



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

1.1 Background and Motivation

1.2 Objectives

1.3 Significance of the Study

2. Phase 1

2. Network Design and Planning

2.1 Proposed Network Topology

2.2 Components and Their Roles

2.2.1 Ubuntu Router

2.2.2 Ubuntu Client

2.2.3 Windows Server (DNS Server)

2.2.4 Windows 10 Client

2.3 Virtual Switches and LAN Segments in VMware

2.4 IP Addressing Scheme

2.5 Security Considerations

3. Implementation Methodology

3.1 Setting Up Virtual Switches (LAN Segments) in VMware

3.2 Configuring the Ubuntu Router

3.2.1 Enabling IP Forwarding

3.2.2 Assigning Static IPs to Router Interfaces

3.2.3 Installing and Configuring DHCP Server

3.2.4 Configuring NAT

3.3 Configuring Windows Server as DNS Server

3.3.1 Installing DNS Server Role

3.3.2 Configuring DNS Zones and Records

3.3.3 Assigning Static IP to Windows Server

3.4 Configuring Clients to Obtain IP via DHCP

3.4.1 Configuring Windows 10 Client

3.4.2 Configuring Ubuntu Client

3.5 Configuring Clients to Use Windows Server DNS

3.6 Configuring Windows Firewall to Allow ICMP (Ping) Requests

4. Testing and Validation

4.1 Connectivity Testing Between Clients

4.2 DHCP Functionality Verification

4.3 DNS Resolution Testing

4.4 NAT and Internet Access Testing

5. Results and Discussion

5.1 Evaluation of DHCP and DNS Services

5.2 Challenges and Troubleshooting

5.3 Security Implications

6. Conclusion

6.1 Summary of Findings

6.2 Achievement of Objectives

6.3 Recommendations for Future Work

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

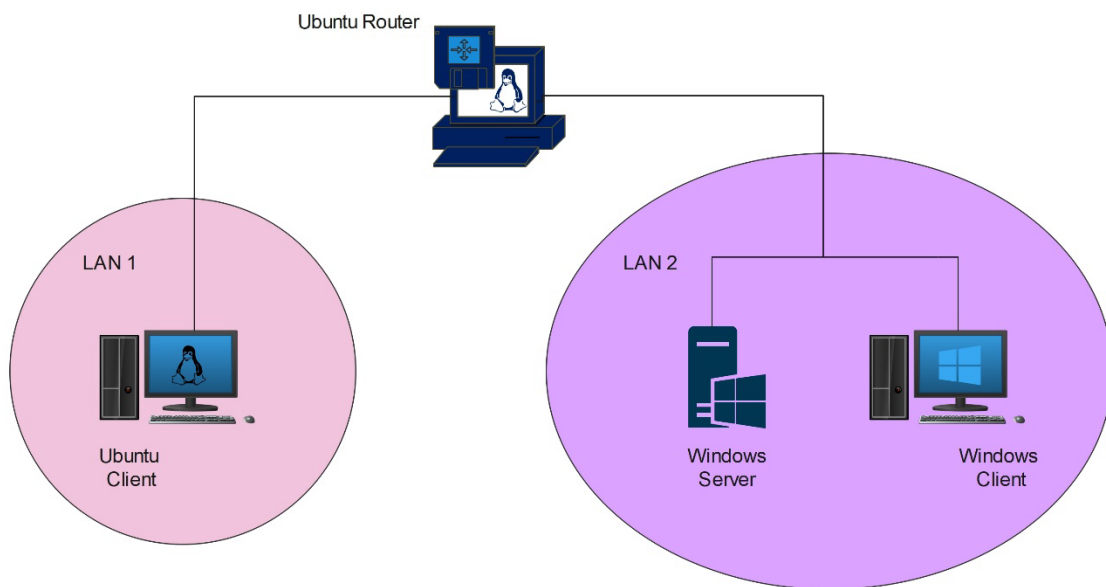
In this section, we outline the design and planning of our virtual network. The goal is to create a simulated environment that mirrors a real-world network, allowing us to explore network functionalities and management without the need for physical hardware.

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].



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.

```
ghazal@Router: ~  
ghazal@Router:~$ sudo nano /etc/sysctl.conf  
ghazal@Router:~$ sudo sysctl -p  
net.ipv4.ip_forward = 1  
ghazal@Router:~$
```

```
GNU nano 6.2 /etc/sysctl.conf *  
net.ipv4.ip_forward=1  
  
# Uncomment the next line to enable packet forwarding for IPv6  
# Enabling this option disables Stateless Address Autoconfiguration  
# based on Router Advertisements for this host  
#net.ipv6.conf.all.forwarding=1  
  
#####  
# Additional settings - these settings can improve the network  
# security of the host and prevent against some network attacks  
# including spoofing attacks and man in the middle attacks through  
# redirection. Some network environments, however, require that these  
# settings are disabled so review and enable them as needed.  
#  
# Do not accept ICMP redirects (prevent MITM attacks)  
#net.ipv4.conf.all.accept_redirects = 0  
#net.ipv6.conf.all.accept_redirects = 0  
# _or_  
# Accept ICMP redirects only for gateways listed in our default  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

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.



