# Obligatory Assignment 1

## DATA 2410: Networking and Cloud Computing

**The deadline is February 24, 2020 at 23:59 p.m.**

## INSTRUCTIONS

This assignment is divided into **2 parts**. **Part 1** is **OBLIGATORY FOR EVERYONE**. That means that when you deliver your files, everyone must have the answer to this set of exercises. For **Part 2**, you have to choose ***only one*** **out of the 2 suggested tracks**. That means when you deliver this obligatory assignment, your file must contain all the answers from Part 1 and the ones corresponding to the track you chose.

When you have completed the first part and one of the tracks in the second part, you are ready to hand in your assignment. **The hand-in should be a PDF**. You should show your configuration and relevant CLI commands in Packet Tracer tasks by providing screenshots and providing necessary information as indicated in other tasks.

# Part 1 - Obligatory for Everyone

## IPv4

### Task 1

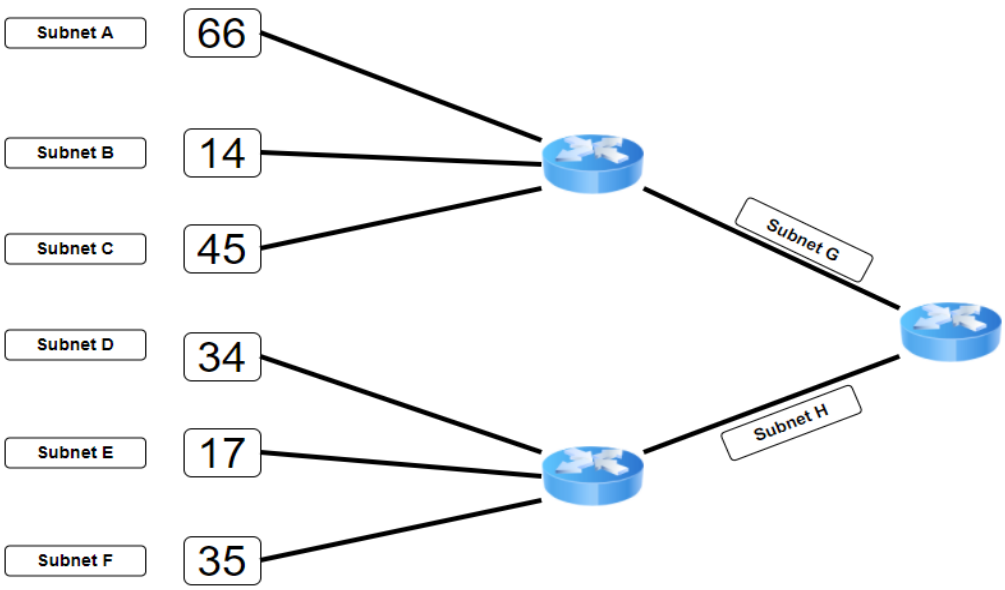
- **Given the IP address 172.18.12.59 (in decimal notation) and a subnet mask of /28, what is:**
  - The IP address in binary notation?
  - The subnet mask in binary notation?
  - The subnet mask in decimal notation?
  - The Network ID?
  - The last available host address and broadcast address for the network?
  - The number of host addresses, given that you need two for router interfaces?

- **Given a network with a /26 subnet mask what is:**
  - The subnet mask in decimal notation?
  - The number of subnets you get if you use FLSM (i.e., Fixed Length Subnet Mask) to divide into subnets, with each subnet having a /28 subnet mask?
  - The subnet mask for each subnet, using FLSM, if you need addresses for 12 hosts in each subnet?

# Task 2

In the following task you will use VLSM to fill out the Table, given the figure below. You are given a /23 subnet mask and a Network ID of **172.18.4.0** . You should assign Network IDs starting from subnet A and ending with subnet H. For the subnets A-H, there should be 1 address for a router interface in each subnet. For subnet A, the information is already filled out for you.

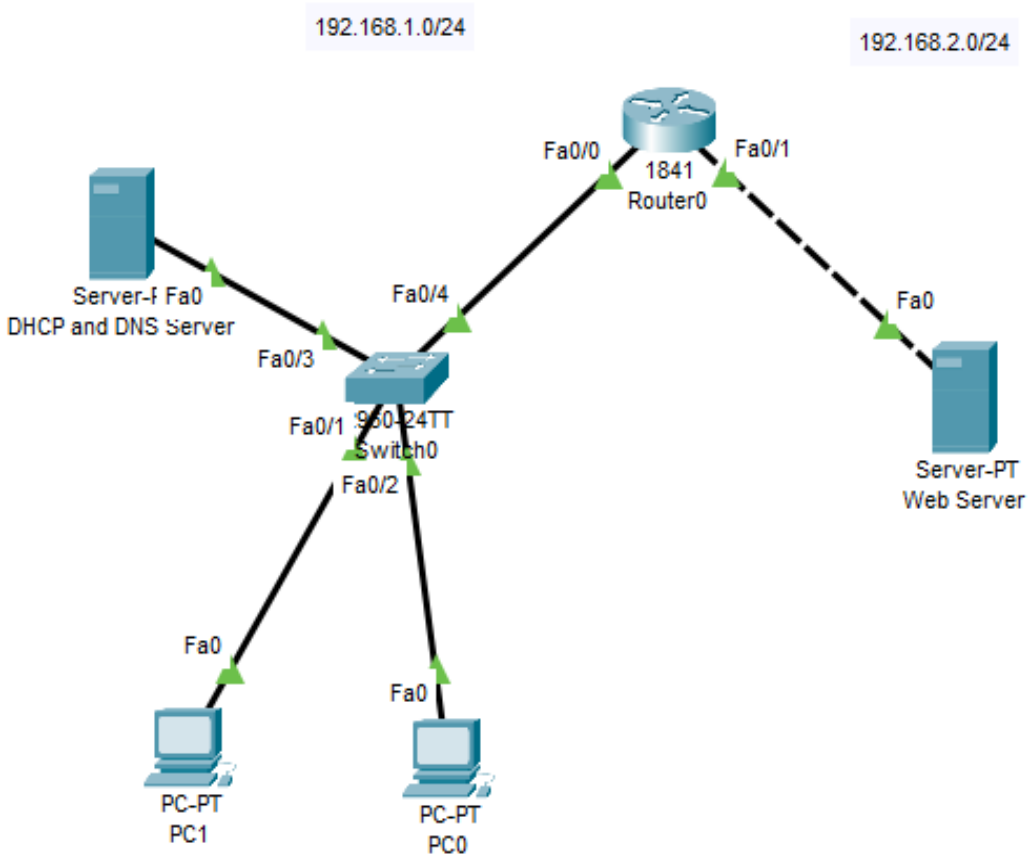
| Subnet | Addresses Needed | Available Hosts | Unused Hosts | Network ID | CIDR | Subnet Mask | Usable Range | Broadcast |
|---|---|---|---|---|---|---|---|---|
| A | 67 | 126 | 59 | 172.18.4.0 | /25 | 255.255.255.128 | 172.18.4.1 - 172.18.4.126 | 172.18.4.127 |
| B | | | | | | | | |
| C | | | | | | | | |
| D | | | | | | | | |
| E | | | | | | | | |
| F | | | | | | | | |
| G | | | | | | | | |
| H | | | | | | | | |



