BUILDING A VIRTUAL SANDBOX NETWORK USING VIRTUALBOX

ANAND MANKUZHA POULOSE

STUDENT ID: A00028816

INTRODUCTION

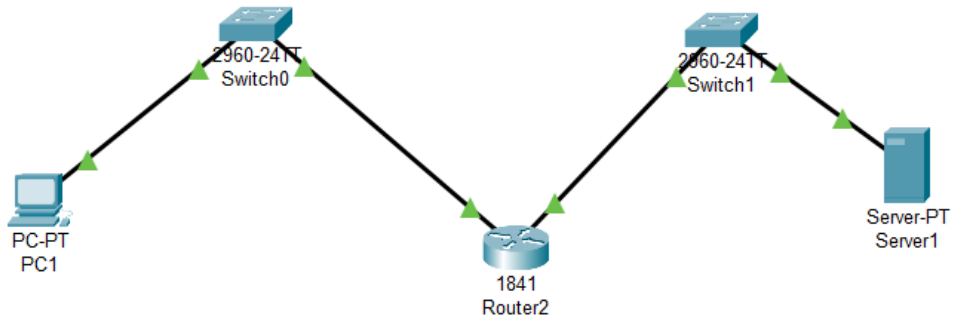
A sandboxed network is a virtual setting that offers a secure and segregated area where you can test out different network setups, protocols, and systems without worrying about interfering with live or production settings. Consider it a virtual playground where you can safely study and experiment with networking principles, solve problems, and model different network situations. You can experiment and practice without fear of damaging important systems or data by setting up a sandboxed network.

Utilize VirtualBox, a well-known and intuitive virtualization software, to create your own private sandboxed virtual network for this project. To replicate how real-world networks function, you will set up several virtual machines (VMs) in this network, each with its own private IP address. You can model a working network with a variety of functions, including web servers, application servers, and client computers, thanks to the interconnection of these virtual machines.

The objective of this task is to enhance your comprehension of fundamental networking principles. You will learn how to break up networks into more manageable chunks by gaining hands-on experience with IP subnetting. Additionally, you will delve into network interface settings, learning how to set up routing, assign IP addresses, and guarantee safe machine-to-machine communication. You will also work on simple server configurations, which entail setting up virtual machines to function as servers and perform services like file sharing and web hosting.

You will get practical knowledge of how networks work as well as abilities in network design, planning, and organization by the end of this assignment. Anyone hoping to work in network administration, IT support, or any other area that includes maintaining and safeguarding network infrastructures has to have these fundamental abilities. Your training ground will be this sandboxed network, which will protect your experiments from affecting real systems while enabling you to confidently apply what you've learned in practical settings.

1. NETWORK DIAGRAM



Configuring the Gateway Router (Router1)

To configure Router1 as a gateway router with two interfaces (one for the local network and one for the serial connection), follow the steps below:

❖ Enter Privileged EXEC Mode:

- From the router's console, enter privileged EXEC mode by typing:

Router1> enable

❖ Enter Global Configuration Mode:

- Enter global configuration mode to make configuration changes:

Router1# configure terminal

❖ Configure FastEthernet Interface (Local Network):

- Configure the **FastEthernet 0/0** interface, which connects to your local network:

Router1(config)# interface fastethernet 0/0

Router1(config-if)# ip address 192.168.21.1 255.0.0.0

Router1(config-if)# no shutdown

Router1(config-if)# exit

❖ Configure Serial Interface (Connection to Another Router or WAN):

- Configure the **Serial 0/1** interface, which connects to another router or WAN network:

```
Router1(config)# interface serial 0/1
Router1(config-if)# ip address 192.168.121.1 255.0.0.0
Router1(config-if)# no shutdown
Router1(config-if)# exit
```

Verification

To ensure your interfaces are correctly configured and active, use the following commands:

