Linux Router

This report presents a comprehensive guide to establishing a Linux router utilizing VirtualBox alongside three Ubuntu virtual machines. The primary function of the router is to serve as a bridge facilitating communication between the client and server virtual machines. While internet connectivity might not be established within this setup, the focus remains on fostering communication among the networked machines. Alongside the detailed steps for configuration, the report emphasizes transparency by providing information on the tools utilized and their sourcing, ensuring clarity and reproducibility of the setup process. Despite the absence of direct internet access, the successful transmission of packets between the interconnected virtual machines underscores the efficacy of the configured network environment.

Step 1: Setting up Virtual Machines in VirtualBox:

1**. Download, Install VirtualBox and Ubuntu Server**:

-Download the VirtualBox installer from the official website:

Source: https://www.virtualbox.org/wiki/Downloads

-Download Ubuntu Server from the official website:

Source: https://ubuntu.com/download/server

-Follow the installation instructions for your operating system.

2. **Create Virtual Machines:**

-Open VirtualBox and click on "New" to create a new virtual machine.

-Choose a name for the virtual machine (e.g., "Router").

-Select "Linux" as the type and "Ubuntu (64-bit)" as the version.

-Allocate RAM and create a virtual hard disk.

-Repeat the process to create two more virtual machines for the client and server.

3. **Configure Network Adapters:**

-Select each virtual machine, then go to "Settings" > "Network".

-For the router VM:

i). Adapter 1: Select "NAT" to provide internet access.

ii). Adapter 2: Choose "Internal Network" and name it (e.g., "internal_network").

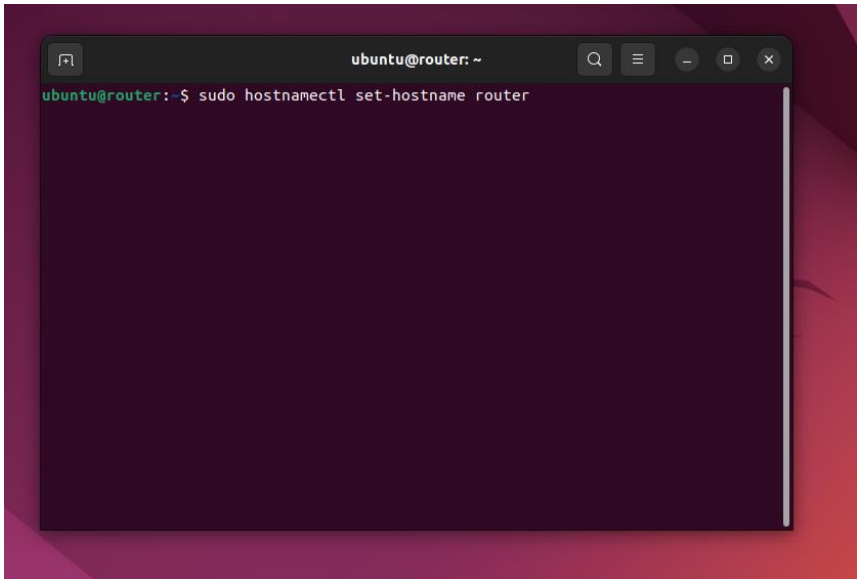
-For the client and server VMs:

Adapter 1: Choose "Internal Network" and select the same network name as the router's second adapter ("internal_network").

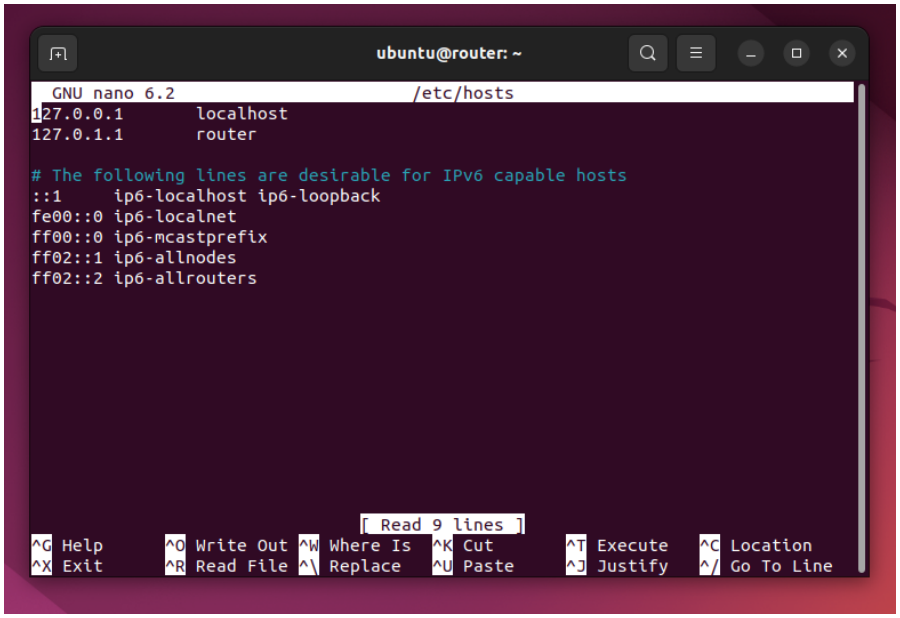
Step 2: Installation and Configuration of Router:

-The first step in setting up our router is to change the hostname of our machine for easy identification, in our case, "router". This step is optional.

Using this command: **sudo hostnamectl set-hostname router**



Additionally, edit the hosts file using the following command: sudo nano /etc/hosts, this will launch a text editor. Rename the second line 127.0.0.1 to router. After making this change, save the file and exit the editor. Finally, restart your machine so that the new name can take effect.

