

Network Operating System and Services
Coursework 1
Research, build, and deploy, a Virtualised network

Table of contents

Table of figures.....	5
Abstract.....	6
Introduction	7
I – Research	8
Hypervisor	8
Introduction	8
Architecture	9
Features.....	9
Hardware requirements.....	10
Advantages	10
Scalability, security and compatibility	10
IP addressing	11
Introduction	11
IP address structure and format.....	12
Data transmission	13
Subnetting.....	13
Efficient utilisation.....	13
Routing.....	14
Introduction	14
Fundamental concepts of routing protocols	15
Static and dynamic routing.....	15
How routing protocols are deployed in a real time network	15
Benefits and challenges on routing protocols.....	16
Network Services	17
Introduction	17
VPNs	18
DHCP	18
Active Directory Domain Services.....	18
Certificate Authorities	18
II – Technical documentation.....	19
Choice of hypervisor.....	19
Download and Setup.....	20
Installation and configuration (Windows)	20
Network deployment and troubleshooting (Windows)	21
Installation and configuration (Debian Linux)	23

Network deployment and troubleshooting (Debian Linux)	26
III – Reflection	30
Technical reflection	30
Personal reflection.....	30
Conclusion.....	31
Appendix A: The OSI Model and network layers	32
Appendix B: Network classification and addressing	33
IP address classes	33
Private and public IP addresses	33
References	34

Table of figures

Figure 1 – Type 1 vs Type 2 hypervisors.....	9
Figure 2 – Table with hardware requirements for each type of hypervisors	10
Figure 3 – Format of IPv4 address	12
Figure 4 – Routing protocols.....	15
Figure 5 – Table of the benefits and the challenges of different protocols.....	16
Figure 6 – Different protocols	16
Figure 7 – S-DC installed on VirtualBox	20
Figure 8 – ClientS1 installed and running.....	20
Figure 9 – Successful ping response from S-DC to ClientS1.....	21
Figure 10 – Ping from ClientS1 to S-DC	21
Figure 11 – Enable rule to allow ICMP traffic	22
Figure 12 – Switching off the firewall profiles	22
Figure 13 – Successful ping response from ClientS1 to S-DC.....	23
Figure 14 – Debian server installation.....	23
Figure 15 – Debian server installed.....	24
Figure 16 – Text written in the wrong file.....	24
Figure 17 – File edited to define IP address	25
Figure 18 – IP addresses assigned	25
Figure 19 – Ping successful from Windows to Debian	26
Figure 20 – Ping successful from Debian to Windows	26
Figure 21 – IP address not assigned	27
Figure 22 – IP address being configured.....	27
Figure 23 – File modified.....	28
Figure 24 – IP address assigned	28
Figure 25 – Ping successful from Debian server to Ubuntu.....	29
Figure 26 – Ping successful from Ubuntu to Debian server.....	29
Figure 27 – Table of IP address class	33

Abstract

This coursework is structured into three tasks. Each task focuses on specific aspects, researching, building, and finally deploying the virtualised network.

The first task is a research phase. It explores key concepts including hypervisors, IP addressing, routing protocols and network services. It includes a comparison of different types of hypervisors. Furthermore, the study explores the role of IP addressing in data transmission, the structure and format of IPv4 and IPv6, subnetting techniques, and the impact of routing protocols on network efficiency and scalability.

The second task focuses on the technical implementation of the virtualised network across different operating systems, including Windows and Debian Linux. This phase includes selecting and configuring the chosen hypervisor, downloading and setting up virtual machines, and establishing an appropriate IP addressing scheme. Network connectivity is tested, and troubleshooting procedures are documented to address any encountered issues, such as firewall restrictions or misconfigured network parameters.

The final task is a reflective analysis of the implementation process. It highlights both technical and personal challenges encountered during network deployment. It discusses the various obstacles faced, such as connectivity errors, IP conflicts.

Through these three tasks, this coursework aims to develop a comprehensive understanding of virtualised network environments while strengthening practical skills in network configuration, troubleshooting, and system administration.

Introduction

In a progressively interconnected world each day, enterprises require high-performance, adaptive network infrastructure. Virtualization is important for effective optimization methods for optimum utilisation of hardware resources and simplification of networks. The current report is a part of the "Network Operating Systems and Services" subject and has tried to delve into and comprehend the essential steps to examine, design, and implement a virtualized network. Through the detailed study of hypervisors, IP addressing, and routing protocols, we are going to identify best practices ensuring that connectivity is optimised, and security is guaranteed. Additionally, an in-depth technical report will expose the actualization of the network in the real world, combined with a critique of the encountered challenges and implementation solutions.

I – Research

Hypervisor

Introduction

Virtualization has transformed modern computing by enabling multiple operating systems to run on a single physical machine. This technology allows a computer to divide and share its hardware resources across multiple separate digital environments [1]. Each virtual machine works within its allocated resources, including memory, CPU, and storage. Thanks to virtualization, companies can switch between different operating systems on a single server without rebooting. Virtualization offers great benefits for business or development environment. It offers great flexibility and improves efficiency [2]. Virtual machines (VMs) and hypervisors are two important concepts in virtualization. This report explores the different types of hypervisors, focusing on their architectures, features, and hardware requirements. It also highlights the advantages of using hypervisors, their scalability, security considerations, and compatibility with different environments.

A hypervisor is a software component which creates and runs VMs. It allows a host computer to support different guest VMs. This software layer is an intermediate between the virtual machine and the hardware. For example, when the VM needs something, the request first goes to the hypervisor, which then passes it to the underlying hardware. There are two types of Hypervisors: Type 1 and Type 2 [1].

Architecture

A type 1 hypervisor, or bare-metal hypervisor, is a hypervisor program installed directly on top of the hardware. It manages VMs and allocates resources efficiently. The kernel handles VM management. It interacts with the hardware through device drivers and a management console for system administration [3].

A type 2 hypervisor – hosted hypervisor – interacts with the underlying host machine hardware through the host machine's OS. It negotiates with the OS to obtain underlying system resources. The host operating system prioritizes its own functions and applications over the virtual workloads.

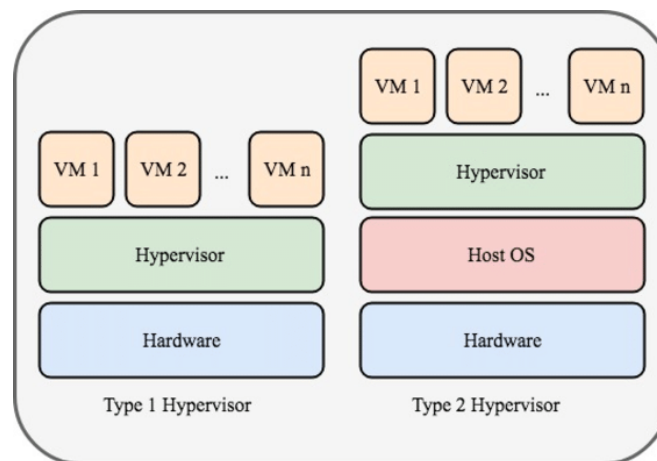


Figure 1 – Type 1 vs Type 2 hypervisors

Features

A type 1 hypervisor improves virtualization with dynamic CPU, memory, and storage allocation. It supports VM snapshots, cloning, and virtualized storage and networking.

For a type 2 hypervisor, it installs easily like software without requiring specialised hardware. It efficiently allocates CPU, memory, and storage, supports multiple OSes, and provides user-friendly graphical interfaces for simplified VM management.

Hardware requirements

The type 1 hypervisors require robust hardware to support multiples VMs. The type 2 hypervisor has more flexible hardware requirements and runs on standard desktops/laptops with moderate resources.

Type 1 hypervisor – taking Hyper-V on Windows (according to Microsoft website) [4]	Type 2 hypervisor – taking VirtualBox (official website) [5]
➤ 64-bit processor with SLAT	➤ Processor: any x86 compatible processor from Intel / AMD
➤ VM Monitor Mode extensions	➤ SSD disk: minimum 10GB
➤ Memory: 4 GB of RAM – more is better	➤ Memory: minimum 1GB + RAM as required by running guests
➤ Virtualization support enabled in the BIOS / UEFI: - Hardware-assisted virtualization – processors with Intel VT / AMD-V technology - Hardware-enforced Data Execution Prevention: enabled	➤ Windows, Linux, Mac OS X, Oracle Solaris

Figure 2 – Table with hardware requirements for each type of hypervisors

Advantages

For type 1 hypervisor, there is a direct access to the hardware which gives the best performance. Furthermore, they can implement their own custom resource allocation strategies to service their VMs [1].

For type 2 hypervisor, it is easier to manage and can be installed on a wider range of hardware, it is easily deployed. The user can manage it as an application on an operating system, it is a simple setup [1].

Scalability, security and compatibility

The type 1 hypervisors are very secure. In fact, there is a strong isolation with VMs which prevents malware. They can also scale to support thousands of VMs. They are well suited to host large and complex workloads and can support lots of guest operating systems, (Windows, Linux, macOS) [1] while type 2 hypervisors support a wide range of operating systems as guest VMs, providing compatibility with diverse software environments. It is less secure than type 1 hypervisor because it depends on the host OS.

Popular examples of type 1 hypervisors include VMware ESXi, Hyper-V, and KVM and we have VirtualBox, and Microsoft Virtual PC for type 2 hypervisors.

IP addressing

Introduction

On the Internet, information is broken down into packets that travel across different networks before being reassembled at their destination. These packets contain important details, such as the IP (Internet Protocol) addresses of the sender and recipient, to ensure that they arrive at the right place [15].

IP addressing is a fundamental part of the communication network and is primarily managed at the Network Layer of the OSI model. The Network Layer is responsible for logical addressing, routing, and forwarding of packets across different networks. Each device connected to a network is assigned a unique IP address, which identifies it and allows it to exchange data smoothly. This part of the report explores the structure of IP addressing with the representation of IPv4 and IPv6 addressing, the basics of subnetting, and ways to optimise network efficiency. Additionally, it examines the difference between public and private IP addresses.

For a more detailed explanation of the OSI model and its layers, please refer to Appendix A.

IP address structure and format

An IP address is a unique numerical identifier assigned to every device connected to a network. It allows communication between systems on the Internet. There are two main parts: the Network Identifier which defines the network to which the device belongs, and the Host Identifier that identifies the specific device within that network.

There are two principal versions of IP addressing: IPv4 and IPv6.