1. Verify Interface Status:

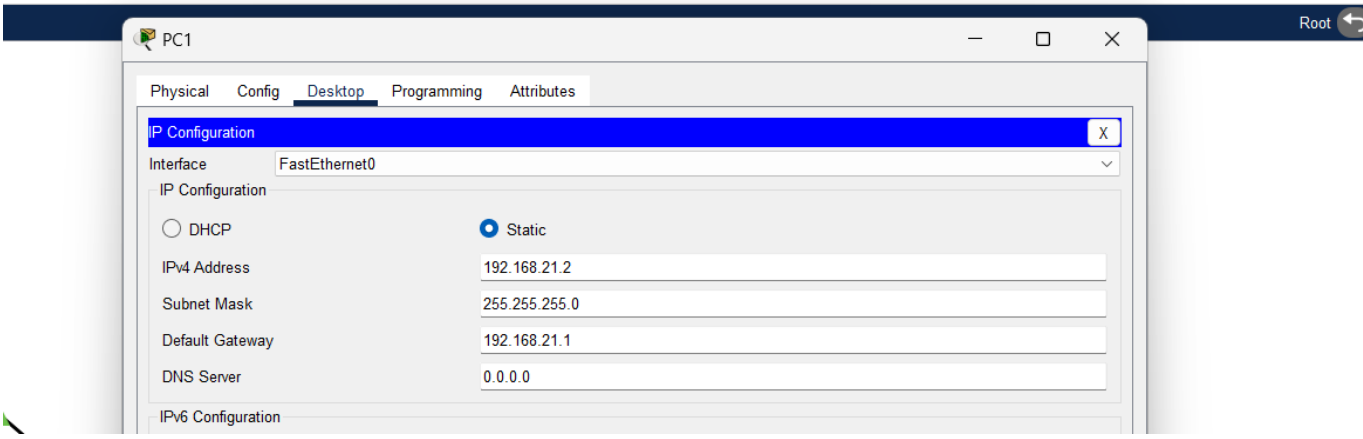
```
Router1# show ip interface brief
```

2. Ping Test: You can also test connectivity by pinging the configured interfaces:

```
Router1# ping 192.168.21.1
Router1# ping 192.168.121.1
```

Desktop (PC):

- Click the pc and open it go to the desktop >IP Configuration >change to static >Set IP address.