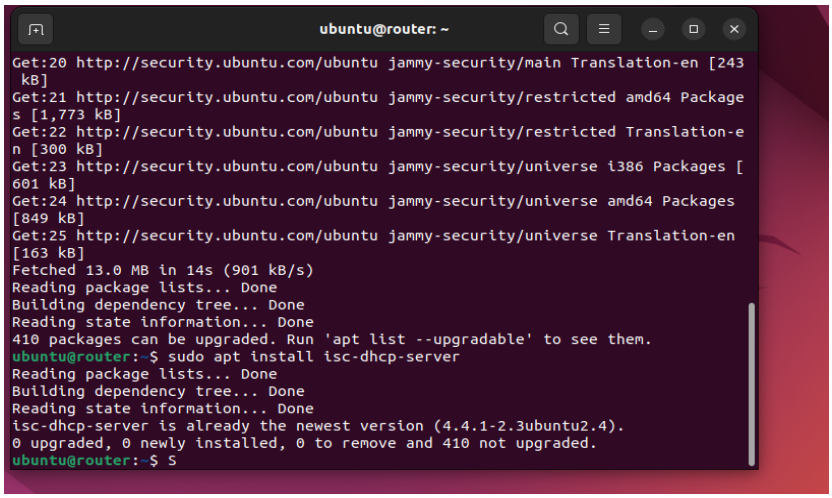


Configuring DHCP Server in our router machine

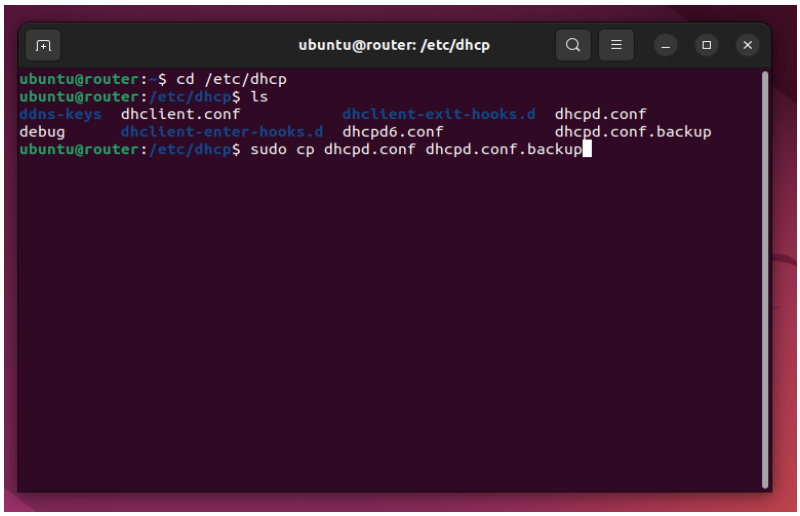
Dynamic Host Configuration Protocol (DHCP): enables centralized IP address management within a network. When machines are added to a network, they send DHCP requests seeking configuration details such as IP address, subnet mask, gateway, and DNS server from any available DHCP server. This automation simplifies network setup and administration, ensuring devices can seamlessly connect and communicate within the network infrastructure.

We need to update packages and install DHCP server: To set up a DHCP server, first, update your package list using: `sudo apt update`

Then, install the ISC DHCP server package with: `sudo apt install isc-dhcp-server`

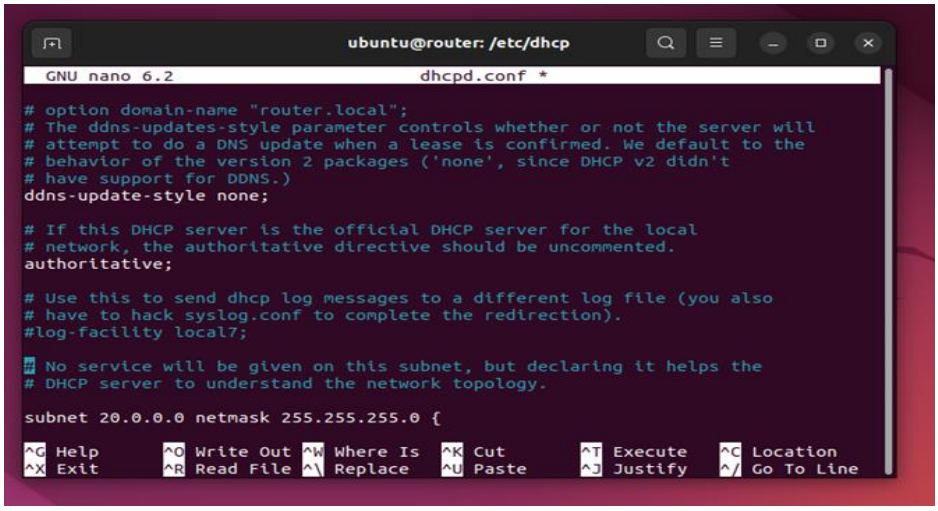


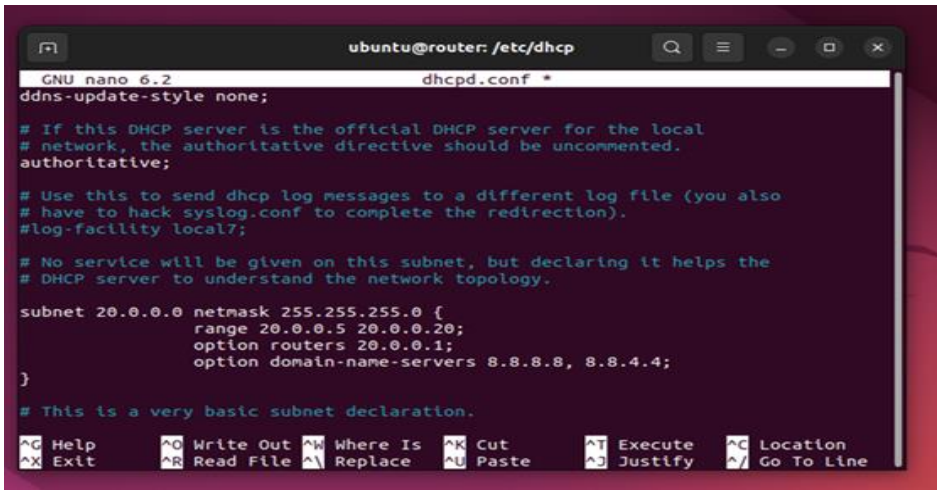
After successfully installing dhcp-server we need to make few changes in the dhcpd.conf file which we can find by navigating to this folder **/etc/dhcp** using this command **cd /etc/dhcp** and copy **dhcpd.conf** file to a backup file using this command `sudo cp dhcpd.conf dhcpd.conf.backup` in case we make errors we can quickly look at backup file for reference as we are goi to edit **dhcpd.conf**.



