Router Network Documentation

Internet Router

Configure each interface present so that they all have a line of communication to each router. The interfaces are as follows:

- eth0(net-1): 192.168.0.1/24
- eth1 (net-2): 192.168.10.1/24
- eth2(external): DHCP
- eth3 (net-3): 192.168.20.1/24
  To configure a vyos router you are going to need to enter the configuration mode within VyOS which is:

```
vyos@vyos:~$ config
[edit]
vyos@vyos#
```

Important: For each command you type in, you should get into the habit of commiting each command you type in:

```
vyos@vyos# <RANDOM VYOS CONFIGURATION>
vyos@vyos# commit
[edit]
```

This will enable us to set up interfaces along with the critical services and configurations needed in order for our networking to function properly.

```
vyos@vyos# set interfaces ethernet <INTERFACE> address <IPADDRESS>
vyos@vyos# set interfaces ethernet eth0 description <DESCRIPTION>
```

Always ensure to accurately label your interfaces so fututre engineers are able to differentiate them. Use the above commands to set up the interfaces for each router. NOTE: when applying this command for the external interface, make sure to include dhcp in place of a static ip address.

Next we are going to have to set up NAT. This will enable each client to communicate through a centralized eternal facing device known as our internet or public router. the configurations are as follows:

```
vyos@vyos# set nat source rule 100 outbound-interface <EXTERNAL INTERFACE>
vyos@vyos# set nat source rule 100 source address 192.168.0.0/16
vyos@vyos# set nat source rule 100 translation address masquerade
```

Then we need to set up the dns service so that clients use common websites such as google or amazon websites.

```
vyos@vyos# set service dns forwarding listen-address 192.168.0.1
vyos@vyos# set service dns forwarding allow-from 192.168.0.0/16
vyos@vyos# set service dns forwarding name-server 10.100.0.1
```

Finally we are going to need to configure this router to accept network traffic flow by default.

```
vyos@vyos# set firewall name WAN-LOCAL default-action 'allow'
```

We are then inclined to configure a system level dns server so that the router can use and complete DNS lookups. More than one DNS server can be added for redundency purposes.

```
vyos@vyos# set system name-server 10.100.0.1
```

Finally after we have commitied each and every command and ensured that there were no errors, we can save to the running config.

```
vyos@vyos# save
Saving configuration to '/config/config.boot'...
Done
[edit]
```

LAN Routers (Networks 1-3)

It is also important to enter configuration mode so you can edit directly to the running configuration.

```
vyos@vyos$ config
[edit]
```

Configuration for the three WAN routers will be a little different but mostly the same. Lets start with the interface configuration:

```
vyos@vyos# set interfaces ethernet <INTERFACE-DEVICE> address <LAN-INT>
vyos@vyos# set interfaces ethernet <INTERFACE-DEVICE> description
<DESCRIPTION>
```

NOTE: The address on this interface will **not** be the same as what you previously configured from the internet/ public facing router. This will be the gateway address and the dns that client devices on this subnet will reference.

It is also necessary to set up NAT on these routers as well so that the client devices in each traffic in the network can be routed to their destinations.

```
vyos@vyos# set nat source rule 100 outbound-interface <INTERFACE-DEVICE>
vyos@vyos# set nat source rule 100 source address <WAN-INT>
vyos@vyos# set nat source rule 100 translation address masquerade
```

You'll also have to configure specific protocols to statically route from subnets so that a client hops from router to router within the internal WAN:

```
vyos@vyos# set protocols static route 0.0.0.0/0 next-hop <WAN-INT>
vyos@vyos# set protocols static route <SUBNETADDRESS>/32 next-hop <LAN-INT>
```

It is also essential that you configure the DNS forwarding on each router as well so that client DNS requests go to the actual dns server and not the router itself since this is not a DNS server.

```
vyos@vyos# set service dns forwarding cache-size '0'
vyos@vyos# set service dns forwarding listen-address <ROUTERINT>
vyos@vyos# set service dns forwarding allow-from '192.168.0.0/16'
vyos@vyos# set service dns forwarding name-server <DNS-SERVER>
```

Remember, for all other routers were are also going to configure the name server at a system level as well for the same reasons as the internet router.

```
vyos@vyos# set system name-server 10.100.0.1
```