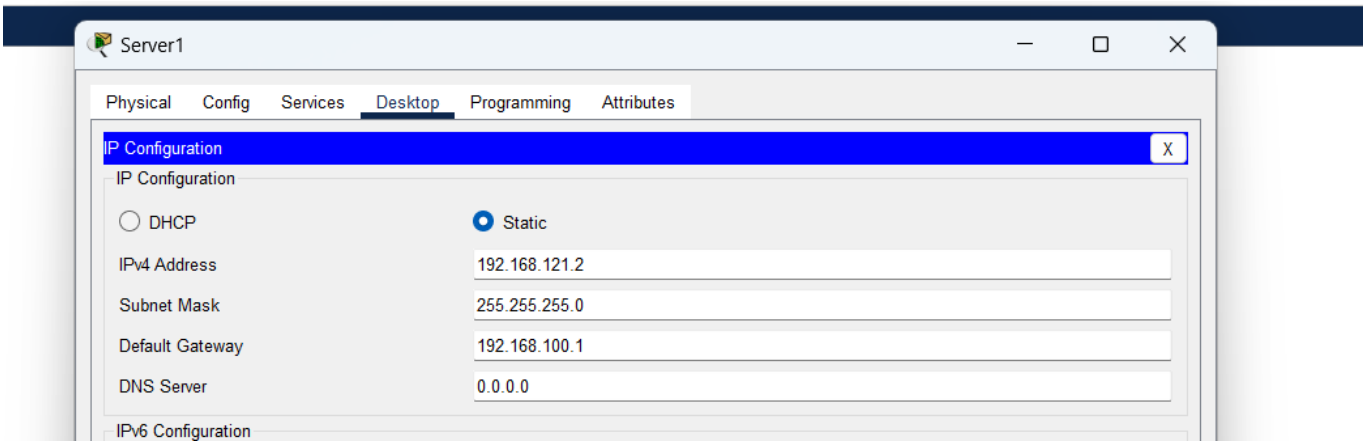
- Set all PC for this.
- And ping 192.168.21 .2

```
Cisco Packet Tracer SERVER Command Line 1.0
C:\>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:

Reply from 192.168.0.2: bytes=32 time<1ms TTL=127
Reply from 192.168.0.2: bytes=32 time<1ms TTL=127
Reply from 192.168.0.2: bytes=32 time<1ms TTL=127
Reply from 192.168.0.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Screenshot For ping PC to Gateway

Application Server:

- Click the Application Server and open it go to the desktop >IP Configuration >change to static >Set IP address.



- Click the Application Server and open it go to the service> Click HTTP> Turn ON HTTP and HTTPS> add HTML page in Application Server.

- Click the Application Server and open it go to the desktop >Web Browser >Type the IP address 192.168.100.2.

2. IP ADDRESS TABLE

Device	Role	IP Address	Subnet Mask
Desktop VM **(Ubuntu Desktop)**	Management	192.168.21.2	255.255.255.0
Gateway Router VM (enp0s3) **(Ubuntu Server)**	Internet Access	192.168.21.1	255.255.255.0
Gateway Router VM (enp0s8) **(Ubuntu Server)**	Subnet 01 - Internal Network	192.168.121.1	255.255.255.0
Gateway Router VM (enp0s9) **(Ubuntu Server)**	Subnet 02 - Internal Network	10.0.2.15/24	255.255.255.0
Application Server VM **(Wordpress)**	Server	192.168.121.2	255.255.255.0

3. Git Pages Lab Report

https://github.com/anandmankuzhapoulose/BUILDING-A-VIRTUAL-SANDBOXED

Configuration Guide for Virtual Machines and Operating Systems

To proceed with this project, you are required to set up three distinct virtual machines (VMs). Below are the detailed steps to guide you through the process.

Step 1: Prepare Your Virtual Environment

Before starting the configuration of individual VMs, ensure that the following prerequisites are completed:

1. **Install VirtualBox**:
    - Download and install the latest version of Oracle VirtualBox on your system.
2. **Download the Necessary Operating Systems (OS)**:
    - **Ubuntu Desktop OS** (ISO format) or another desktop OS such as Kali Linux, Windows, etc.
    - **Ubuntu Server OS** (ISO format).
    - **Bitnami WordPress Application Server** (OVA format).
3. **Import the Operating Systems into VirtualBox**:
    - Create three VMs within VirtualBox, each allocated for one of the above operating systems.

Ubuntu Server OS Configuration:

Step 1: Create a New Virtual Machine for Ubuntu Server

1. Launch **VirtualBox**.
2. Click **New** to create a new VM.
3. Name the VM (e.g., "Ubuntu Server").
4. Set the **Type** to **Linux** and the **Version** to **Ubuntu (64-bit)**.

5. Allocate **memory** (e.g., 2048 MB or higher depending on your system capacity).
6. Opt to **Create a virtual hard disk now** and specify a disk size (e.g., 10 GB or more).
7. Click **Create**.

Step 2: Configure Network Interfaces

To enable the Ubuntu Server to function as a router between two subnets, configure the following network interfaces:

1. Access the VM's **Settings**.
2. Select the **Network** tab.
3. **Adapter 1**:
    - Set to **Internal Network** (e.g., "intnet").
    - Set the **Adapter Type** to **PCnet-FAST III** or any supported type.
    - In Ubuntu, this adapter will usually be named **enp0s8**.
4. **Adapter 2**:
    - Enable the second adapter and set it to another **Internal Network** (e.g., "intnet1").
    - In Ubuntu, this adapter will be identified as **enp0s9**.
5. **Adapter 3**:
    - Set to **NAT** (Network Address Translation).
    - In Ubuntu, this will be named **enp0s3** and its IP address will be assigned automatically via DHCP.

Step 3: Install Ubuntu Server

1. Boot the VM and select the **Ubuntu Server ISO** as the installation disk.
2. Follow the on-screen prompts:
    - Set your **time zone** and **keyboard layout**.
    - Create a user account and assign a strong password.
    - When prompted, select the option to install **OpenSSH Server** for remote access.
3. Complete the installation process and **reboot** the VM once done.

Step 4: Configure Static IPs on the Network Interfaces

After installation, assign static IP addresses to the two internal network interfaces:

1. Log in to your **Ubuntu Server**.
2. Edit the network configuration file:

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

3. Insert the appropriate configuration for your interfaces (replace with actual IP addresses for each subnet):

4. Save and exit (Ctrl + X, then confirm with 'Yes').
5. Apply the network changes:
   sudo netplan apply
   ip a