Modify /etc/dhcp/dhcpd.conf to configure the network information you want to serve:

```
ddns-update-style none;
option domain-name-servers 8.8.8.8, 8.8.4.4;
default-lease-time 600;
max-lease-time 7200;
authoritative;
subnet 20.0.0.0 netmask 255.255.255.0 {
     range 20.0.0.5 20.0.0.10;
     option routers 20.0.0.1;}
```





In this configuration:

ddns-update-style none; : Disables Dynamic DNS updates.

option domain-name-servers 8.8.8.8, 8.8.4.4;: Specifies the DNS servers as Google's public DNS servers.

default-lease-time 600;: Sets the default lease time for IP addresses to 600 seconds (10 minutes).

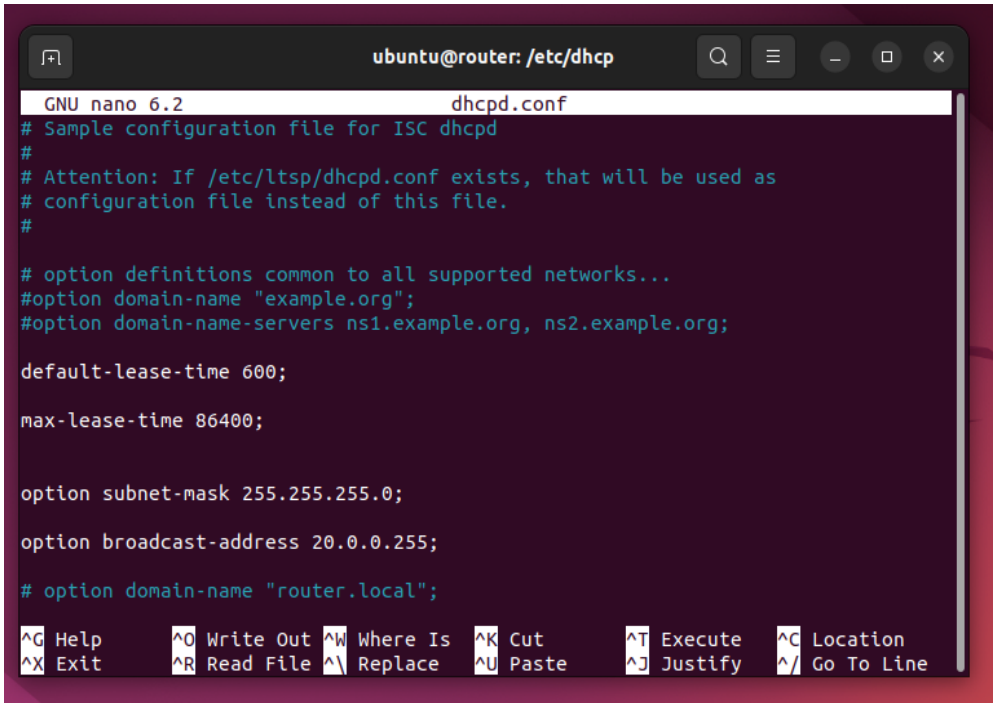
max-lease-time 7200;: Sets the maximum lease time for IP addresses to 7200 seconds (2 hours).

authoritative;: Declares this DHCP server as authoritative for the specified subnet.

ubnet 20.0.0.0 netmask 255.255.255.0 { ... }: Defines the subnet and its associated configuration.

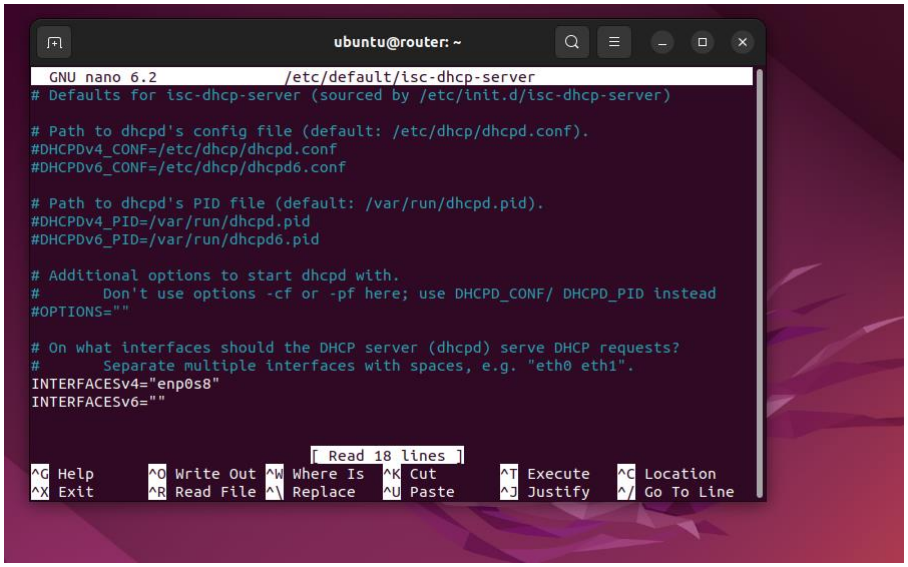
range 20.0.0.5 20.0.0.10;: Specifies the range of IP addresses to be assigned by the DHCP server.

option routers 20.0.0.1;: Specifies the default gateway/router for clients on this subnet.



