

Islamic Azad University

Network Automation in Action: A Virtual Lab Approach

by

Ghazal Fayaztorshizi

A Bachelor’ s Thesis

Advisor: Dr. Mojtaba Asgari

Islamic Azad University – Mashhad Branch

Winter 2024

Table of Contents

[1. Introduction](#introduction)

[1.1 Background and Motivation](#Background)

[1.2 Objectives](#Objectives)

[1.3 Significance of the Study](#Significance)

[2. Phase 1](#Phase1)

[2. Network Design and Planning](#NetworkDesign)

[2.1 Proposed Network Topology](#ProposedNetwork)

[2.2 Components and Their Roles](#Components)

[2.2.1 Ubuntu Router](#UbuntuRouter)

[2.2.2 Ubuntu Client](#UbuntuClient)

[2.2.3 Windows Server (DNS Server)](#WindowsServer)

[2.2.4 Windows 10 Client](#Windows10Client)

[2.3 Virtual Switches and LAN Segments in VMware](#VirtualSwitches)

[2.4 IP Addressing Scheme](#IPAddressingScheme)

[2.5 Security Considerations](#SecurityConsiderations)

[3. Implementation Methodology](#ImplementationMethodology)

[3.1 Setting Up Virtual Switches (LAN Segments) in VMware](#SettingUpVirtualSwitches)

[3.2 Configuring the Ubuntu Router](#ConfiguringtheUbuntuRouter)

[3.2.1 Enabling IP Forwarding](#EnablingIPForwarding)

[3.2.2 Assigning Static IPs to Router Interfaces](#AssigningStaticIPstoRouterInterfaces)

[3.2.3 Installing and Configuring DHCP Server](#InstallingandConfiguringDHCP)

[3.2.4 Configuring NAT](#ConfiguringNAT)

[3.3 Configuring Windows Server as DNS Server](#ConfiguringDNS)

[3.3.1 Installing DNS Server Role](#InstallingDNS)

[3.3.2 Configuring DNS Zones and Records](#DNSZonesRecords)

[3.3.3 Assigning Static IP to Windows Server](#StaticIPWindowsServer)

[3.4 Configuring Clients to Obtain IP via DHCP](#ConfiguringClientsDHCP)

[3.4.1 Configuring Windows 10 Client](#ConfiguringWindows10Client)

[3.4.2 Configuring Ubuntu Client](#ConfiguringUbuntuClient)

[3.5 Configuring Clients to Use Windows Server DNS](#ConfiguringClientstoUseWindows)

[3.6 Configuring Windows Firewall to Allow ICMP (Ping) Requests](#Allow)

[4. Testing and Validation](#TestingandValidation)

[4.1 Connectivity Testing Between Clients](#ConnectivityTesting)

[4.2 DHCP Functionality Verification](#DHCPFunctionalityVerification)

[4.3 DNS Resolution Testing](#DNSResolutionTesting)

[4.4 NAT and Internet Access Testing](#NATandInternetAccessTesting)

[5. Results and Discussion](#ResultsandDiscussion1)

[5.1 Evaluation of DHCP and DNS Services](#EvaluationDHCPandDNS)

[5.2 Challenges and Troubleshooting](#ChallengesandTroubleshooting1)

[5.3 Security Implications](#SecurityImplications1)

[6. Conclusion](#Conclusion1)

[6.1 Summary of Findings](#SummaryFindings1)

[6.2 Achievement of Objectives](#AchievementofObjectives1)

[6.3 Recommendations for Future Work](#RecommendationsforFutureWork1)

[7. References](#References1)

**Introduction**

**1.1 Background and Motivation**

In pursuing a better understanding of network structures and how different services work, the implementation of network components often calls for real-world laboratories. However, the lack of resources raises a crucial question: **"How can we simulate a local network?"** With the help of virtualization, we are able to simulate an actual network without worrying about physical resources. Virtualization provides a flexible and cost-effective platform to experiment with network configurations, allowing us to delve deep into networking concepts in a controlled environment.

After understanding the basics about networks, we are faced with another challenge. With the rapid growth of networks, **how can we manage them efficiently?** Is it possible to add all the changes one by one manually? This scenario calls for the **automation of networks**. Automation not only streamlines the management process but also reduces the potential for human error, enhances scalability, and improves overall network reliability.

This project is mainly divided into two phases:

1. **Implementation of the Network**
2. **Automation of the Network**

**1.2 Objectives**

* **Simulate a Local Network Using Virtualization:**
  + Utilize VMware to create a virtual LAN environment.
  + Implement an Ubuntu-based router, clients, and servers.
  + Configure network services like DHCP, DNS and NAT.
* **Explore Network Automation Techniques:**
  + Automate network configurations using scripts.
  + Demonstrate the efficiency gains from automating repetitive tasks.
  + Assess how automation impacts network scalability and management.

**1.3 Significance of the Study**

Understanding how to simulate and manage networks virtually is essential in today's technology-driven world. With virtualization, we can overcome resource limitations and experiment with complex network scenarios. Furthermore, automation is becoming increasingly important as networks grow in size and complexity. This project not only enhances our comprehension of network functionalities but also equips us with the skills to manage modern networks efficiently.