Step 5: Enable IP Forwarding

To allow the Ubuntu Server to route traffic between the subnets:

1. Open the sysctl configuration file:

   sudo nano /etc/sysctl.conf

2. Uncomment (or add if missing) the line:

net.ipv4.ip_forward=1

3. Apply the changes:

   sudo sysctl -p

Step 6: Set Up IPTables for Routing

For the server to properly route packets between the two subnets, configure **iptables**:

1. Allow packet forwarding between interfaces:

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

2. To persist these rules across reboots:

   sudo apt install iptables-persistent
   sudo netfilter-persistent save
   sudo netfilter-persistent reload

Ubuntu Desktop OS Configuration:

Step: Install Ubuntu Desktop in a Virtual Machine

To create a VM with a GUI-based operating system, follow these steps:

1. Open **VirtualBox**.
2. Click **New** to initiate the VM creation process.
3. Specify the following details:
   - **Name**: Ubuntu Desktop.
   - **ISO**: Select the ubuntu-24.04.1-desktop-amd64.iso.
   - **Unattended Installation**: Ensure the box for "Skip Unattended Installation" is checked.
4. Click **Next** and assign the following resources:
   - **Base Memory**: 2048 MB.

- **Processors**: 2 (adjust based on your system's performance).

5. Create a **Virtual Hard Disk** with a minimum size of **25 GB**.
6. Click **Next**, verify the settings, and then click **Finish**.
7. Start the VM and select **Try or Install Ubuntu**.
8. Once booted, click **Install Ubuntu**.
9. Create a user account, complete the installation process, and restart the VM when prompted.

Step 2: Configure the Network Interfaces for Ubuntu Desktop VM

To configure the network interface for the **Ubuntu Desktop VM**:

1. **Access VM Settings**:
   - Go to the **Settings** of your Ubuntu Desktop VM within **VirtualBox**.
   - Select the **Network** tab.
2. **Network Adapter Configuration**:
   - Set **Adapter 1** to **Internal Network** (you can name it, e.g., intnet).
3. **Configure Network Interface on Ubuntu Desktop**:
   - Launch the **Ubuntu Desktop** VM and log in.
   - Navigate to **Settings** > **Network**.
   - Select **enp0s3** (your primary network interface) and click the settings gear.
   - In the **IPv4** tab, change the **IPv4 Method** to **Manual**.
   - Set the following:
     - **Address**: 192.168.21.2
     - **Netmask**: 255.255.255.0 or 24
     - **Gateway**: 192.168.21.1
   - Click **Apply** to save the changes.

4. **Reconnect the Network**:
    - Disconnect the network and then reconnect to apply the new settings.

Bitnami Web Application Configuration

Step 1: Install Bitnami Application in the VM

1. **Import Bitnami OVA File**:
    - In **VirtualBox**, go to **File** > **Import Appliance**.
    - Browse to the file **bitnami-wordpress-6.3.1-r0-debian-11-amd64.ova** and select **Open**.
    - Click **Next** and then **Finish** to complete the import process.
2. **Start the Virtual Machine**:
    - Once the appliance is imported, click **Start** to launch the Bitnami WordPress VM.
3. **Initial Login**:
    - Upon first boot, the login credentials will be displayed on the screen.
    - You will need to reset the password the first time you log in.
4. **Close the VM**:
    - Once the setup is complete, close the Bitnami application.

Step 2: Configure the Network Interfaces for Bitnami VM

1. **Access the VM Settings**:
    - Go to the **Settings** of your Bitnami WordPress VM.
    - Select the **Network** tab.
2. **Configure Network Adapter**:
    - Set **Adapter 1** to **Internal Network** (you can name it, e.g., intnet1).
3. **Configure Network Settings on Bitnami VM**:

- Open the **Bitnami WordPress VM** and log in.
- Run the following command to edit the network interfaces file:

sudo nano /etc/network/interfaces

- Add the following configuration for **eth1**:

auto eth1

iface eth1 inet static

address 192.168.121.2

netmask 255.255.255.0

gateway 192.168.121.1

- Save the file by pressing **CTRL + X**, then type **Y** to confirm, and hit **Enter**.

4. **Apply Network Configuration**:
   - Run the following commands to apply the changes:

sudo ifdown eth1 && sudo ifup eth1

sudo systemctl restart networking

   - Verify the IP address configuration:

ip a

4. Functional Test Results

Evidence that all VMs can communicate as per the design (e.g., ping results, screenshots of application access).

Screenshots for Functional Test Results:

Ubuntu Server OS:

- ✓ Network IP configuration For Ubuntu Server.

