

Network Operating Systems and Services (ACCA5023)

LECTURER: MD. SADEQUE IMAM SHAIKH

STUDENT: ANDREI-STEFANITA STOICA (2413927)

DATE: 4TH OF DECEMBER, 2025 | University of Wales Trinity Saint
David (London Campus)

Contents

1. Introduction	2
2. Research Report.....	2
2.1 Hypervisor	2
2.2 IP Addressing	3
2.3 Research on Routing	5
2.4 Research on Network Services	7
3. Technical Documentation	7
4. Reflection	27
5. References	27

1. Introduction

The aim of this Assessment is to research and build/deploy a Virtualized network. The first task at hand will be to research the different types of hypervisors available as well as do brief analysis on IP Addressing and Routing. The second task will consist of choosing the hypervisors most suitable for this assignment considering available resources. On top of that, it is needed to download and set up the chosen hypervisor, configure IP addressing and finally do tests and troubleshooting. The third and final task will be comprised of a short reflection regarding the assessment e.g.: whether there were any problems whilst troubleshooting, downloading or setting up the hypervisor. The scenario given for this assignment is to assume I have been hired as an IT Tech by a Multinational Corporation called NOSS2026 which operates at two locations: Swansea and London.

2. Research Report

2.1 Hypervisor

A hypervisor is a software which enables various virtual machines (VMs) to run on one physical server. Each virtual machine has its own operating system (OS) and the main hypervisor's task is to allocate physical computing resources as instructed and/or needed. Essentially, what this piece of software does is enable its users to run multiple operating systems at once sharing the same physical computer resources. In terms of variety, there are Type 1 and Type 2 hypervisors. Starting with Type 1, commonly referred to as metal hypervisor or native hypervisor, its main purpose is to take place of the host operating system and interact directly with the central processing unit

(CPU), memory and physical storage. Due to its nature, it is highly efficient and offers an extra layer of security because it is directly connected to the CPU which makes it harder for an attacker to compromise the system. A notable drawback, however, is that these types of hypervisors often require a separate management machine to control all the different VMs and the host hardware. In comparison with a type 1 hypervisor, a type 2 hypervisor (also referred to as hosted hypervisor) does not run directly on the underlying hardware. It runs as an application in an OS. These hypervisors are often used by individual PC users such as engineers, security experts testing malware and business users in need of access to an application only available on other operating systems. Hosted hypervisors offer a quick and intuitive installation of an alternative OS perfect for enthusiasts and experts alike.

2.2 IP Addressing

Every device which is part of a network must be addressable, this means that it can be reached by referring to it under a preset of different addresses. In other words, IP addresses already exist in a network and must be given to each device or location either automatically or manually. IP addresses help devices communicate with one another.

Currently, we have two revisions of the IP protocol which are used worldwide on systems: IPv4 and IPv6. IPv4 is a 32-bit address split into four octets [2].

Example: 192.168.1.10

Binary representation: 192 = 11000000 168 = 10101000 1 = 00000001 10 = 00001010

Originally, IPv4 used address classes:

IPv4 Address Classes and Ranges

A	Class A	0.0.0.0 – 127.255.255.255
B	Class B	128.0.0.0 – 191.255.255.255
C	Class C	192.0.0.0 – 223.255.255.255
D	Class D	224.0.0.0 – 239.255.255.255
E	Class E	240.0.0.0 – 255.255.255.255

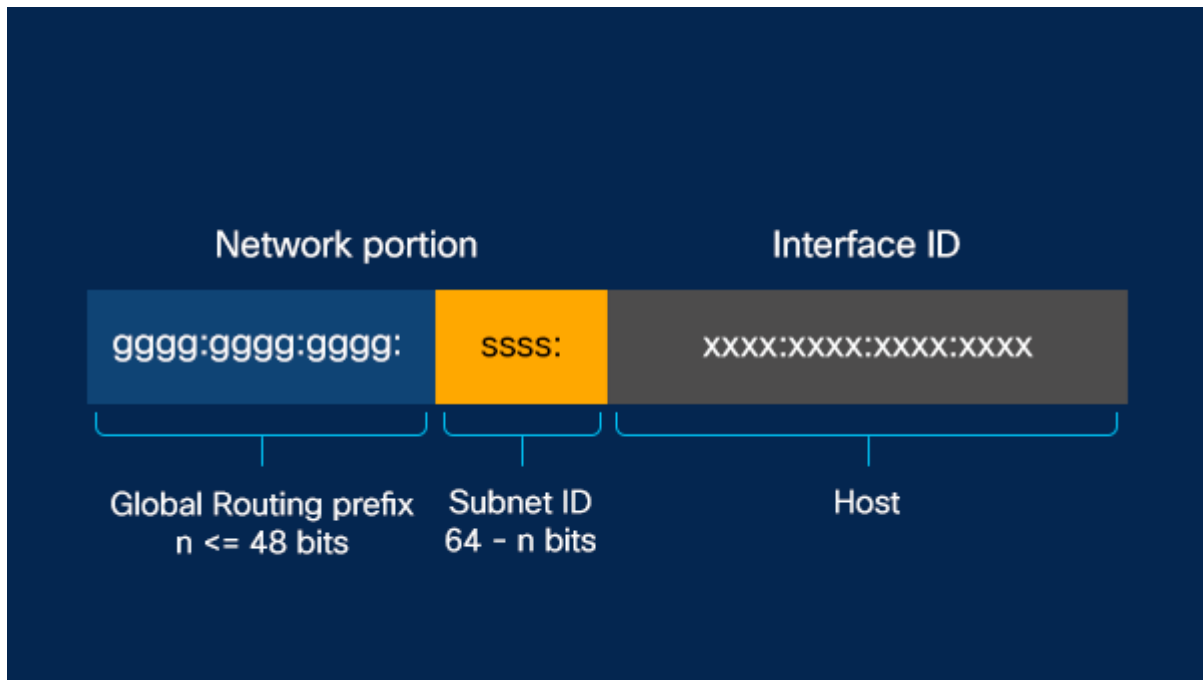
Class A is reserved for very large networks, Class B for medium networks, Class C for small networks, Class D for multicast and lastly Class E for research purposes [3].