**Phase 1**

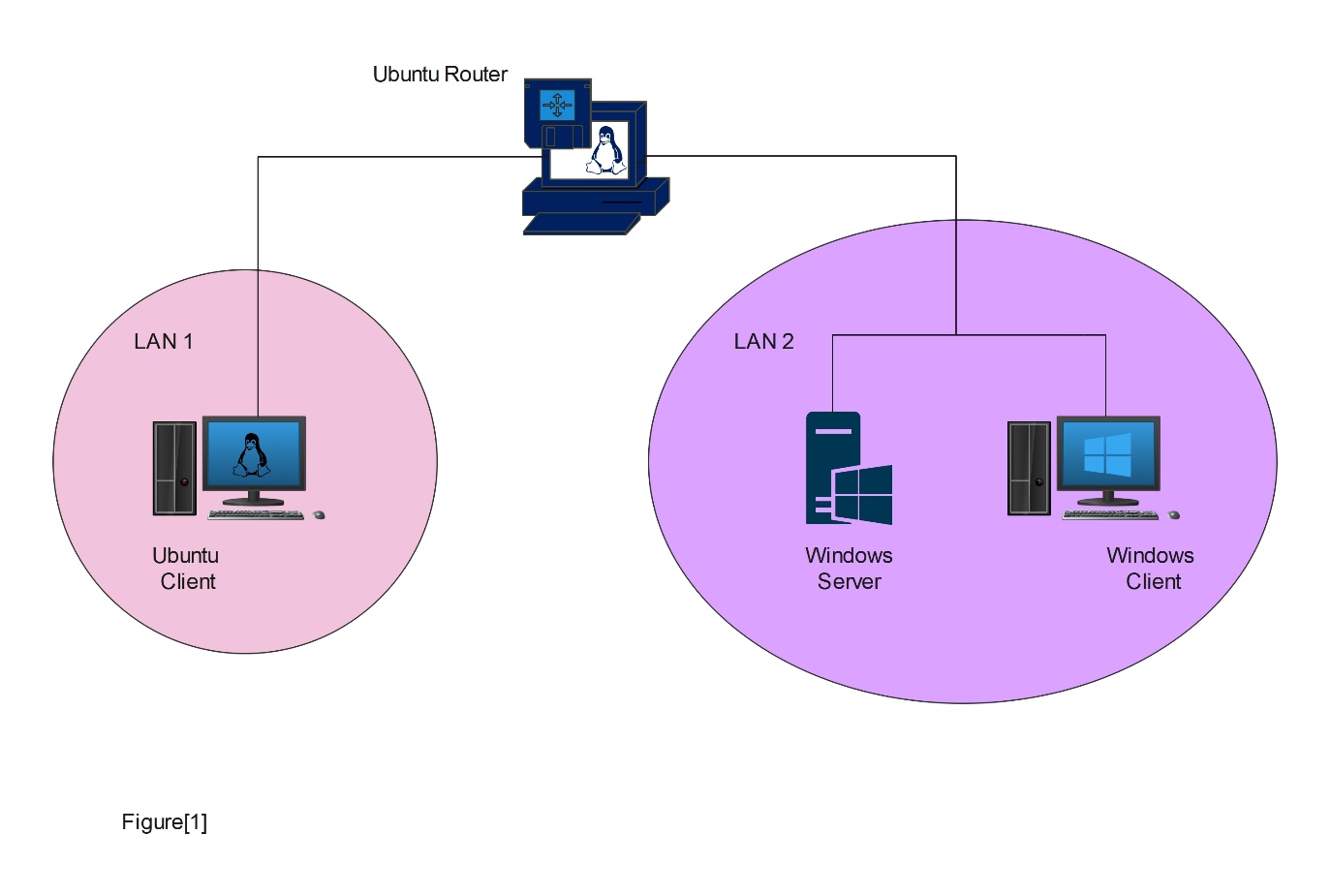
**2. Network Design and Planning**

In this section, we outline the design and planning of our virtual network. The goal is to create a simulated environment that mirrors a real-world network, allowing us to explore network functionalities and management without the need for physical hardware.

**2.1 Proposed Network Topology**

The network topology consists of two separate LANs connected through an Ubuntu-based router. Each LAN represents a different network segment, allowing us to simulate inter-network communication and routing. The topology includes:

* **LAN 1**: Contains the Windows 10 client.
* **LAN 2**: Includes the Ubuntu client and the Windows Server acting as a DNS server.
* **Ubuntu Router**: Connects LAN 1 and LAN 2, handling packet forwarding and network services like DHCP and NAT.

A visual representation of this topology is provided in Figure [1].

**2.2 Components and Their Roles**

**2.2.1 Ubuntu Router:**

The Ubuntu system acts as the router, forwarding packets so the two LANs can communicate with each other. This setup allows devices on one LAN to reach devices on the other, effectively connecting the two separate networks.

Its primary functions include:

* **Packet Forwarding**: Routes traffic between LAN 1 and LAN 2.
* **DHCP Server**: Assigns IP addresses to devices on both LANs, simplifying network configuration.
* **NAT (Network Address Translation)**: Masks the internal IP addresses of the clients, allowing secure internet access through a single public IP address (if connected to an external network).

**2.2.2 Ubuntu Client:**

The Ubuntu client resides on LAN 2 and acts as a standard workstation within the network. It will:

* Obtain its IP configuration automatically via DHCP.
* Use the Windows Server for DNS resolution.

**2.2.3 Windows Server (DNS Server):**

Located on LAN 2, the Windows Server is responsible for DNS services and is set up so that all other components in the network get their DNS information from it.

Its responsibilities include:

* **DNS Services**: Resolves domain names for network clients, ensuring proper name resolution within the network.

**2.2.4 Windows 10 Client:**

The Windows 10 client is situated on LAN 1 and functions as an end-user machine. Its features include:

* Automatic IP configuration via DHCP.
* DNS queries directed to the Windows Server on LAN 2.

**2.3 Virtual Switches and LAN Segments in VMware**

To create different LANs, we're using virtual switches or LAN segmentation in VMware. With this built-in option in VMware, we're able to have different LANs within our virtual environment.

In our setup:

* **LAN Segment 1**: Connects the Windows 10 client and one interface of the Ubuntu router.
* **LAN Segment 2**: Connects the Ubuntu client, Windows Server, and the second interface of the Ubuntu router.

By configuring the network adapters of each VM to connect to the appropriate LAN segment, we ensure proper network isolation and control over traffic flow.

**2.4 IP Addressing Scheme**

Each LAN in our topology has a private IP address range, which is assigned by the DHCP server set up on the Ubuntu router.

The Ubuntu router's interfaces act as the default gateways for their respective LANs. The DHCP server on the router provides IP addresses to clients and specifies the DNS server (Windows Server) for name resolution.

**2.5 Security Considerations**

Security is a key aspect of our network design:

* **NAT Implementation**: We've configured NAT on the Ubuntu router. This means all the clients are hidden from the outside world, and they get their internet connection through the router. NAT provides security by masking the internal IP addresses of the clients when accessing external networks.

**3. Implementation Methodology**

The implementation of the virtual network infrastructure involved a series of configurations and setups to simulate a realistic networking environment within a virtualized context. The methodology encompassed setting up virtual LAN segments, configuring a router, establishing DNS services, and ensuring proper communication among all network devices.

**3.1 Setting Up Virtual Switches (LAN Segments) in VMware**

Virtual switches, also known as LAN segments in VMware, were created to emulate separate network segments. These virtual LANs allowed for the isolation of network traffic and the test of routing and communication between different subnets. By configuring multiple LAN segments, the network topology mirrored that of a physical network with distinct broadcast domains, enabling a comprehensive examination of inter-network connectivity and security controls within a virtualized environment.

**Create LAN Segments:**

By editing any Virtual Machine's Network Settings, two LAN segments are created:  
 LAN\_Segment\_1

LAN\_Segment\_2

**Assigning LAN Segments to Virtual Machines**

* **Ubuntu Router**:
  + **3 network adapters**:
    - **Adapter 1**: Connect to LAN\_Segment\_1 (ens37)
    - **Adapter 2**: Connect to LAN\_Segment\_2 (ens38)
    - **Adapter 3**: NAT (ens33)
* **Windows 10 Client**:
  + Set network adapter to LAN\_Segment\_1
* **Ubuntu Client**:
  + Set network adapter to LAN\_Segment\_2 (ens33)
* **Windows Server**:
  + Set network adapter to LAN\_Segment\_2