## Packet tracer

## Topology 1: Network Protocols: DHCP, DNS and HTTP

In this topology, you do not need to configure the PCs manually. They will get IP addresses, default gateways, and other network parameters automatically.



**To implement topology 1, follow the instructions below:**

• IP address for Router0 f0/0: 192.168.1.1

• IP address for Router0 f0/1: 192.168.2.1

• IP address for DHCP server: 192.168.1.254

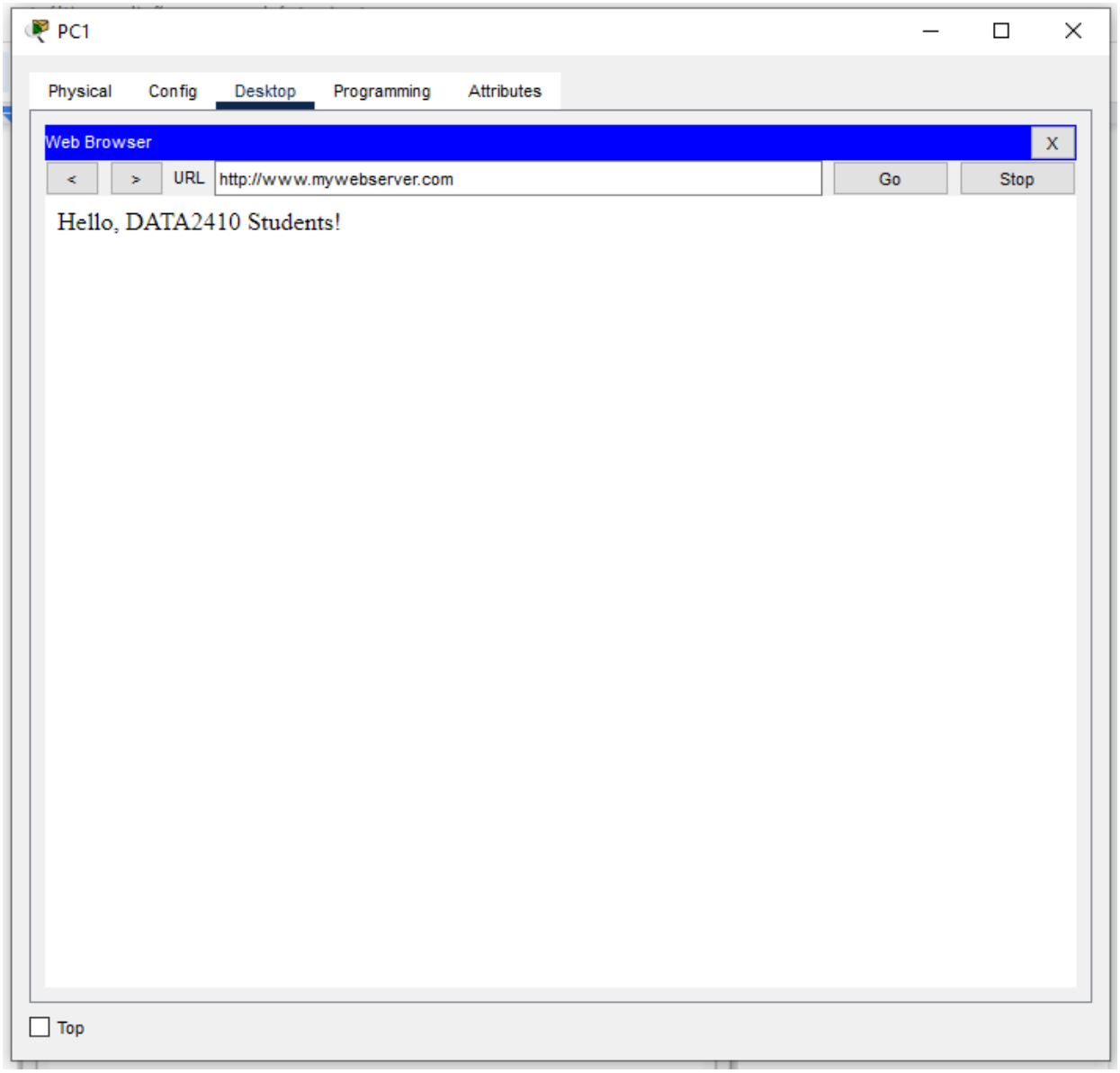
• Start IP address for DHCP pool 1: 192.168.1.100

• IP address for DNS server: 192.168.1.254

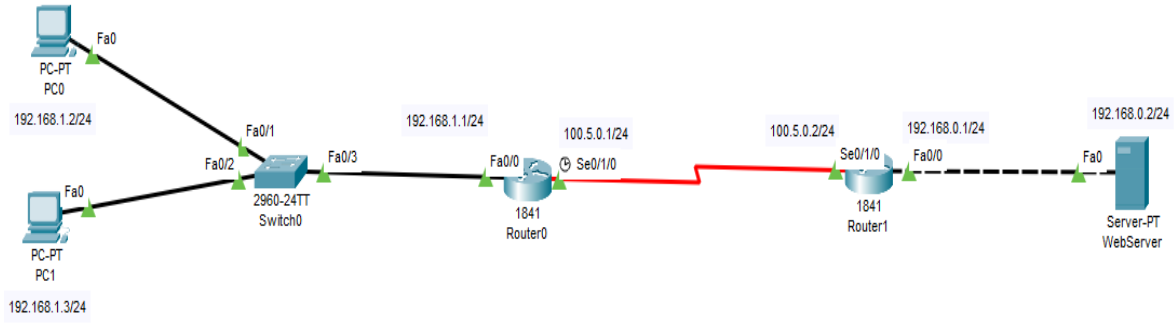
• Name of Webserver: www.mywebserver.com

• Make sure that PC1 and PC2 get IP addresses, default gateways and DNS automatically.

