**C19704439**

**Conor Farrell**

**Systems Integration**

**Assignment 2**

**How-To Document**

Contents

[1. DNS Resolver & Lookup 2](#_Toc120710295)

[2. DHCP Server 2](#_Toc120710296)

[3. NFS File Transfer 2](#_Toc120710297)

[4. FTP File Download 2](#_Toc120710298)

[5. Server Router/Gateway 3](#_Toc120710299)

# 1. DNS Resolver & Lookup

To set up youre machine to have the functionality of DNS resolver you need to follow these commands in order:

**sudo apt install bind9**

**named -v**

Checks to see if bind9 is installed

**systemctl status named**

**sudo systemctl start named**

Check to see its running, if it’s not then manually start it

**dig @<” IP”> google.com**

dig followed by your own machines IP and then the query address

Now we want to create a new zone so as when we run the command “dig example.com” it will return a specific IP address that we set.

Create a new zone file in “/etc/bind/” called “db.example.com” and enter the following into the file:

$ORIGIN example.com.  
$TTL 123  
@ IN SOA ns.example.com. admin.example.com. (  
20211013  
2d  
15m  
2w  
1h  
)  
@ IN A 10.20.30.40  
ns IN A 10.20.30.40  
www IN CNAME example.com.  
@ IN NS ns.example.com.

Following this we want to modify named’s config to let it know we’ve made a new zonefile. Run this command “sudo nano /etc/bind/named.conf” and add these lines to the file:

zone "example.com" in {  
type master;  
file "/etc/bind/db.example.com";  
};

# 2. DHCP Server

To properly set up the DHCP server you will need to be working in either a virtual machine environment with a clone or you must have access to another machine on the same network as your host machine.

Run “sudo apt install isc-dhcp-server” to install the DHCP server on your host machine.

To tell the server which interface to listen on run “ifconfig” and “ip link” command and take note of the IP address here that didn’t show up on “ifconfig”. Open the file “sudo nano /etc/default/isc-dhcp-server” and where it says “INTERFACESv4= ‘’ “, add the IP that was in the ip link command.

Next, we have to configure the IP’s the server gives out. Run this command “sudo nano /etc/dhcp/dhcpd.conf” and then add the following:

subnet 192.168.40.0 netmask 255.255.255.0 {  
range 192.168.1.150 192.168.1.200;  
option subnet-mask 255.255.255.0;  
option routers 192.168.40.1;  
option broadcast-address 192.168.40.255;  
default-lease-time 600;  
max-lease-time 30000;  
}

This lets us assign address between 192.168.1.150 and 192.168.1.200

# 3. NFS File Transfer

NFS file servers allow us to sync files between devices. To set one of these up on your host machine run the command “sudo apt install nfs-kernel-server” then make a new directory to work in such as “shared”. To apply permissions so as we can use this run these two commands:

sudo chown -R nobody:nogroup shared/

sudo chmod 777 shared/

This makes it so as the folder we made isn’t associated with any user account or group.

Next, we open the config file for our NFS server at “/etc/exports” and add the following line:

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

Replace username with your host machines username and then run ifconfig on your client to get its IP address which you will replace “client IP” in the above command with.

Run “sudo exportfs -a” and “sudo systemctl restart nfs-kernel-server” to restart the services and put into effect the config changes we just made.

Moving on to the client machine, run “sudo apt install nfs-common” to install the software required for NFS transfer. Next, we make a new directory that we will work in such as “NFS\_DIR”.

Finally, we mount the server folder to this directory with the following command “sudo mount <server IP>:/home/<username>/shared <directory name>”

Any changes made on either machine in this folder should now be synced between the two.

# 4. FTP File Download

FTP file servers allow us to send files anonymously to a client. On your host machine run the command “sudo apt install vsftpd” to install the software needed.

Open the config file found in “/etc/vsftpd.conf”, find the line that contains anonymous\_enable=No and change it to YES. Then add the following lines at the end:

pasv\_enable=YES  
pasv\_min\_port=10000  
pasv\_max\_port=10010

Once complete restart the server using “sudo systemctl restart vsftpd”.

Moving over to your client simply run the command “wget ftp://server\_ip/file\_to\_download” replacing server IP and file to download with the correct values.

# 5. Server Router/Gateway

Now we will make our host machine act as a router for any client that connects. This allows the clients to connect to the internet through the server.

We’ll need to enable packet forwarding first so on the host machine open the file “/etc/sysctl.conf” and add “net.ipv4.ip\_forward=1”

The server now needs to how the forwarding should work. Run the command “sudo iptables -A FORWARD -i <server> -o <client> -j ACCEPT” replacing server with the hosts interface name and client with the client’s interface name.

Then run “sudo iptables -A FORWARD -i <client> -o <server> -m state --state  
RELATED,ESTABLISHED -j ACCEPT”

followed by,

“sudo iptables -t nat -A POSTROUTING -o <server> -j MASQUERADE”

Finally we run “sudo bash -c "iptables-save > /etc/iptables/rules.v4"” to save our settings and make our changes permanent.

Moving on to the client machine, you need to modify the network config files so open the file found at /etc/netplan/00-installer-config.yaml and remove its contents.

Now we go to /etc/netplan/99\_config.yaml, it should look like:

network:  
version: 2  
renderer: networkd  
ethernets:  
<client interface>:  
dhcp4: true  
routes:  
- to: default  
via: <server ip>  
nameservers:  
addresses: [<server ip>]

Replace “server ip” with your hosts machines IP.

Save and close the file and then restart the client.