```
ghazal@Ubuntu: ~  
ghazal@Ubuntu:~$ sudo nano /etc/netplan/01-netcfg.yaml  
  
GNU nano 6.2 /etc/netplan/01-netcfg.yaml *  
network:  
  version: 2  
  renderer: networkd  
  ethernet:  
    ens37:  
      addresses: [192.168.10.1/24]  
    ens38:  
      addresses: [192.168.20.1/24]  
  
ghazal@Ubuntu:~$ sudo netplan apply
```

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:

```

ghazal@Ubuntu: ~
ghazal@Ubuntu:~$ sudo apt update
Hit:1 http://ir.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ir.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ir.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
25 packages can be upgraded. Run 'apt list --upgradable' to see them.
ghazal@Ubuntu:~$ sudo apt install isc-dhcp-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
isc-dhcp-server is already the newest version (4.4.1-2.3ubuntu2.4).
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 25 not upgraded.
ghazal@Ubuntu:~$

```

- Subnet Declarations:

```

ghazal@Ubuntu: ~
ghazal@Ubuntu:~$ sudo nano /etc/dhcp/dhcpd.conf

GNU nano 6.2 /etc/dhcp/dhcpd.conf *
# Global Parameters
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
authoritative;

# Subnet for LAN Segment 1 (ens37)
subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.100 192.168.10.200;
    option routers 192.168.10.1;
    option domain-name-servers 192.168.20.2; # Windows Server DNS IP
}

# Subnet for LAN Segment 2 (ens38)
subnet 192.168.20.0 netmask 255.255.255.0 {
    range 192.168.20.100 192.168.20.200;
    option routers 192.168.20.1;
    option domain-name-servers 192.168.20.2; # Windows Server DNS IP
}

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line

```

- Specify the Network Interfaces for DHCP Server:

```
ghazal@Ubuntu: ~  
ghazal@Ubuntu:~$ sudo nano /etc/default/isc-dhcp-server
```

```
GNU nano 6.2 /etc/default/isc-dhcp-server *  
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)  
  
# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).  
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf  
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf  
  
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).  
#DHCPDv4_PID=/var/run/dhcpd.pid  
#DHCPDv6_PID=/var/run/dhcpd6.pid  
  
# Additional options to start dhcpd with.  
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead  
#OPTIONS=""  
  
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?  
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".  
INTERFACESv4=" ens37 ens38"  
INTERFACESv6=""
```

- Apply changes:

```
ghazal@Ubuntu: ~  
ghazal@Ubuntu:~$ sudo systemctl restart isc-dhcp-server
```

- Verify the DHCP Server Status:

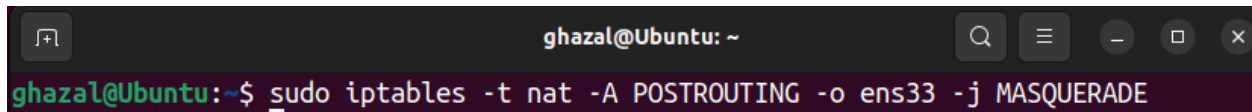
```
ghazal@Ubuntu: ~  
ghazal@Ubuntu:~$ sudo systemctl status isc-dhcp-server
```

```
ghazal@Ubuntu: ~  
● isc-dhcp-server.service - ISC DHCP IPv4 server  
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)  
   Active: active (running) since Tue 2024-11-12 22:12:51 +0330; 27s ago  
     Docs: man:dhcpd(8)  
    Main PID: 19312 (dhcpd)  
      Tasks: 4 (limit: 4552)  
     Memory: 4.6M  
        CPU: 71ms  
    CGroup: /system.slice/isc-dhcp-server.service  
            └─19312 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf ens37 ens38  
  
22:12:51 12 نؤوامير Ubuntu sh[19312]: Listening on LPF/ens38/00:0c:29:d8:da:60/192.168.20.0/24  
22:12:51 12 نؤوامير Ubuntu sh[19312]: Sending on LPF/ens38/00:0c:29:d8:da:60/192.168.20.0/24  
22:12:51 12 نؤوامير Ubuntu dhcpd[19312]: Sending on LPF/ens38/00:0c:29:d8:da:60/192.168.20.0/24  
22:12:51 12 نؤوامير Ubuntu dhcpd[19312]: Listening on LPF/ens37/00:0c:29:d8:da:56/192.168.10.0/24  
22:12:51 12 نؤوامير Ubuntu sh[19312]: Listening on LPF/ens37/00:0c:29:d8:da:56/192.168.10.0/24  
22:12:51 12 نؤوامير Ubuntu sh[19312]: Sending on LPF/ens37/00:0c:29:d8:da:56/192.168.10.0/24  
22:12:51 12 نؤوامير Ubuntu sh[19312]: Sending on Socket/fallback/fallback-net  
22:12:51 12 نؤوامير Ubuntu dhcpd[19312]: Sending on LPF/ens37/00:0c:29:d8:da:56/192.168.10.0/24  
22:12:51 12 نؤوامير Ubuntu dhcpd[19312]: Sending on Socket/fallback/fallback-net  
22:12:51 12 نؤوامير Ubuntu dhcpd[19312]: Server starting service.  
~  
~  
lines 1-21/21 (END)
```

3.2.4 Configuring NAT:

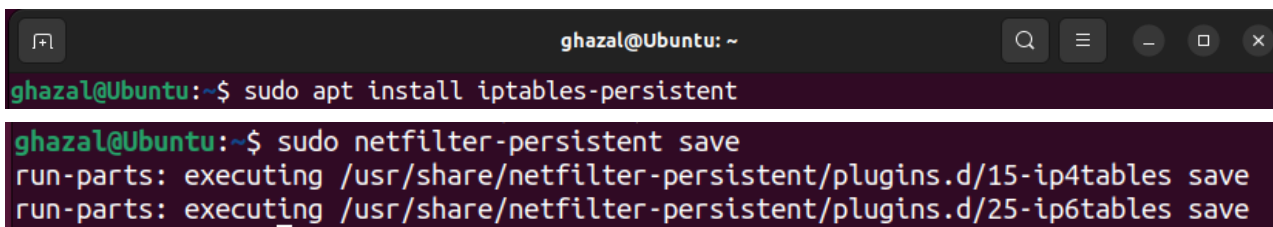
Network Address Translation (NAT) was configured on the Ubuntu router to allow devices on the private network to access external networks, including the internet, using a single public IP address. NAT masquerading was employed to translate internal private IP addresses to the router's public IP address 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:**

A terminal window titled 'ghazal@Ubuntu: ~' with search, menu, and window control icons. The command 'sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE' is entered and executed.