Modify `/etc/default/isc-dhcp-server` to add the interface **enp0s8** which you should serve requests on: # `sudo nano /etc/default/isc-dhcp-server`

Interfaces can be located on this command # `ifconfig`

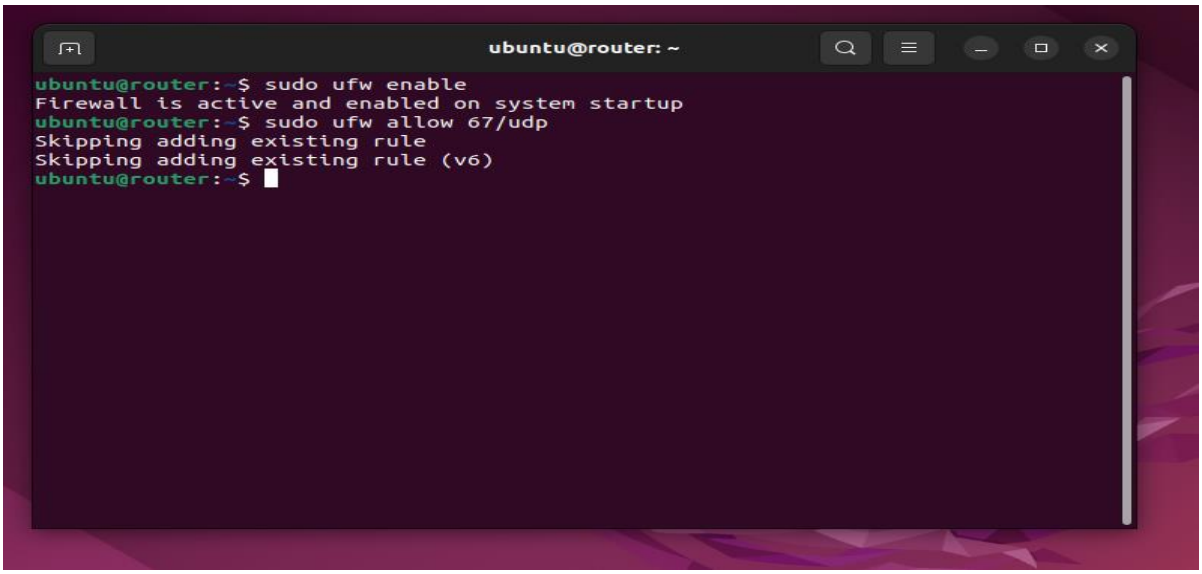


To set up a firewall using iptables, you can follow these steps:

-Enable the firewall using this command : `sudo ufw enable`

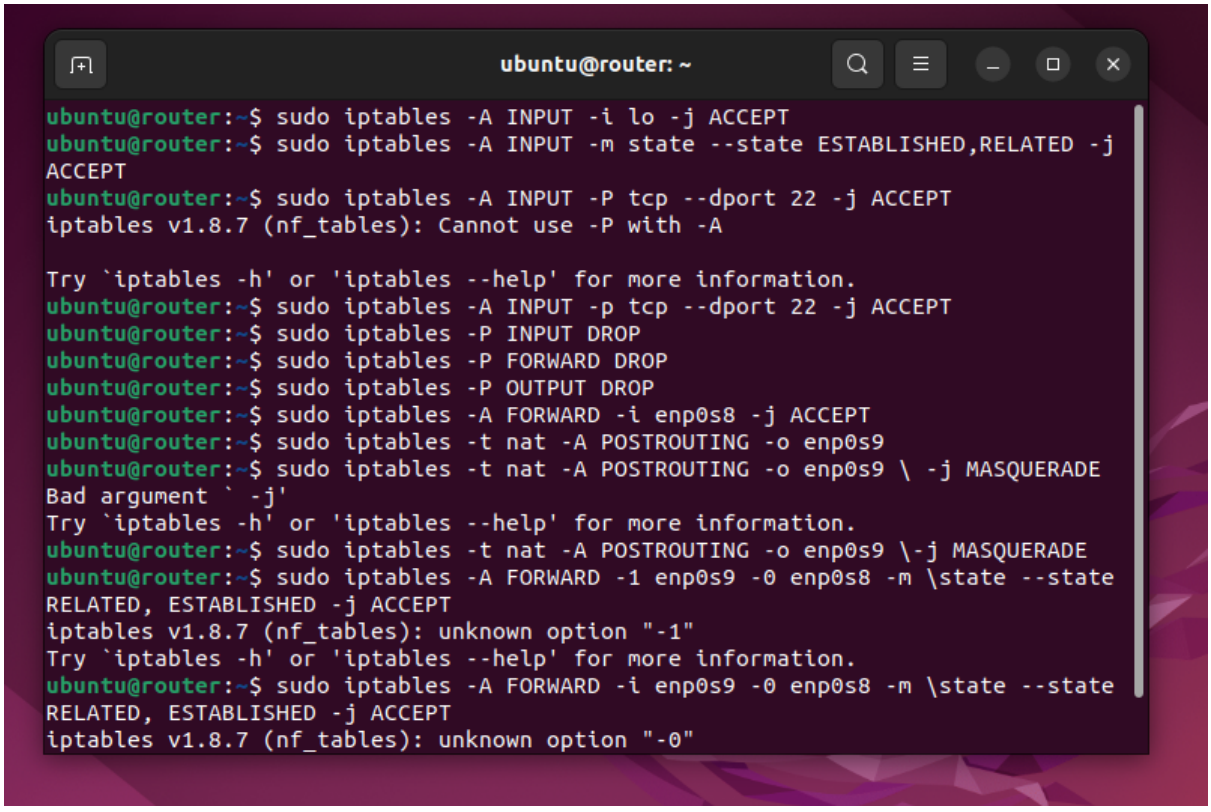
-Allow port 67 UDP, which will be used by DHCP, using this command : `sudo ufw allow 67/udp`

These commands will activate the firewall and permit incoming UDP traffic on port 67, which is essential for DHCP functionality.