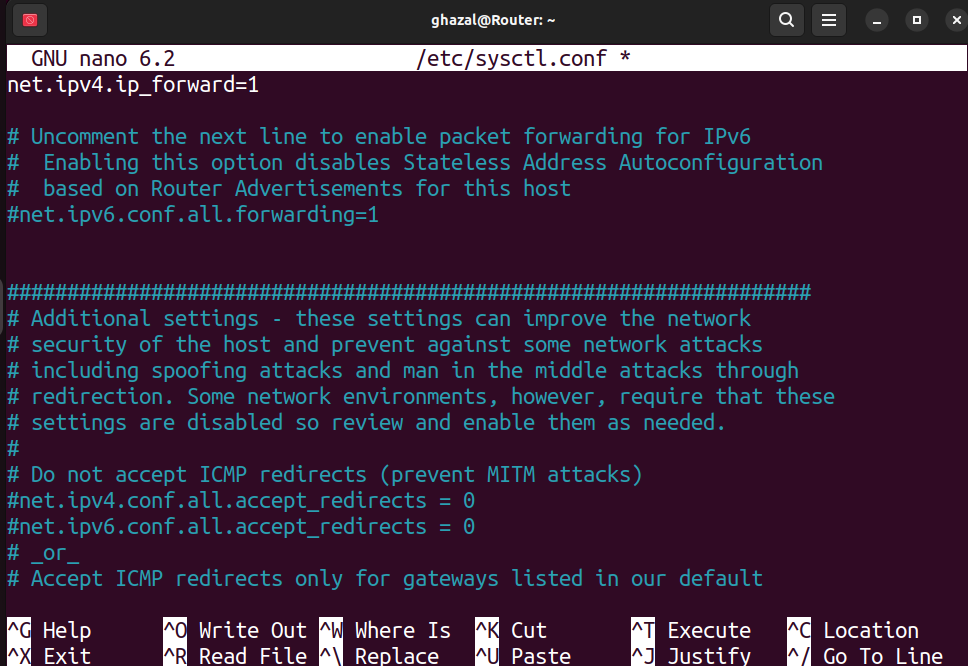
**3.2 Configuring the Ubuntu Router**

An Ubuntu server was configured to function as a router, interconnecting the virtual LAN segments and providing essential network services such as DHCP and NAT.

**3.2.1 Enabling IP Forwarding:**

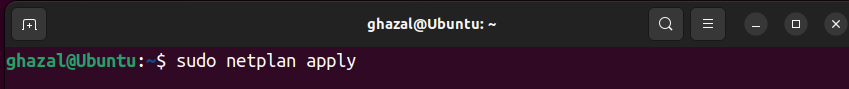
To allow the Ubuntu to forward packets between the two LANs, IP forwarding should be enabled. This setting altered the kernel's behavior to forward incoming packets destined for other networks, effectively turning the Ubuntu server into a router capable of connecting multiple LAN segments.

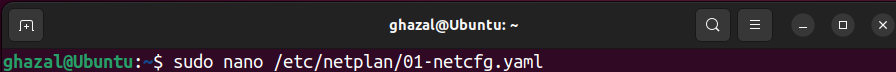
A screenshot of a computer

Description automatically generated

**3.2.2 Assigning Static IPs to Router Interfaces:**

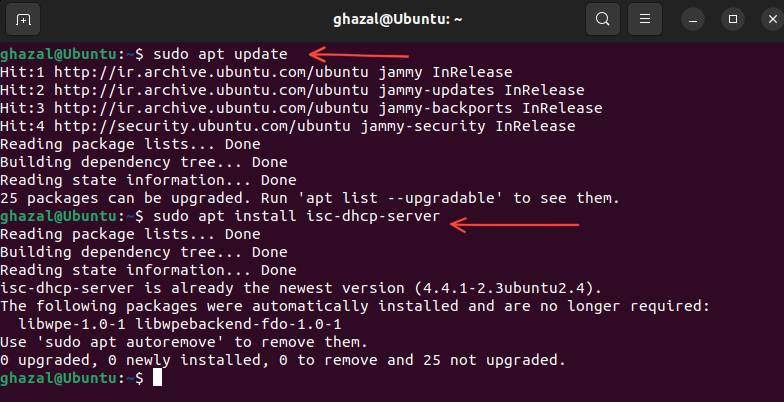
Static IP addresses were assigned to each network interface of the Ubuntu router. This ensured consistent addressing and reliable routing between the connected LAN segments. Each interface corresponded to a different subnet, aligning with the respective LAN segment's IP address scheme, thereby maintaining proper network segmentation and simplifying routing table configurations.

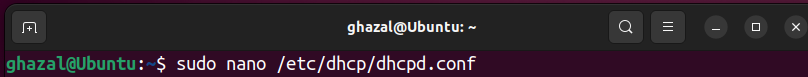
A screenshot of a computer

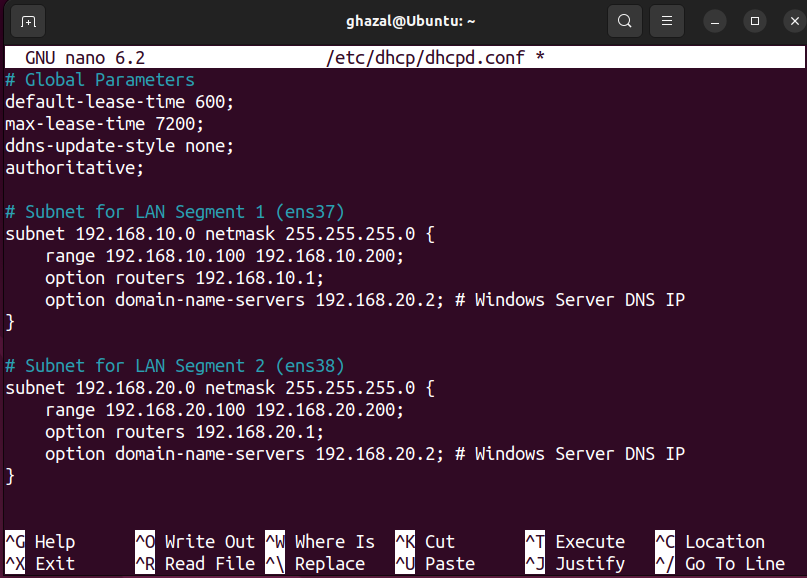
Description automatically generated

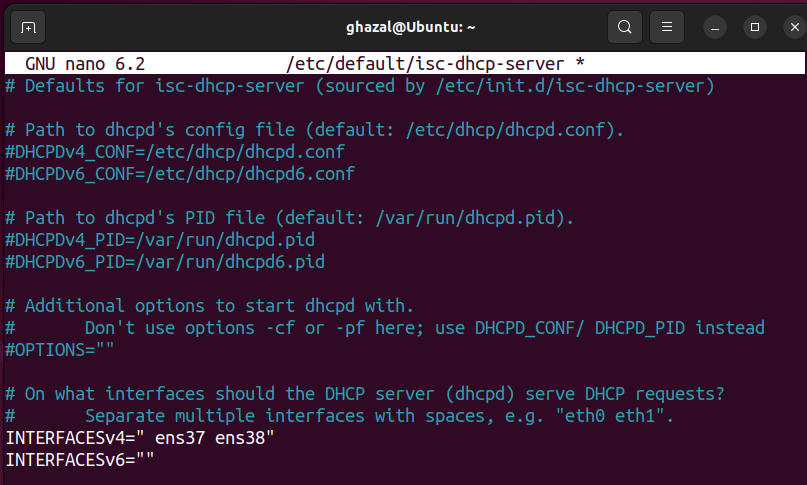
**3.2.3 Installing and Configuring DHCP Server:**

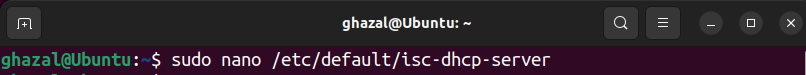
A DHCP server was installed and configured on the Ubuntu router to automate IP address assignment and network configuration for client devices. The DHCP server was set up with scopes corresponding to each subnet, providing clients with necessary network parameters such as IP addresses, subnet masks, default gateways, and DNS server addresses. This automated approach reduced manual configuration errors and streamlined the network setup process for clients joining the network.