```
ghazal@Ubuntu:~$ sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
```

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

A terminal window titled 'ghazal@Ubuntu: ~' with search, menu, and window control icons. Two commands are entered and executed: 'sudo apt install iptables-persistent' and 'sudo netfilter-persistent save'. The output of the second command shows that the rules were saved for both ip4tables and ip6tables.

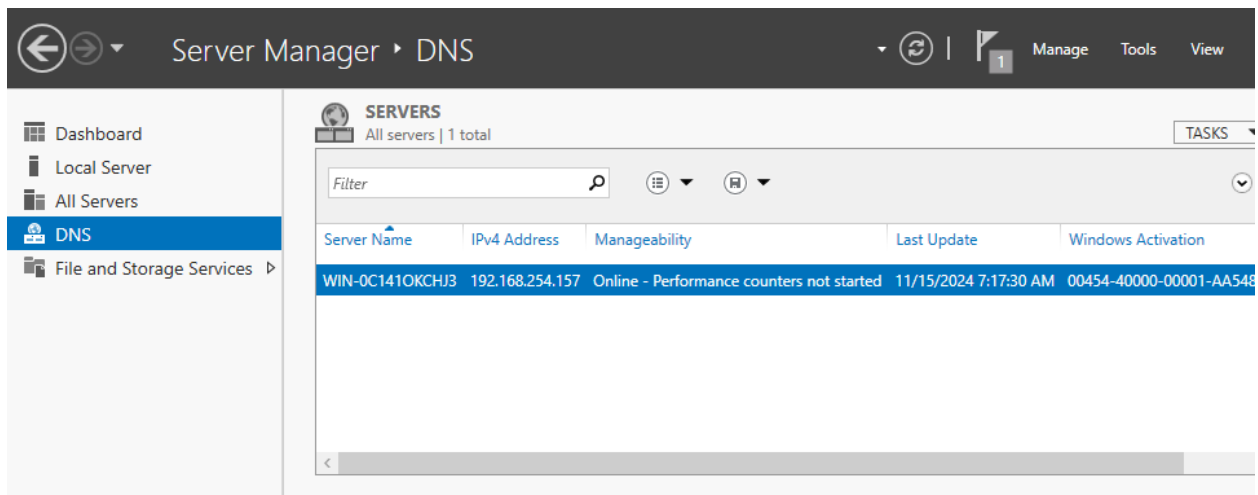
```
ghazal@Ubuntu:~$ sudo apt install iptables-persistent
ghazal@Ubuntu:~$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
```


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.

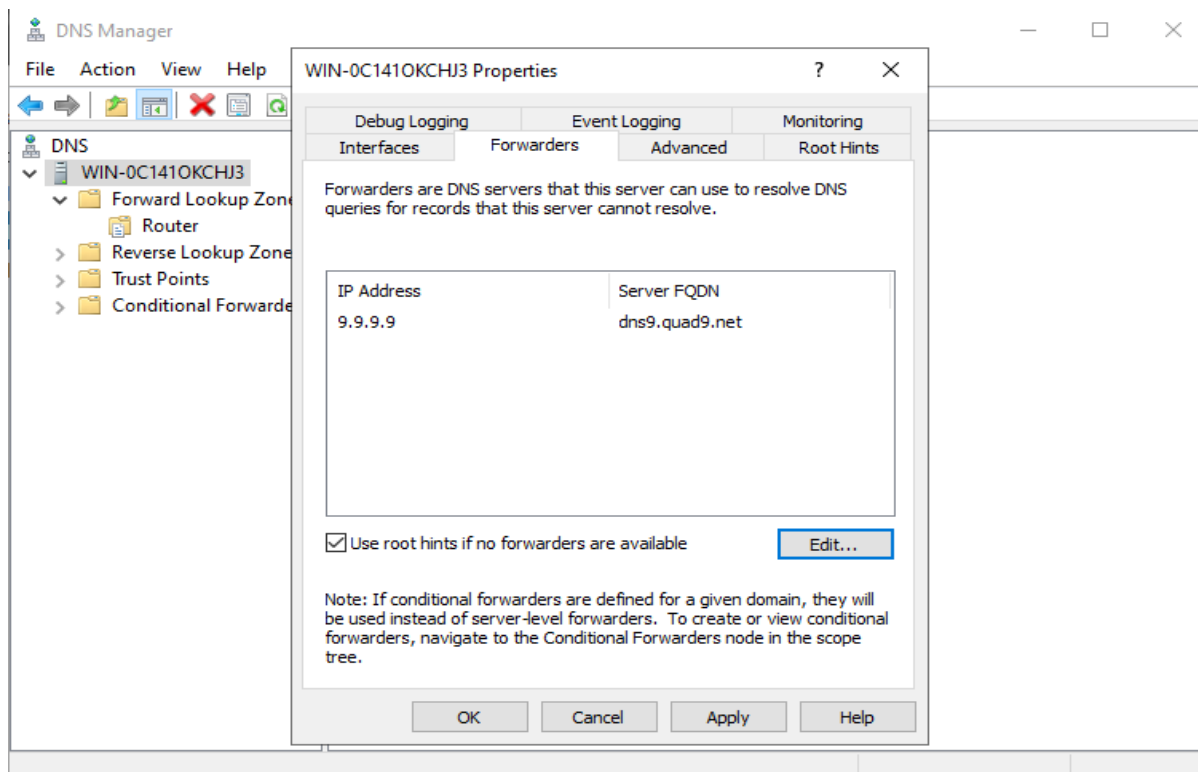
3.3.1 Installing DNS Server Role:

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.



New Zone Wizard

Completing the New Zone Wizard

You have successfully completed the New Zone Wizard. You specified the following settings:

Name: mydomain.local

Type: Standard Primary

Lookup type: Forward

File name: mydomain.local.dns

Note: You should now add records to the zone or ensure that records are updated dynamically. You can then verify name resolution using nslookup.

To close this wizard and create the new zone, click Finish.

< Back

Finish

Cancel

	Name	Type	Status	DNSSEC Status
DNS <ul style="list-style-type: none"> WIN-0C1410KCHJ3 <ul style="list-style-type: none"> Forward Lookup Zones <ul style="list-style-type: none"> mydomain.local 	mydomain.local	Standard Primary	Running	Not Signed

New Zone Wizard

Completing the New Zone Wizard

You have successfully completed the New Zone Wizard. You specified the following settings:

Name: 10.168.192.in-addr.arpa

Type: Standard Primary

Lookup type: Reverse

File name: 10.168.192.in-

Note: You should now add records to the zone or ensure that records are updated dynamically. You can then verify name resolution using nslookup.

To close this wizard and create the new zone, click Finish.

< Back

Finish

Cancel

	Name	Type	Status	DNSSEC Status