Nowadays, modern networks use CIDR, also known as Classless Inter-Domain Routing, instead of classes.

The structure of IPv6 is in the form of a 128-bit address shown in hexadecimal.

Example: **2001:0db8:85a3:0000:0000:8a2e:0370:7334**

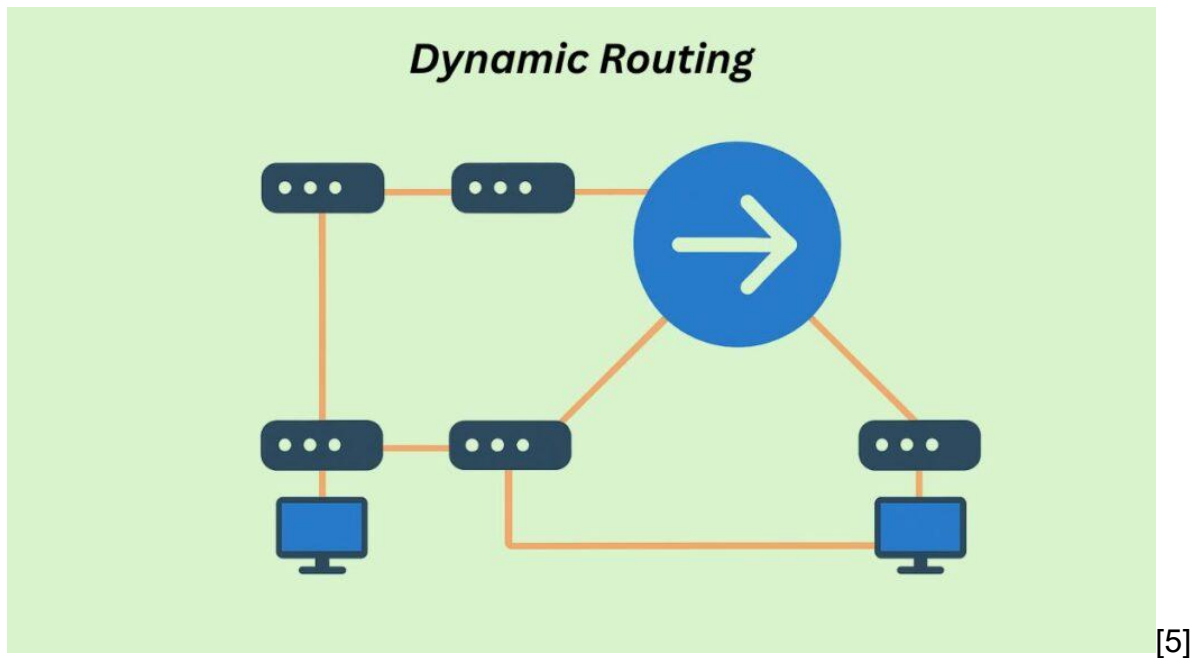
This address can also be simplified by removing leading zeros and consecutive groups of zeros. Therefore, the example address I used previously would end up looking like this: **2001:db8:85a3::8a2e:370:7334**



The IPv6 protocol offers different types of advantages including virtually unlimited addresses, optional NAT (Network Address Translation), easier administration with DHCPv6, built-in security (IPsec support) ... among others which make IPv6 superior to the outdated IPv4 protocol [4].

2.3 Research on Routing

Routing protocols determine how data travels between networks. Their main purposes are to find the best path for data packets across networks, to exchange routing information between routers, to maintain updated routing tables and to adapt to network changes thus ensuring reliable connectivity. To put it simply, routing protocols make networks more efficient, resilient and scalable.



The way routing protocols operate is firstly, routers identify directly connected networks and communicate them to other routers, then the routers exchange full routing tables, and they share the state of their links. The routing tables are built and maintained dynamically, based on metrics such as hop count, bandwidth, delay... In the event of a link failing, routers detect the failure, recalculate routes and converge on the best available paths. All of this happens automatically, which is a major advantage of dynamic routing protocols.

Static routing means routes are managed by an administrator, they are simple to set up for small networks, they are predictable and secure and use low amounts of CPU and bandwidth. On the other hand, static routing is not scalable, and it requires manual updates for any network change.

In dynamic routing, routes are learned automatically using routing protocols like OSPF (Open Shortest Path First), EIGRP (Enhanced Interior Gateway Routing Protocol) ... Dynamic routing offers automatic adaptability to network changes, scalability and reduced administrative engagement. However, it uses more CPU, memory and bandwidth, it is more complex to configure and there is a risk of routing loops if it is not properly configured [6] [7].

Considering I have a Debian Server acting as a DHCP Relay, Static routing would be the most suitable because I would like to easily document my findings and I also do not plan on benefiting from the scalability Dynamic routing offers.