IPv4 (Internet Protocol version 4) Addressing

The IPv4 address is the older one. It has space for up to 4 billion IP addresses and it is assigned to all computers [9]. IPv4 addresses use a 32-bit structure, divided into four sets of numbers, and separated by periods (e.g., 192.168.1.0). Each set, called an octet, represents 8 bits, and can be between 0 and 255. This means that an IPv4 is always going from 0.0.0.0 to 255.255.255.255. Their division into network and host depends on the network class. IPv4 addresses are categorised into five classes (A, B, C, D, E). Each class determines the size of the network, and how many devices it can accommodate [9].

Refer to the appendix B for more details on the classes and additional information.

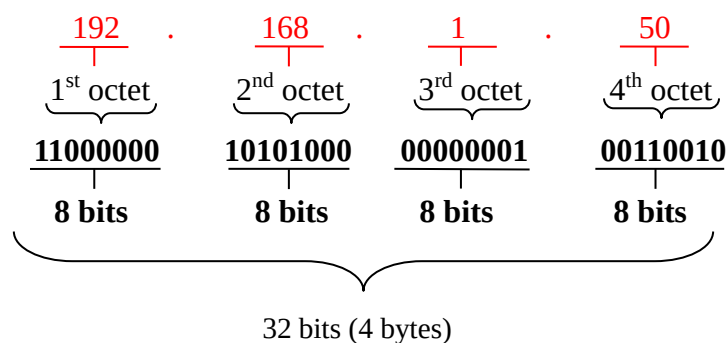


Figure 3 – Format of IPv4 address

Due to IPv4's limited address space, a more scalable solution was needed. Therefore, IPv6 was developed. It offers a larger pool of addresses and improved efficiency.

IPv6 (Internet Protocol version 6) Addressing

IPv6 is more recent and has space for trillions of IP addresses. It has 128-bit addresses and it is written in hexadecimal notation, separated by colons (e.g., 2001:db8::ff00:42:8329). It has a larger address space (~340 undecillion), that ensures long-term scalability [9]. IPv6 reduces the need of NAT (Network Address Translation) and enables direct communication. This improves efficiency, security and performance [8] [9].

Data transmission

When data is sent, it is broken up into packets. Each packet includes a header containing the source and destination IP addresses, which guide the packet to its recipient. These packets travel through multiple networks via routers and switches that determine the best path. The packets are then reassembled. This process ensures efficient delivery. Networks are divided into subnets to optimise routing [6].

Subnetting

Subnetting is the process of dividing a larger network into smaller, subnetworks (subnets). This technique improves the security by isolating some segments within a network and reduces network congestion [7].

Subnet Mask

A subnet mask defines which part of an IP address is the network and which is the host, using a series of bits to separate them.

Efficient utilisation

Subnetting allows a better IP address allocation, preventing waste of IP addresses, especially in large networks. It can also improve network performance ensuring that data packets travel a shorter distance, reducing unnecessary routing.

Routing

Introduction

Routing is the process of selecting the best path across one or multiple networks. It applies to many networks, from telephone networks to transportation. On the Internet, the main device that routes data is the router. They decide how IP packets move across networks to reach their destination. They are essential to ensure efficient communication and to maintain network performance and reliability. This part of the report explores the core concepts of routing protocols, their purpose, and functionality.

Fundamental concepts of routing protocols

Routing protocols are essential for directing data packets across networks. These protocols adapt to network changes by maintaining and updating routing tables. This optimises traffic flow and prevents congestion [10]. Routers receive IP packets, decide where to send them, and send them to their destinations. This process is called IP routing. The router will examine the routing table to determine the interface through which the packet is to be sent. There is a routing table in every router [10].

Two type of routing protocols are required: Interior Gateway Protocols (IGP), “*used for routing within an AS*” [11], and Exterior Gateway Protocols (EGP), which manage routing between autonomous systems [11]. Different protocols use various metrics such as hop count, bandwidth, and delay to select the most efficient routes. Routing can be static or dynamic. [13]

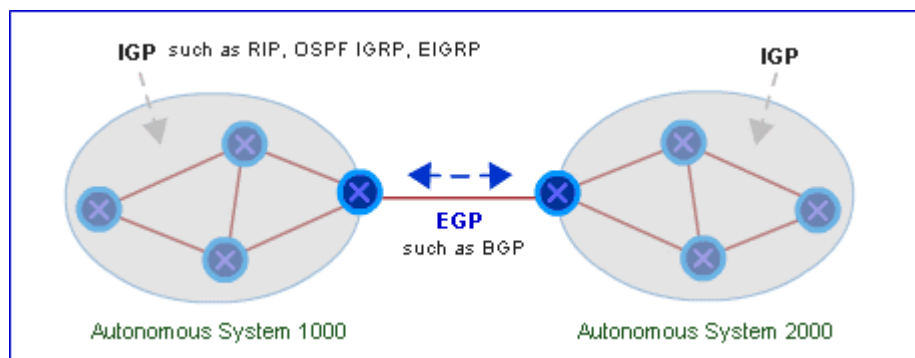


Figure 4 – Routing protocols

Static and dynamic routing

Static routes “*are not automatically updated and must be manually reconfigured* – by network administrators – *if the network topology changes*” [11]. Entries are fixed. They are suitable for small networks like a small office network. Dynamic routing uses algorithms to “*automatically exchange information in order to create entries for the routing tables*” [11]. They are more suitable for larger network, and network with a lot of changes.

Static routing provides manual control, stability, and security. However, it lacks scalability and requires manual updates. This makes it impractical for frequently changing networks. In contrast, dynamic routing offers scalability, and automatic adaptability to network changes but comes with higher resource consumption and it is more complex [11].

How routing protocols are deployed in a real time network

BGP (Border Gateway Protocol) is used by ISPs and cloud providers like Google and AWS (Amazon Web Service) to manage internet traffic between autonomous systems, dynamically adjusting routes based on congestion and availability. Inside corporate networks, OSPF (Open Shortest Path First) is deployed to enable real-time route recalculations, ensuring minimal

downtime. For example, Facebook uses BGP to reroute traffic instantly during network failures, it allows to maintain the availability of the service [13].

Benefits and challenges on routing protocols

Different network scenarios need different routing protocols. For example, if we need to run a big global network, we need BGP to manage internet traffic between ISPs, but it is tough to configure and maintain. OSPF is great for larger hierarchical networks because it recalculates quickly routes when a link fails, however it can be complex in large environments [10]. EIGRP is fast and efficient, but it is mostly limited to Cisco devices. A simple one is RIP, but it struggles with modern network demands [12]. Every protocol has benefits and challenges, the thing is to choose the right one for the right environment [17].

PROTOCOLS	BENEFITS	CHALLENGES
BGP	Big global network, more scalable and flexible	Complex configuration and to maintain
OSPF	Fast convergence, large network	Complex in large environment
EIGRP	Quick updates, efficient, scalability	Limited to Cisco devices
RIP	Simple	Struggles with modern network

Figure 5 – Table of the benefits and the challenges of different protocols

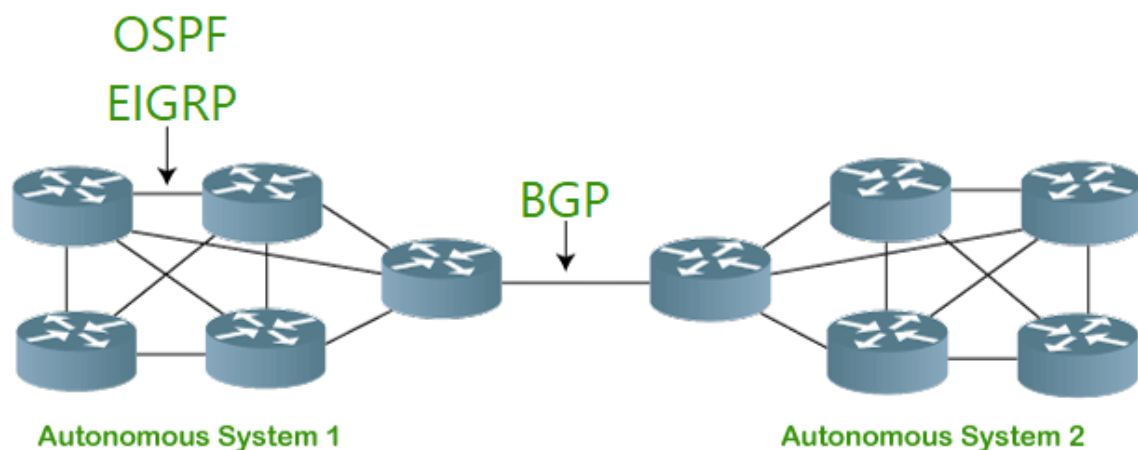


Figure 6 – Different protocols

Network Services

Introduction

Network services are fundamental to modern IT environments. They enable secure communications, automated network management, and user authentication. This part of the report provides an overview of different network services, such as VPN, DHCP, AD DS, and Certificate Authorities. It summarizes the key concepts and features associated with these services.

VPNs

VPN (Virtual Private Network) is a network architecture that secures internet communication by creating an encrypted tunnel between a device and a remote network. The technologies and protocols on which a VPN is built determine how it works [15] [16]. Key protocols include OpenVPN (flexible, open source) and IPsec (widely used for corporate VPNs). Best practices for VPNs include using strong encryption (AES-256), implementing multi-factor authentication (MFA), enabling a kill switch to prevent data leaks, and regularly updating VPN software.

DHCP

DHCP (Dynamic Host Configuration Protocol) is the automatic assignment of IP addresses and other communication parameters to connected devices. It operates at the application layer, dynamically assigning IP addresses and network configurations (subnet mask, gateway, DNS) to clients. It follows a client-server model, where clients request an address, and the server assigns one from its pool with a lease duration. The best practices are to secure DHCP servers and to use IP exclusions to prevent conflicts [19].

Active Directory Domain Services

AD DS manages users, groups, and resources in a Windows network. It provides centralized authentication, authorization, and policy management using protocols like LDAP and Kerberos. AD DS enables secure access control through Group Policy and Organizational Units (OUs). Best practices include maintaining redundant domain controllers and using strong password policies to enhance security and reliability [18].

Certificate Authorities

Certificate Authorities (CAs) issue and manage digital certificates that authenticate identities and secure communications. They operate within a hierarchy: root CAs (highly trusted, self-signed), intermediate CAs (issuing certificates), and end-entity certificates (used by websites or individuals). Secure private key storage and the use of the Online Certificate Status Protocol (OCSP) for real-time revocation checks help maintain security and trust [20].