```
student@router:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qle
n 1000
    link/ether 08:00:27:17:36:79 brd ff:ff:ff:ff:ff:ff
    inet 192.168.21.1/24 brd 192.168.21.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe17:3679/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:cc:b1:6a brd ff:ff:ff:ff:ff:ff
    inet 192.168.121.1/24 brd 192.168.121.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fecc:b16a/64 scope link
       valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:28:9c:11 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s9
       valid_lft 86396sec preferred_lft 86396sec
    inet6 fd00::a00:27ff:fe28:9c11/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86398sec preferred_lft 14398sec
    inet6 fe80::a00:27ff:fe28:9c11/64 scope link
       valid_lft forever preferred_lft forever
```

- ✓ Ping Ubuntu Server to Ubuntu Desktop Using IP address 192.168.21.2.

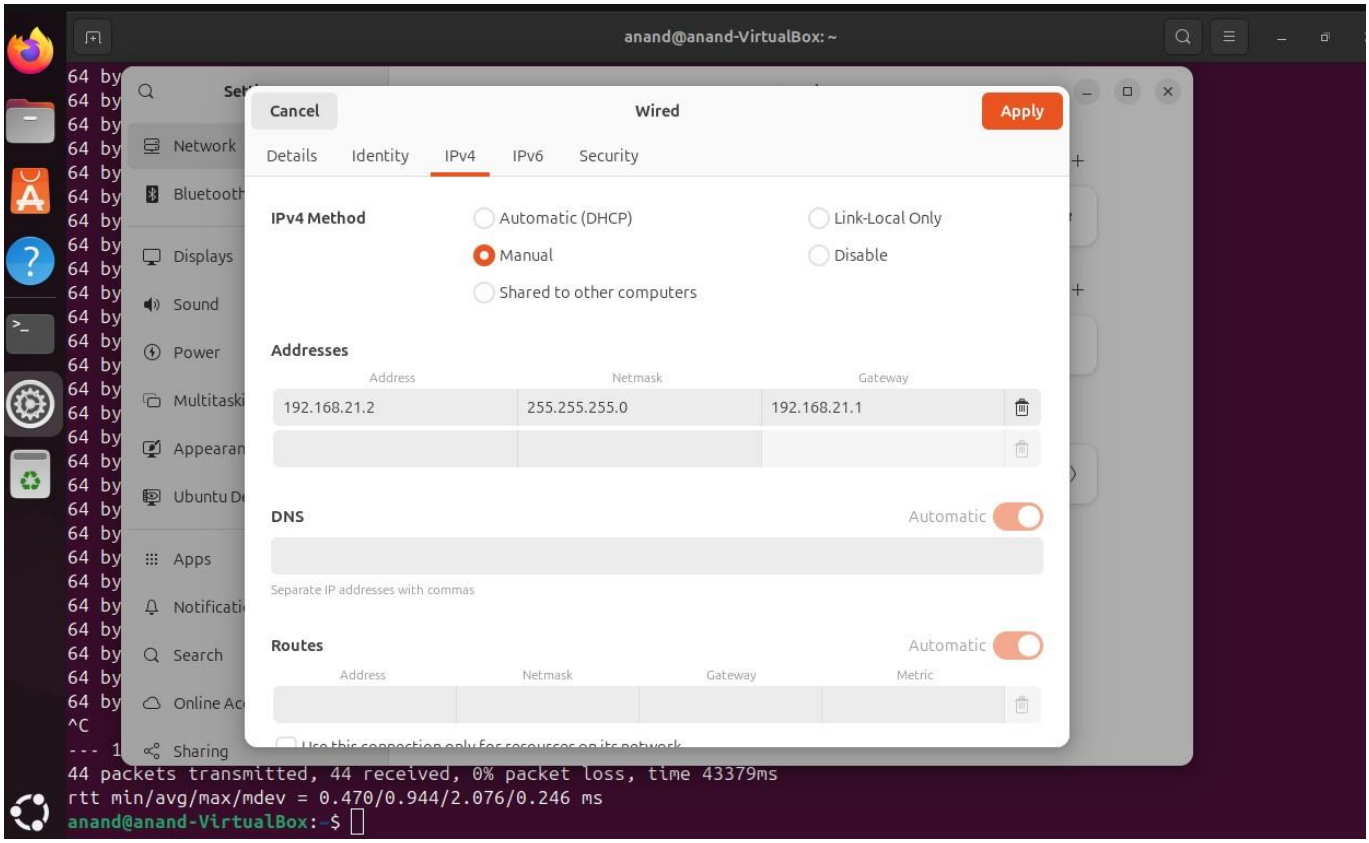
```
student@student:~$ ping 192.168.121.1
PING 192.168.121.1 (192.168.121.1) 56(84) bytes of data.
64 bytes from 192.168.121.1: icmp_seq=1 ttl=64 time=1.00 ms
64 bytes from 192.168.121.1: icmp_seq=2 ttl=64 time=0.595 ms
64 bytes from 192.168.121.1: icmp_seq=3 ttl=64 time=0.608 ms
64 bytes from 192.168.121.1: icmp_seq=4 ttl=64 time=0.751 ms
64 bytes from 192.168.121.1: icmp_seq=5 ttl=64 time=0.496 ms
64 bytes from 192.168.121.1: icmp_seq=6 ttl=64 time=0.644 ms
64 bytes from 192.168.121.1: icmp_seq=7 ttl=64 time=0.561 ms
64 bytes from 192.168.121.1: icmp_seq=8 ttl=64 time=0.555 ms
64 bytes from 192.168.121.1: icmp_seq=9 ttl=64 time=0.582 ms
64 bytes from 192.168.121.1: icmp_seq=10 ttl=64 time=0.705 ms
64 bytes from 192.168.121.1: icmp_seq=11 ttl=64 time=0.467 ms
64 bytes from 192.168.121.1: icmp_seq=12 ttl=64 time=0.611 ms
64 bytes from 192.168.121.1: icmp_seq=13 ttl=64 time=0.547 ms
64 bytes from 192.168.121.1: icmp_seq=14 ttl=64 time=0.593 ms
64 bytes from 192.168.121.1: icmp_seq=15 ttl=64 time=0.607 ms
64 bytes from 192.168.121.1: icmp_seq=16 ttl=64 time=0.433 ms
64 bytes from 192.168.121.1: icmp_seq=17 ttl=64 time=0.530 ms
64 bytes from 192.168.121.1: icmp_seq=18 ttl=64 time=0.564 ms
^C
--- 192.168.121.1 ping statistics ---
18 packets transmitted, 18 received, 0% packet loss, time 17017ms
rtt min/avg/max/mdev = 0.433/0.602/1.000/0.124 ms
student@student:~$
```

- ✓ Ping Ubuntu Server to Bitnami Application Server Using IP address 192.168.121.2.