DNS <ul style="list-style-type: none"> WIN-0C1410KCHJ3 <ul style="list-style-type: none"> Forward Lookup Zones <ul style="list-style-type: none"> mydomain.local Reverse Lookup Zones <ul style="list-style-type: none"> 10.168.192.in-addr.arpa 20.168.192.in-addr.arpa 	10.168.192.in-addr.arpa 20.168.192.in-addr.arpa	Standard Primary Standard Primary	Running Running	Not Signed Not Signed

New Zone Wizard

Completing the New Zone Wizard

You have successfully completed the New Zone Wizard. You specified the following settings:

Name: 20.168.192.in-addr.arpa

Type: Standard Primary

Lookup type: Reverse

File name: 20.168.192.in-

Note: You should now add records to the zone or ensure that records are updated dynamically. You can then verify name resolution using nslookup.

To close this wizard and create the new zone, click Finish.

< Back

Finish

Cancel

DNS	Name	Type	Data
WIN-0C141OKCHJ3	(same as parent folder)	Start of Authority (SOA)	[14], win-0c141okchj3., ho...
Forward Lookup Zones	(same as parent folder)	Name Server (NS)	win-0c141okchj3.
mydomain.local	Router	Host (A)	192.168.10.1
Reverse Lookup Zones	windows10	Host (A)	192.168.20.100
10.168.192.in-addr.ar	ubuntuclient	Host (A)	192.168.10.100
20.168.192.in-addr.ar	dnsserver	Host (A)	192.168.20.2
Trust Points			
Conditional Forwarders			

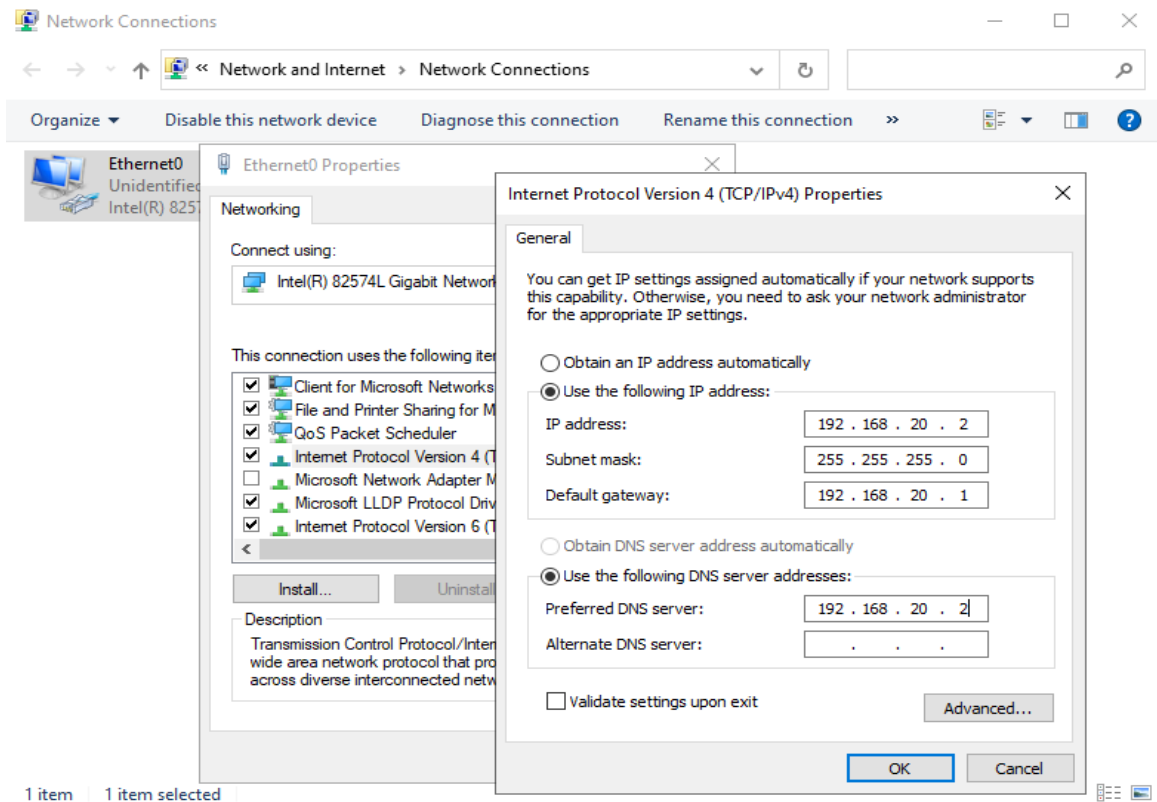
DNS	Name	Type	Data
WIN-0C141OKCHJ3	(same as parent folder)	Start of Authority (SOA)	[3], win-0c141okchj3., hos...
Forward Lookup Zones	(same as parent folder)	Name Server (NS)	win-0c141okchj3.
mydomain.local	192.168.10.1	Pointer (PTR)	Router.mydomain.local.
Reverse Lookup Zones	192.168.10.100	Pointer (PTR)	ubuntuclient.mydomain.l...
10.168.192.in-addr.ar			
20.168.192.in-addr.ar			
Trust Points			
Conditional Forwarders			

DNS Manager			
File Action View Help			
DNS	Name	Type	Data
WIN-0C141OKCHJ3	(same as parent folder)	Start of Authority (SOA)	[3], win-0c141okchj3., hos...
Forward Lookup Zones	(same as parent folder)	Name Server (NS)	win-0c141okchj3.
mydomain.local	192.168.20.100	Pointer (PTR)	windows10.mydomain.loc...
Reverse Lookup Zones	192.168.20.2	Pointer (PTR)	dnsserver.mydomain.local.
10.168.192.in-addr.ar			
20.168.192.in-addr.ar			
Trust Points			
Conditional Forwarders			