Ubuntu Client-Side Configuration

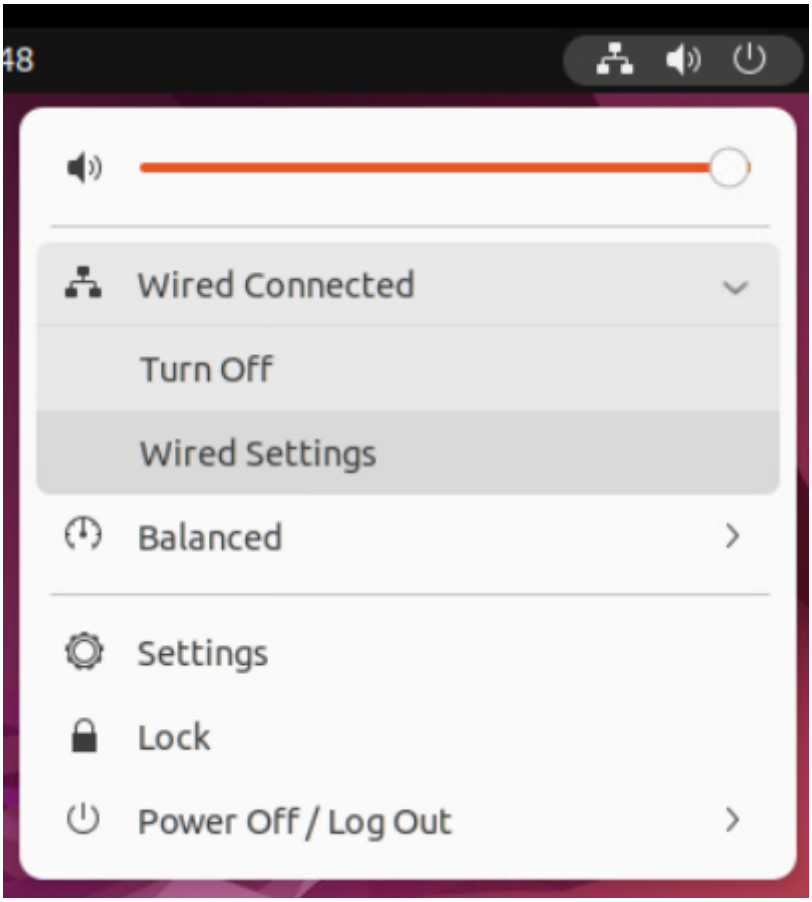
For client side operations, there are three tasks you must complete:

- Statically assign Network Configuration
- Update Ubuntu operating system

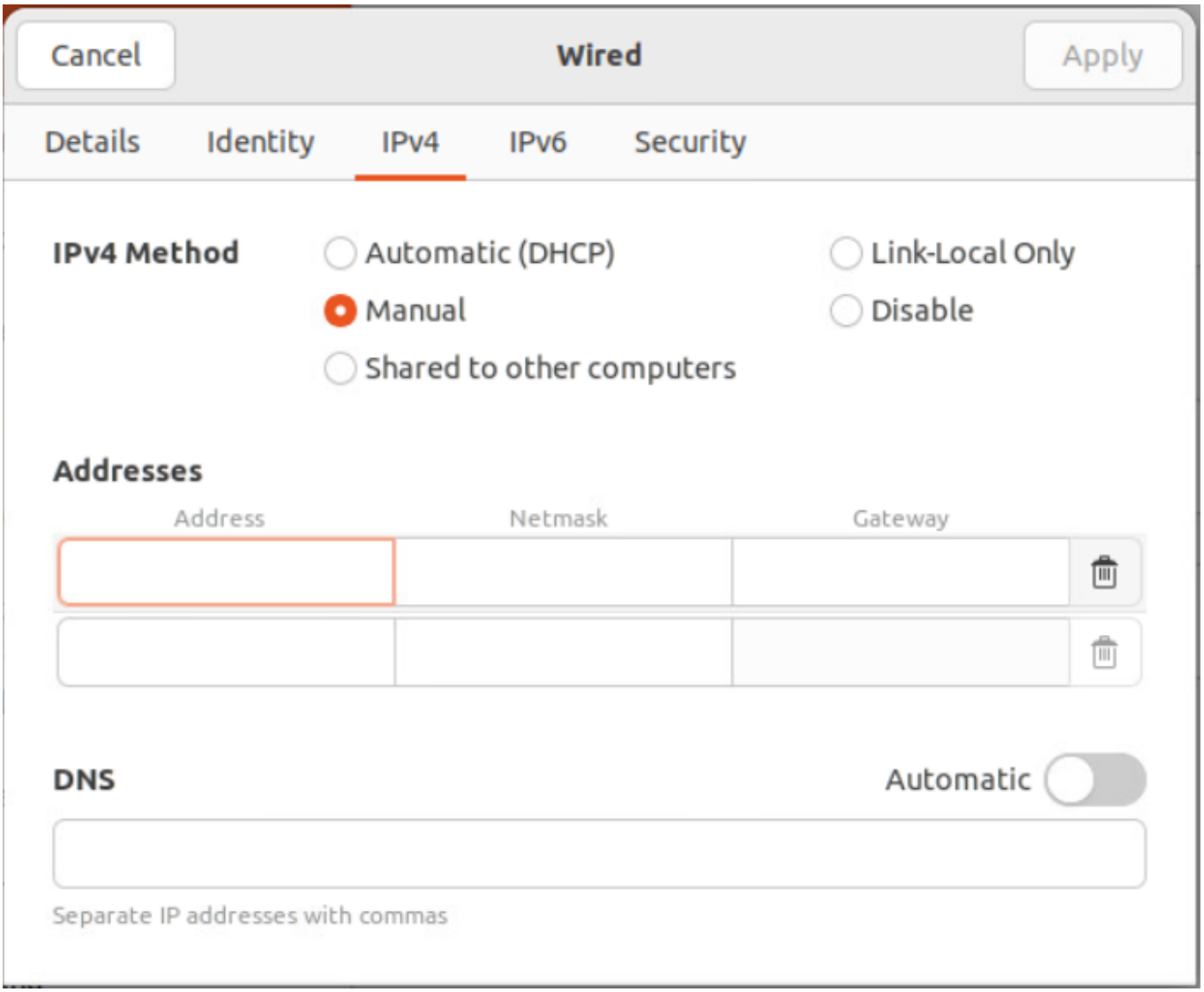
- Use SCP to exchange files between clients