```
student@router:~$ ping 192.168.121.2
PING 192.168.121.2 (192.168.121.2) 56(84) bytes of data.
64 bytes from 192.168.121.2: icmp_seq=1 ttl=64 time=0.785 ms
64 bytes from 192.168.121.2: icmp_seq=2 ttl=64 time=0.456 ms
64 bytes from 192.168.121.2: icmp_seq=3 ttl=64 time=0.626 ms
64 bytes from 192.168.121.2: icmp_seq=4 ttl=64 time=0.434 ms
64 bytes from 192.168.121.2: icmp_seq=5 ttl=64 time=0.556 ms
64 bytes from 192.168.121.2: icmp_seq=6 ttl=64 time=0.649 ms
64 bytes from 192.168.121.2: icmp_seq=7 ttl=64 time=0.786 ms
64 bytes from 192.168.121.2: icmp_seq=8 ttl=64 time=0.488 ms
64 bytes from 192.168.121.2: icmp_seq=9 ttl=64 time=0.434 ms
64 bytes from 192.168.121.2: icmp_seq=10 ttl=64 time=0.576 ms
64 bytes from 192.168.121.2: icmp_seq=11 ttl=64 time=0.512 ms
64 bytes from 192.168.121.2: icmp_seq=12 ttl=64 time=0.876 ms
64 bytes from 192.168.121.2: icmp_seq=13 ttl=64 time=0.566 ms
64 bytes from 192.168.121.2: icmp_seq=14 ttl=64 time=0.663 ms
64 bytes from 192.168.121.2: icmp_seq=15 ttl=64 time=0.801 ms
64 bytes from 192.168.121.2: icmp_seq=16 ttl=64 time=0.602 ms
64 bytes from 192.168.121.2: icmp_seq=17 ttl=64 time=0.414 ms
^C
--- 192.168.121.2 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 17293ms
rtt min/avg/max/mdev = 0.414/0.601/0.876/0.138 ms
student@router:~$
```

Ubuntu Desktop OS:

- ✓ Network IP configuration For Ubuntu Desktop.



- ✓ Ping Ubuntu Desktop to Ubuntu Server Using IP address 192.168.21.1.