II – Technical documentation

Choice of hypervisor

For this coursework, I chose to use a type 2 hypervisor, specifically Oracle VM VirtualBox. This decision is based on different key factors.

One of the arguments for using a type 2 hypervisor is because it is easy to install and to manage. Compared to type 1 hypervisors, type 2 hypervisors do not require dedicated hardware and complex configurations, they run on top of an existing operating system. This means that no specialized hardware is needed, and deployment can be done on standard machines without any additional setup complexity. Furthermore, it is cheaper. In general, type 2 hypervisor are more affordable than the type 1. They do not require licensing fees or specialised infrastructures. Additionally, Oracle VM VirtualBox is free and provides all the essential virtualization features at no cost. This makes it the perfect option for this project. The flexibility and compatibility offered by Oracle VM VirtualBox is good. Indeed, it supports a wide range of guest operating systems, like Linux, Windows, and MacOS. This will be particularly useful in Coursework 2, where we will need to manage different network services. VirtualBox offers also snapshot and cloning features, which make testing and troubleshooting easier. Finally, portability is another advantage of type 2 hypervisors. Unlike type 1 hypervisors, which are tied to specific hardware, type 2 hypervisors can be easily installed on different machines. This means that virtual machines can be transferred easily between computers by simply moving virtual disk files on a drive for example. It also simplifies backups.

If we consider all these benefits: ease of use, cost, flexibility and portability, the type 2 hypervisors are the most favourable option for this coursework. More specifically, Oracle VM VirtualBox ensures an adaptable and not complex solution.

Download and Setup

Installation and configuration (Windows)

To begin setting up the network infrastructure, I installed and configured a VM running Windows Server 2022 as the domain controller (S-DC). The Windows Server 2022 ISO file was downloaded from the official Microsoft website.

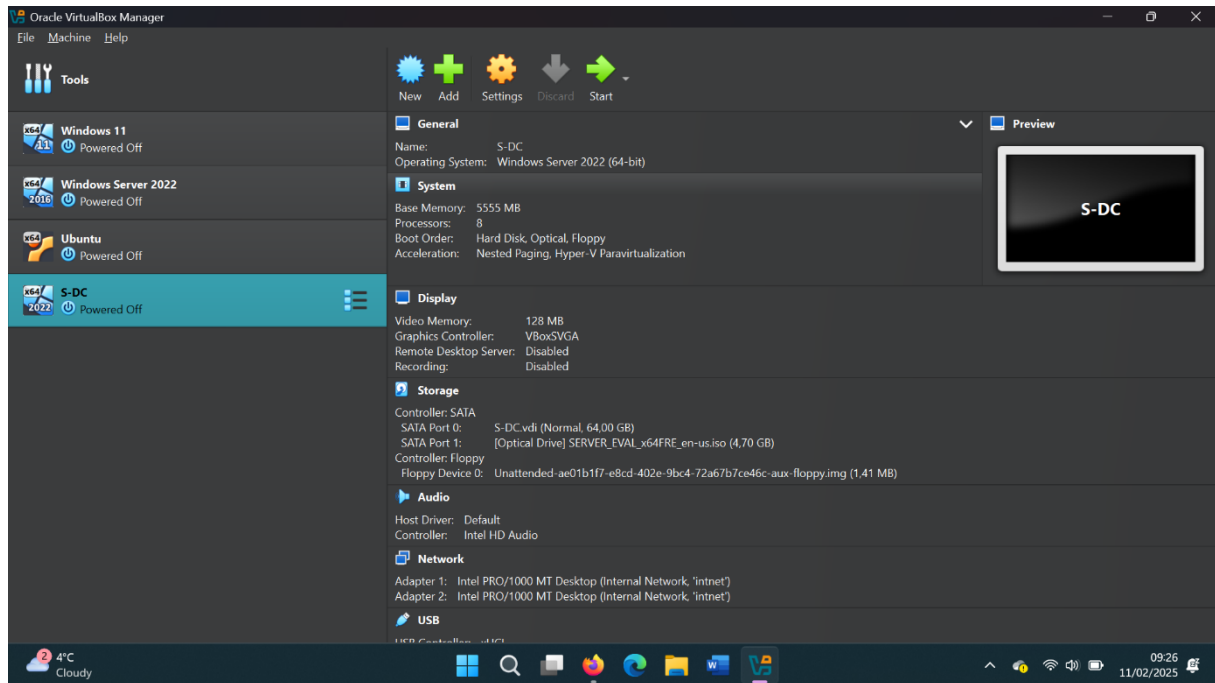


Figure 7 – S-DC installed on VirtualBox

After setting up the domain controller, a client machine named ClientS1 was deployed using Windows 10 as the operating system. The Windows 10 ISO file was downloaded from Microsoft's official website.

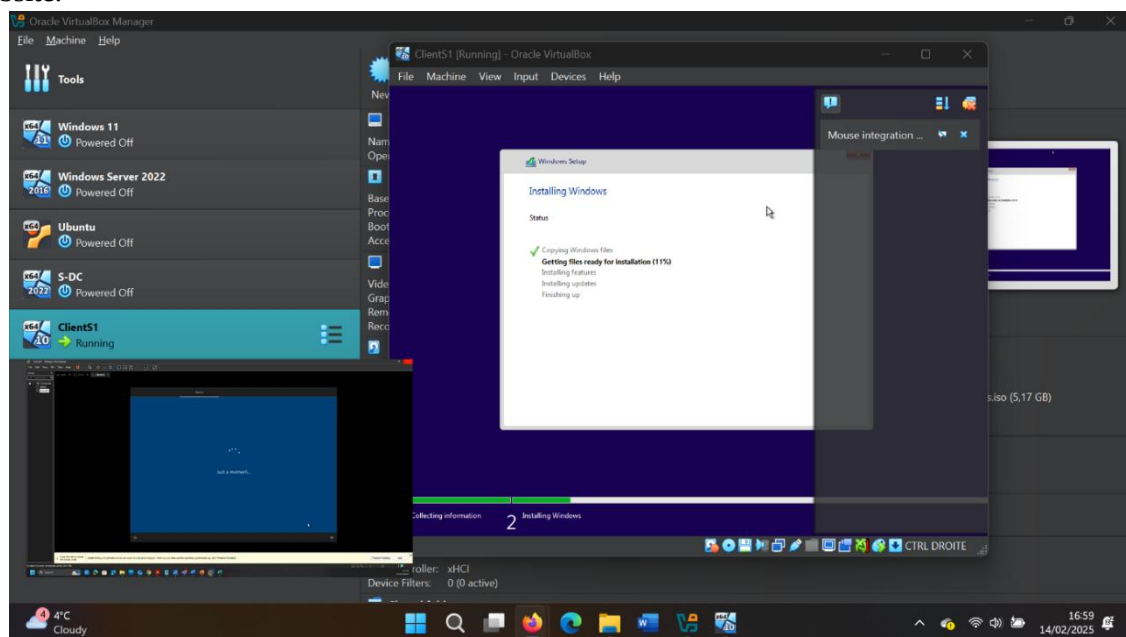


Figure 8 – ClientS1 installed and running

Network deployment and troubleshooting (Windows)

After deploying both S-DC and ClientS1, I configured the network settings to ensure proper communication between the machines. I manually assigned static IP addresses to both systems to align with the network topology. After that, a ping test was conducted to verify network connectivity between the machines. Initially, the ping command from S-DC to ClientS1 resulted in a "Request Timed Out", indicating that the client machine was not responding.

To resolve this issue, I troubleshooted it. I opened Windows Defender Firewall on ClientS1 and switched off all firewall profiles (domain profile, private profile, and public profile) because the Windows Defender Firewall on ClientS1 was blocking ICMP traffic. I retried the ping test from S-DC, and this time, the request was successfully received, confirming network connectivity.

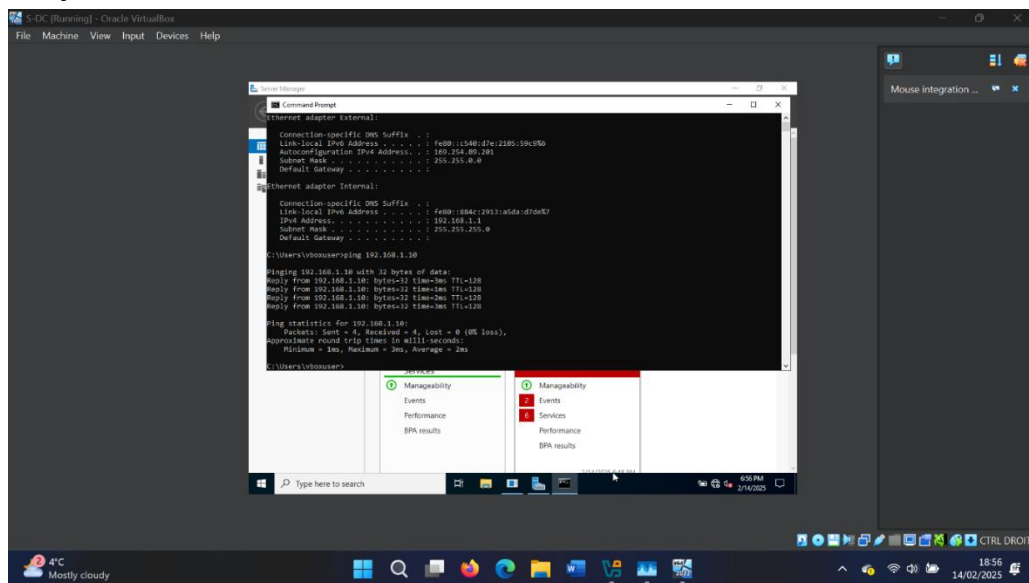


Figure 9 – Successful ping response from S-DC to ClientS1

After successfully verifying network connectivity from S-DC to ClientS1, I tried to do a reverse ping test from ClientS1 to S-DC. However, I had a "Request Timed Out", indicating that ClientS1 was not responding to ICMP requests.