Network Configuration

Depending on what version of Ubuntu you are operating, the NetworkManager service is responsible for handling network configurations. Access the control panel from the Ubuntu system dropdown menu:



From here select the gear icon for your wired interface and navigate to the IPv4 tab which will look like this:



Populate Addess, Netmask, Gateway, and DNS with their appropriate fields. After this is completed. It is necessary to restart the NetworkManager service in the terminal to ensure your changes are actually taking effect.
Right click an empty area on your desktop, and choose "Open in Terminal"
type in this command:

```
$ sudo systemctl restart NetworkManager
```

NOTE: 'sudo' is not necessarily part of the command we need to restart this service but since the user we are operating this on is the root system user, we will have to use the 'sudo' binary to temporarily elevate our privlieges.
After this, ensure the ip address has changed by running a common network diagnostic tool:

```
$ ip a
# OR #
$ ip -brief a
```

From here, test our network connection by either opening the Firefox web browser, or straight to updating ubuntu.

Updating Ubuntu

After ensuring there is a stable internet connection, open the terminal (or open the last terminal you had open) and type in this command:

```
$ sudo apt update
```

This will fetch the necessary dependencies required for an update but not actually install them. For that, use this command:

```
$ sudo apt upgrade
```

This second command might take a little bit longer to complete.
These sequence of commands will successfully update the operating system.

SCP Transfer

You will see a unique .png on each of the ubuntu desktops, you're task will be (on any client from any network) copy the .png from the other two clients on to the client you are currently using. First you should ensure 'openssh-server' is installed and configured correctly on all clients. In order to achive this, complete these sequence of commands:

```
$ sudo apt install openssh-server
$ sudo ufw allow from any
$ sudo ufw enable
$ sudo ufw status
```

These commands will allow for connections from any ip address to any port. **NOTE:** This is not at all advisable/ poor network security practice. This is to only be used for this simulated lab environment. From here are are able to use scp (a tool accompanied with ssh) to transfer files between clients. You will have to know where the location of the file is located and for our use case, it should be in the "/home/user/Desktop/" Directory for all clients.

```
$ scp <USER>@<HOST>:/PATH/TO/REMOTE /PATH/TO/LOCAL/
```

The above will copy a remote file from and place it where you'd want it to go.

```
$ scp /PATH/TO/LOCAL/ <USER>@<HOST>:/PATH/TO/REMOTE
```

The above will send a local file to a remote location to the client you specify.

This documentation should provide you with all the necessary commands and tools for completing this simulated network lab.