2.4 Research on Network Services

Virtual Private Networks (VPNs), Dynamic Host Configuration Protocol (DHCP), Active Directory Domain Services (AD DS), and Certificate Authorities (CAs) are essential technologies for secure and efficient enterprise network operations.

A VPN (Virtual Private Network) creates an encrypted tunnel which protects data travelling between remote users or sites and the internal network. VPNs commonly use TLS or IPsec for authentication and encryption, with certificate authentication considered best practice for strong identity assurance [8].

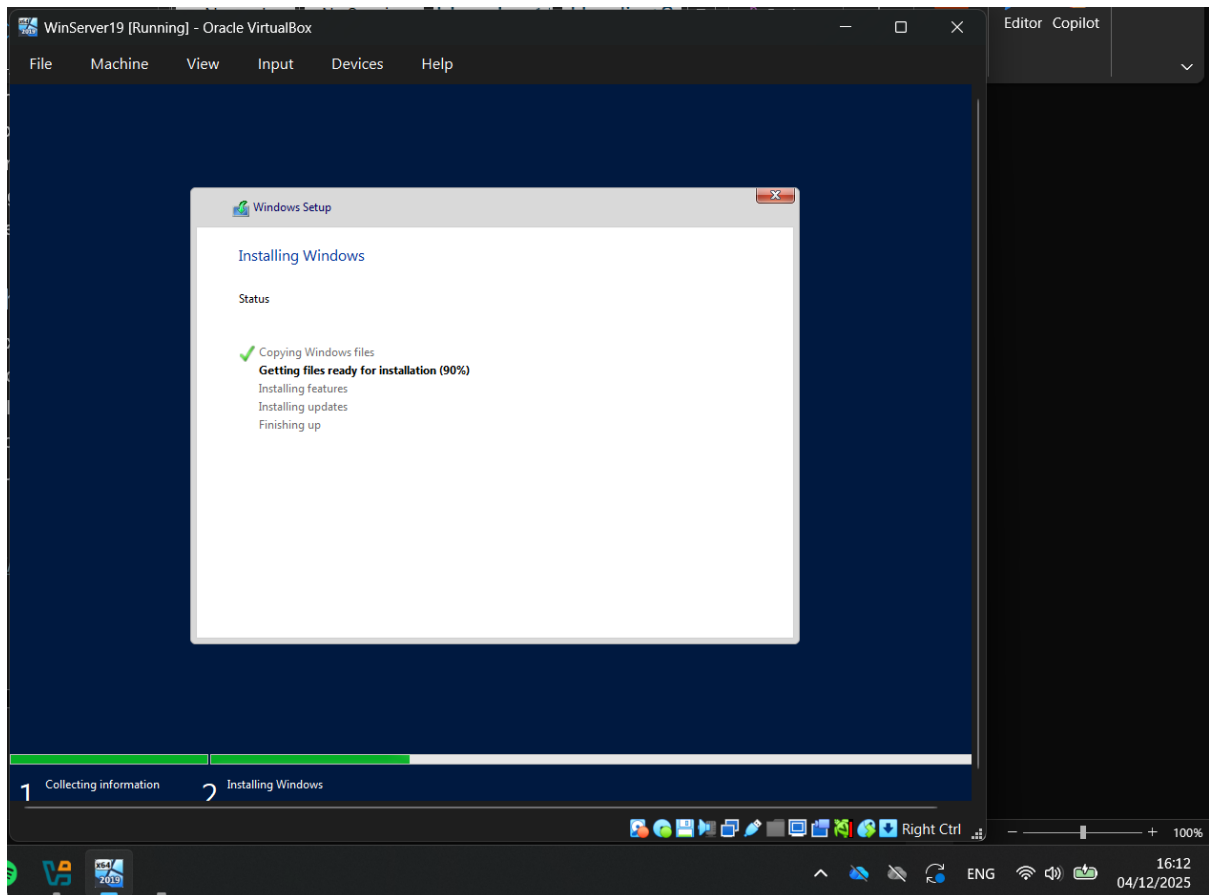
DHCP (Dynamic Host Configuration Protocol) automatically assigns IP configuration such as IP address, subnet mask, default gateway and DNS to client devices. This reduces manual errors and simplifies administration, especially when paired with a DHCP Relay [9] [10].

Active Directory Domain Services (AD DS) provides centralized authentication, authorization and policy management within a Windows domain. AD DS stores information about users, computers and resources. Best practices include enforcing strict passwords and access controls, auditing privileged accounts among others [11].