```
option domain-name "mydomain.local";
option domain-name-servers 192.168.20.2; # Windows Server DNS IP
```

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.

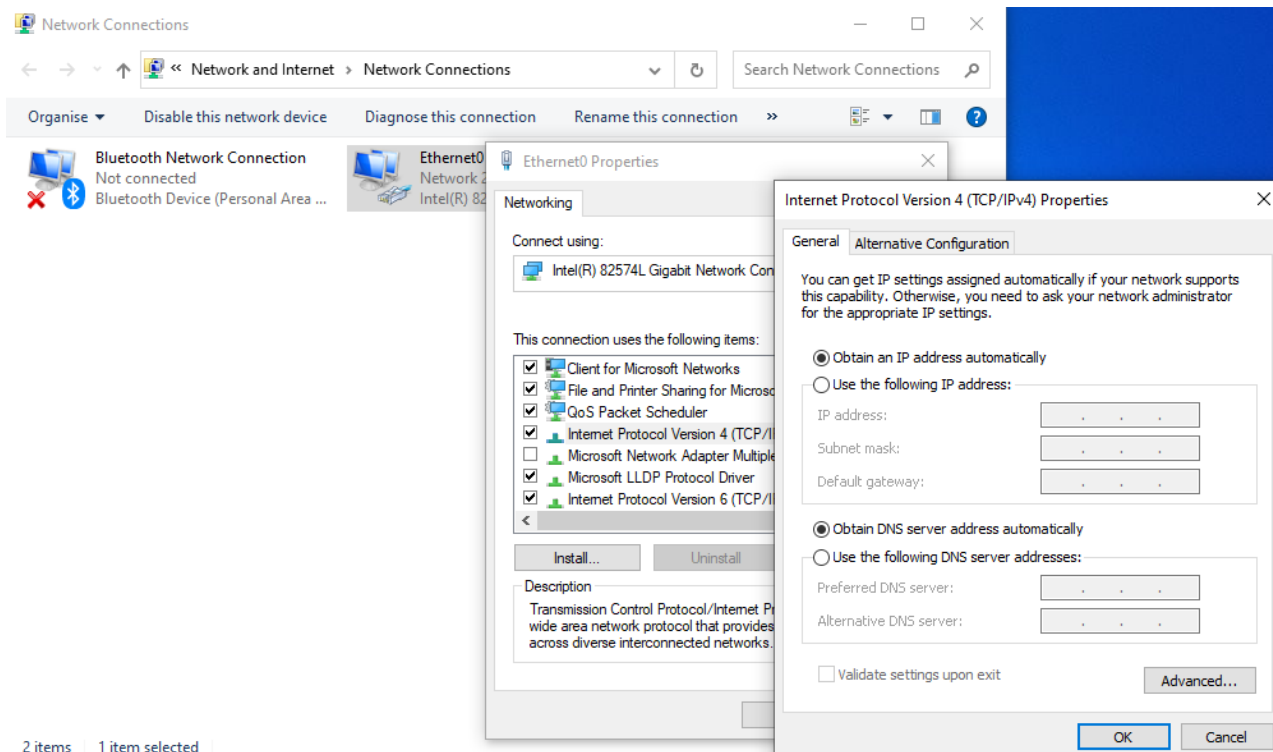


3.4 Configuring Clients to Obtain IP via DHCP

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:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ghaza>ipconfig /release
Windows IP Configuration

No operation can be performed on Bluetooth Network Connection while it has its media disconnected.

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::bd69:6834:a0a3:43d7%6
    Default Gateway . . . . . : 

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

C:\Users\ghaza>ipconfig /renew
Windows IP Configuration

No operation can be performed on Bluetooth Network Connection while it has its media disconnected.

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::bd69:6834:a0a3:43d7%6
    IPv4 Address. . . . . : 192.168.20.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.20.1

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

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.