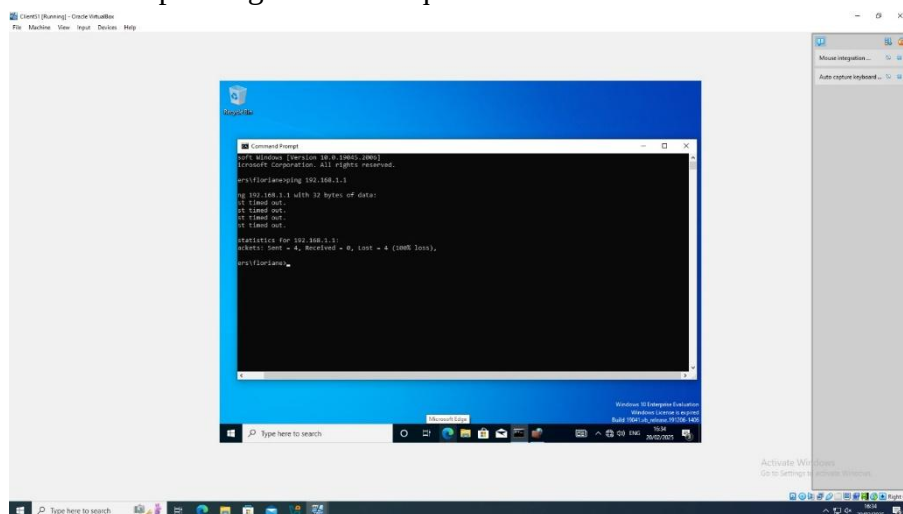


Figure 10 – Ping from ClientS1 to S-DC

Troubleshooting Steps

To resolve this issue, I opened Windows Firewall settings on ClientS1 and found that ICMP requests were being blocked.

1. Opened Windows Defender Firewall with Advanced Security on ClientS1.
2. Inbound Rules → File and Printer Sharing (Echo Request - ICMPv4-In)
3. Enabled rule to allow ICMP traffic.

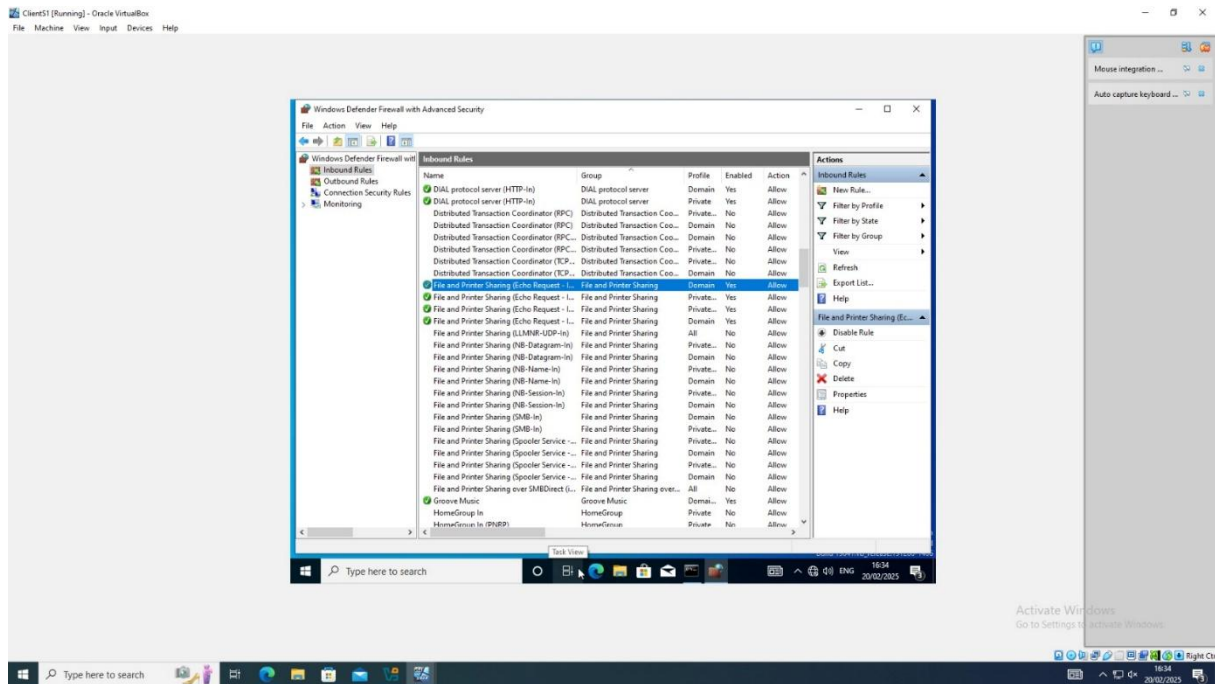


Figure 11 – Enable rule to allow ICMP traffic

After applying this change and switching off all firewall profiles, the ping test from S-DC to ClientS1 was successful. This confirms that both machines can now communicate over the network.

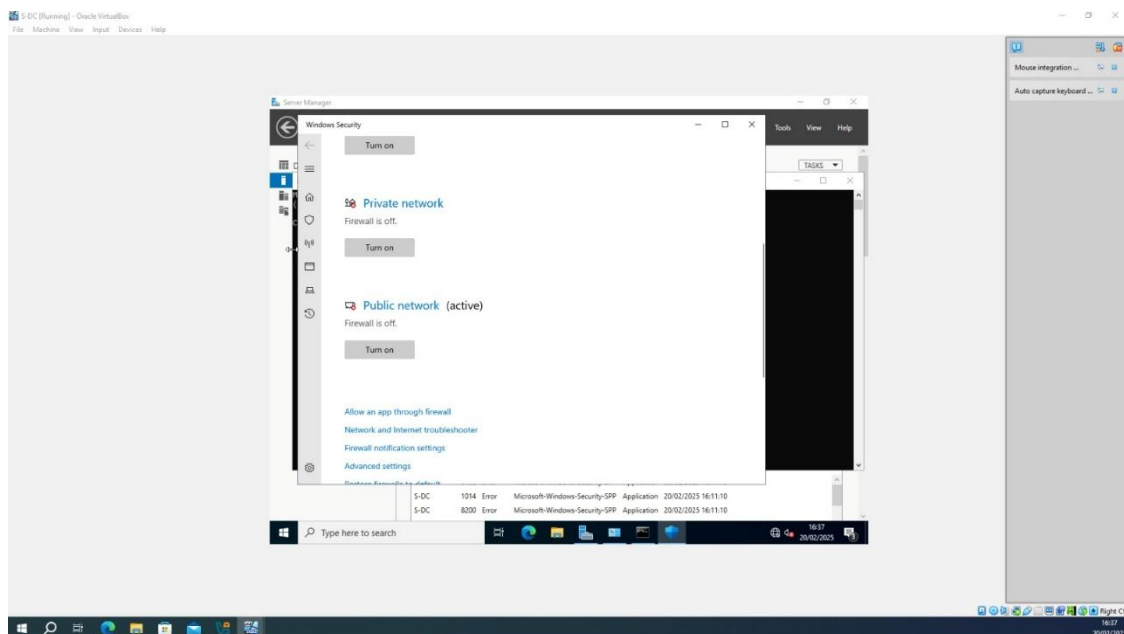


Figure 12 – Switching off the firewall profiles

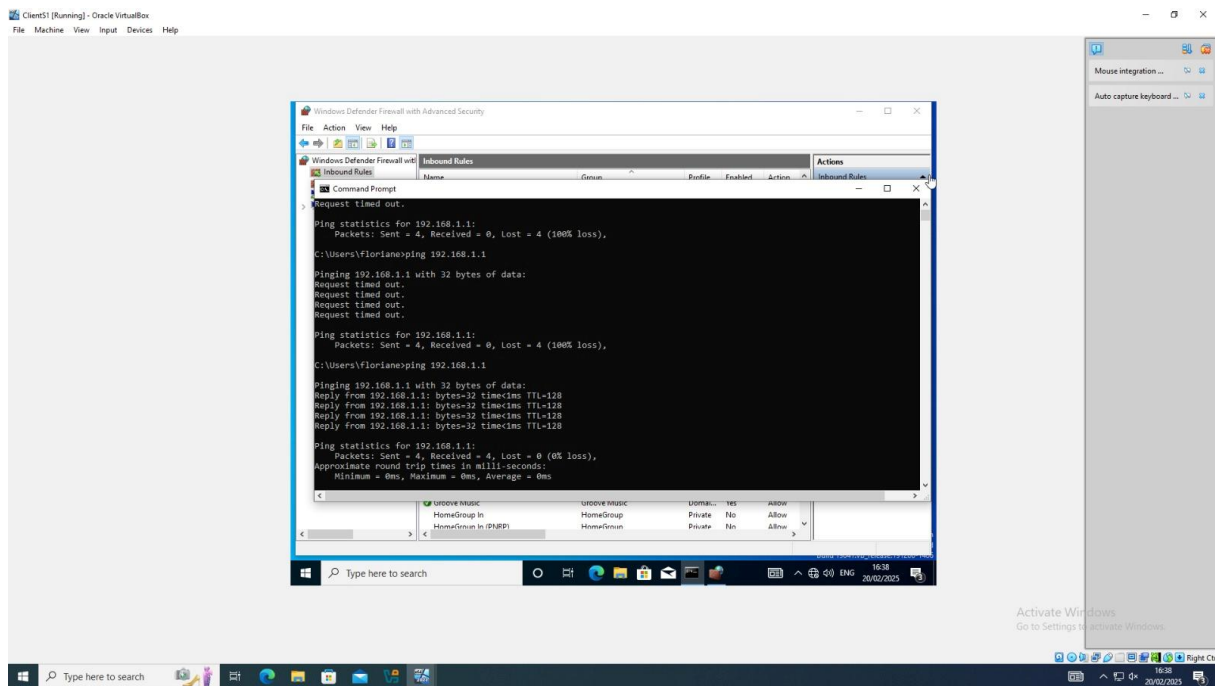


Figure 13 – Successful ping response from ClientS1 to S-DC

595

Installation and configuration (Debian Linux)

After setting up the Windows Server and Client and having a proper connectivity with successful pings, we now move on to Debian Linux. We have to install the Debian server. For the installation, I downloaded the Debian ISO DVD-1 for PC 64-bit from the official Debian website.