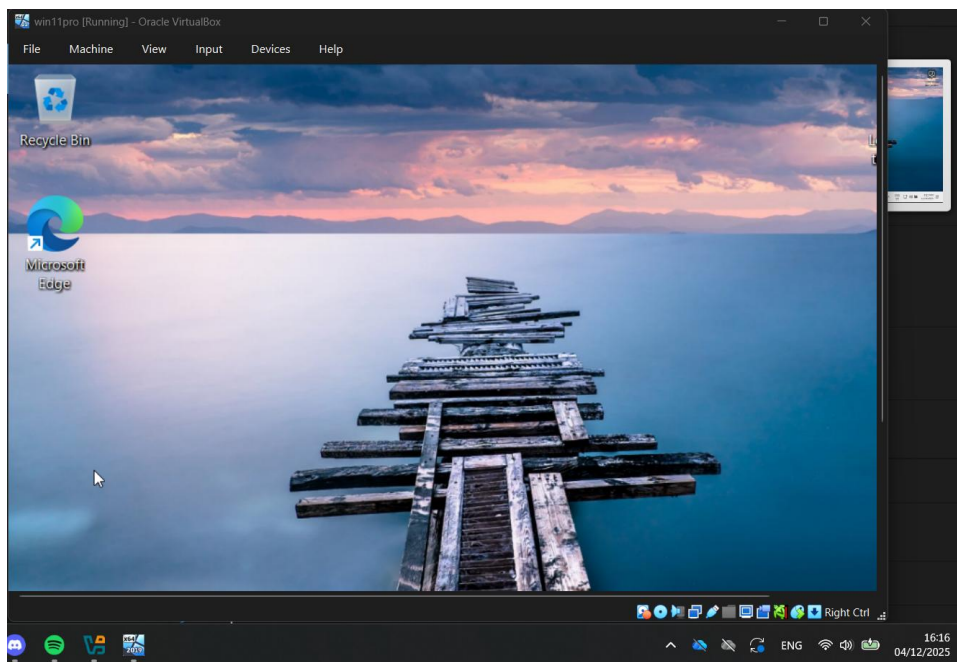
Certificate Authorities support Public Key Infrastructure (PKI) by issuing and managing digital certificates used for encryption, authentication, and data integrity. Their primary responsibilities include certificate lifecycle management, revocation handling, and maintaining trust chains. Best practices emphasize protecting CA private keys, monitoring the PKI environment, developing a robust certificate registration procedure... [12] [13]

3. Technical Documentation

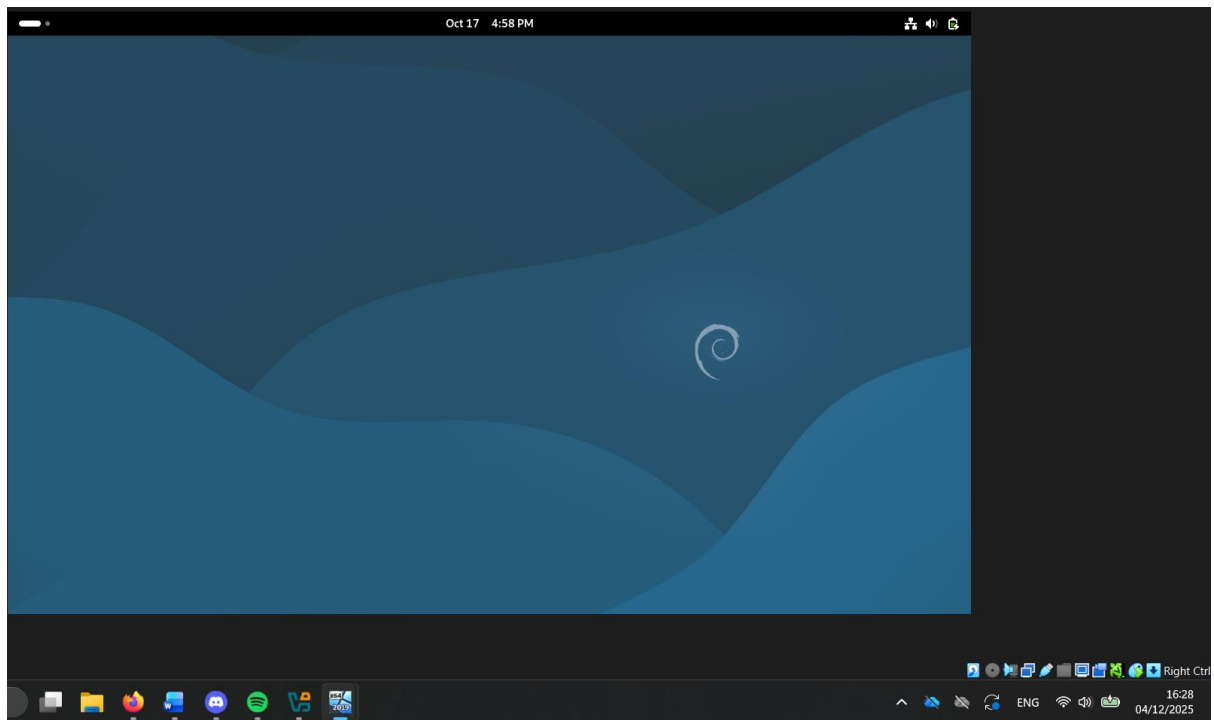
In the next part of the assignment, I will be creating 4 virtual machines: 1 Windows Client and 1 Windows Server, and 1 Debian client along with 1 Debian Server using Oracle Virtual Machine. The main goal will be to deploy a working network, configure the IP correctly and do testing and troubleshooting making sure the connectivity between clients and servers is successful.