```
anand@anand-VirtualBox:~$ ping 192.168.21.1
PING 192.168.21.1 (192.168.21.1) 56(84) bytes of data.
64 bytes from 192.168.21.1: icmp_seq=1 ttl=64 time=1.14 ms
64 bytes from 192.168.21.1: icmp_seq=2 ttl=64 time=1.15 ms
64 bytes from 192.168.21.1: icmp_seq=3 ttl=64 time=0.908 ms
64 bytes from 192.168.21.1: icmp_seq=4 ttl=64 time=0.726 ms
64 bytes from 192.168.21.1: icmp_seq=5 ttl=64 time=1.09 ms
64 bytes from 192.168.21.1: icmp_seq=6 ttl=64 time=0.720 ms
64 bytes from 192.168.21.1: icmp_seq=7 ttl=64 time=1.15 ms
64 bytes from 192.168.21.1: icmp_seq=8 ttl=64 time=1.05 ms
64 bytes from 192.168.21.1: icmp_seq=9 ttl=64 time=0.978 ms
64 bytes from 192.168.21.1: icmp_seq=10 ttl=64 time=0.715 ms
64 bytes from 192.168.21.1: icmp_seq=11 ttl=64 time=2.08 ms
64 bytes from 192.168.21.1: icmp_seq=12 ttl=64 time=1.08 ms
64 bytes from 192.168.21.1: icmp_seq=13 ttl=64 time=0.970 ms
64 bytes from 192.168.21.1: icmp_seq=14 ttl=64 time=0.471 ms
64 bytes from 192.168.21.1: icmp_seq=15 ttl=64 time=1.03 ms
64 bytes from 192.168.21.1: icmp_seq=16 ttl=64 time=0.863 ms
64 bytes from 192.168.21.1: icmp_seq=17 ttl=64 time=0.878 ms
64 bytes from 192.168.21.1: icmp_seq=18 ttl=64 time=0.941 ms
64 bytes from 192.168.21.1: icmp_seq=19 ttl=64 time=0.862 ms
64 bytes from 192.168.21.1: icmp_seq=20 ttl=64 time=0.849 ms
64 bytes from 192.168.21.1: icmp_seq=21 ttl=64 time=1.10 ms
64 bytes from 192.168.21.1: icmp_seq=22 ttl=64 time=1.14 ms
64 bytes from 192.168.21.1: icmp_seq=23 ttl=64 time=0.849 ms
```

Bitnami Application Server:

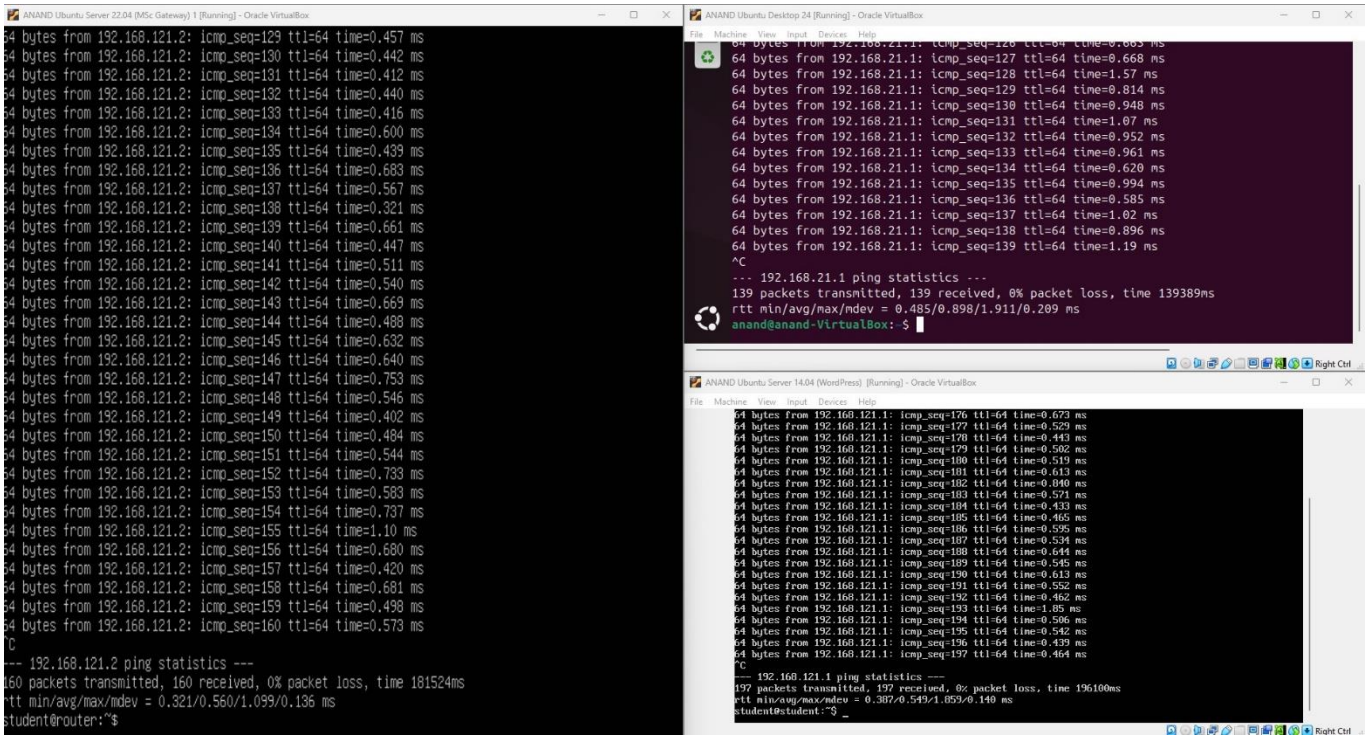
- ✓ Network IP configuration For Bitnami Application Server.