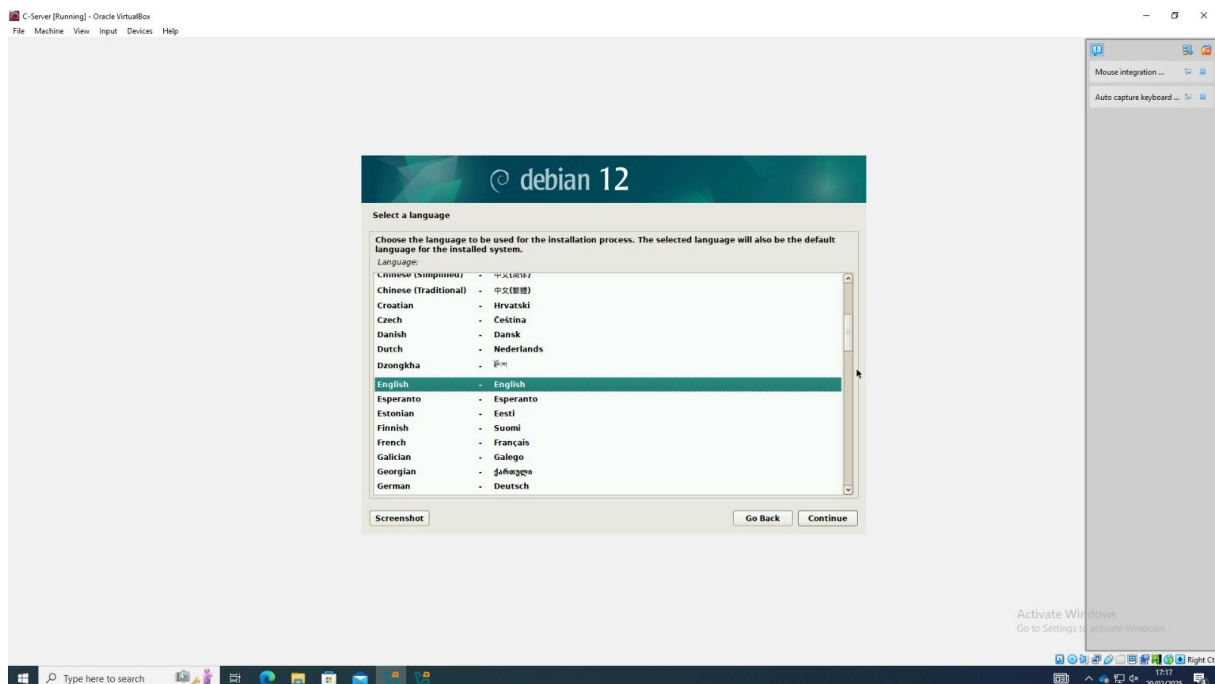


Figure 14 – Debian server installation

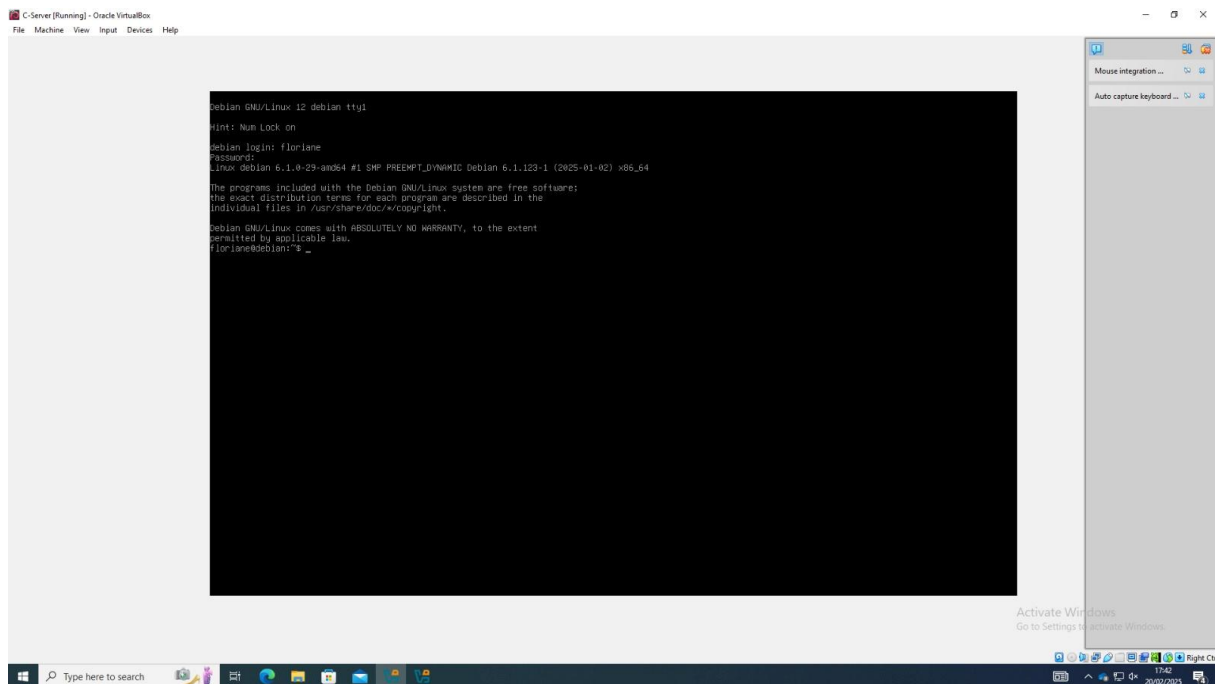


Figure 15 – Debian server installed

At first, only two internal networks were configured. Later, I added two more networks:

- NAT for internet access
- Host-only adapter for communication between the host and the vm

Encountered Issues

During the network setup, Debian required non-free firmware to properly recognize the network adapter. To install these missing firmware packages, it was necessary to add the "non-free" and "contrib" repositories to Debian's package manager sources. Modifications were made but in the wrong file. The correct file is `/etc/apt/sources.list`.

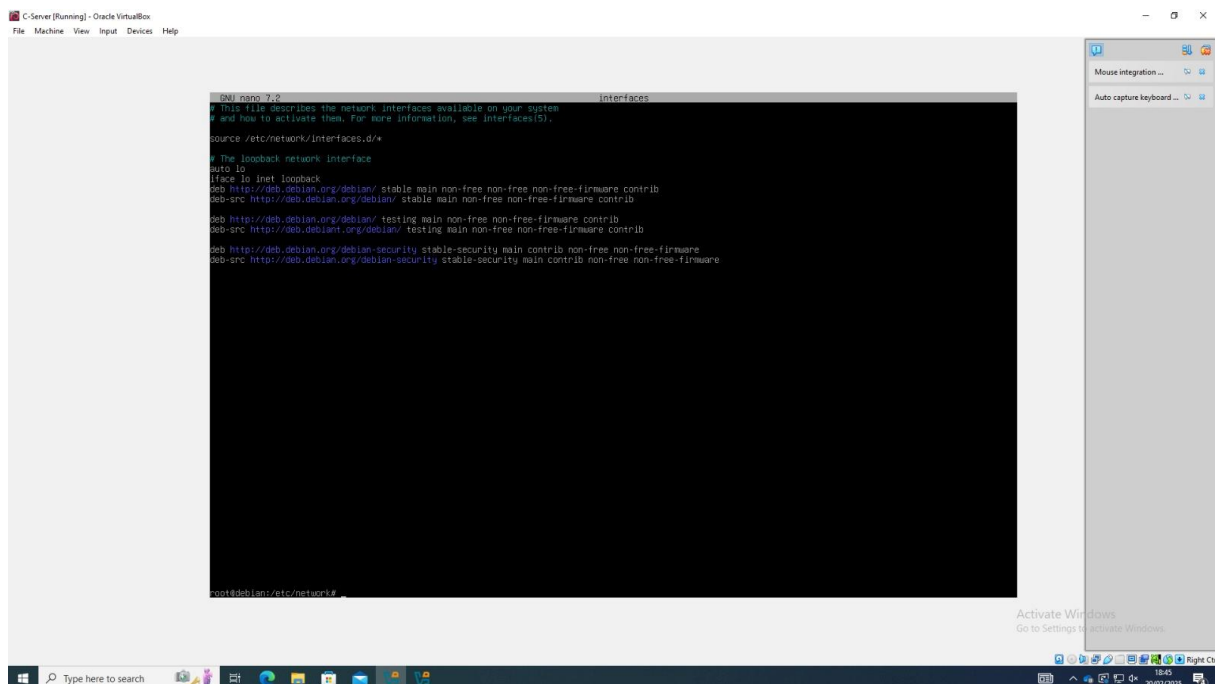


Figure 16 – Text written in the wrong file

I have configured the network settings. I used commands like “ip a” and “ipconfig” to display network interfaces, their IP addresses and connection types. The /etc/network/interfaces file was edited to define static IP configurations. Some interfaces have static IP addresses, and some use DHCP to have dynamic addresses.