• Make changes to the webserver index page, so when you browse the webserver you



- **Topology 2: NAT**




**To configure NAT, do the following:**

**On Router1:**

Router1(config)#**ip nat inside source static< Inside local IP address> < Inside global IP address>**

Router1(config)#**interface <Inside local Interface>**

Router1(config-if)#**ip nat inside**

Router1(config)#**interface <Outside local Interface>**

Router1(config-if)#**ip nat outside**

**On Router0:**

Router0(config)#**interface <Inside local Interface>**

Router0(config-if)#**ip nat inside**

Router0(config)#**interface <Outside local Interface>**

Router0(config-if)#**ip nat outside**

Router0(config)#**access-list 1 permit <Local address> <Wildcard bits>**

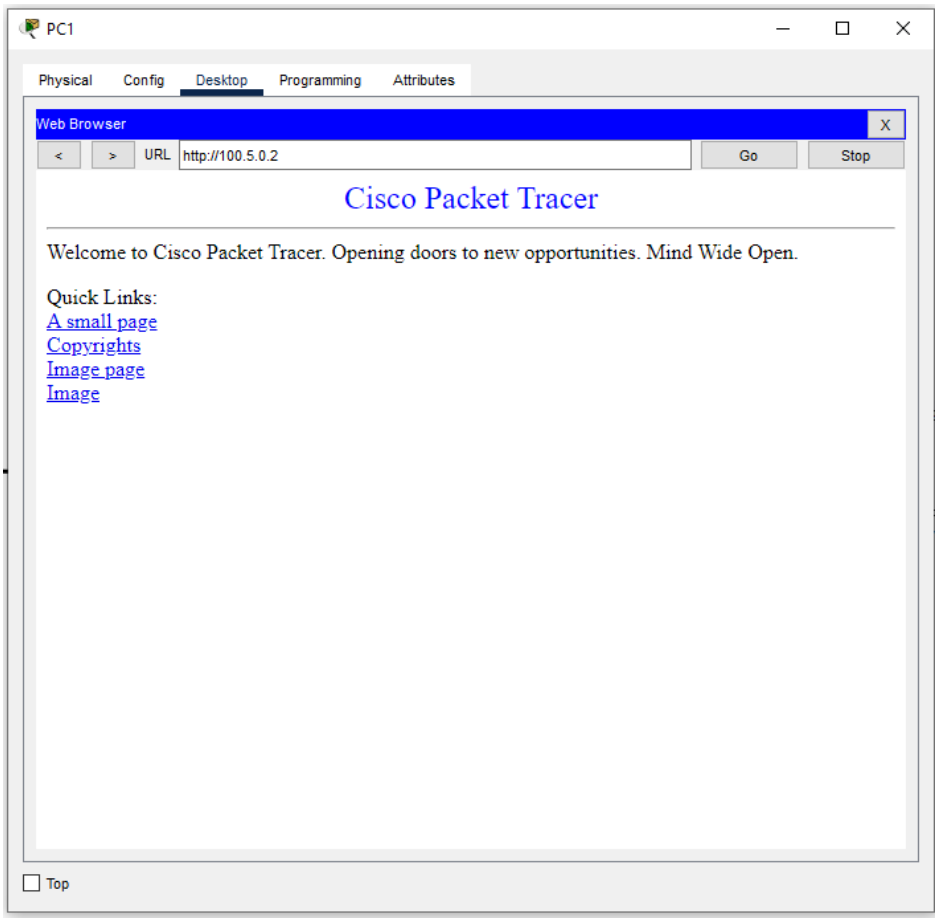
Router0(config)#**ip nat pool <Name> <Start IP> <End IP> netmask <Subnet mask>** Router0 (config)#**ip nat inside source list <Access-id> pool <Name>**

**Notes:**

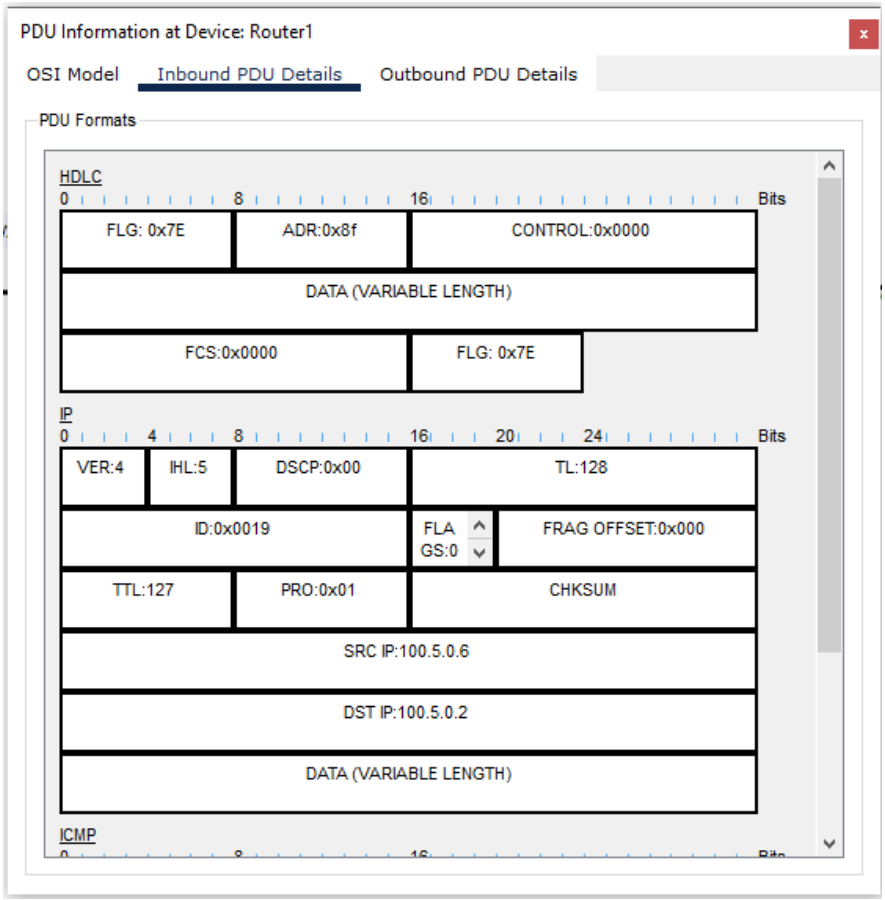
- Wildcard bits are the opposite of the subnet mask. When you subtract the Subnet Mask in Binary notation from 255.255.255.255 (Or take the Binary complement) you get the Wildcard Mask. A wildcard of 255.255.0.0 is therefore 0.0.255.255
- You need to configure some routes on each router to direct the packets to the WebServer and back to the Clients. You can also use default routes to direct any packet which is not directly connected to the router.