* **Install DHCH:**
* **Subnet Declarations:**

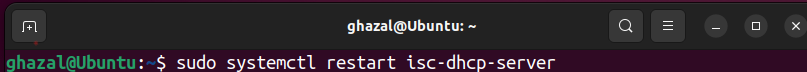




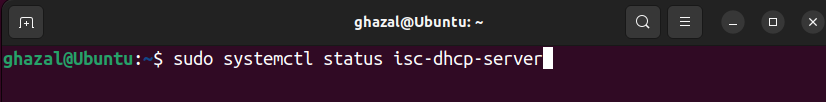
* **Specify the Network Interfaces for DHCP Server:**

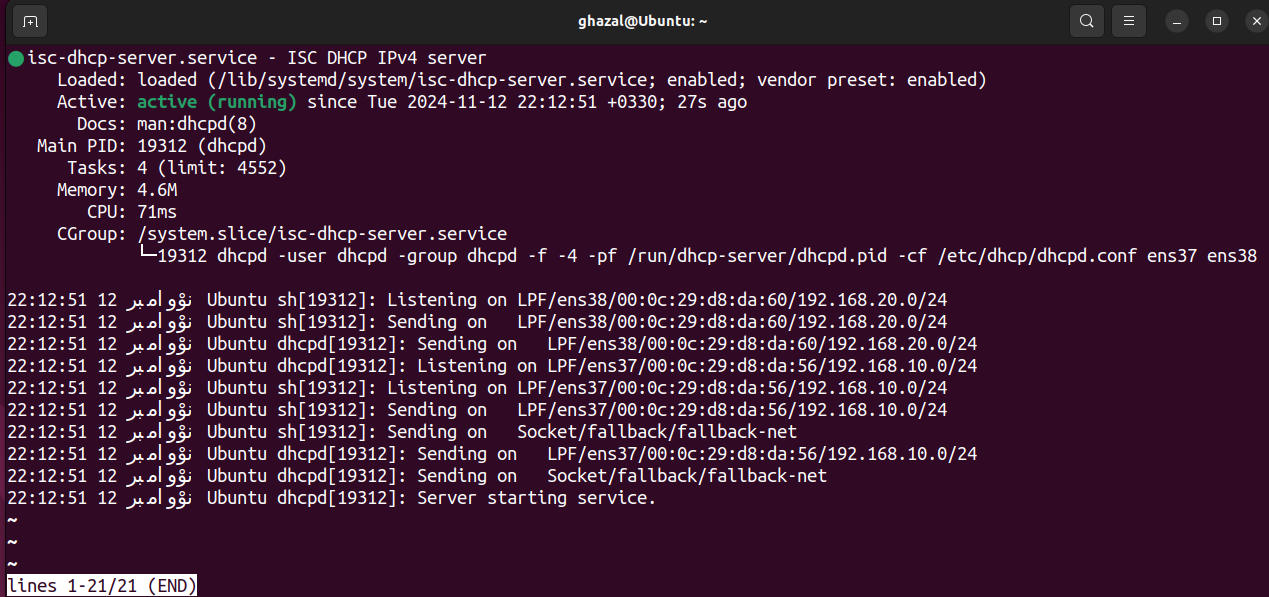


* **Apply changes:**



* **Verify the DHCP Server Status:**

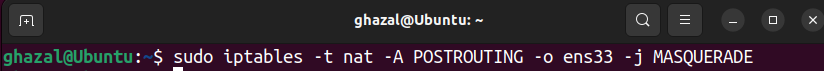
****



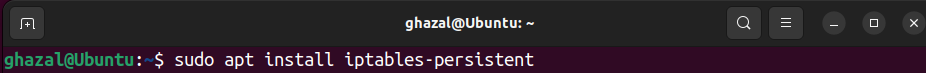
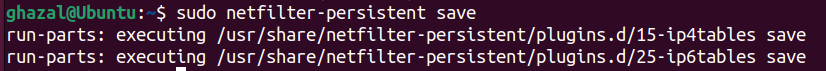
**3.2.4 Configuring NAT:**

Network Address Translation (NAT) was configured on the Ubuntu router to allow devices on the private network to access external networks, including the internet, using a single public IP address. NAT masquerading was employed to translate internal private IP addresses to the router's public IP address for outbound traffic. This setup preserved the limited public IP addresses and enhanced network security by hiding internal network structures from external entities.

* **Add NAT rule to iptables:**

****

* **Save the iptables rules so they persist after reboot:**

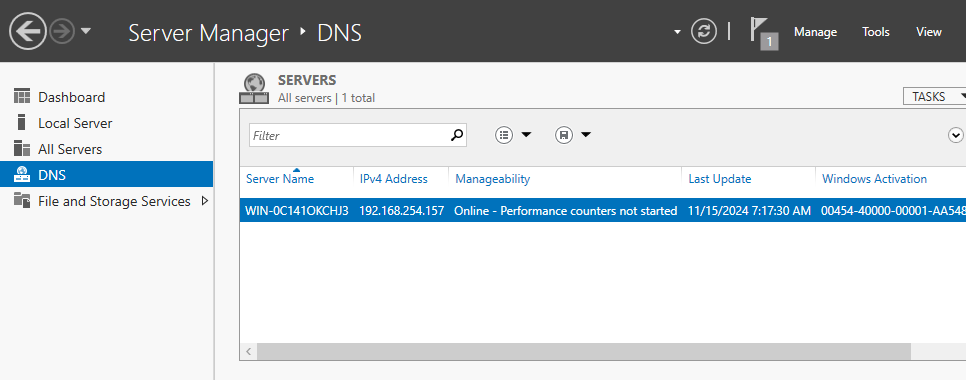


**3.3 Configuring Windows Server as DNS Server**

A Windows Server was configured to serve as the network's DNS server, providing domain name resolution services critical for network operations.

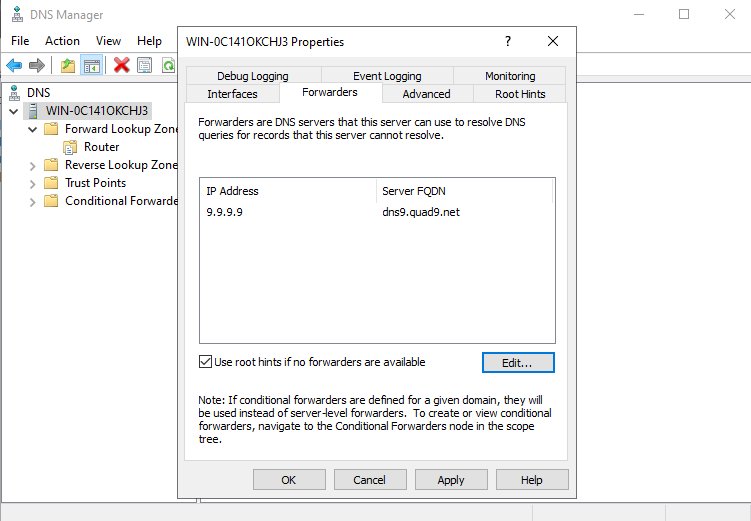
**3.3.1 Installing DNS Server Role:**

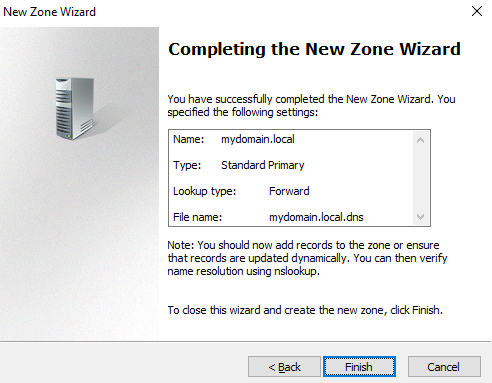
The DNS Server role was installed on the Windows Server to enable it to respond to DNS queries from clients. This role provided the necessary infrastructure to host DNS zones and manage DNS records, facilitating both forward and reverse DNS lookups within the network. This capability is crucial for the internal network to function efficiently, as it reduces dependency on external DNS servers and allows for customized DNS management tailored to the network's topology.