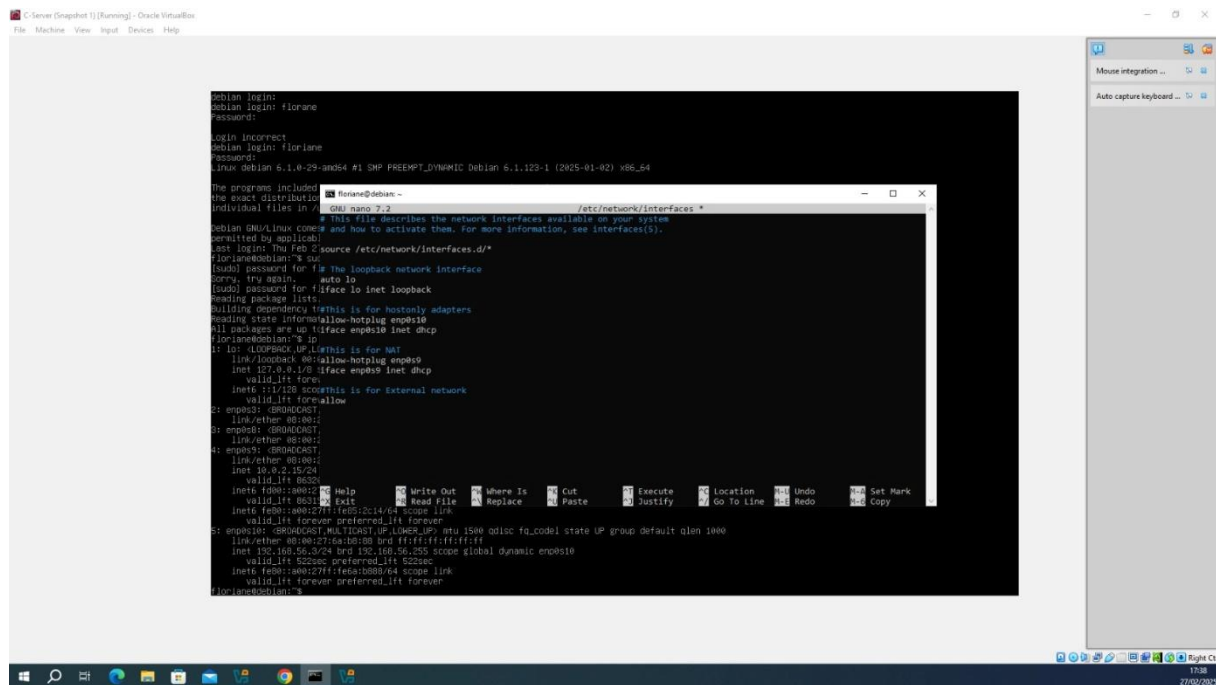


Figure 17 – File edited to define IP address

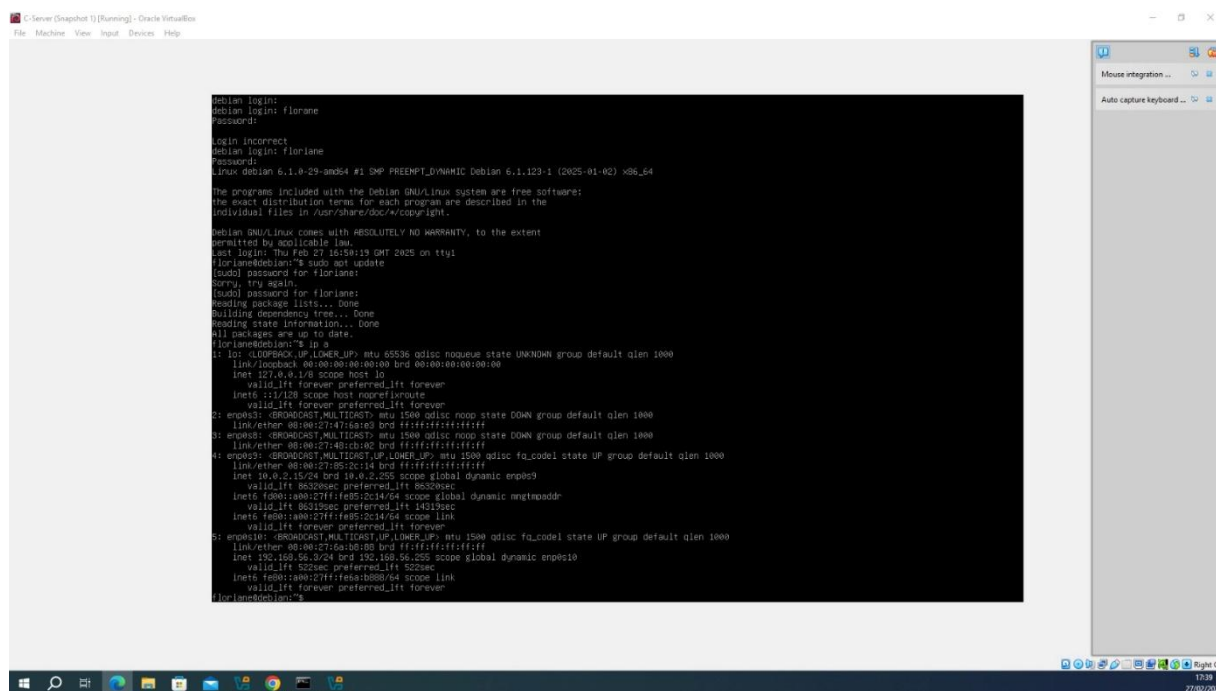


Figure 18 – IP addresses assigned

Debian does not have a way to copy and paste files directly. An SSH connection has been set up from the host machine to the Debian server. This allows files or commands to be transferred easily between the two systems.

Network deployment and troubleshooting (Debian Linux)

I set up the network connectivity between Windows server and Debian system. It pings successfully from S-DC to Debian.

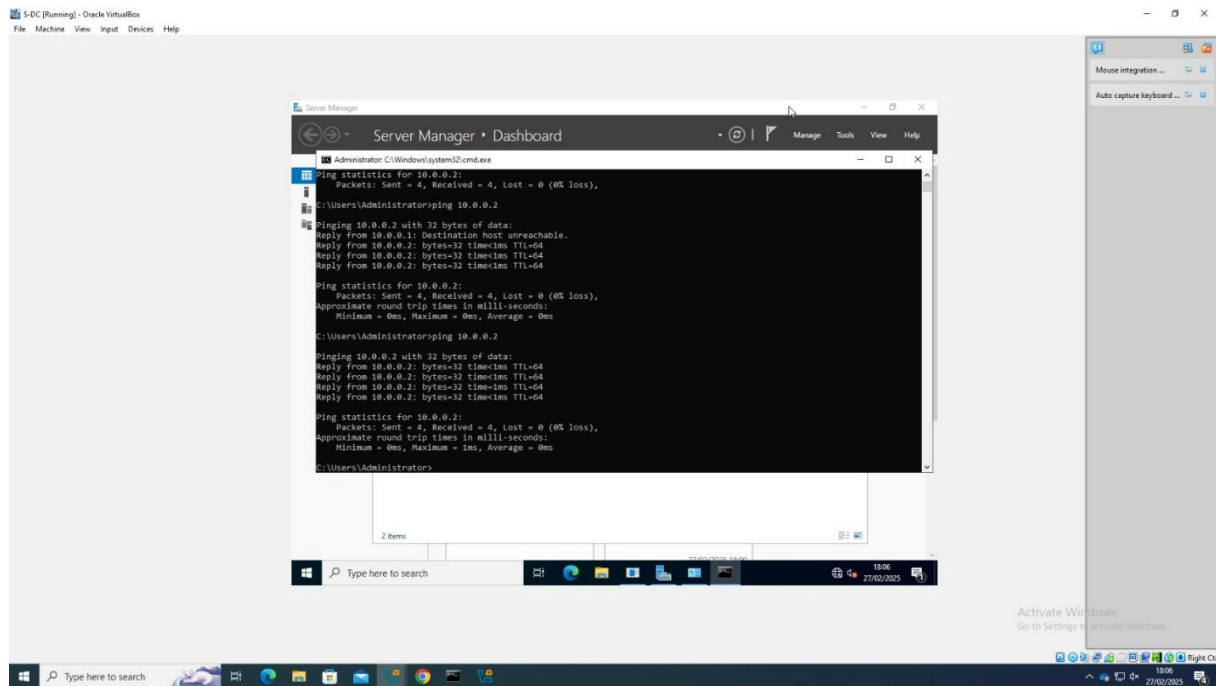


Figure 19 – Ping successful from Windows to Debian

Now from Debian, the terminal shows continuous ping responses from Debian to S-DC.

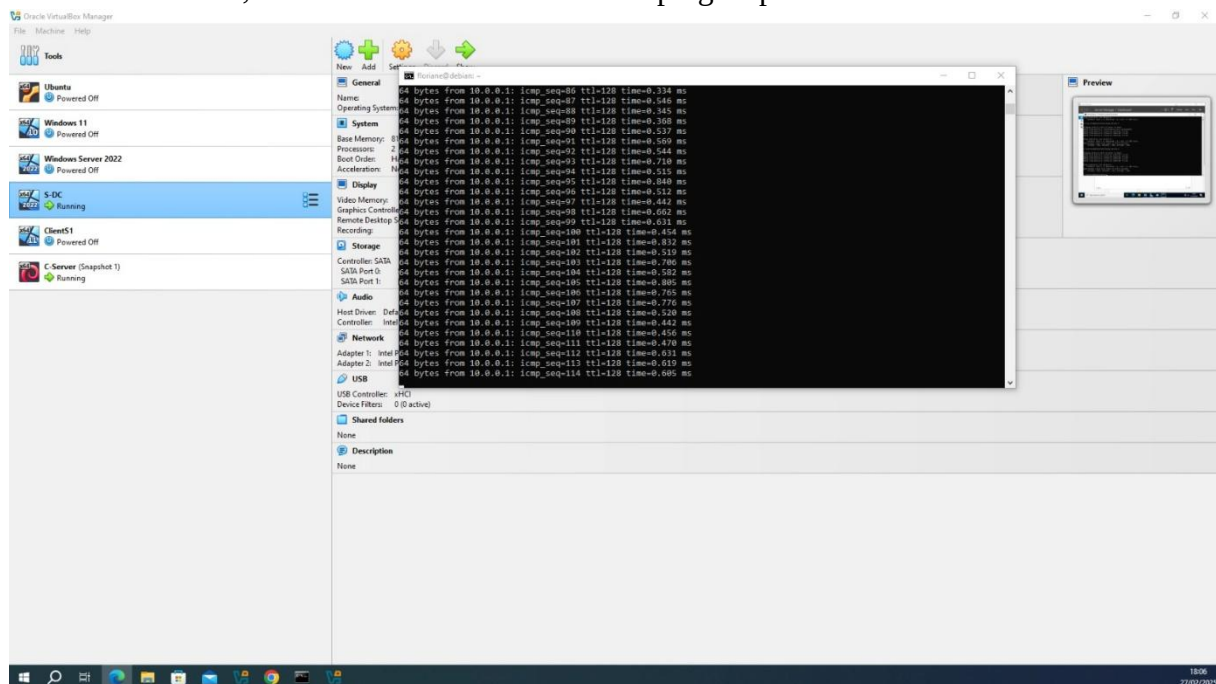


Figure 20 – Ping successful from Debian to Windows

The enp0s3 interface is active but does not have an IP address assigned so that it can communicate with the Debian server.

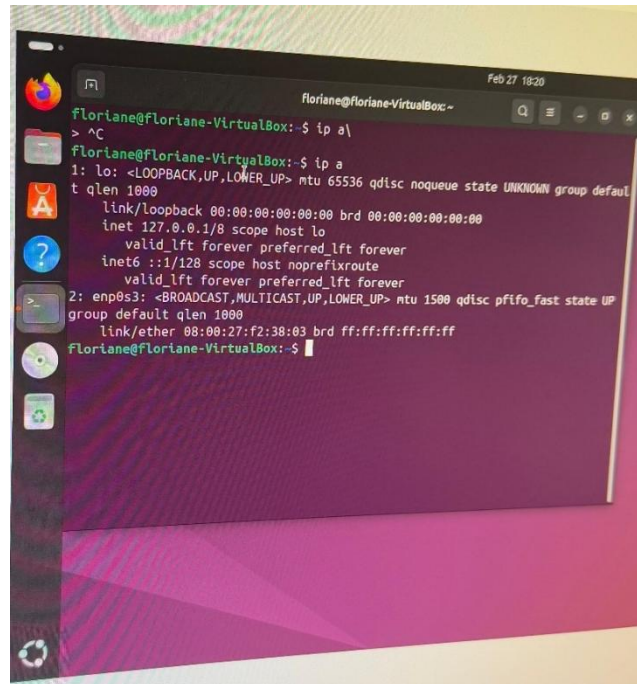


Figure 21 – IP address not assigned

I've changed the Netplan configuration (Ubuntu 18.04+). I typed this command in the terminal `sudo nano /etc/netplan/50-cloud-init.yaml` and added the static IP configuration. I used the `ip r` command to check the router's IP.