**Result:**

- Browse the server with this public IP address: 200.10.0.2




- Ping from PC0 to 100.5.0.2 and capture ICMP in simulation mode to examine source and destination IP address, you will get the following result:

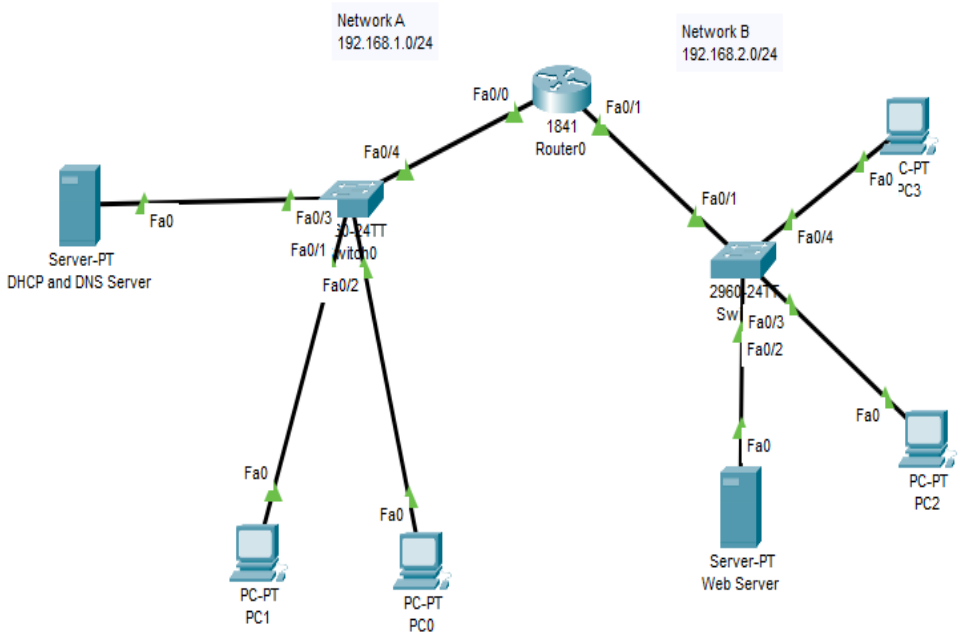

PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0
8
16
Bits
FLG: 0x7E
ADR:0x8f
CONTROL:0x0000
DATA (VARIABLE LENGTH)
FCS:0x0000
FLG: 0x7E
IP
0
4
8
16
20
24
Bits
VER:4
IHL:5
DSCP:0x00
TL:128
ID:0x0019
FLA
GS:0
FRAG OFFSET:0x000
TTL:127
PRO:0x01
CHKSUM
SRC IP:100.5.0.6
DST IP:100.5.0.2
DATA (VARIABLE LENGTH)
ICMP