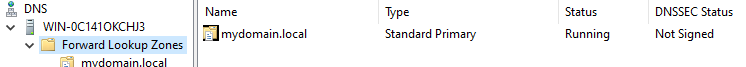
****

**3.3.2 Configuring DNS Zones and Records:**

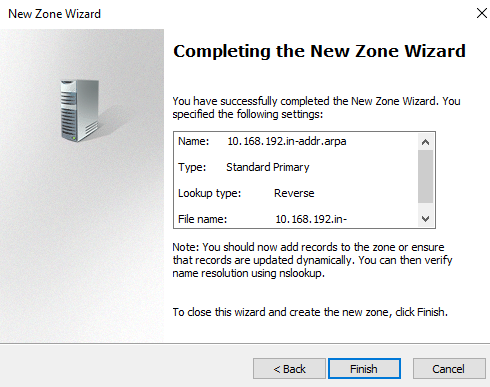
DNS zones were created and configured to define the namespace and manage domain name resolution. A forward lookup zone was established to map hostnames to IP addresses, while reverse lookup zones were set up to map IP addresses back to hostnames. Within these zones, DNS records such as A (Address) records and PTR (Pointer) records were added for each network device. Additionally, DNS forwarders were configured to direct unresolved queries to external DNS servers (e.g., 9.9.9.9), enabling internet name resolution for clients.

****

****

****

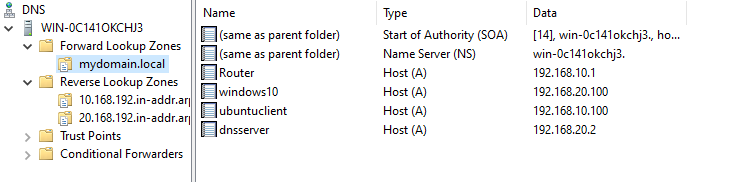
**A screenshot of a computer

Description automatically generated**

**A close-up of a white background

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

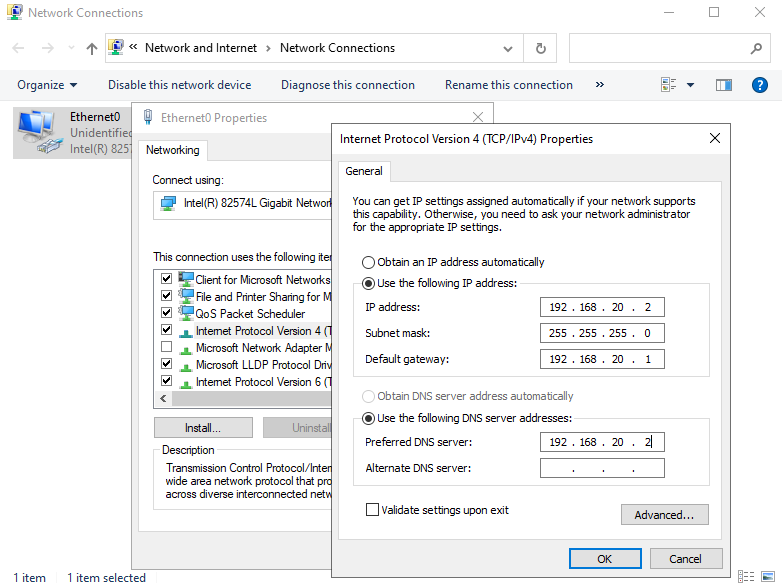
**A screenshot of a computer

Description automatically generated**

****

**3.3.3 Assigning Static IP to Windows Server:**

A static IP address was assigned to the Windows Server to ensure consistent accessibility as the DNS server. This fixed addressing prevented disruptions in DNS services that could occur if the server's IP address changed, thereby maintaining network stability and reliability for domain name resolution.