```
ghazal@Client: ~  
ghazal@Client:~$ sudo nano /etc/netplan/01-netcfg.yaml
```

```
GNU nano 6.2 /etc/netplan/01-netcfg.yaml  
network:  
  version: 2  
  renderer: networkd  
  ethernets:  
    ens33:  
      dhcp4: yes
```

```
ghazal@Client: ~  
ghazal@Client:~$ sudo netplan apply
```

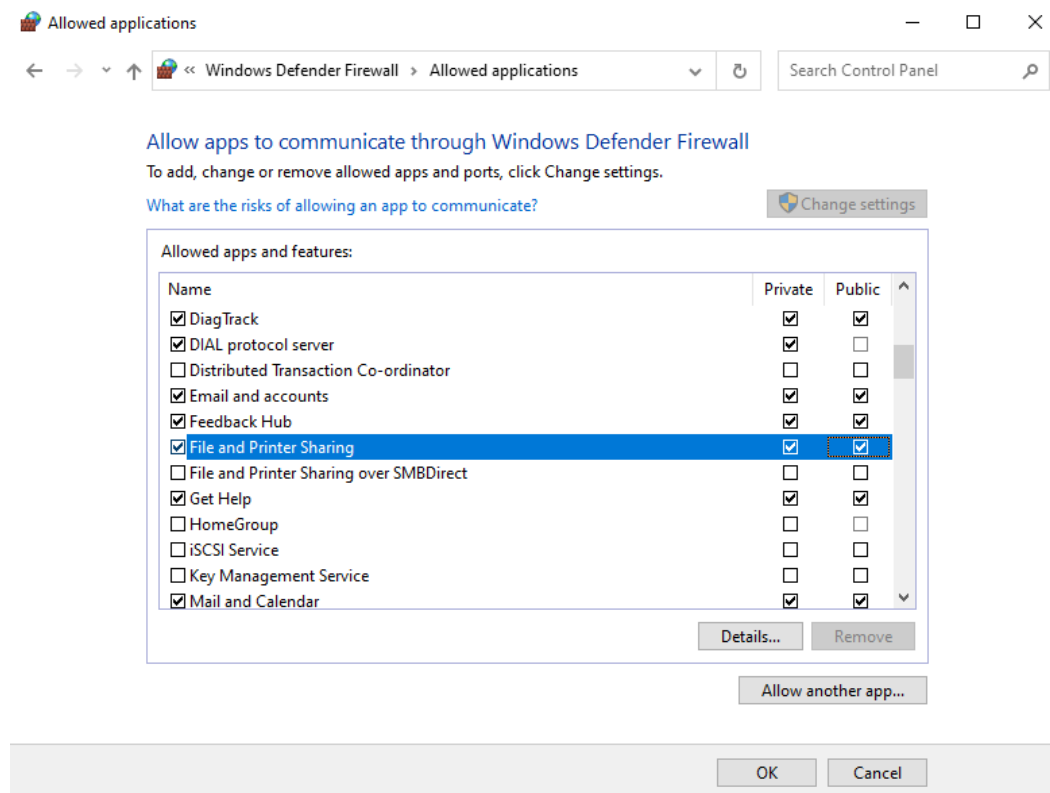
```
ghazal@Client: ~  
ghazal@Client:~$ ip addr show ens33  
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:0c:29:2c:0d:c6 brd ff:ff:ff:ff:ff:ff  
    altname enp2s1  
    inet 192.168.10.100/24 brd 192.168.10.255 scope global dynamic noprefixroute ens33  
        valid_lft 337sec preferred_lft 337sec  
    inet6 fe80::20c:29ff:fe2c:dc6/64 scope link  
        valid_lft forever preferred_lft forever
```


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

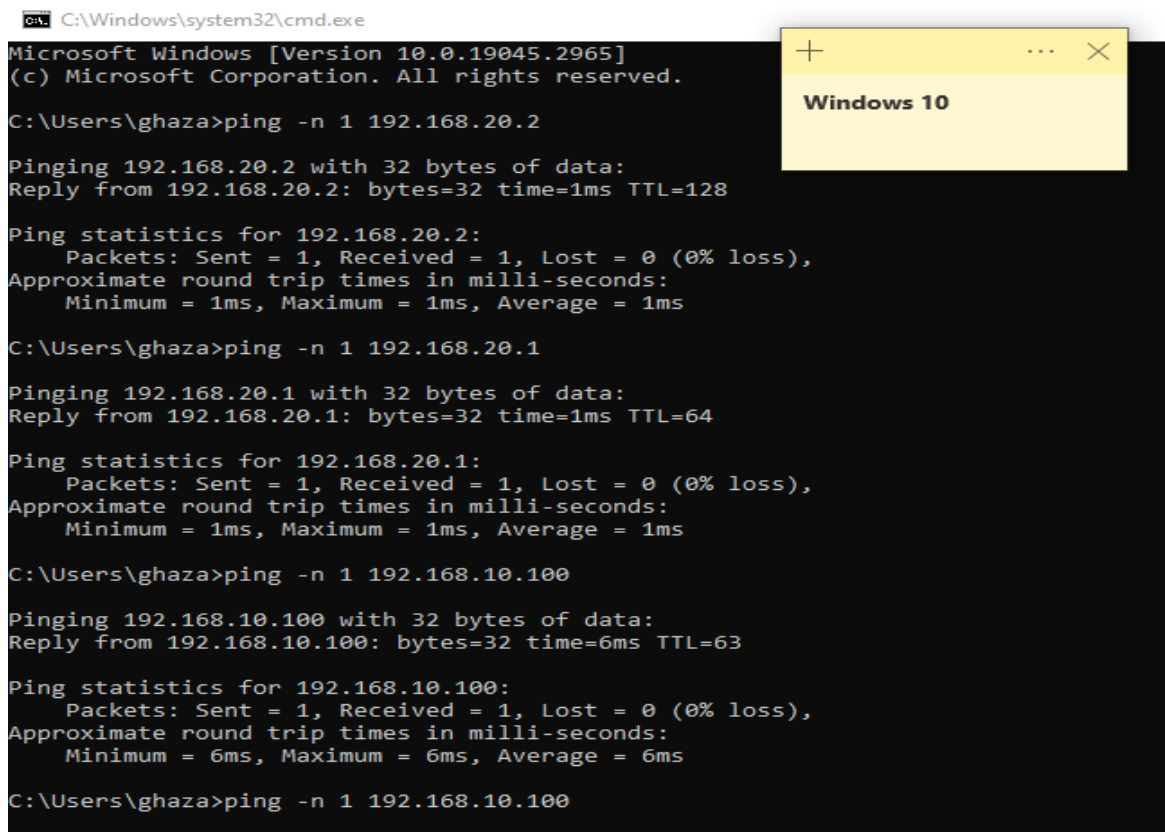
Thorough testing and validation are critical to ensure that the implemented virtual network functions as intended. This section outlines the methods and results of testing various network components, including connectivity, DHCP functionality, DNS resolution, NAT configuration, and performance metrics.

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:



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ghaza>ping -n 1 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:
Reply from 192.168.20.2: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.20.2:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Users\ghaza>ping -n 1 192.168.20.1

Pinging 192.168.20.1 with 32 bytes of data:
Reply from 192.168.20.1: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.20.1:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Users\ghaza>ping -n 1 192.168.10.100

Pinging 192.168.10.100 with 32 bytes of data:
Reply from 192.168.10.100: bytes=32 time=6ms TTL=63

Ping statistics for 192.168.10.100:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 6ms, Maximum = 6ms, Average = 6ms

C:\Users\ghaza>ping -n 1 192.168.10.100
```

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.20348.587]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ping -n 1 192.168.20.100

Pinging 192.168.20.100 with 32 bytes of data:
Reply from 192.168.20.100: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.20.100:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms

C:\Users\Administrator>ping -n 1 192.168.10.100

Pinging 192.168.10.100 with 32 bytes of data:
Reply from 192.168.10.100: bytes=32 time=2ms TTL=63

