Setting up a DNS server

First you need to install a DNS server which in this case will be bind9 to do that enter the following command

**sudo apt install bind9**

Then check that it is running by entering the following command

**named -v**

this should output the version number of named that was installed

next check if the daemon is running by running the following command

**systemctl status named**

if it isn’t active enter the following command to get it started

**sudo systemctl start named**

the database that DNS servers use are called zone files; they consist of entries which are called resource records. Resource records contain the domains and map them to the IP address

Now you need to create a zone file you can call it db.example.com, to do this enter the following command

**sudo nano /etc/bind/db.example.com**

then inside the file make sure the content is the same as what’s in the following picture

Text

Description automatically generated

Then save and close the file and run the following command to make sure the syntax is correct

**named-checkzone example.lan. /etc/bind/db.example.com**

This command makes sure that the contents of the file constitute a valid zone file for   
the FQDN example.lan. by checking for any typos.

If the syntax is right it should output OK

Now to make the reverse lookup works we need to create a file in /etc/bind called rev.40.30.20.10.in-addr.arpa to do this enter the following command

sudo nano /etc/bind called rev.40.30.20.10.in-addr.arpa

then inside the file make sure its contents match the contents in the following image

Text

Description automatically generated

Then save and close the file and run the following command to make sure the syntax is correct

**named-checkzone example.lan. /etc/bind/db.example.com**

This command makes sure that the contents of the file constitute a valid zone file for   
the FQDN example.lan. by checking for any typos.

If the syntax is right, it should output OK

After making these new zones they need to be added to the named’s config file to ensure that it knows about the new zonefiles.

To do this open the named.conf file by entering the following command

**sudo nano /etc/bind/named.conf**

then ensure that its contents match the contents in the following image

Text

Description automatically generated

After that is done save and exit the file

Next we need to ensure that there were no typos made in the named.conf file to do this enter the following command

**named-checkconf /etc/bind/named.conf**

if there was no issues it should exit silently

Next we need to restart named to make it read the updated files. To do this enter the following command

**sudo systemctl restart named**

We should now be able to dig example.lan and also the reverse dig through the local host

Dig @127.0.0.1 example.lan

Dig @127.0.0.1 -x 10.20.30.41

DHCP server

Set up

Since the VM is set up with bridged networking it should have access to the DHCP server of TU Dublin. Therefore, we need to create a new VM and network it to the existing VM.

To do this you need to need to add a new network adaptor to the VM and clone it.

To add the new network adaptor you need to shutdown the VM and then go into its settings. Once you are in the settings go into the network tab then adapter 2.

Once you are there enable network adaptor box and set Attached to internal network.

After that click advanced and make sure Cable Connected is checked then save the settings.

Now clone this VM and for the clone’s settings make sure you set the MAC Address policy to generate a new MAC address for all network adaptors. Then clone it and now you are done with the set up

Now go into the server VM (the original VM not the clone)

First we need to Install a DHCP server in this case we will be using isc-dhcp-server

Install it using the following command

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

Then make sure its running using the following command

**systemctl status install isc-dhcp-server**

if its not running start it using the following command

**sudo systemctl start isc-dhcp-server**

If you run the command **ip link** you should see an existing network interface which you don’t see when you run the command **ifconfig.** That’s the network interface which we want.

In this case that network interface is enp0s8

Now we need to tell the DHCP server to listen on the enp0s8 interface to do that go into the /etc/default/isc-dhcp-server by entering the following command

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

then in there you will see a line that looks like INTERFACESV4=””

put the network interface you found earlier using ip link in between the quotation’s marks, should look like the following

Text

Description automatically generated

Next we need to tell the server what the range of IP addresses it should lease to the clients, to do this we need to go to the dhcpd.conf file. To get there enter the following command

sudo nano /etc/dhcp/dhcpd.conf

then make sure its content includes the lines which are uncommented in the following image

Text

Description automatically generated

Then save and exit. This lets tells the server we want to lease the addresses in the range of 192.168.1.150 and 192.168.1.200 with a subnet mask of 255.255.255.0 and a gateway of 192.168.1.1. It also gives those IPs a broadcast address of 192.168.1.255. The default lease time is set to 600 seconds and the max lease time is 30000 seconds.

Next we need to set up the network interface so the server can use it. To do this you need to enter the 99\_config.yaml file which you can enter by entering the following command

**sudo nano /etc/netplan/99\_config.yaml**

Once you are in the file make sure it contains the content inside following the image

Text

Description automatically generated

This gives the host vm a static ip of 192.168.1.10

Now save and exit the file

To make the server load the new network settings enter the following command

**sudo netplan apply**

and restart the DHCP server as well by entering the following command

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

Then enter the following command to make sure the server is running

**sudo systemctl status isc-dhcp-server**

Now on the clone VM the 99\_config.yaml file also needs to be configured but before you do that you need to find the network interface name again so just like before run the command **ip link** you should see an existing network interface which you don’t see when you run the command **ifconfig.** That’s the network interface which you want. It will most likely be the same as in the other VM

In this case it is enp0s8. Now go into the 99\_config.yaml file by entering the following command

sudo nano /etc/netplan/99\_config.yaml

then make sure its contents are the same as in the following image

Text

Description automatically generated

This says we want the IP address to be assigned using DHCP rather than just using a static ip address

Now run the following command to apply the changes

**sudo netplan apply**

Now if you restart the clone and run **dhcp-lease-list** in the original VM you should see the clone’s DHCP transaction with the server and you should be able to ping the two VMs from each other.