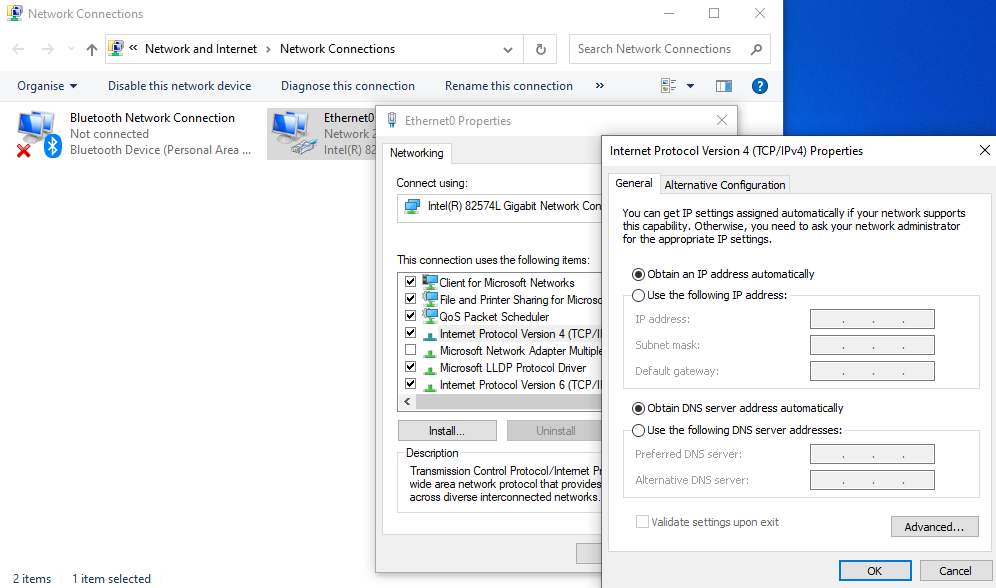
****

**3.4 Configuring Clients to Obtain IP via DHCP**

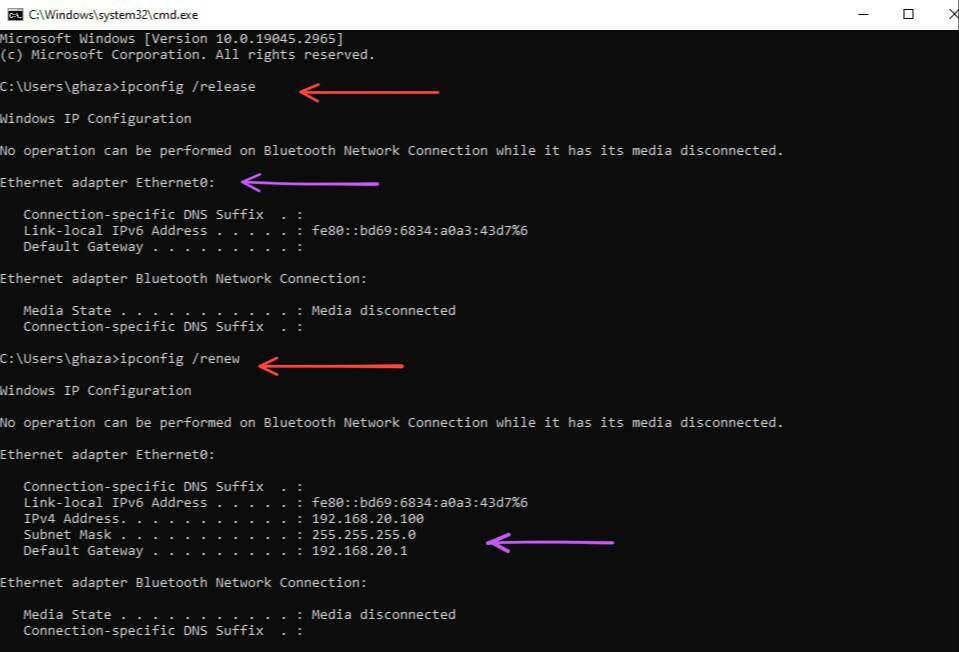
Client devices were configured to automatically obtain their IP configuration from the DHCP server on the Ubuntu router, simplifying network management and ensuring consistent network settings across all devices.

**3.4.1 Configuring Windows 10 Client:**

The Windows 10 client was set to use DHCP by adjusting its network adapter settings. This configuration allowed the client to receive an IP address, subnet mask, default gateway, and DNS server information automatically from the DHCP server.

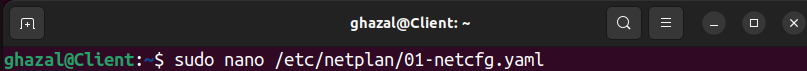


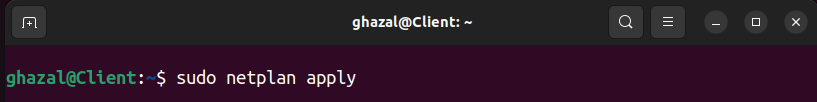
* **Apply changes:**

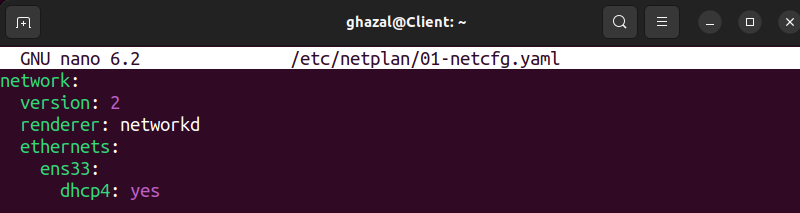


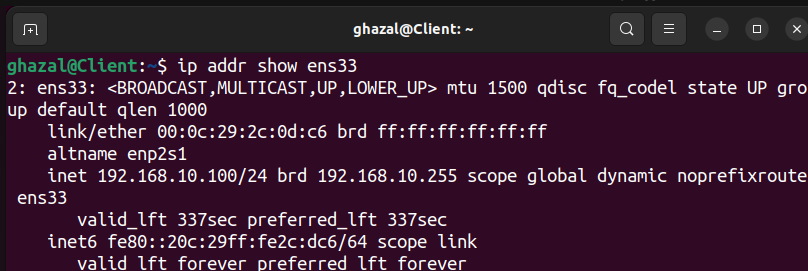
**3.4.2 Configuring Ubuntu Client:**

Similarly, the Ubuntu client was configured to obtain its network settings via DHCP. By enabling DHCP on its network interface, the client automatically received the necessary network parameters, ensuring proper communication within the network and access to network services.







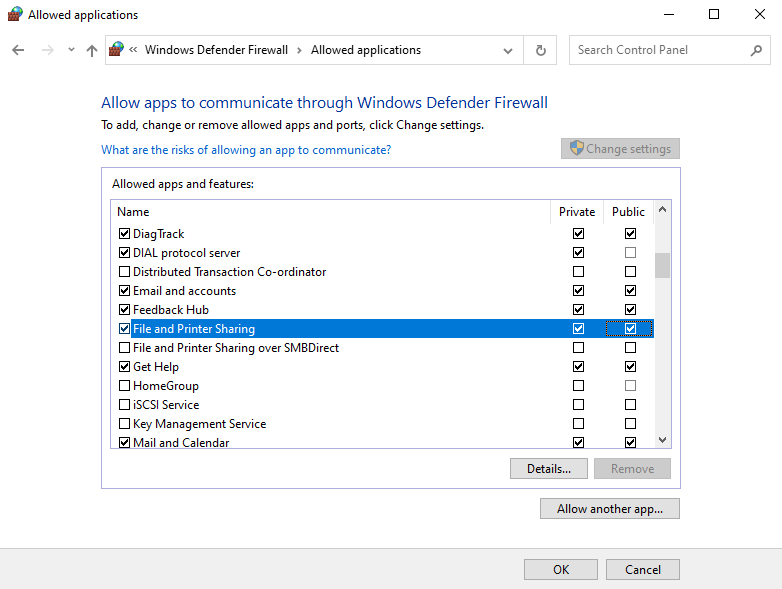


**3.5 Configuring Clients to Use Windows Server DNS**

Clients were configured to use the Windows Server as their primary DNS server for domain name resolution. This configuration was achieved by specifying the DNS server's IP address in the network settings provided by the DHCP server. By directing DNS queries to the Windows Server, clients could resolve internal hostnames efficiently and access external domains through the server's DNS forwarding capabilities.

**3.6 Configuring Windows Firewall to Allow ICMP (Ping) Requests**

By default, Windows Firewall may block ICMP echo requests (ping), which can prevent devices on the network from pinging Windows machines. This can hinder network diagnostics and connectivity tests. To enable proper communication and troubleshooting, the firewall settings should be adjusted on Windows machines.

****

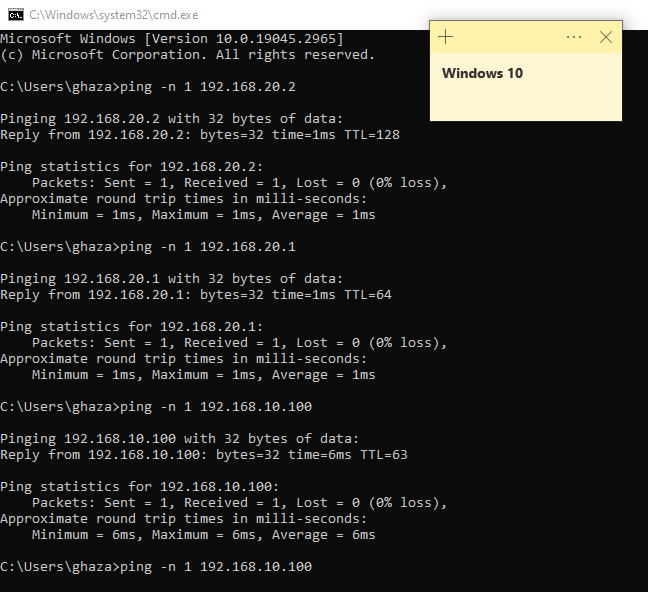
**4. Testing and Validation**

Thorough testing and validation are critical to ensure that the implemented virtual network functions as intended. This section outlines the methods and results of testing various network components, including connectivity, DHCP functionality, DNS resolution, NAT configuration, and performance metrics.