## Part 2

## For this Part 2: you should choose ONLY ONE track.

## Track 1

### Topology: DHCP Relay



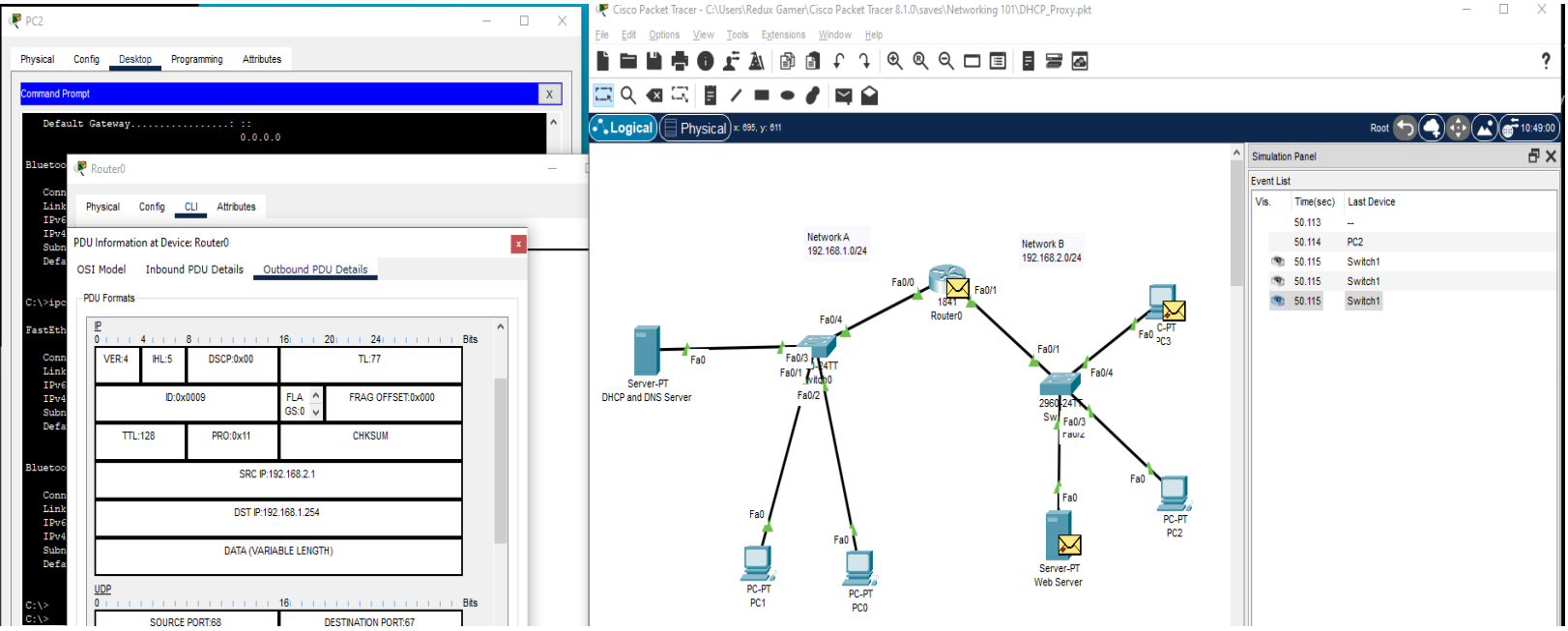

**To implement the above topology, follow the instructions below:**

- IP address for Router0 f0/0: 192.168.1.1/24
- IP address for Router0 f0/1: 192.168.2.1/24
- IP address for DHCP server: 192.168.1.254/24
- Start IP address for DHCP pool1: 192.168.1.100
- Start IP address for DHCP pool2: 192.168.2.100
- You need to configure the router as a DHCP relay agent to make network 2 get IP addresses, default gateways

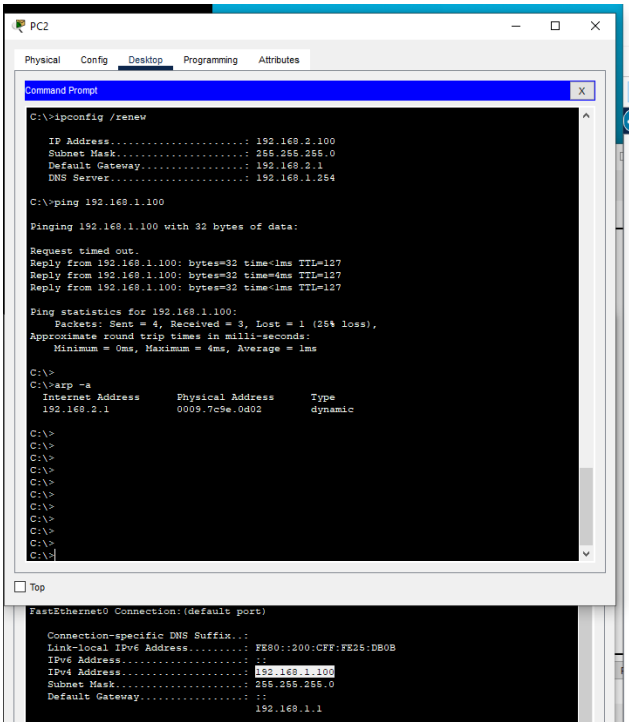
and other network parameters automatically the same as Network1. To do that, you need to configure Router0 f0/1 with this command:

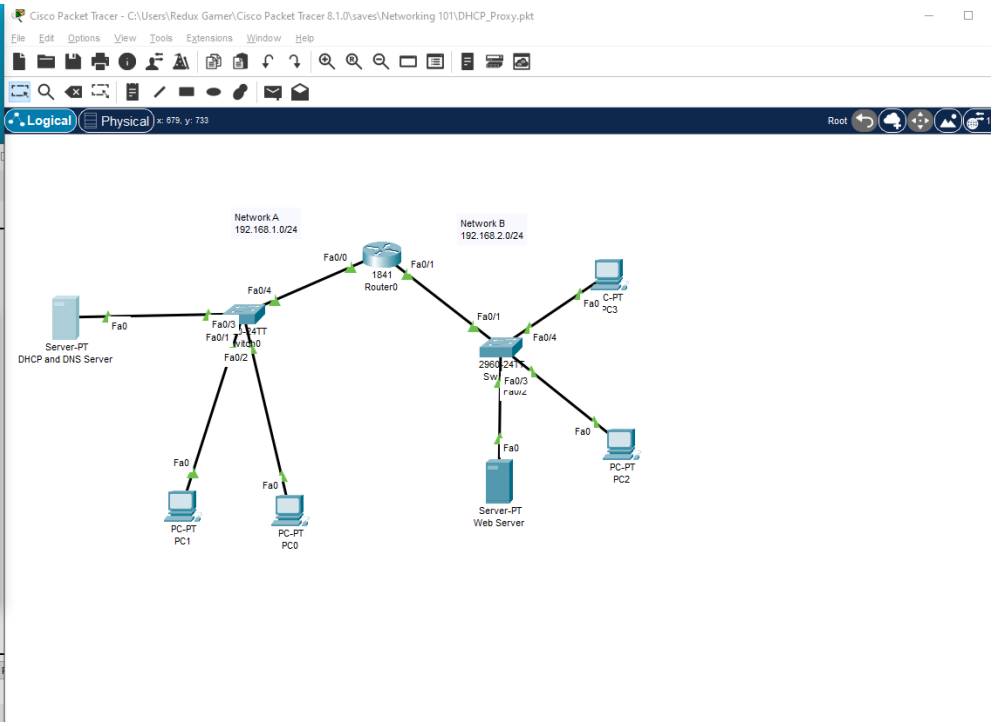
Router0(config-if)#**ip helper-address <DHCP IP address>**

- In simulation mode, remove the link from PC2 and place it back. Analyze the DHCP packet flow.
- Document all your work by taking screenshot.