Ping statistics for 192.168.10.100:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms

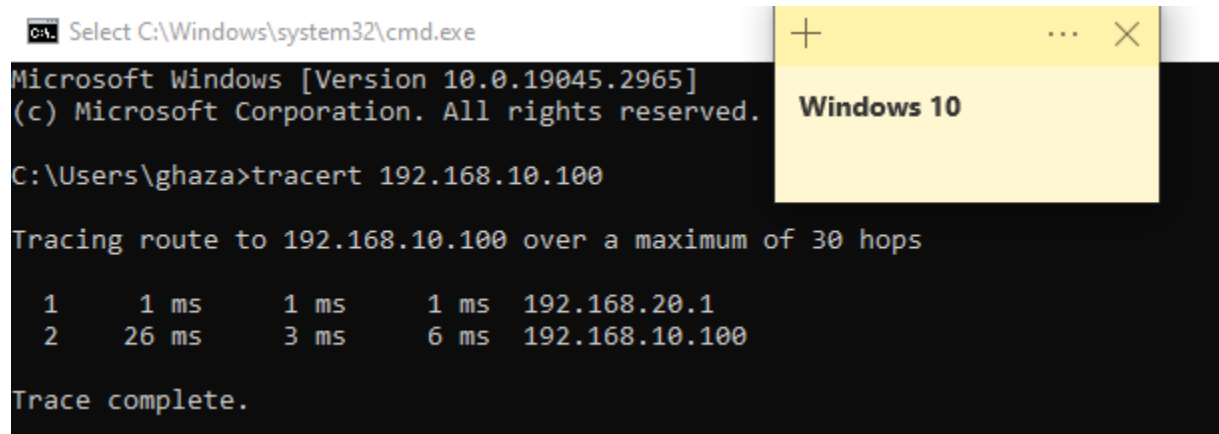
C:\Users\Administrator>ping -n 1 192.168.10.1

Pinging 192.168.10.1 with 32 bytes of data:
Reply from 192.168.10.1: bytes=32 time=4ms TTL=64

Ping statistics for 192.168.10.1:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 4ms, Average = 4ms
```

```
ghazal@Client: ~
ghazal@Client:~$ ping 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=127 time=5.45 ms
64 bytes from 192.168.20.2: icmp_seq=2 ttl=127 time=2.76 ms
^C
--- 192.168.20.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 2.756/4.103/5.451/1.347 ms
ghazal@Client:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=4.18 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=3.14 ms
^C
--- 192.168.10.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 3.135/3.655/4.176/0.520 ms
```

2. Traceroute Tests:



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ghaza>tracert 192.168.10.100

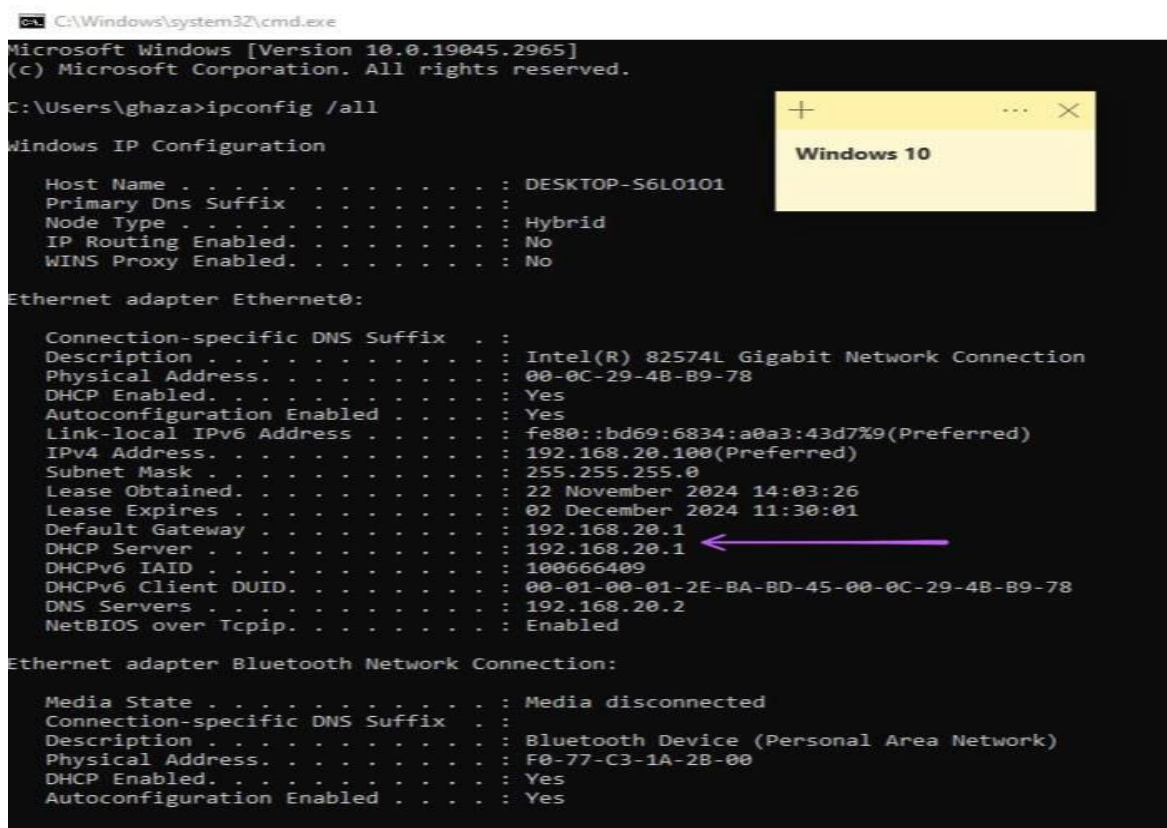
Tracing route to 192.168.10.100 over a maximum of 30 hops

  1      1 ms      1 ms      1 ms      192.168.20.1
  2     26 ms      3 ms      6 ms      192.168.10.100

Trace complete.
```