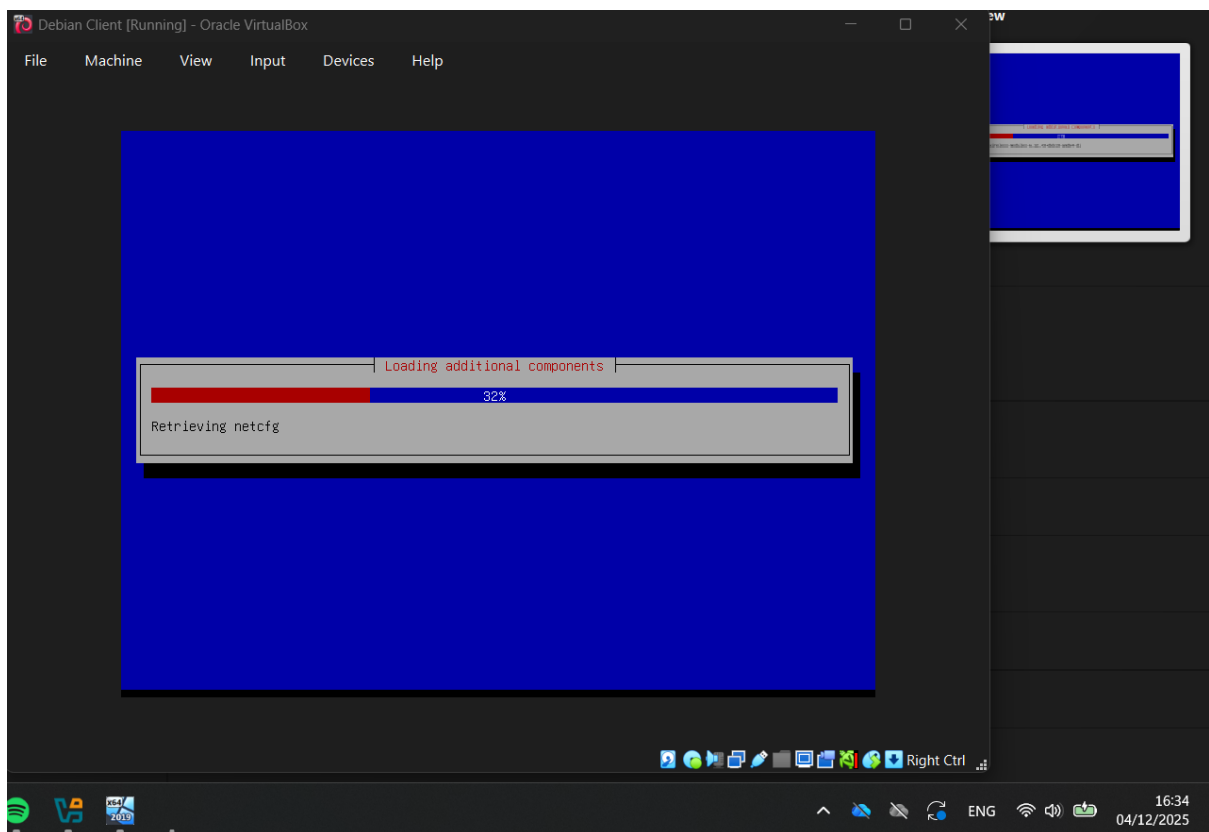
This is the installation process of the Windows Server, it was advised we install the 2019 version due to it being the most stable.



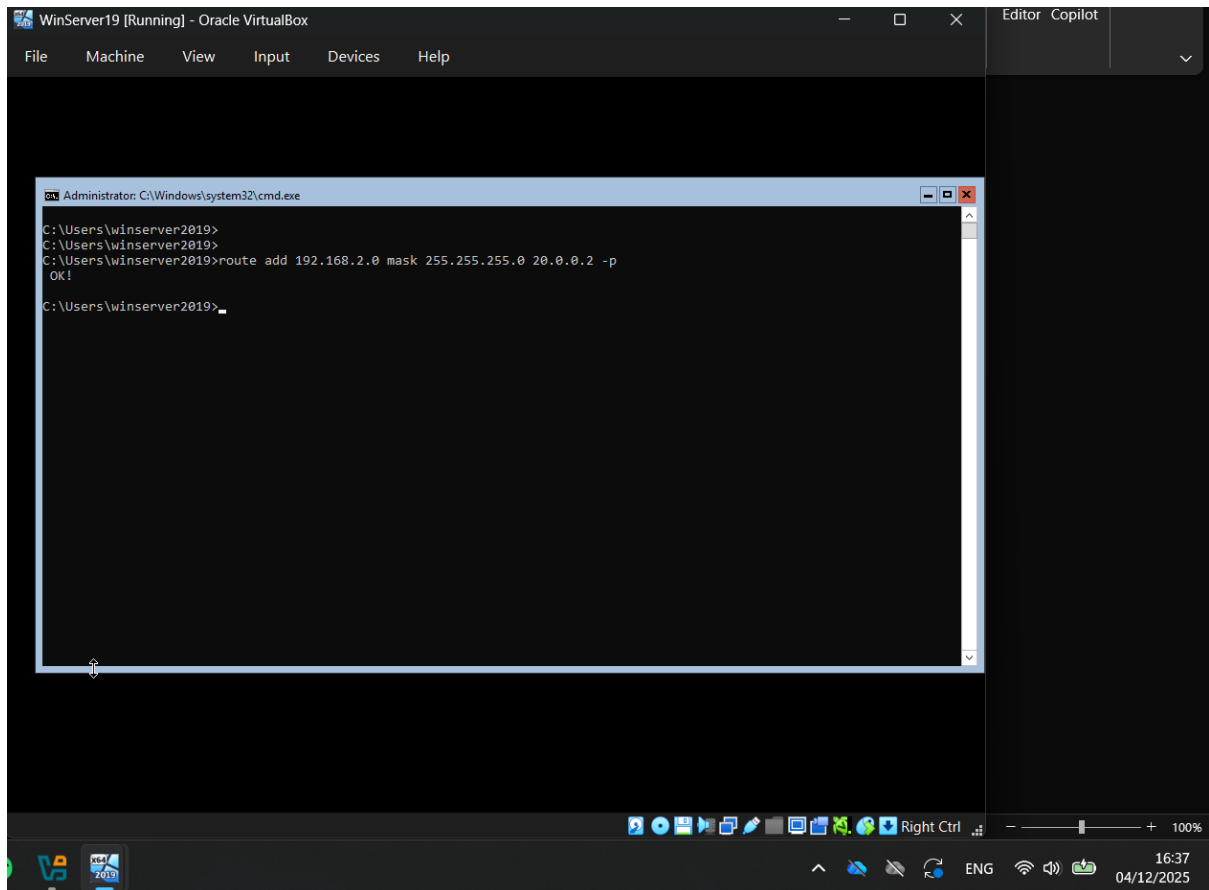
The Windows Client was already installed. In this case I chose Windows 11 Pro, the choice of Windows Client was not as relevant in comparison to the Windows Server.