Confirm PC2 has IP address and can ping PC1:


PC2
Physical
Config
Desktop
Programming
Attributes
Command Prompt
C:\>ipconfig /renew
IP Address.....: 192.168.2.100
Subnet Mask.....: 255.255.255.0
Default Gateway.....: 192.168.2.1
DNS Server.....: 192.168.1.254
C:\>ping 192.168.1.100
Pinging 192.168.1.100 with 32 bytes of data:
Request timed out.
Reply from 192.168.1.100: bytes=32 time<1ms TTL=127
Reply from 192.168.1.100: bytes=32 time=4ms TTL=127
Reply from 192.168.1.100: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.1.100:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4ms, Average = 1ms
C:\>
C:\>arp -a
Internet Address Physical Address Type
192.168.2.1 0009.7c9e.0d02 dynamic
Top
FastEthernet0 Connection:(default port)
Connection-specific DNS Suffix..:
Link-local IPv6 Address.....: FE80::200:CFF:FE25:DB0B
IPv6 Address.....: ::
IPv4 Address.....: 192.168.1.100
Subnet Mask.....: 255.255.255.0
Default Gateway.....: ::
192.168.1.1

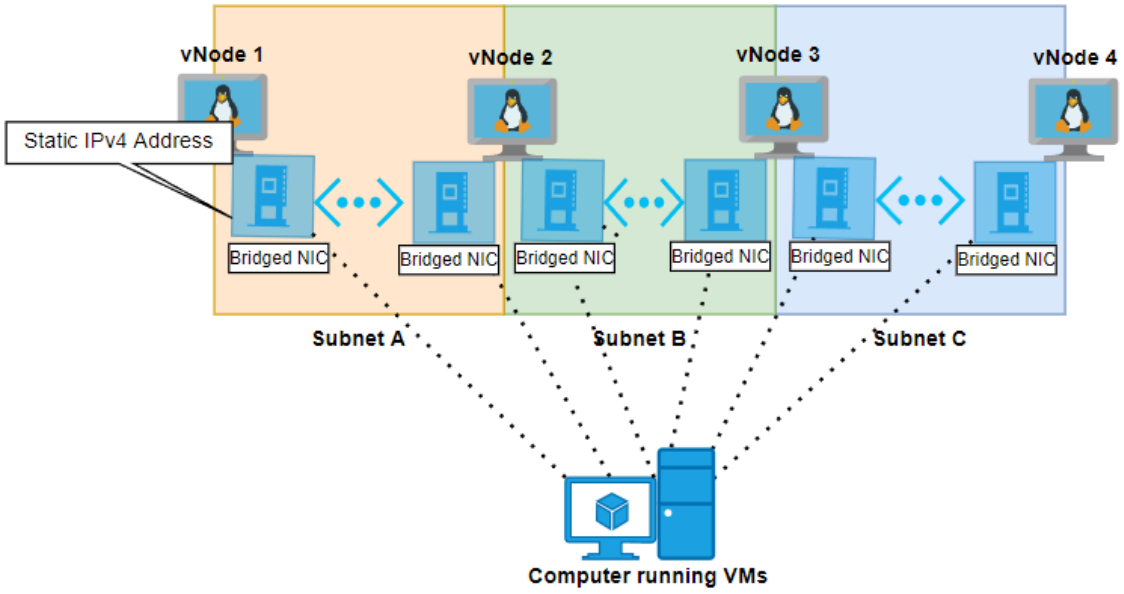

Cisco Packet Tracer - C:\Users\Redux Gamer\Cisco Packet Tracer 8.1.0\saves\Networking 101\DHCP_Proxy.pkt
File Edit Options View Tools Extensions Window Help
Logical
Physical
Root
Network A
192.168.1.0/24
Network B
192.168.2.0/24
Fa0/0
Fa0/1
1841
Router0
Fa0/4
Fa0/3
Fa0/1
Fa0/2
Fa0
Server-PT
DHCP and DNS Server
C-PT
PC3
Fa0/1
Fa0/4
Fa0/3
Fa0
PC-PT
PC2
Fa0
Server-PT
Web Server
Fa0
PC-PT
PC1
Fa0
PC-PT
PC0

## Part 2

### Track 2

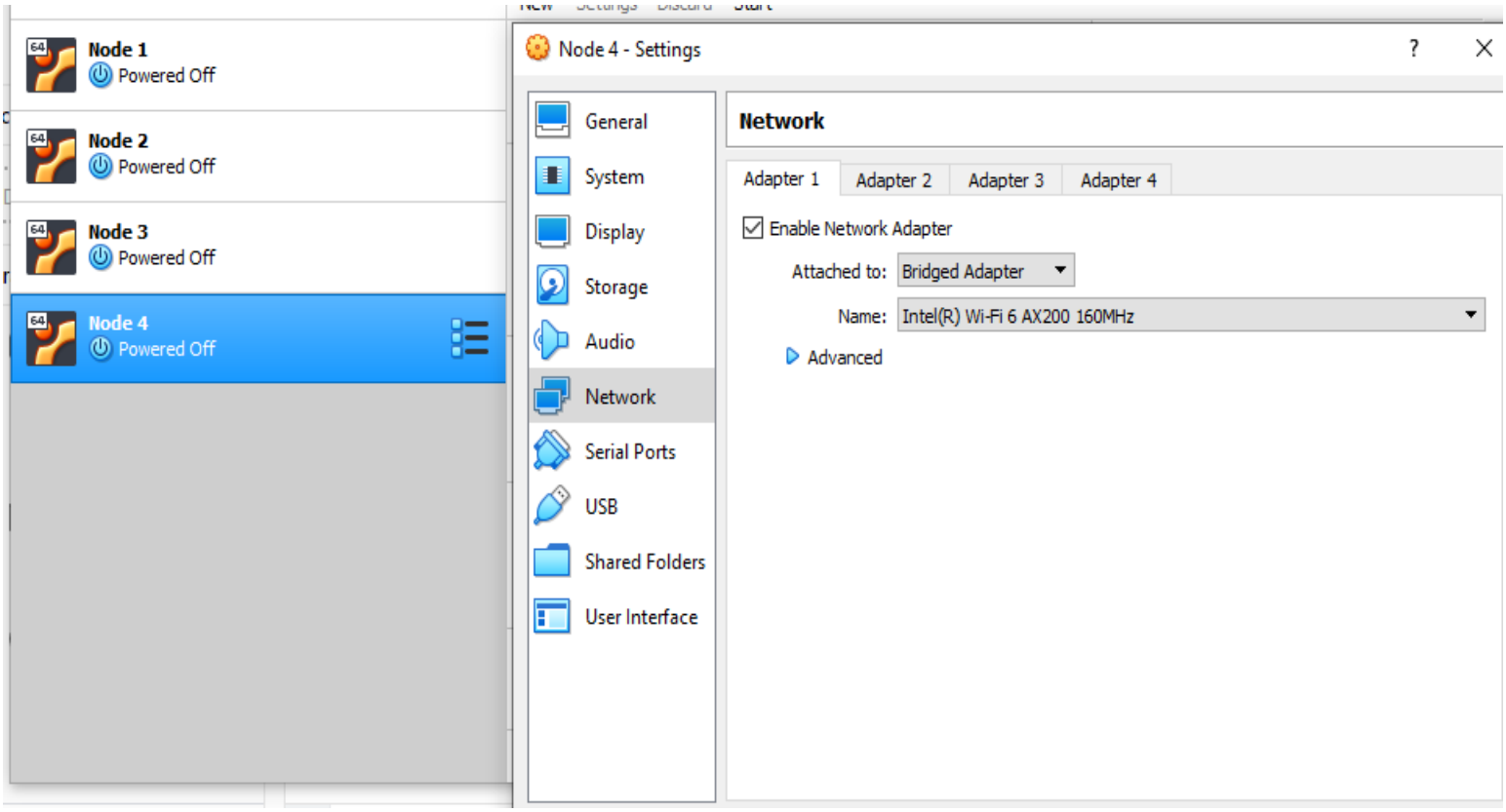
### Static Routing and Bridged Adapters – VirtualBox

