following should show up if the server is working as it should

Text

Description automatically generated

Testing:

To test that your DNS server is setup, alongside the forwarder and reverse lookup

Enter the command “**Dig @<dns ip> example.lan”** to test the forward lookup. It should look like this.

Text

Description automatically generated

Enter the command **“Dig@ -x <vm ip address>”** to test the reverse lookup. The command should have this output

Text

Description automatically generated

**DHCP Server**

This section will show you how to set up a DHCP server that will lease ip addresses between 192.168.1.150 and 192.168.1.200

First, you’re going to make a clone of the base(DHCP server) virtual machine. Now boot up the server vm and run this command

**“sudo apt install isc-dhcp-server”**

Now run this command to check that the server is running

**systemctl status isc-dhcp-server**

run the command **“ifconfig”** to see your network interface

run **ip link** to see the existing interface

To tell the DHCP server to listen on that interface, run:

**sudo nano /etc/default/isc-dhcp-server**

You should see a line that looks like:

INTERFACESv4=""

Make sure you enter the name of your interface that you found using the ip link command

Now, you have to tell the server what IP addresses it should give a client. Run this command:

sudo nano /etc/dhcp/dhcpd.conf

And enter the following

**subnet 192.168.1.0 netmask 255.255.255.0 {  
 range** **192.168.1.150 192.168.1.200;  
 option subnet-mask 255.255.255.0;  
 option routers 192.168.1.1;  
 option broadcast-address 192.168.1.200;  
 default-lease-time 600;  
 max-lease-time 30000;  
}**

it should look like this

A screenshot of a computer

Description automatically generated

now save and exit the file

this code will tell the server that we want to lease ip addresses between 192.168.1.150 and 192.168.1.200 it also sets the default and max lease times to be 600 and 30000 seconds.

Now, Run **sudo nano /etc/netplan/99\_config.yaml** to set up the network interface  
And enter the following:  
**network:  
 version: 2  
 renderer: networkd  
 ethernets:  
 <interface name>:  
 addresses:  
 - 192.168.1.150/24  
 gateway4: 192.168.1.1**

Note: make sure to replace “interface name” with the interface name you entered on the isc-dhcp-server file

Now, run the command **sudo netplan apply** to apply the changes

Now restart the server by running the following command

**sudo systemctl restart isc-dhcp-server**

Now hop on the clone(client) vm and firstly, run the “**IP link”** command to see the interface name on the vm. Then enter the command “**sudo nano /etc/netplan/99\_config.yaml”** . Add the following code to the file

**network:**

**version: 2**

**renderer: networkd**

**ethernets:**

**<interface name>:**

**dhcp4: true**

make sure you save and exit. Now enter the command **“sudo netplan apply”**

to test whether the dhcp server is working or not, run the command **“dhcp-lease-list”** the list should look like this

Text

Description automatically generated

**NFS Server**

Firstly, make sure to create both a server(base) vm and a client(clone). Firstly, hop on the server vm and run the following commands:

**sudo apt install nfs-kernel-server**

**systemctl status nfs-kernel-server**

after this, make a directory called “shared” by running the command **“mkdir shared”.** Make sure you a file in this directory so that the nfs server can be tested, e.g. hello.txt

Now run these commands to set permissions that allow NFS to proceed

**sudo chown -R nobody:nogroup shared/**

**sudo chmod 777 shared/**

now you will edit the exports file so that the server will export the nfs\_dir by entering the following: /home/<username>/shared <client IP address>(rw,async,no\_subtree\_check)

next, you must run these commands to update the configuration, and restart the server so the change is applied

**sudo exportfs -a**

**sudo systemctl restart nfs-kernel-server**

next, you must run the following commands to make the firewall allow NFS connections

**sudo ufw allow from <client IP address> to any port nfs**

**sudo ufw enable**

now run this command to check if the rules have been added successfully

**sudo ufw status**

Move onto the client vm and run the following command to install the NFS client software

**sudo apt install nfs-common**

now, you will make a directory that will inherit the contents of the NFS directory on the server, run the command

**mkdir nfs\_dir**

Now, to test that the set up was successful, run this command to mount the folder of the nfs server to the directory on the client

**sudo mount <server IP address>:/home/<username>/shared nfs\_dir**

if the command had no errors, you should see the same file that was in the shared directory of the NFS server VM in the nfs\_dir directory on the client. It should look like this

**FTP server**

This section will show you how to set up a FTP server which will allow file transfer.

Firstly, you must install the FTP server by running the following command

**sudo apt install vsftpd**

And check that it is running using systemctl:

**systemctl status vsftpd**

now, you must enter the directory /etc/vsftpd.conf by running the command

**sudo nano /etc/vsftpd.conf**

First, edit the following line anonymous\_enable=NO, by changing the no to yes and add the following lines:

**pasv\_enable=YES**

**pasv\_min\_port=10000**

**pasv\_max\_port=10010**

this lets the server know that its minimum port is 10000 and its max port is 10010

now restart the server with this command:

**sudo systemctl restart vsftpd**

now enter the following command to allow file transfers

**sudo ufw allow 21/tcp**

**sudo ufw allow 10000:10010/tcp**

now you will hop on the client vm and run this command to get the file

**wget ftp://server IP address/my\_file.txt**

the response should look like this

Text

Description automatically generated

Router lab

This section will show you how to make your vm act as a router for a clone vm that will be a client

You will first open this file (etc/sysctl.conf) and uncomment the following line

**net.ipv4.ip\_forward=1**

now exit the file and run the following command to tell the server to forward packets from enp0s8 to enp0s3

**sudo iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT**

now run this command to tell it to forward from enp0s3 to enp0s8:

**sudo iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state RELATED,ESTABLISHED -j ACCEPT**

Now, run this command to allow network translations

**sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE**

you need to install the following package to save the network rules that you set up. run this command

**sudo apt iptables-persistent**.

Run the **iptables-save** command to save your settings

Now, run this command

**sudo bash -c “iptables-save > /etc/iptables/rules.v4”**

this command will let the firewall allow NAT packets

now enter the **file /etc/default/ufw,**

you must edit the file to change the firewall policy. You will see theFirst, change the following line below

**DEFAULT\_FORWARD\_POLICY="DROP"**

And change the string "DROP" to "ACCEPT".

open the file **/etc/ufw/sysctl.conf**,

and add the following line:

**net/ipv4/ip\_forward=1**

Like the similar setting in /etc/sysctl.conf above, you might find that this line already exists

in the file, but is commented out.

Save and exit this file, now enter the following command to restart the firewall:

**sudo ufw disable**

**sudo ufw enable**

save the changes by running this command:

**sudo netfilter-persistent save**

now hop onto the client(clone) vm

First, enter the /etc/netplan/00-installer-config.yaml and comment out the entire file.

The setup of the 99\_config.yaml file should be substantially changed. Below is the entire

contents of the file:

**network:**

**version: 2**

**renderer: networkd**

**ethernets:**

**enp0s8:**

**dhcp4: true**

**gateway4: <vm ip address>**

**nameservers:**

**addresses:**

**- <vm ip address>**

Now run the command “**sudo netplan apply”** to save the changes.

To test the router, turn off the network connection on the client and turn the clone vm back on. On the clone, try to ping a website, e.g. google.com to test if the client is connected to the internet through the router vm.