This Virtual Machine will act as the Debian Server.

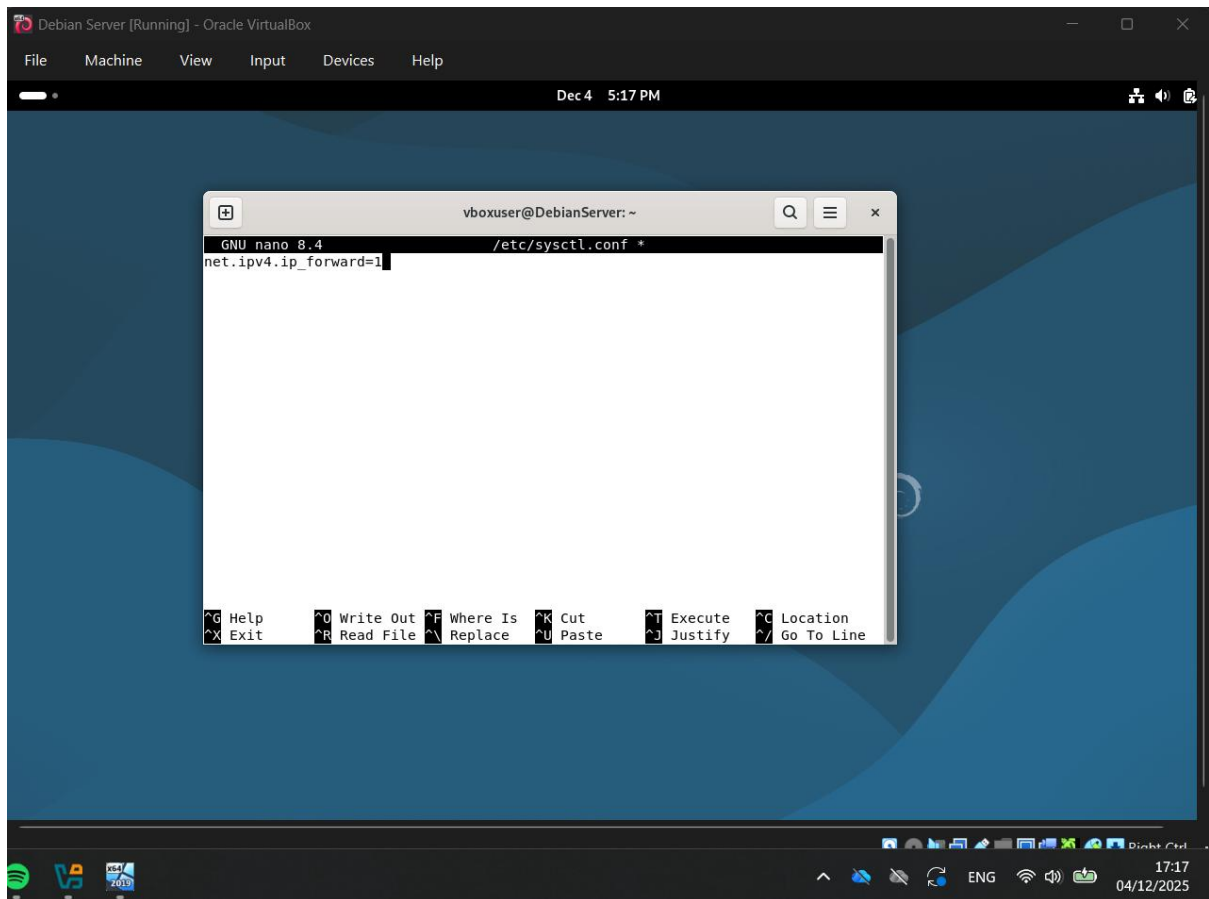


This is the installation process of the Debian Client.