4.2 DHCP Functionality Verification

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



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ghaza>ipconfig /all

Windows IP Configuration

Host Name . . . . . : DESKTOP-S6L0101
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Ethernet0:

Connection-specific DNS Suffix . :
Description . . . . . : Intel(R) 82574L Gigabit Network Connection
Physical Address. . . . . : 00-0C-29-4B-B9-78
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::bd69:6834:a0a3:43d7%9(Preferred)
IPv4 Address. . . . . : 192.168.20.100(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 22 November 2024 14:03:26
Lease Expires . . . . . : 02 December 2024 11:30:01
Default Gateway . . . . . : 192.168.20.1
DHCP Server . . . . . : 192.168.20.1
DHCPv6 IAID . . . . . : 100666409
DHCPv6 Client DUID. . . . . : 00-01-00-01-2E-BA-BD-45-00-0C-29-4B-B9-78
DNS Servers . . . . . : 192.168.20.2
NetBIOS over Tcpip. . . . . : Enabled

Ethernet adapter Bluetooth Network Connection:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Bluetooth Device (Personal Area Network)
Physical Address. . . . . : F0-77-C3-1A-2B-00
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
```

```
ghazal@Client: ~  
ghazal@Client:~$ ifconfig  
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.10.100 netmask 255.255.255.0 broadcast 192.168.10.255  
    inet6 fe80::20c:29ff:fe2c:dc6 prefixlen 64 scopeid 0x20<link>  
    ether 00:0c:29:2c:0d:c6 txqueuelen 1000 (Ethernet)  
    RX packets 109 bytes 54034 (54.0 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 247 bytes 31051 (31.0 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 133 bytes 11644 (11.6 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 133 bytes 11644 (11.6 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

4.3 DNS Resolution Testing

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

```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 10.0.19045.2965]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\ghaza>nslookup Router  
Server: dnsserver.mydomain.local  
Address: 192.168.20.2  
  
Name: Router.mydomain.local  
Address: 192.168.10.1  
  
C:\Users\ghaza>ping -n 1 windows10  
  
Pinging windows10.mydomain.local [192.168.20.100] with 32 bytes of data:  
Reply from 192.168.20.100: bytes=32 time<1ms TTL=128  
  
Ping statistics for 192.168.20.100:  
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
C:\Users\ghaza>ping -n 1 dnsserver.mydomain.local  
  
Pinging dnsserver.mydomain.local [192.168.20.2] with 32 bytes of data:  
Reply from 192.168.20.2: bytes=32 time=1ms TTL=128  
  
Ping statistics for 192.168.20.2:  
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

```
ghazal@Client: ~  
ghazal@Client:~$ nslookup windows10  
Server:      127.0.0.53  
Address:     127.0.0.53#53  
  
Non-authoritative answer:  
Name:   windows10.mydomain.local  
Address: 192.168.20.100  
  
ghazal@Client:~$ ping -c 1 Router  
PING Router.mydomain.local (192.168.10.1) 56(84) bytes of data.  
64 bytes from Router.mydomain.local (192.168.10.1): icmp_seq=1 ttl=64 time=1.25  
ms  
  
--- Router.mydomain.local ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 1.251/1.251/1.251/0.000 ms  
ghazal@Client:~$ ping -c 1 dnsserver  
PING dnsserver.mydomain.local (192.168.20.2) 56(84) bytes of data.  
64 bytes from dnsserver.mydomain.local (192.168.20.2): icmp_seq=1 ttl=127 time=3  
.91 ms  
  
--- dnsserver.mydomain.local ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 3.913/3.913/3.913/0.000 ms
```

4.4 NAT and Internet Access Testing

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

Test Internet Connectivity:

```
ghazal@Client: ~  
ghazal@Client:~$ ping 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=127 time=247 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=127 time=221 ms  
^C  
--- 8.8.8.8 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1003ms  
rtt min/avg/max/mdev = 221.420/234.188/246.956/12.768 ms
```

C:\Windows\system32\cmd.exe

Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ghaza>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:

Reply from 8.8.8.8: bytes=32 time=231ms TTL=127

Reply from 8.8.8.8: bytes=32 time=283ms TTL=127

Reply from 8.8.8.8: bytes=32 time=171ms TTL=127

Reply from 8.8.8.8: bytes=32 time=445ms TTL=127

Ping statistics for 8.8.8.8:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 171ms, Maximum = 445ms, Average = 282ms



Windows 10

Administrator: C:\Windows\system32\cmd.exe

Microsoft Windows [Version 10.0.20348.587]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:

Reply from 8.8.8.8: bytes=32 time=228ms TTL=127

Reply from 8.8.8.8: bytes=32 time=147ms TTL=127

Reply from 8.8.8.8: bytes=32 time=162ms TTL=127

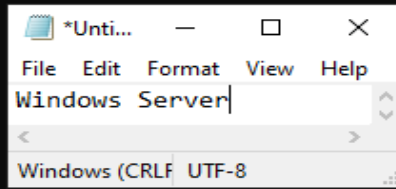
Reply from 8.8.8.8: bytes=32 time=207ms TTL=127

Ping statistics for 8.8.8.8:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 147ms, Maximum = 228ms, Average = 186ms



Verify NAT Functionality on Router:

```
ghazal@Ubuntu: ~  
ghazal@Ubuntu:~$ sudo iptables -t nat -L -n -v  
Chain PREROUTING (policy ACCEPT 2850 packets, 226K bytes)  
  pkts bytes target    prot opt in     out     source        destination  
  
Chain INPUT (policy ACCEPT 266 packets, 48920 bytes)  
  pkts bytes target    prot opt in     out     source        destination  
  
Chain OUTPUT (policy ACCEPT 980 packets, 82275 bytes)  
  pkts bytes target    prot opt in     out     source        destination  
  
Chain POSTROUTING (policy ACCEPT 822 packets, 63404 bytes)  
  pkts bytes target    prot opt in     out     source        destination  
 2702  191K MASQUERADE all  --  *      ens33  0.0.0.0/0     0.0.0.0/0
```


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 according