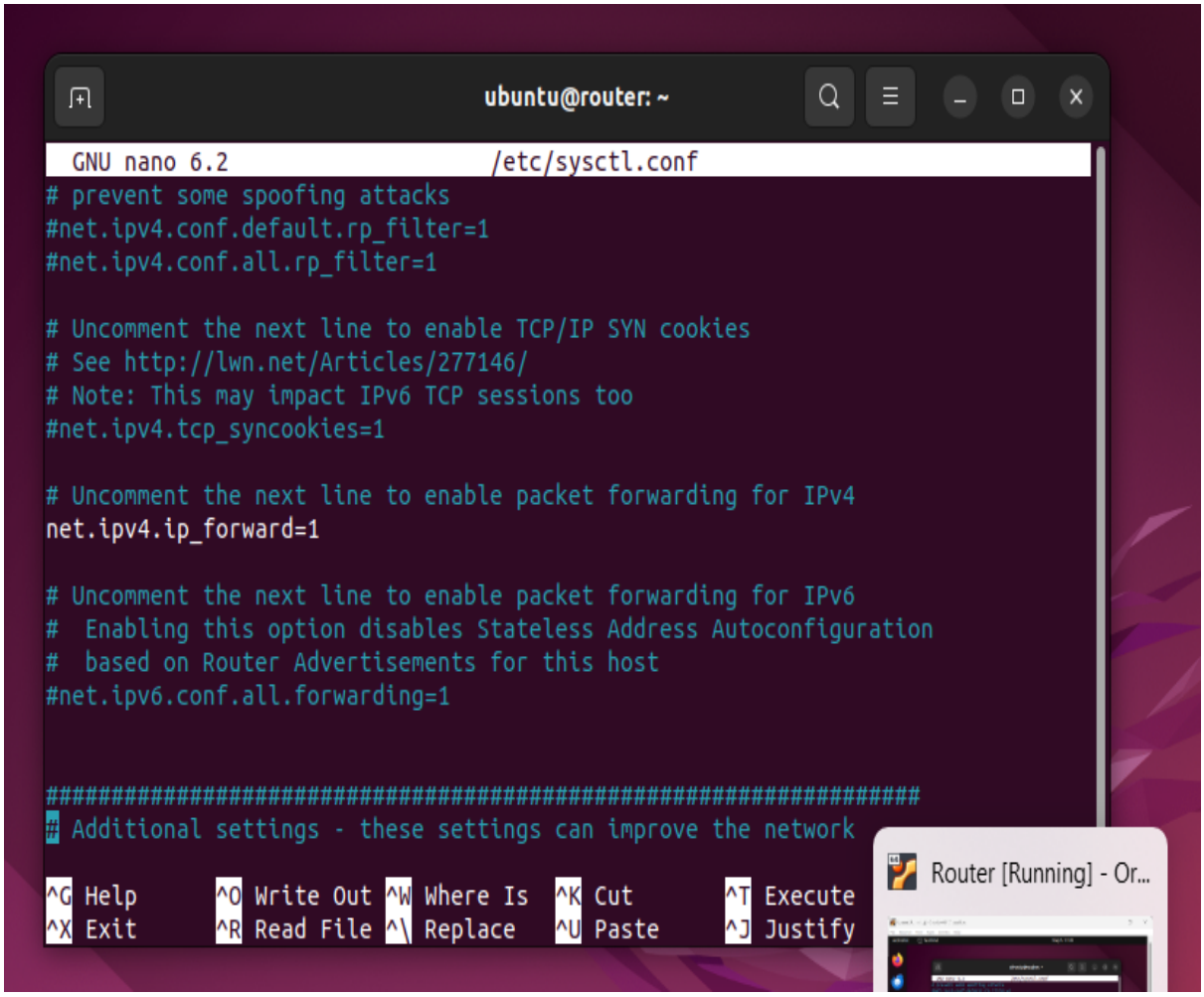
A properly configured firewall should be configured in a default deny configuration with specific allows (Whitelist) for what you want to accept.

```
# sudo iptables -A INPUT -i lo -j ACCEPT

# sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT

# sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

# sudo iptables -P INPUT DROP

# sudo iptables -P FORWARD DROP

# sudo iptables -P OUTPUT ACCEPT

# sudo iptables -A FORWARD -i eth0 -j ACCEPT

# sudo iptables -t nat -A POSTROUTING -o eth2 \-j MASQUERADE

# sudo iptables -A FORWARD -i eth2 -o eth0 -m \state --state
RELATED,ESTABLISHED -j ACCEPT

# sudo iptables -A FORWARD -i eth0 -j ACCEPT
```



IP Forwarding using net.ipv4.ip_forward

We need to edit the **/etc/sysctl.conf** file using **# sudo nano /etc/sysctl.conf,** to make sure the new setting survives a reboot.
so we need to uncomment the line # **net.ipv4.ip_forward = 0** and change its value from **net.ipv4.ip_forward = 0** to **net.ipv4.ip_forward = 1**