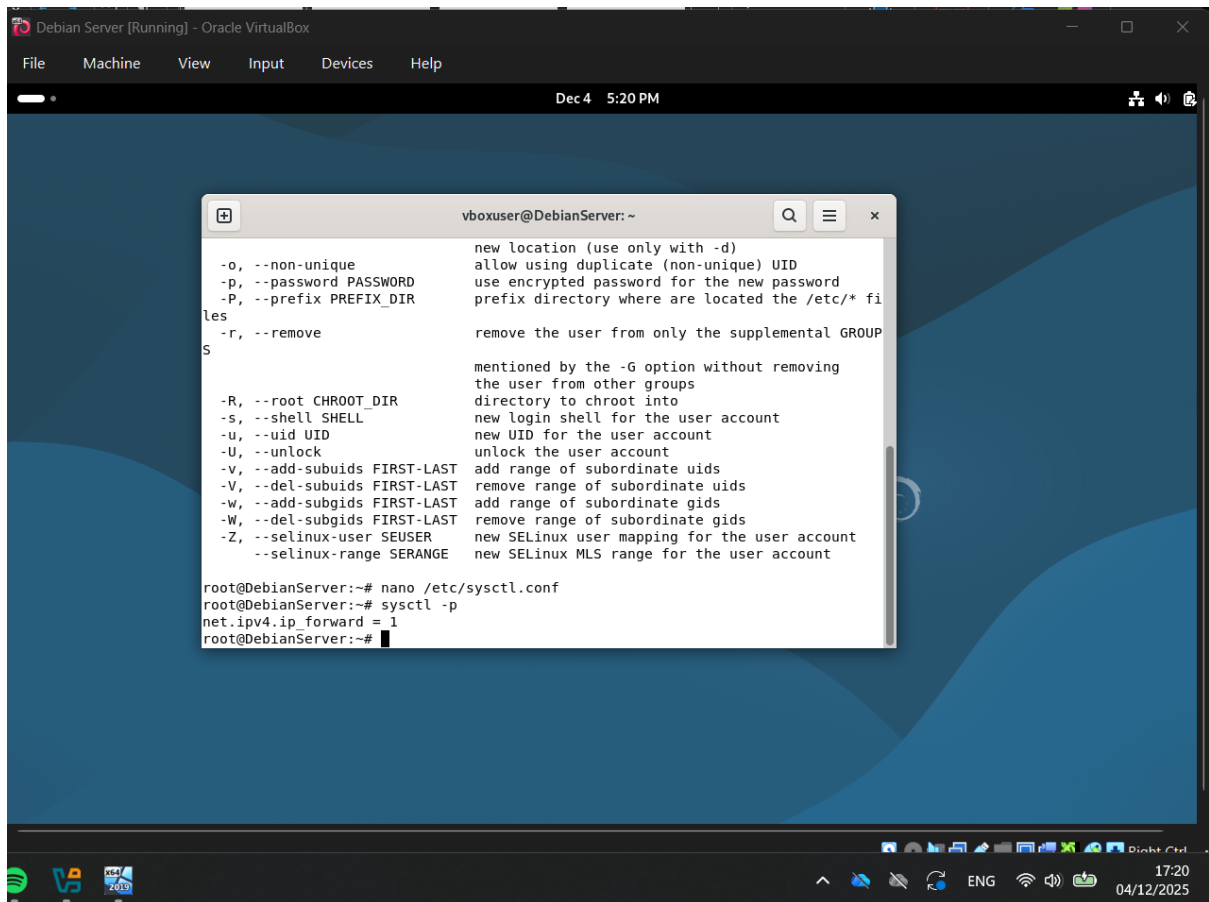


Command used: route add 192.168.2.0 mask 255.255.255.0 20.0.0.2 -p

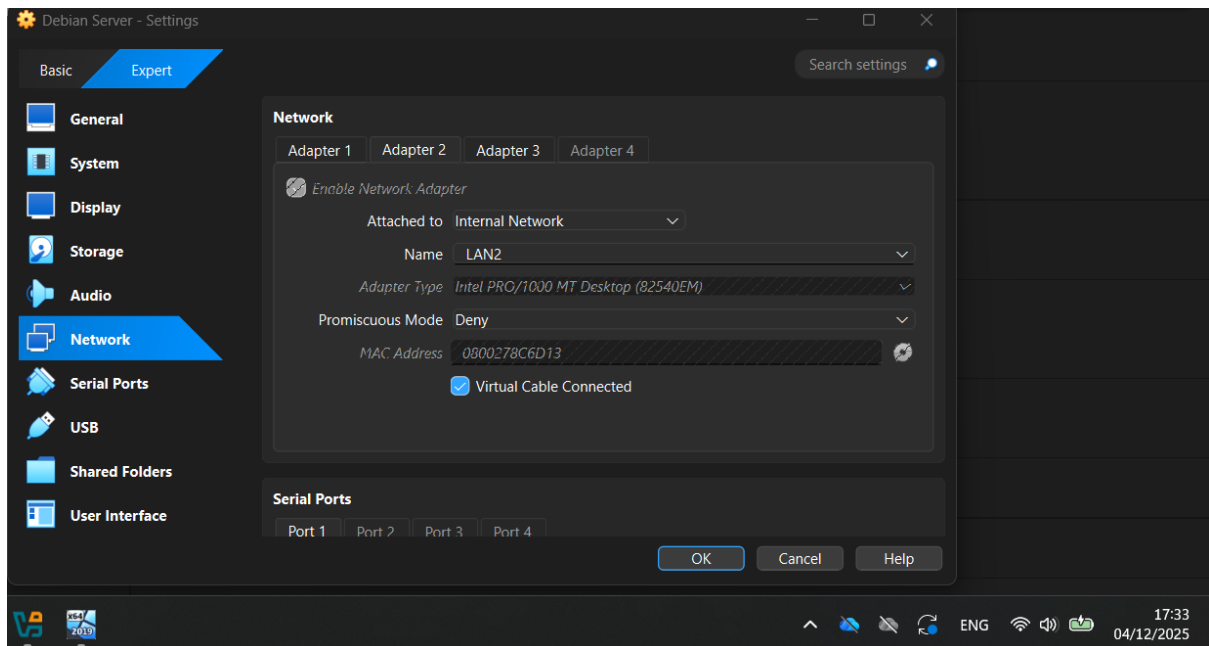
The purpose of this command is to tell the Windows Server to send any Debian LAN traffic to the Debian Server.