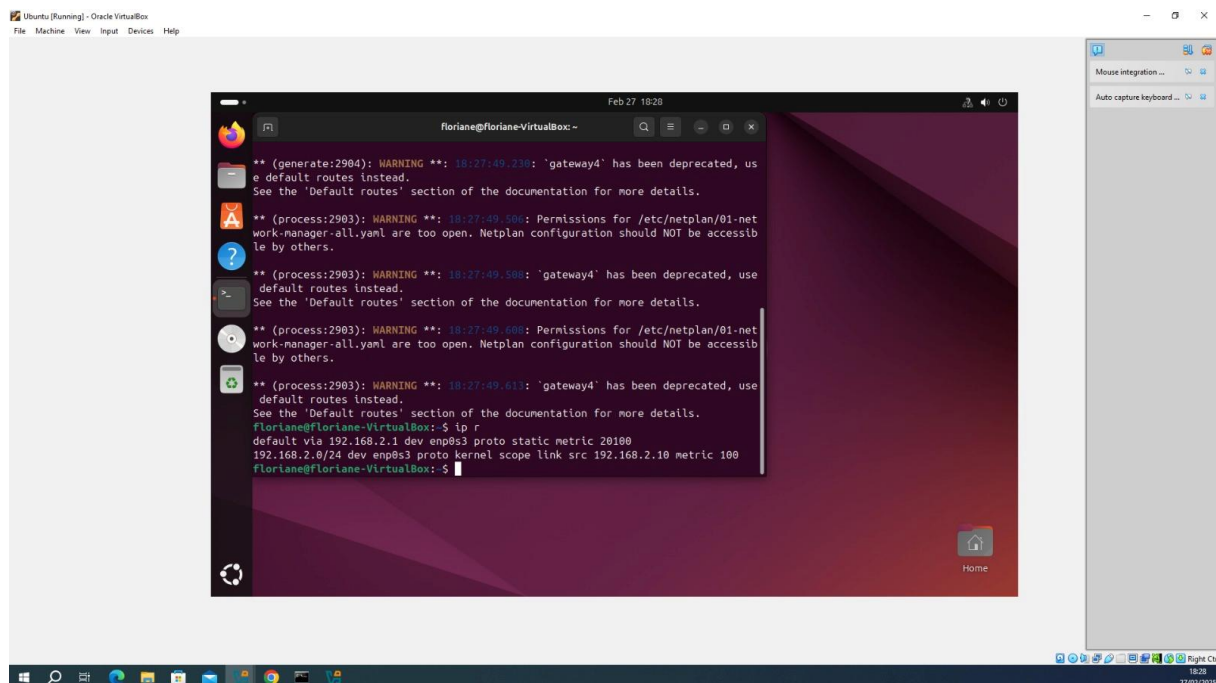


Figure 22 – IP address being configured

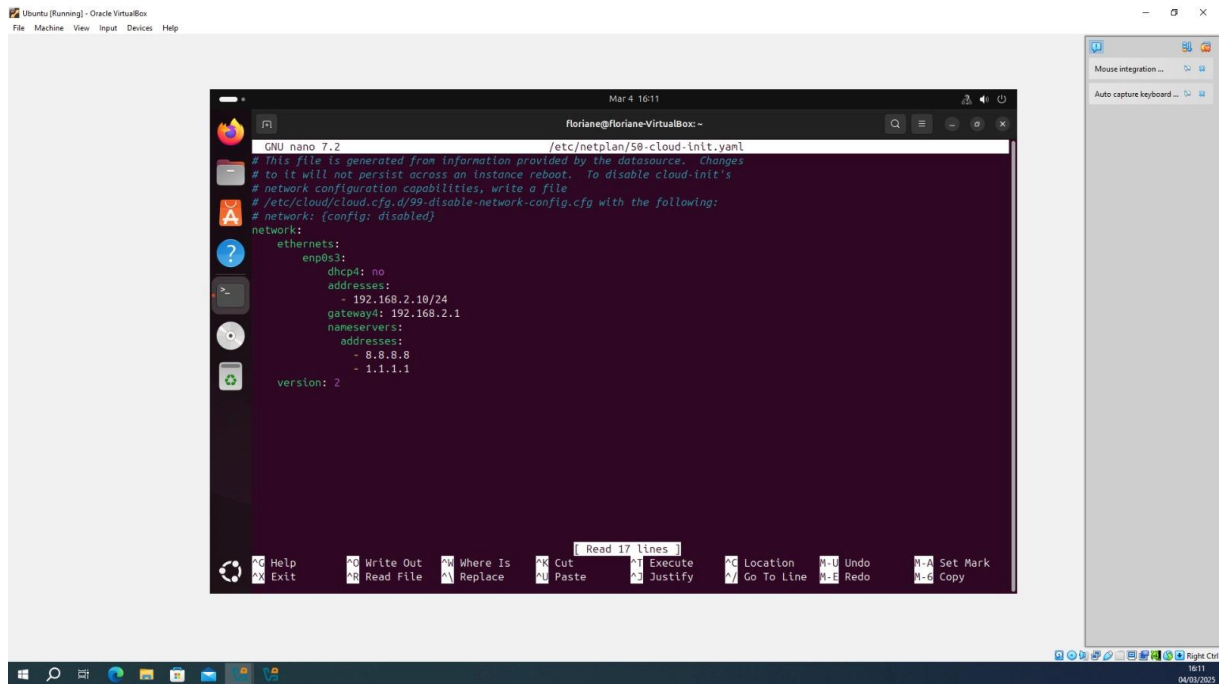


Figure 23 – File modified

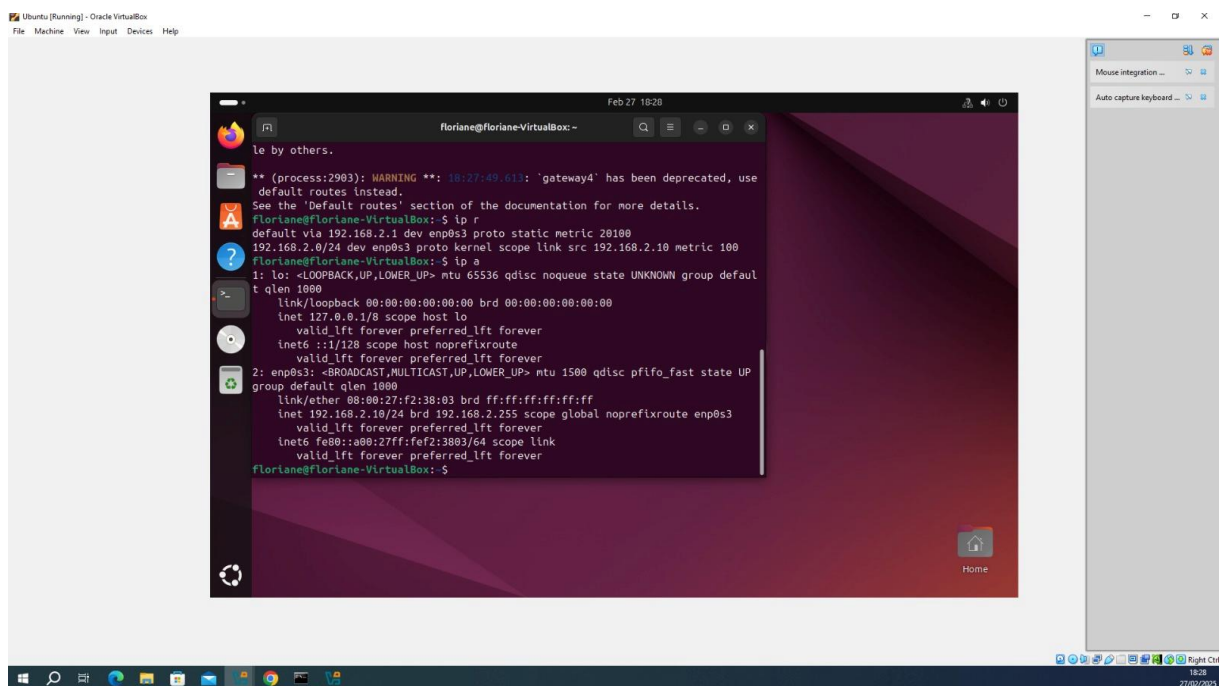


Figure 24 – IP address assigned

After configuring the IP address for ubuntu, I tried pinging from the Debian server and it works.