Configure 4 VMs with static IP addresses and three distinct subnets. VMs #2 and #3 should have two bridged NICs and VMs #1 and #4 should have 1 bridged NIC, allowing the VMs with 1 NIC to reach the subnets #1 and #3 only when both the NICs on the bridge are up. Make a drawing, similar to the figure below, indicating the Network IDs, subnet masks and IP addresses you used for your setup



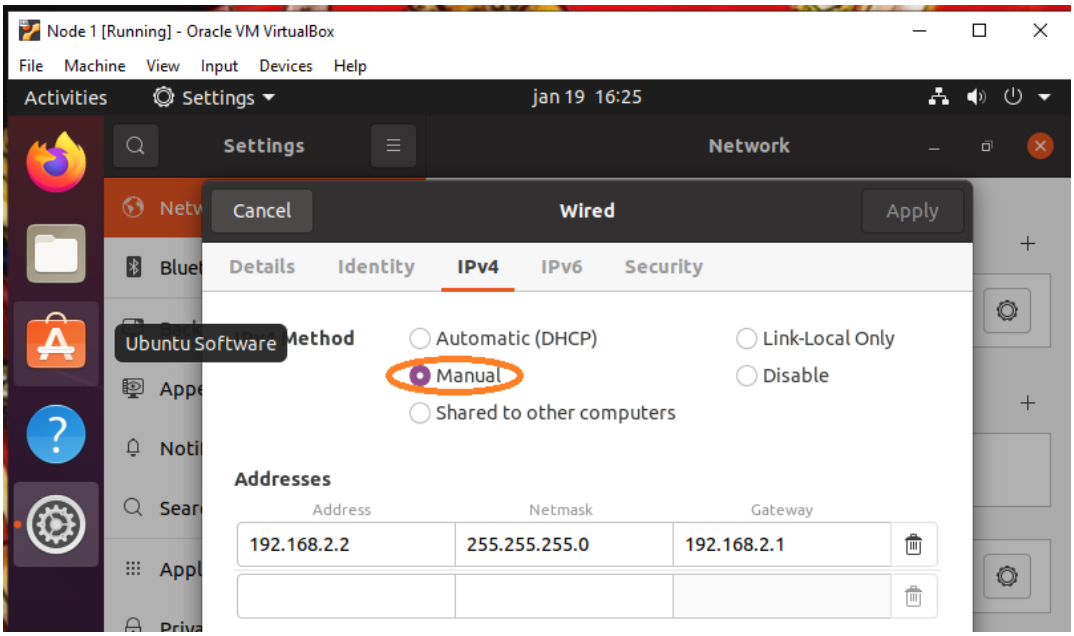
**Access VirtualBox Preferences and create new NAT Network for each subnet**

**Configuring bridged adapters in VirtualBox Manager:**




**Setting static IPv4 addresses:**

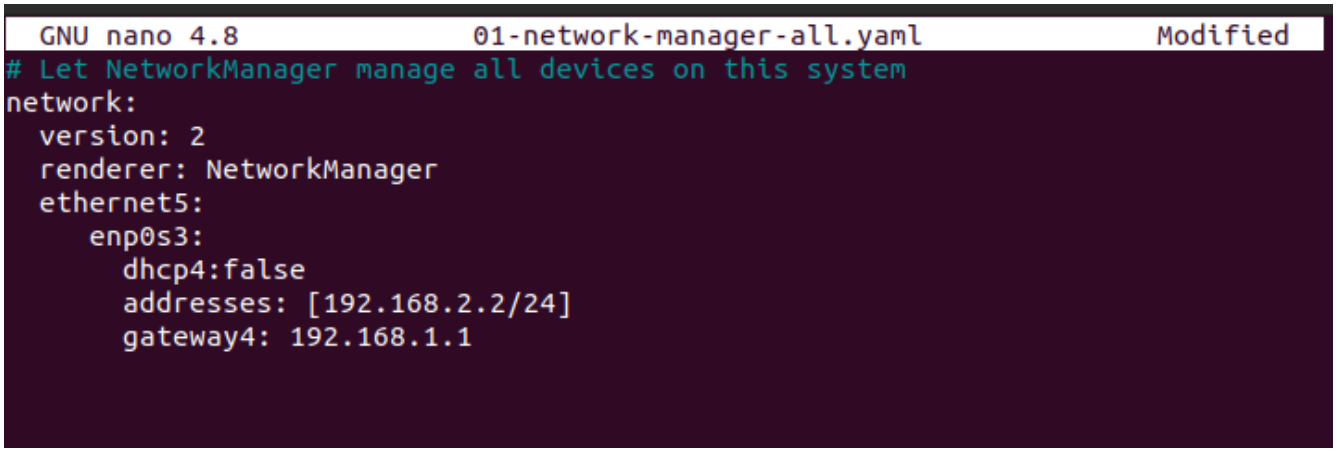
To set a static IP for a VM, power it on and enter “Advanced Network Configuration”. Make sure "Manual" method is enabled, press “Add” and enter the IP address for the corresponding interface along with the Netmask for the subnet. The inputs in the screenshot need not be used. When you have entered what is required for the corresponding interface, press Enter and save the configuration. You may need to reboot for changes to take effect.