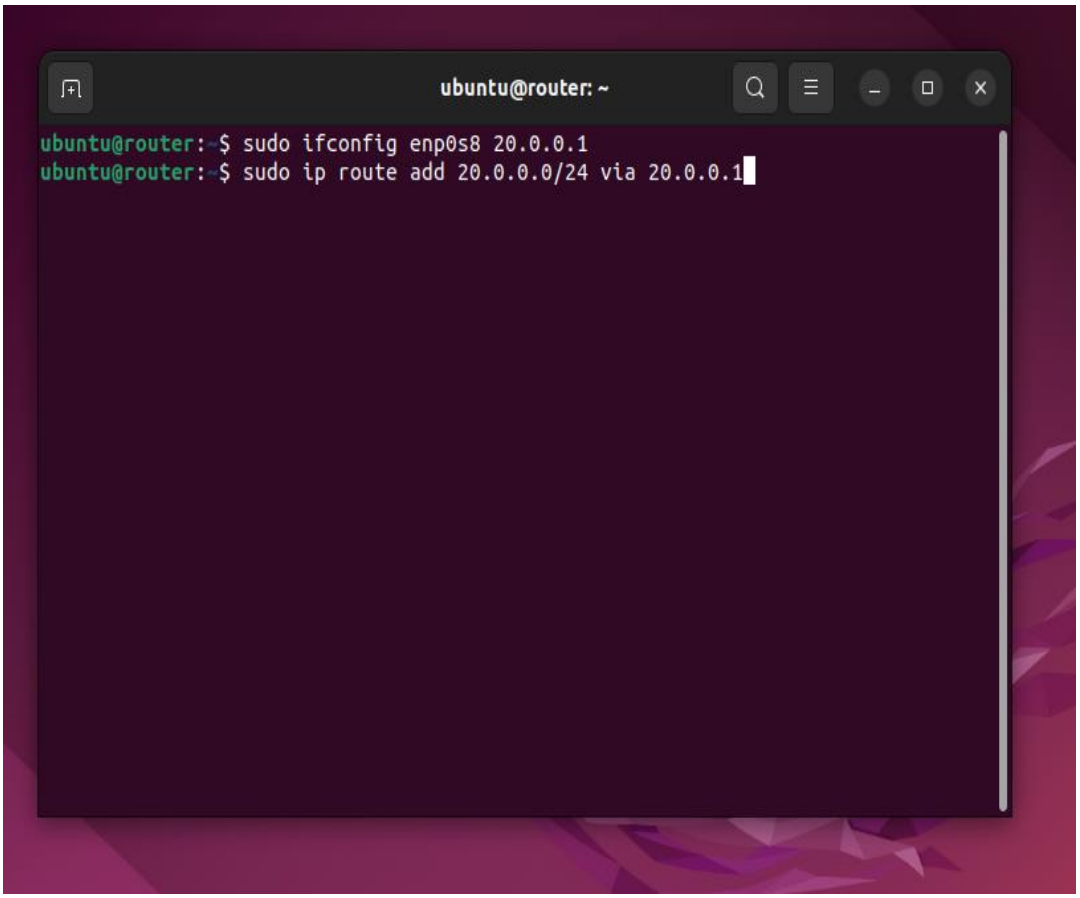


Give the interface **enp0s8 IP address** that matches the router option defined in the dhcpd.conf file and add the route via the IP address given to the interface **enp0s8** using commands.

```
sudo ifconfig enp0s8
```

```
sudo ip route add 20.0.0.0/24 via 20.0.0.1
```

Alternatively, we can add static address to interface enp0s8 to save us a lot of time to run these commands each time the router reboots.

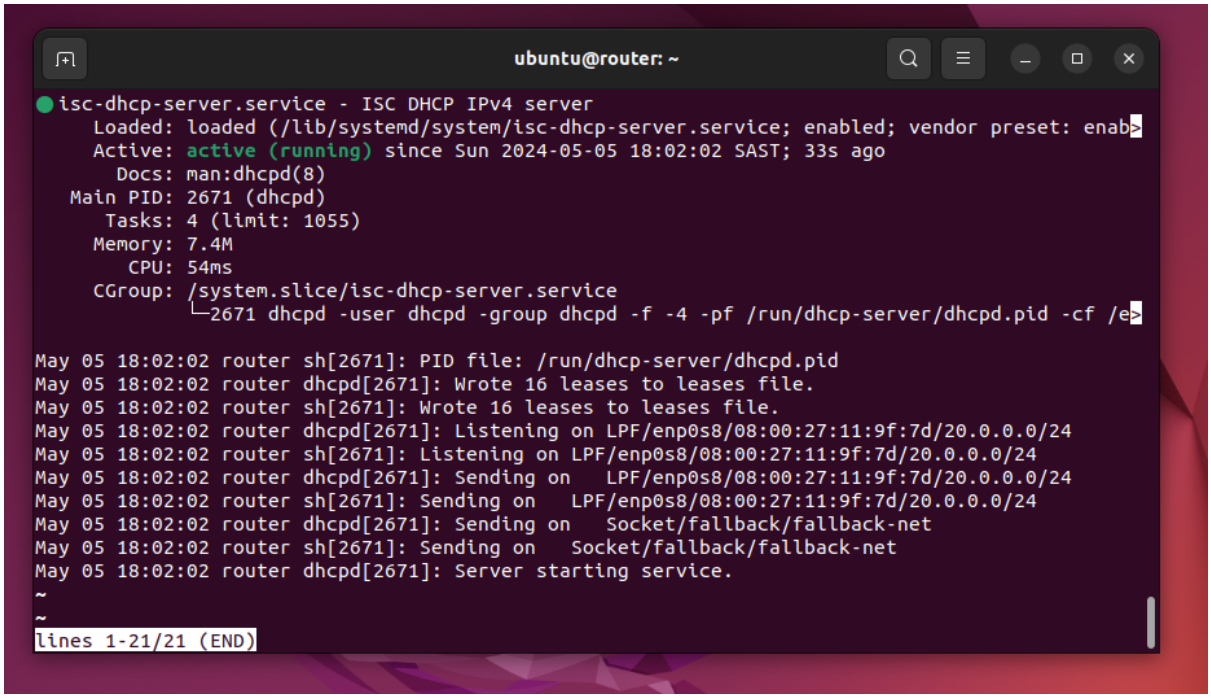


-Restart NetworkManager and DHCP server, execute the following commands:

```
sudo systemctl restart NetworkManager
```

```
sudo systemctl restart isc-dhcp-server
```

These commands will restart both services, ensuring that any changes made to the network configuration or DHCP server settings take effect.