```
student@student:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 100
0
    link/ether 08:00:27:f3:be:98 brd ff:ff:ff:ff:ff:ff
    inet 192.168.123.20/24 brd 192.168.123.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fef3:be98/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 100
0
    link/ether 08:00:27:ef:e4:c3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.121.2/24 brd 192.168.121.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 2a02:6b67:d457:9500:a00:27ff:feef:e4c3/64 scope global dynamic
       valid_lft 647083sec preferred_lft 560683sec
    inet6 fe80::a00:27ff:feef:e4c3/64 scope link
       valid_lft forever preferred_lft forever
student@student:~$
```

- ✓ Ping Bitnami Application Server to Ubuntu Server Using IP address 192.168.121.1.

```
student@router:~$ ping 192.168.121.2
PING 192.168.121.2 (192.168.121.2) 56(84) bytes of data.
64 bytes from 192.168.121.2: icmp_seq=1 ttl=64 time=0.785 ms
64 bytes from 192.168.121.2: icmp_seq=2 ttl=64 time=0.456 ms
64 bytes from 192.168.121.2: icmp_seq=3 ttl=64 time=0.626 ms
64 bytes from 192.168.121.2: icmp_seq=4 ttl=64 time=0.434 ms
64 bytes from 192.168.121.2: icmp_seq=5 ttl=64 time=0.556 ms
64 bytes from 192.168.121.2: icmp_seq=6 ttl=64 time=0.649 ms
64 bytes from 192.168.121.2: icmp_seq=7 ttl=64 time=0.786 ms
64 bytes from 192.168.121.2: icmp_seq=8 ttl=64 time=0.488 ms
64 bytes from 192.168.121.2: icmp_seq=9 ttl=64 time=0.434 ms
64 bytes from 192.168.121.2: icmp_seq=10 ttl=64 time=0.576 ms
64 bytes from 192.168.121.2: icmp_seq=11 ttl=64 time=0.512 ms
64 bytes from 192.168.121.2: icmp_seq=12 ttl=64 time=0.876 ms
64 bytes from 192.168.121.2: icmp_seq=13 ttl=64 time=0.566 ms
64 bytes from 192.168.121.2: icmp_seq=14 ttl=64 time=0.663 ms
64 bytes from 192.168.121.2: icmp_seq=15 ttl=64 time=0.801 ms
64 bytes from 192.168.121.2: icmp_seq=16 ttl=64 time=0.602 ms
64 bytes from 192.168.121.2: icmp_seq=17 ttl=64 time=0.414 ms
^C
--- 192.168.121.2 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 17293ms
rtt min/avg/max/mdev = 0.414/0.601/0.876/0.138 ms
student@router:~$
```

PING OF ALL MACHINE



The video demonstration of machine pings

https://drive.google.com/drive/u/1/folders/1uveevAvgy_9FfyIofCipHbK-sH8-rBcZ

ALL TOGETHER PING.mp4