NFS Server

We need to start by installing and configuring the server.

To install the server enter the following command

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

and then make sure it is running using the following command

systemctl status nfs-kernel-server

if it is not running start it using the following command

sudo systemctl start nfs-kernl-server

next we create the directory on the server which we want to share, in this case it will be called shared. To make it enter the following command

sudo mkdir shared

NFS requires the shared directory to have liberal permissions and the directory can’t be associated with any user account or primary group to do this run the following commands

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

**sudo chmod 777 shared/**

Now we need to open the NFS config file to tell the server to share our directory. To open the config file run the following command

sudo nano /etc/exports

next enter in the following line into the file but replace <username> and <client ip address> with the values for your setup

**/home/<username>/shared <client IP address>(rw,async,no\_subtree\_check)**

Now you need to use the exportfs command to read in the updated configuration and you also need to restart the server for the changes to take effect. To do those enter in the following commands

**sudo exportfs -a**

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

next you need to set up NFS on the client

first we need to install the nfs client software to do that enter in the following line

sudo apt install nfs-common

next a directory that will mount the shared directory from the server onto using NFS needs to be made. In this case it will be called nfs\_dir. To do that run the following command

sudo mkdir nfs\_dir

then mount the shared folder from the server to the nfs\_dir using the following command but replace <username> and <client ip address> with the values for your setup

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

Now the shared directory on the server and the nfs\_dir should contain the same content and any changes made to one should be passed to the other over the network.

FTP

First you need to install the server

FTP daemon is a free FTP server which will be used for this part

To install the FTP daemon use the command

sudo apt install vsftpd

then you need to make sure that its running by using the following command

systemctl status vsftpd

the output should look like something like this

Text

Description automatically generated

Next we need to configure the server anonymous and passive connections. Making the server anonymous allows the user to access the server without an account. Making the FTP in passive mode makes the server tell the client which port to use for the data via commands sent over the data channel.

To do this we need to edit the config file which is in /etc/vsftpd.conf

We need to change the line that says **anonymous\_enable=NO** to **anonymous\_enable=YES**

Then add you need to add:

**pasv\_enable=YES**

**pasv\_min\_port=10000**

**pasv\_max\_port=10010**

it should look like this

Text

Description automatically generated

To make this take effect we need to restart the server to do this enter the following command

**sudo systemctl restart vsftpd**

next we need to make the firewall allow connections on port 21 and ports 1000 to 10010

to do this you need to enter the two following commands

**sudo ufw allow 21/tcp**

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

Text

Description automatically generated

Then you can download files that are in the /srv/ftp directory, to do this use the following command

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

Text

Description automatically generated

Next you can change the directory that has the files that ftp can access, to change it use the following command

**sudo usermod -d /example ftp**

**(the command is sudo usermod -d <insert directory> ftp)**

Router

First you need to enable packet forwarding on the server. To do this open the sysctl.conf file by entering the following line

**sudo nano /etc/sysctl.conf**

Once you’re in the file enter in the following line

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

now we need to tell the server how it should forward the packets. We need to tell the server that it will be forwarding packets from the enp0s8 interface to the enp0s3 interface. To do this run the following command

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

then we need to tell it to forward in the other direction as well so that packets can go from enp0s3 to enp0s8. To do this run the following command

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

next we need to enable network address translation which you can do by running the following command

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

The previous command does not make it persistent, to make the changes persistent we need to use iptables-save. This is an apt package so we need to install it first which you can do using the following command

**sudo apt install iptables-persistent**

then use iptables-save to make the changes persistent using the following command

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

next we need to make sure the firewall allows NAT packets to do this open the ufw file using the following command

sudo nano /etc/default/ufw

it will contain a line which says **DEFAULT\_FORWARD\_POLICY="DROP"** you need to change the **“DROP”** to **“ACCEPT”**

next you need to enter the sysctl.conf file of the firewall by entering the following command

**sudo nano /etc/ufw/sysctl.conf**

then add in the following line to the file

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

then restart the firewall to apply these changes. To do this run the following commands

**sudo ufw disable**

**sudo ufw enable**

then make these changes persistent by running the following command

**sudo netfilter-persistent save**

next we need to set up the client

you need to modify the network configuration files on the client to make it so it routes all traffic through the enp0s8 device.

First thing you need to do is go to the 00-installer-config.yaml by running the following command

**sudo nano /etc/netplan/00-installer-config.yaml**

next comment everything in this file. Then save and exit. After that go to the 99\_config.yaml by running the following command

sudo nano /etc/netplan/99\_config.yaml and make sure its contains the same content as the following image

Graphical user interface, text, application

Description automatically generated

You can change the ip address which comes after via to whatever your server’s ip address is. After that save and exit. Then run the following command

**sudo netplan apply**

then shutdown the client and disable adaptor then start the client again and test if you can ping 8.8.8.8 using the following command

**ping 8.8.8.8**

if the ping is successful that means you successfully set up the server VM as a router