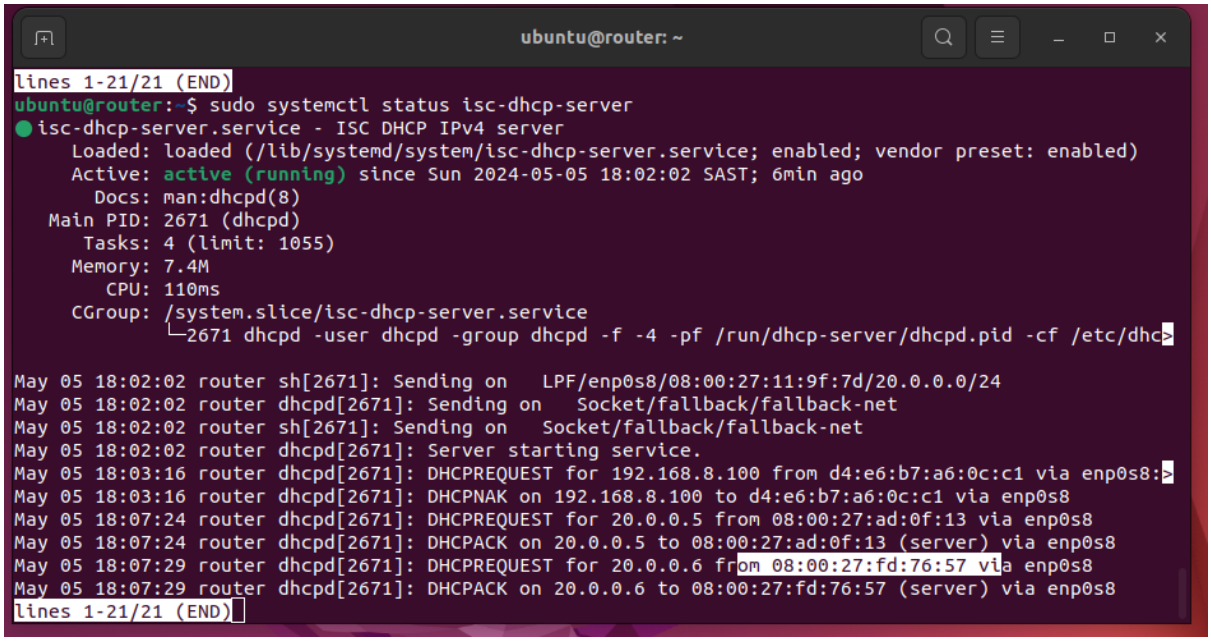


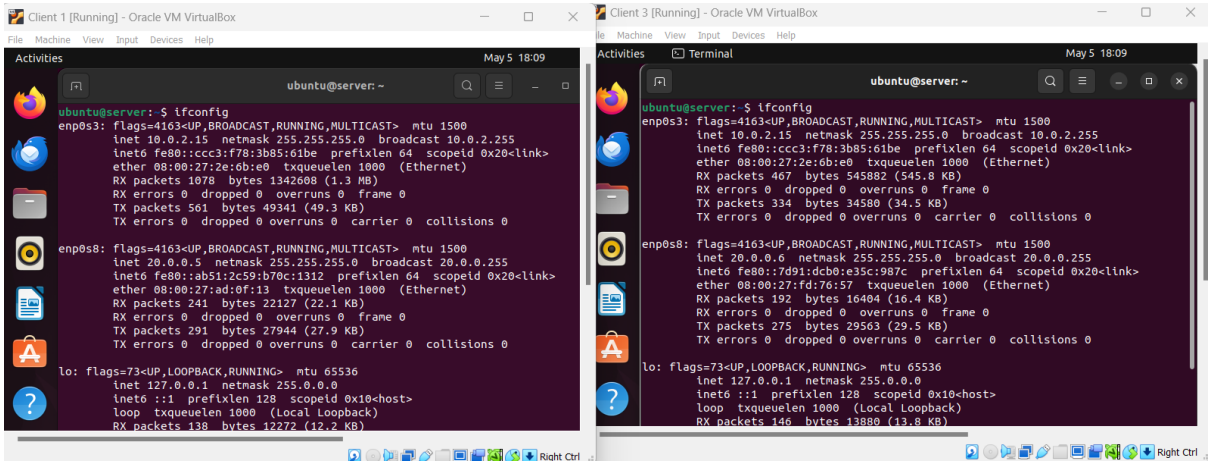
-To check if the ISC DHCP server is active, you can use the following command:

```
Sudo systemctl status isc-dhcp-server
```

Executing this command will provide information about the current status of the ISC DHCP server service, including whether it is active (running) or inactive (stopped). It also displays additional details such as the service's process ID (PID), memory usage, and recent logs.

-Now that the DHCP server is up and running, we can proceed to deploy client and server machines for testing our DHCP configuration settings and packet transmission. It's imperative to connect the client machines to the same network as the router. The DHCP server has successfully allocated IP addresses 20.0.0.5 and 20.0.0.6 to the client machines, demonstrating its functionality as intended.

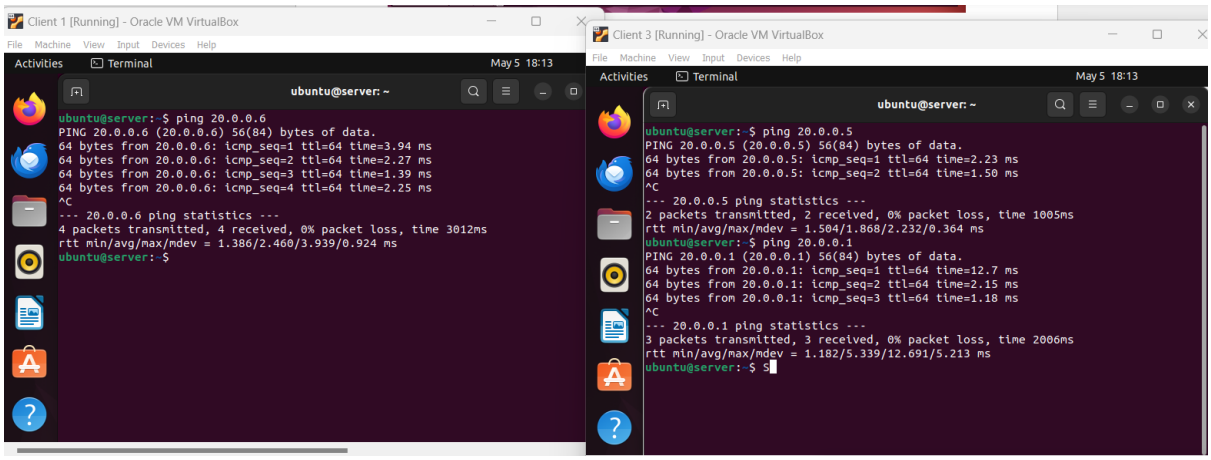




We can test the communication between the clients by executing this command:

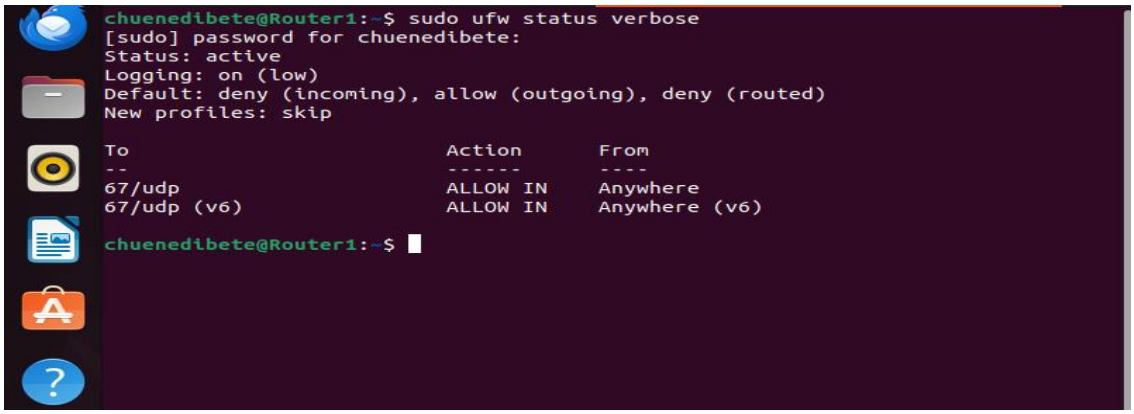
Ping 20.0.0.6 for client 2 and Ping 20.0.0.5 for client 1

clients can communicate with each other and router.