**4.1 Connectivity Testing Between Clients**

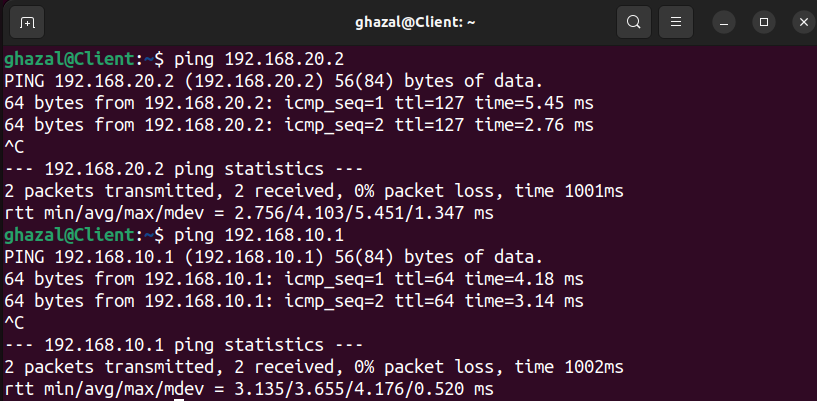
Verifying that all clients within and across LAN segments can communicate with each other as per the network design.

**Steps:**

1. **Ping Tests:**

A screenshot of a computer program

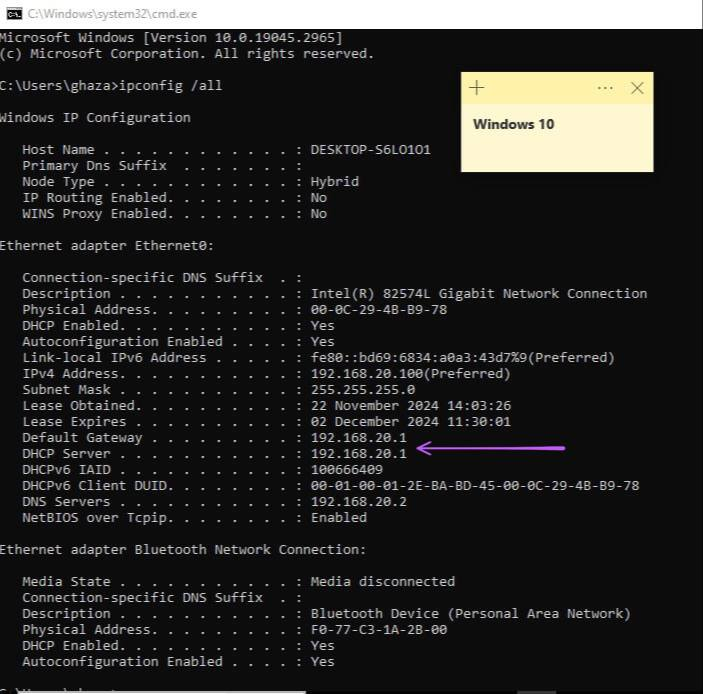
Description automatically generated

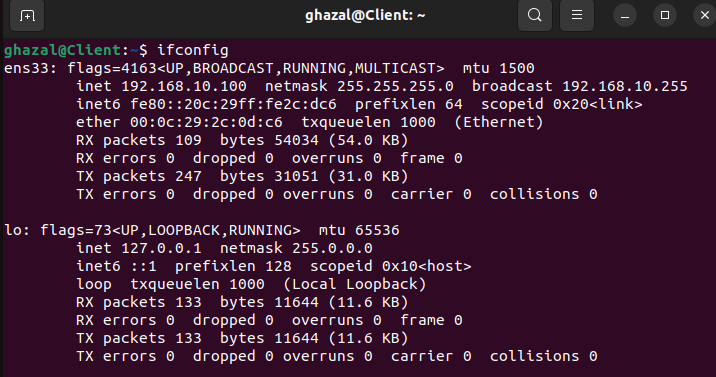


1. A screenshot of a computer

   Description automatically generated**Traceroute Tests:**

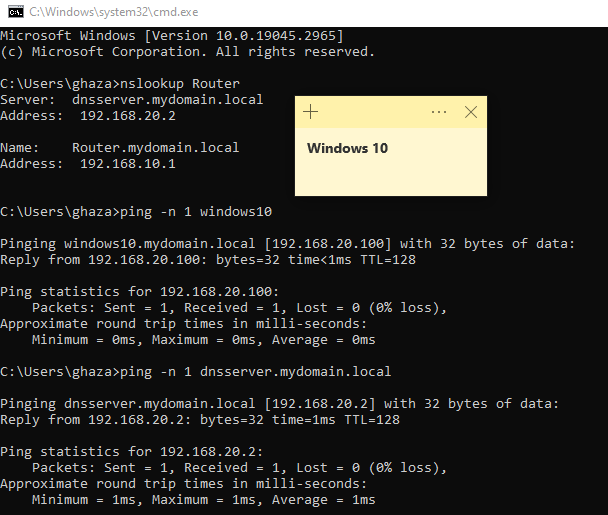
**4.2 DHCP Functionality Verification**

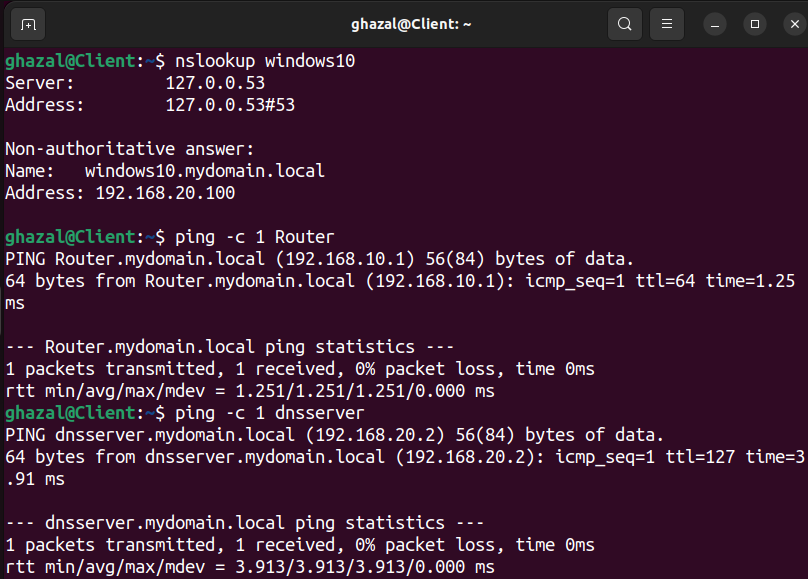
Ensuring that DHCP is correctly assigning IP addresses and network configurations to clients.



**4.3 DNS Resolution Testing**

Validating that the DNS server correctly resolves internal and external domain names.