Alternatively, if you prefer the terminal, you may edit the configuration file /etc/netplan/01- network-manager-all.yaml

Navigate to /etc/netplan by writing the command **cd /etc/netplan**

Start editing the file with **sudo nano 01-network-manager-all.yaml**

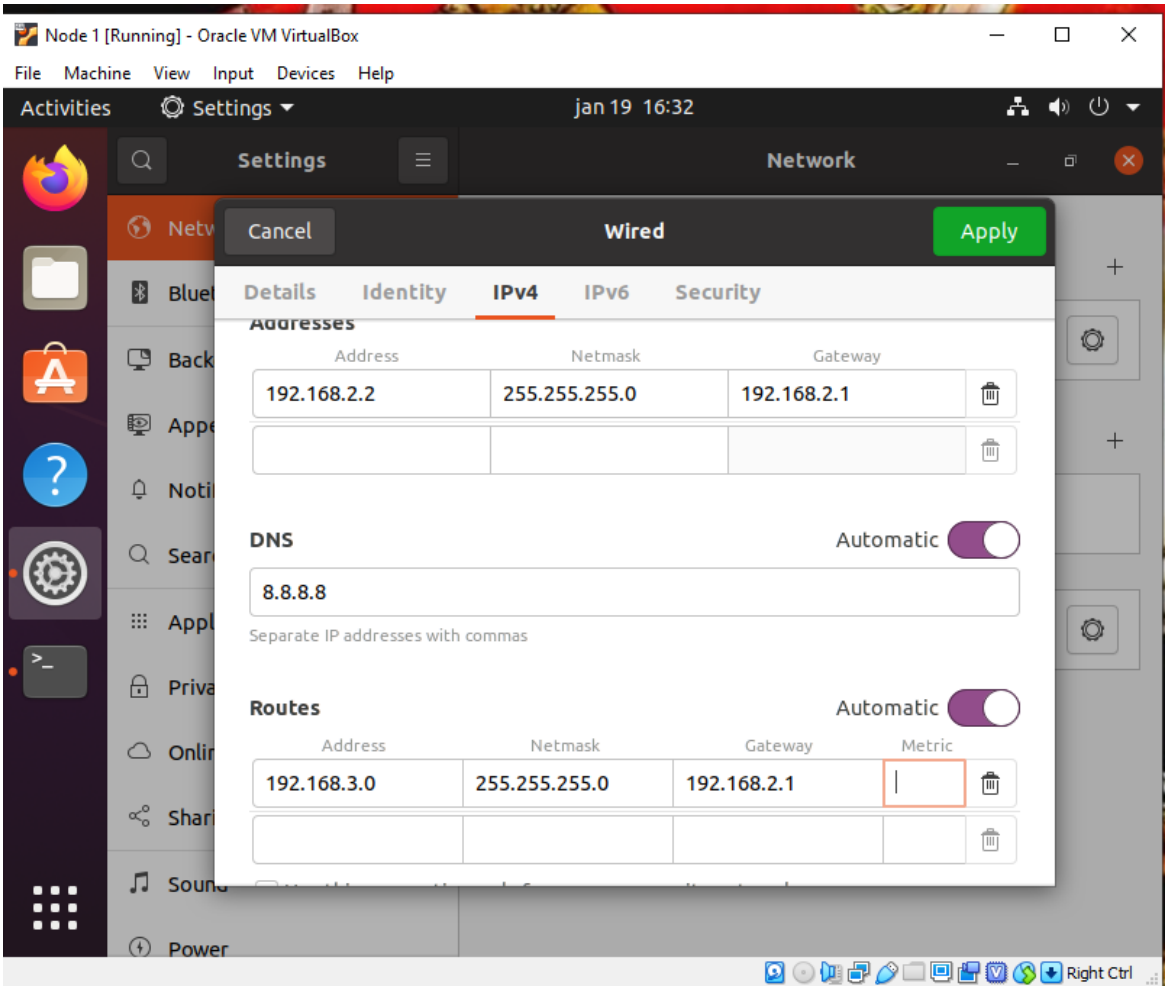



Make sure that when you edit the file, indentation is done by double spaces, anything else will result in an error. Exit the file with **Ctrl+X** and press **Y** to save changes.

After exiting the file, execute the following command: **sudo netplan apply**

**Static routing:**

If you configured IP address in the "Advanced Network Configuration", then you may also configure IP routes by simply pressing the "Routes…" option and adding as you did with the IP addresses.




Alternatively, you can enter the terminal in your Linux VM and type the following command: **sudo ip route add 192.168.3.0/24 via 192.168.2.1**

In the command above, the Network ID, Subnet Mask and IP address are just examples. The command does not make settings persistent; you are advised to use the "Advanced Network Configuration" if you wish to keep the settings on reboot.

Packet forwarding on Linux is disabled by default. To enable forwarding you can enter the command:

**sudo sysctl -w net.ipv4.ip_forward=1**

Since, this command does not make settings persistent, you can for example create a script which can be run on startup - can be created by executing **nano enableforward.sh** and copying the script given on the next page.

```
  GNU nano 4.8                    enableforward.sh
#!/bin/dash

sudo sysctl -w net.ipv4.ip_forward=1
```

You may need to change permissions to run it, which can be done like this: **chmod 700 enableforward.sh**

To run the script type **./enableforward.sh** and press Enter (given that you are in the same folder). You can also add the "ip route" command to this script.

Verify that everything is set up correctly by pinging between VMs #1 and #4. Show the output you get.

Now that everything is setup correctly, try installing web servers in VMs #2 and #3 and access them using browsers in VMs #1 and #4, show an output of the welcoming page for each. You may for example install apache2 by executing **sudo apt-get install apache2**

Make sure that you have configured adapters for VMs #2 and #3 so that these VMs can access the Internet.

You may need to change permissions to run it, which can be done like this: **chmod 700 enableforward.sh**

To run the script type **./enableforward.sh** and press Enter (given that you are in the same folder). You can also add the "ip route" command to this script.

Verify that everything is set up correctly by pinging between VMs #1 and #4. Show the output you get.

Now that everything is setup correctly, try installing web servers in VMs #2 and #3 and access them using browsers in VMs #1 and #4, show an output of the welcoming page for each. You may for example install apache2 by executing **sudo apt-get install apache2**

Make sure that you have configured adapters for VMs #2 and #3 so that these VMs can access the Internet.