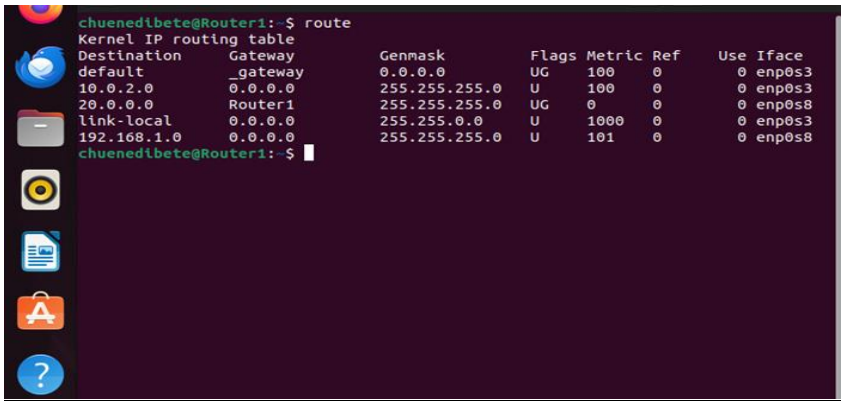


To check the status of the firewall and view detailed information about its rules, including whether it's active or inactive, use the command: sudo ufw status verbose.

The firewall is active according to this output:



To check the routing table, we use this command: route.



Challenges Faced

The initial stages of resource management presented considerable difficulties because it was difficult to manage host machine resources like CPU memory and storage to guarantee sufficient performance for every virtual machine particularly when running multiple VMs at once. Furthermore, the computers slowness made these issues worse making resource allocation and optimization tasks even more difficult. To ensure that there are no connectivity problems network interfaces and routing tables must be carefully configured. In order to successfully enable IP forwarding initial routing errors must be troubleshooted.

It was difficult to make sure that the network interfaces on the router and the client/server virtual machines were configured correctly especially when there were several adapters and different network kinds involved. It was difficult to set up firewall rules on the router to permit or prohibit types of traffic especially when taking both incoming and outgoing traffic into account. It was difficult to configure the routers DHCP (Dynamic Host Configuration Protocol) to assign IP addresses to client devices automatically lease times and IP address ranges needed to be carefully considered. Specifically, when integrating with external DNS servers it was difficult to configure DNS (Domain Name System) resolution on the router to translate domain names to IP addresses and vice versa. It took careful planning and configuration to ensure network security including encrypting sensitive data and guarding against different kinds of cyber threats. Several settings and configurations needed to be adjusted in order to maximize throughput and minimize latency while optimizing the routers overall performance.

During setup and configuration there were difficulties dealing with incompatibilities between various software versions hardware configurations and virtualization platforms (like VirtualBox). It was very difficult and time-consuming to fill in the knowledge and documentation gaps regarding networking principles Ubuntu Server administration and VirtualBox configuration.

Lessons Learned

In the process of setting up the router, I learned that having a deeper understanding of IP addressing and routing is crucial. Troubleshooting skills became invaluable as I encountered various configuration issues that needed prompt resolution. Documenting every step taken, including commands executed and configurations made, proved essential for future reference and troubleshooting. I also realized the importance of implementing version control systems like Git for configuration files and regularly backing up virtual machine snapshots to prevent data loss. Managing time effectively was a challenge, especially when balancing tasks like planning, implementation, testing, and troubleshooting.

Collaboration with other members and seeking assistance from peers provided valuable insights and support throughout the project. Embracing a growth mindset and continuously learning about the latest technologies and best practices were essential for success. Ethical considerations, such as privacy, security, and data protection, were always at the forefront of my mind during the configuration process.

I had to ensure responsible and professional conduct in every aspect of the project. Being resourceful and adaptable in solving problems and overcoming challenges was crucial, especially when encountering unexpected issues. seeking feedback from peers and reflecting on both successes and failures helped me grow both personally and professionally. It allowed me to learn from my experiences and continuously improve my skills. Overall, the process of creating the router taught me valuable lessons in networking, troubleshooting, time management, collaboration, continuous learning, ethical considerations, and resilience.

Recommendations for Improvement

Based on the experience gained during this project, I suggest several improvements for future router installation and configuration efforts. First, developing a standardized configuration process can ensure consistency and maintain a high level of security and performance across the organization. This entails establishing a well-defined, step-by-step approach to router configuration. Secondly, investing in comprehensive training for network administrators on router configuration best practices is crucial. Hands-on experience with various router models and features enhances their skills and confidence in managing network infrastructure effectively.

Addressing resource limitations proactively is another key aspect. This involves optimizing configurations, minimizing resource-intensive processes, and considering hardware upgrades or replacements where feasible to improve overall system performance and efficiency. implementing collaborative learning platforms or online forums where students can share experiences, ask questions, and collaborate on problem-solving fosters peer-to-peer learning and creates a supportive learning community. Integrating troubleshooting resources directly into documentation, such as troubleshooting guides, flowcharts, and common error solutions, empowers students to diagnose and resolve issues independently, promoting self-directed learning. Improving the quality and accessibility of documentation by incorporating multimedia elements like video tutorials, interactive guides, and annotated screenshots caters to diverse learning styles and facilitates understanding for students with varying levels of expertise.

Expanding the scope of router installation and configuration projects to include a broader range of scenarios and challenges, such as multi-site deployments or integration with cloud services, exposes students to real-world complexities and prepares them for diverse networking environments. And establishing feedback mechanisms, such as surveys or feedback forms, to gather input from students about their experiences with router installation and configuration projects, helps identify areas for improvement and refine future project iterations. This continuous feedback loop ensures that the learning process remains dynamic and responsive to student needs.