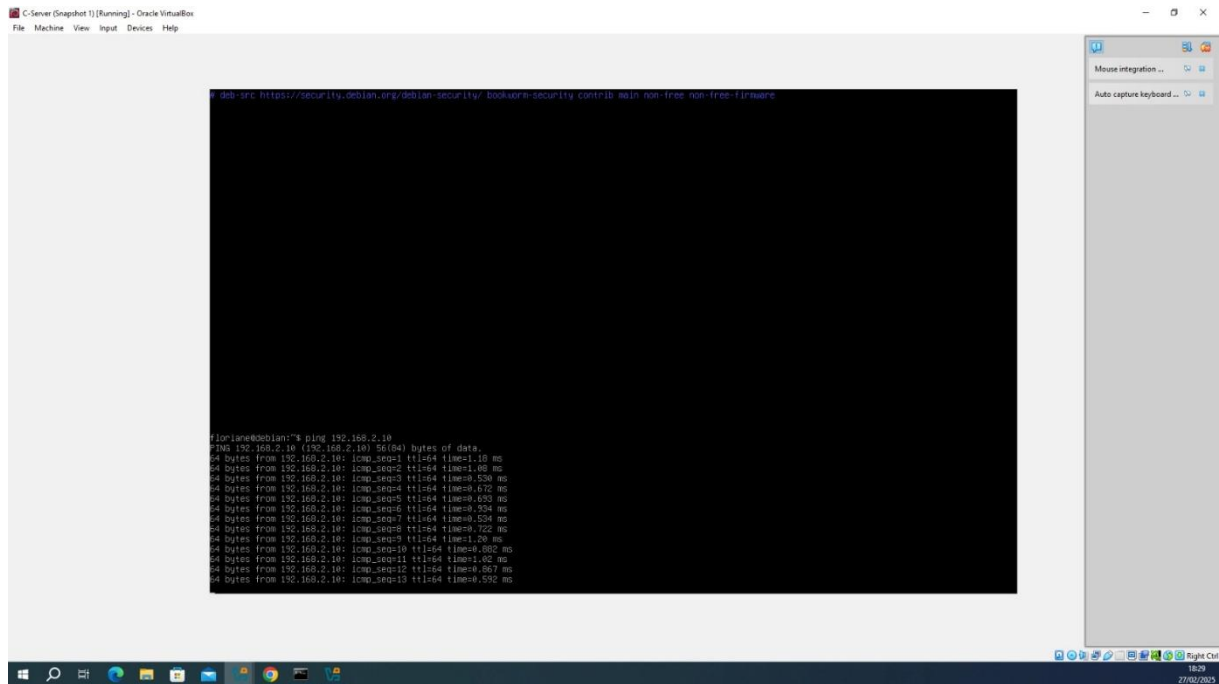


Figure 25 – Ping successful from Debian server to Ubuntu

After successful ping from Debian to the client, I ping from the Ubuntu client to the server. The ping is successful.

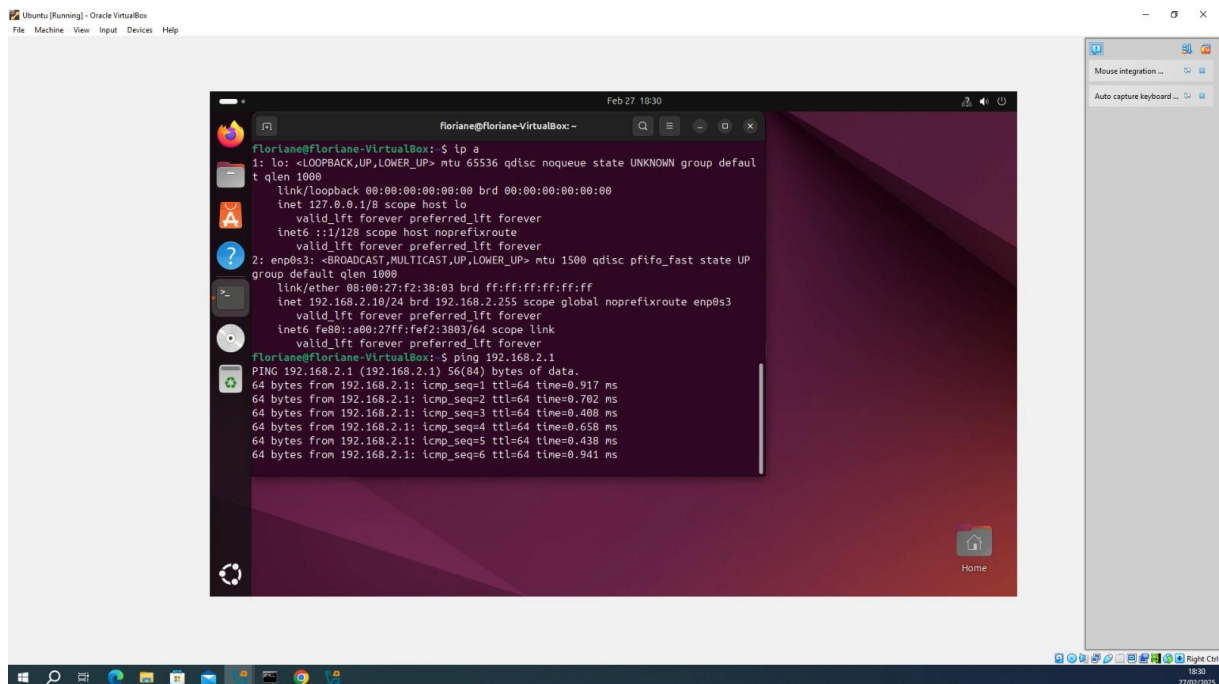


Figure 26 – Ping successful from Ubuntu to Debian server

III – Reflection

Technical reflection

During the network deployment and testing phase, several issues were encountered that made the process challenging. Some problems were easy to resolve, while others were more difficult and required several attempts to find a solution. One problem was that the client could not ping the Windows server, due to ICMP requests being blocked by default.

Personal reflection

This coursework was a good learning experience. Studying firewall configurations and IP addressing provided me with better understanding of network administration. It was interesting to learn about hypervisors and how to manage and use them. Troubleshooting network issues required patience and adaptability. Overall, this project helped me develop essential technical skills for future network implementations.

Conclusion

Working on this coursework was both a technical challenge and a great learning experience. The deployment of a virtual network using a type 2 hypervisor was successful despite some problems. The choice of this hypervisor was good, it offered flexibility, and it was easy to use. The implementation process involved setting up both Windows and Linux environments, configuring network parameters, and troubleshooting connectivity issues. Throughout the deployment, various challenges were encountered, like the firewall restrictions that blocked ICMP traffic and the misconfigured network adapters. These issues were resolved thanks to the modifications of the firewall rules and the adaptation of the IP configurations.

In the end, the virtualized network functioned, with proper communication between servers and client connections. More than just configuring machines, this project reinforced my problem-solving and adaptability skills.

Appendix A: The OSI Model and network layers

OSI (Open Systems Interconnection) is a model that normalises the communication in a network by dividing it into seven layers. Each layer has a function to perform and communicates with the layer above and below it. [21]

The layer 1 is the physical layer. It is the first layer, the most closely associated with physical connections between devices. It includes hardware components such as network cables and switches.

The second one is the Data Link Layer, it ensures reliable data transfer. It uses MAC addresses to identify devices on a network.

Network Layer, layer 3, does addressing, routing, and packet forwarding between different networks. It assigns IP addresses – IPv4 and IPv6 – to devices and determines the best path for data to follow. It has key routing protocols such as OSPF, BGP, and RIP. They route data between networks and manage traffic congestion.

The layer 4 is the Transport Layer. It is responsible for end-to-end communication with the assurance of reliability and flow control. It divides data into segments and retransmits lost packets. There are some protocols such as TCP that offer guaranteed delivery.

Layer 5, the Session Layer, manages sessions between applications, making devices continuously communicate with each other. It establishes and terminates connections, making data exchanges flow smoothly without breaks.

The Presentation Layer, the layer 6, translates data formats between applications and networks, so that data sent from one system can be read by another.

The last one is the Application Layer, it provides end-users with direct network services. It includes used protocols like HTTP for browsing the web and DNS for converting domain names into IP addresses.

Appendix B: Network classification and addressing

IP address classes

IP addresses are categorized into different classes based on their network and host allocation.

CLASS	NETWORK BITS	HOST BITS	ADDRESS RANGE	EXAMPLE
A	8 bits	24 bits	1.0.0.0 - 126.255.255.255	10.0.0.1 → Network: 10, Host: 0.0.1
B	16 bits	16 bits	128.0.0.0 - 191.255.255.255	172.16.5.1 → Network: 172.16, Host: 5.1
C	24 bits	8 bits	192.0.0.0 - 223.255.255.255	192.168.0.1 → Network: 192.168.0, Host: 1

Figure 27 – Table of IP address class

Each IP address class is designed for specific network sizes and purposes. For example, class A uses 8 bits for the network portion and 24 bits for the host portion. It supports large-scale networks. Class D and E are special classes. Class D is reserved for multicast and class E is reserved for experimental purposes. They cannot be used for general networking.

Class A networks support millions of devices, they are ideal for large-scale enterprises, while Class C networks are used for small networks.

Private and public IP addresses

Public IP addresses are unique and assigned by the Internet Assigned Numbers Authority (IANA). They are used to reference devices on the internet and are required for web servers, mail servers, and other public systems.

Private IP addresses are used within local networks (LANs) and are not routable over the internet. They are assigned to devices within organizations to facilitate internal communication.

References

- [1] “Type 1 vs Type 2 Hypervisor”, PureStorage
<https://blog.purestorage.com/purely-informational/type-1-vs-type-2-hypervisor/?print=pdf>
- [2] “What is virtualization”, AWS
https://aws.amazon.com/what-is/virtualization/?nc1=h_ls
- [3] “What is server virtualization? The ultimate guide – *What’s the difference between Type 1 vs Type 2 hypervisor?*”, Stephen J. Bigelow
<https://www.techtarget.com/searchitoperations/tip/Whats-the-difference-between-Type-1-vs-Type-2-hypervisor>
- [4] <https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-overview?pivots=windows>
- [5] <https://www.virtualbox.org/>
- [6] <https://blog.devgenius.io/ever-wonder-how-data-is-transferred-over-the-internet-ec4eb3d6d66f>
- [7] <https://www.solarwinds.com/resources/it-glossary/subnetting>
- [8] <https://www.fortinet.com/resources/cyberglossary/what-is-ip-address>
- [9] “IPv6 Fundamentals – *A straightforward Approach to understanding IPv6*”, Cisco, Rick Graziani
- [10] “Scaling Networks – *Companion Guide*” Cisco Networking Academy
- [11] “Routing and Switching Essentials – *Companion Guide*” Cisco Networking Academy
- [12] <https://www.networkcomputing.com/network-management/comparing-dynamic-routing-protocols>
- [13] <http://units.folder101.com/cisco/sem2/Notes/ch6-routing/routing.htm>
- [14] <https://www.catchpoint.com/dynamic-routing-protocols>
- [15] “Computer and communication network”, Second Edition, Nader F. Mir
- [16] <https://www.kaspersky.com/resource-center/definitions/what-is-a-vpn>
- [17] <https://www.spiceworks.com/tech/networking/articles/bgp-vs-ospf-vs-eigrp/>
- [18] <https://www.varonis.com/blog/active-directory-domain-services>
- [19] <https://www.techtarget.com/searchnetworking/definition/DHCP>
- [20] <https://www.digicert.com/blog/what-is-a-certificate-authority>
- [21] https://en.wikipedia.org/wiki/OSI_model

Using of ChatGPT to better understand some concepts