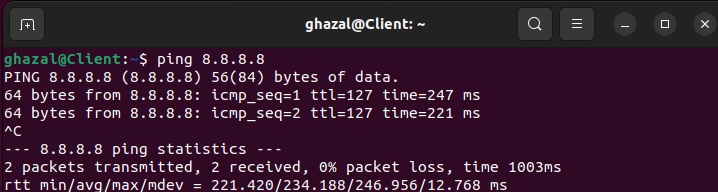
****

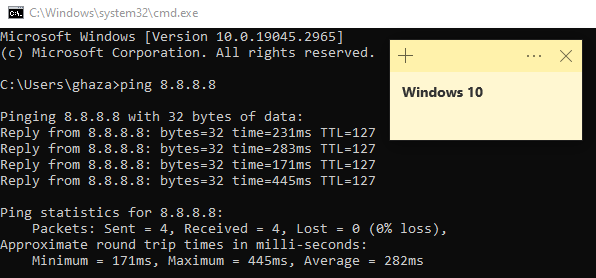
****

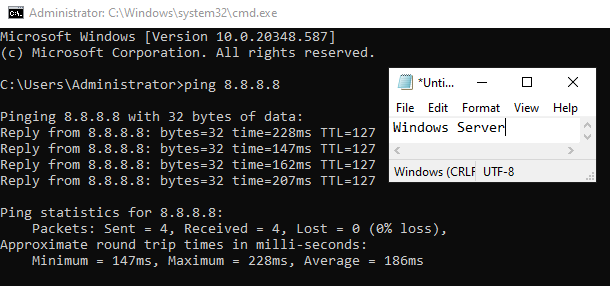
**4.4 NAT and Internet Access Testing**

Confirming that clients can access the internet via the router's NAT configuration.

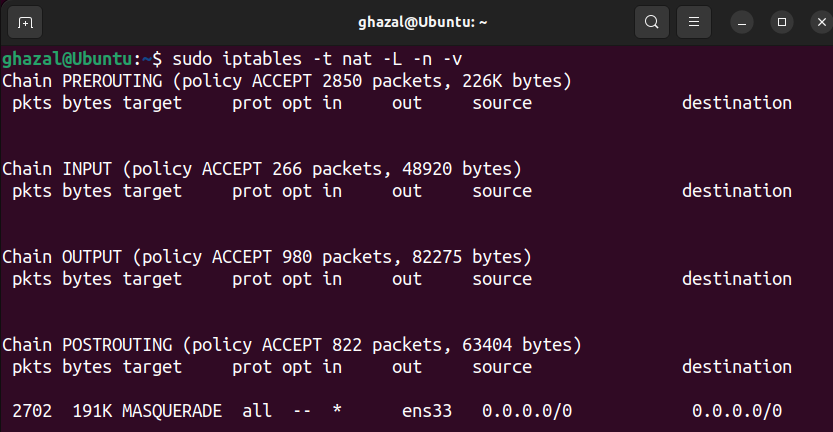
**Test Internet Connectivity:**

****

****

****

**Verify NAT Functionality on Router:**



**5. Results and Discussion**

This section analyzes the outcomes of the testing procedures, evaluates the performance and reliability of network services, discusses challenges faced during implementation, and considers security implications.

**5.1 Evaluation of DHCP and DNS Services**

**Findings:**

* **DHCP:**
  + Clients received correct IP configurations automatically.
  + DHCP lease times and renewals functioned as expected.
* **DNS:**
  + Internal hostnames resolved correctly, facilitating seamless network navigation.
  + External DNS queries were successfully forwarded and resolved, providing internet access.

**Discussion:**

* The DHCP server efficiently managed IP assignments, reducing manual configuration efforts.
* DNS resolution enhanced network usability, but initial issues with PTR records and DNS suffixes required attention.
* Proper DNS configuration is critical, as misconfigurations can lead to significant connectivity problems.

**5.2 Challenges and Troubleshooting**

**Challenges Faced:**

1. **Windows Firewall Blocking ICMP Traffic:**
   * **Issue:** Clients could not ping Windows machines due to firewall settings.
   * **Resolution:** Enabled ICMP echo requests in Windows Defender Firewall.
2. **PTR Record Creation Error:**
   * **Issue:** Error when creating PTR records due to missing reverse lookup zones.
   * **Resolution:** Created reverse lookup zones before adding PTR records.
3. **DNS Resolution Failures:**
   * **Issue:** Clients couldn't resolve hostnames due to missing DNS suffixes.
   * **Resolution:** Configured DHCP to provide the domain name and updated client settings.

**Discussion:**

* These challenges highlighted the importance of comprehensive firewall and DNS configurations.
* Systematic troubleshooting and step-by-step validation were essential in identifying and resolving issues.
* Documentation and adherence to best practices mitigated prolonged disruptions.

**5.4 Security Implications**

**Considerations:**

* **Firewall Configurations:**
  + Adjustments to firewall settings improved connectivity but potentially increased exposure.
  + Ensuring that only necessary ports and protocols are allowed minimizes security risks.
* **Access Controls:**
  + Proper user authentication and permissions on network devices are crucial.
  + Implementing strong passwords and regular updates enhances security posture.

**Discussion:**

* Balancing functionality and security is vital; enabling services should not compromise network integrity.
* Regular security assessments and adherence to security guidelines are recommended to protect the network.

**6. Conclusion**

**6.1 Summary of Findings**

The project successfully implemented a virtual multi-segment LAN environment using VMware, Ubuntu, and Windows systems. Through systematic testing and validation:

* **Connectivity** between clients was established and verified.
* **DHCP and DNS services** functioned correctly after proper configuration.
* **NAT and internet access** were operational, allowing clients to reach external networks.

**6.2 Achievement of Objectives**

All project objectives were met:

* **Design and Implementation:** A virtual network simulating real-world LAN scenarios was created.
* **Service Configuration:** Essential network services (DHCP, DNS, NAT) were configured and operational.
* **Testing and Validation:** Comprehensive testing ensured that the network met the desired performance and functionality criteria.
* **Troubleshooting Skills:** Challenges were addressed effectively, demonstrating problem-solving capabilities.

**6.3 Recommendations for Future Work**

* **Automation Enhancements:**
  + Implement scripts to automate network configuration tasks, reducing manual effort.
  + Explore tools like Ansible for configuration management.
* **Security Improvements:**
  + Introduce intrusion detection systems (IDS) and intrusion prevention systems (IPS).
  + Conduct regular vulnerability assessments and penetration testing.
* **Scalability Testing:**
  + Expand the network to include additional clients and servers.
  + Assess performance under increased load and adjust resources accordingly.
* **Documentation and Training:**
  + Develop comprehensive guides for network setup and maintenance.
  + Provide training sessions for team members to ensure knowledge transfer.

**7. References**

1. **Stallings, W. (2016). *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Addison-Wesley Professional.**
   * Provides foundational knowledge on modern networking concepts relevant to virtualization and automation.
2. **Bolla, R., Bruschi, R., Davoli, F., & Cucchietti, F. (2011). Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures. *IEEE Communications Surveys & Tutorials*, 13(2), 223-244.**
   * Discusses efficient network infrastructure management, which is pertinent to virtual network environments.
3. **Williams, O., & Kavanagh, R. (2019). Network Function Virtualization: An Introduction and its Benefits. *International Journal of Computer Networks & Communications*, 11(2), 45-58.**
   * Explores virtualization benefits, aligning with the project's virtualization focus.
4. **Poulos, G. S., & Katsaros, D. (2018). Orchestrating the deployment of virtual network functions with Docker. *IEEE Transactions on Network and Service Management*, 15(4), 1441-1455.**
   * Provides insights into containerization and automation, relevant for future work recommendations.
5. **Cisco Systems. (2017). *Cisco Guide to Harden Cisco IOS Devices*. Cisco Systems.**
   * Offers best practices for securing network devices, relevant to security implications.