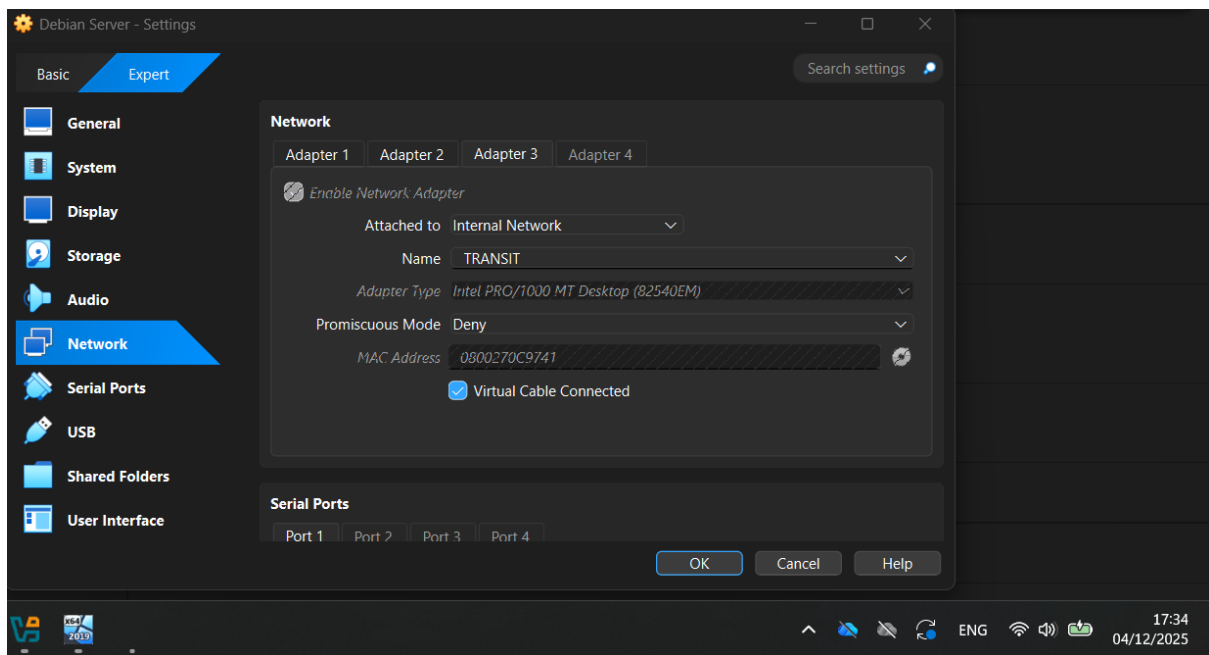
Next, I move to the Debian Server and I enable IP forwarding making use of the command `nano /etc/sysctl.conf`. Inside, I type the command `net.ipv4.ip_forward=1` and I proceed to save and exit.



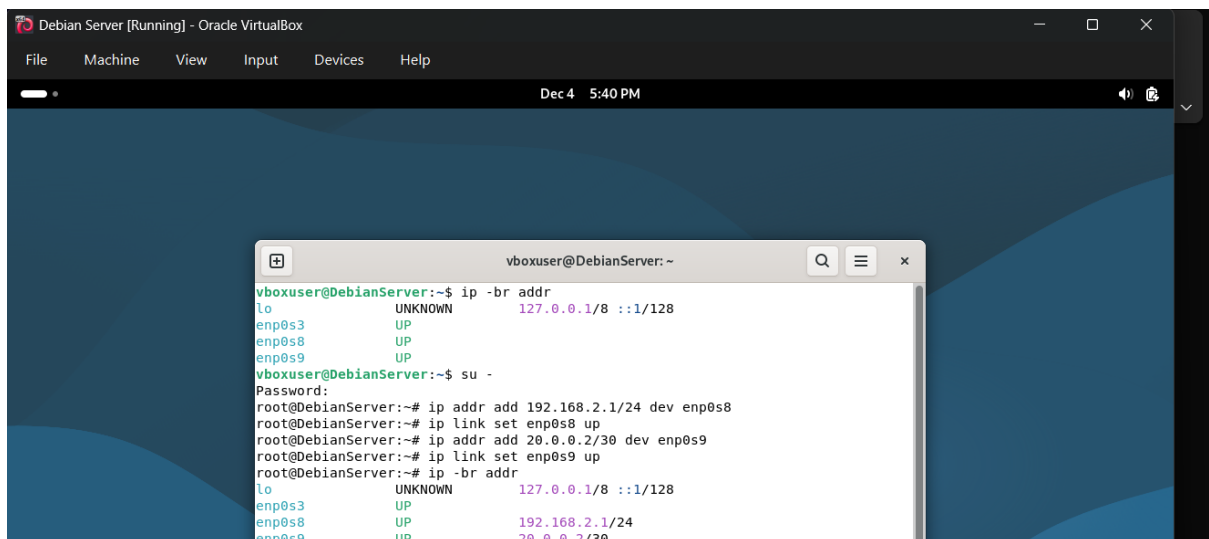
Moving forward, I make sure to verify it is enabled using the command `sysctl -p`. It seems to be working correctly.



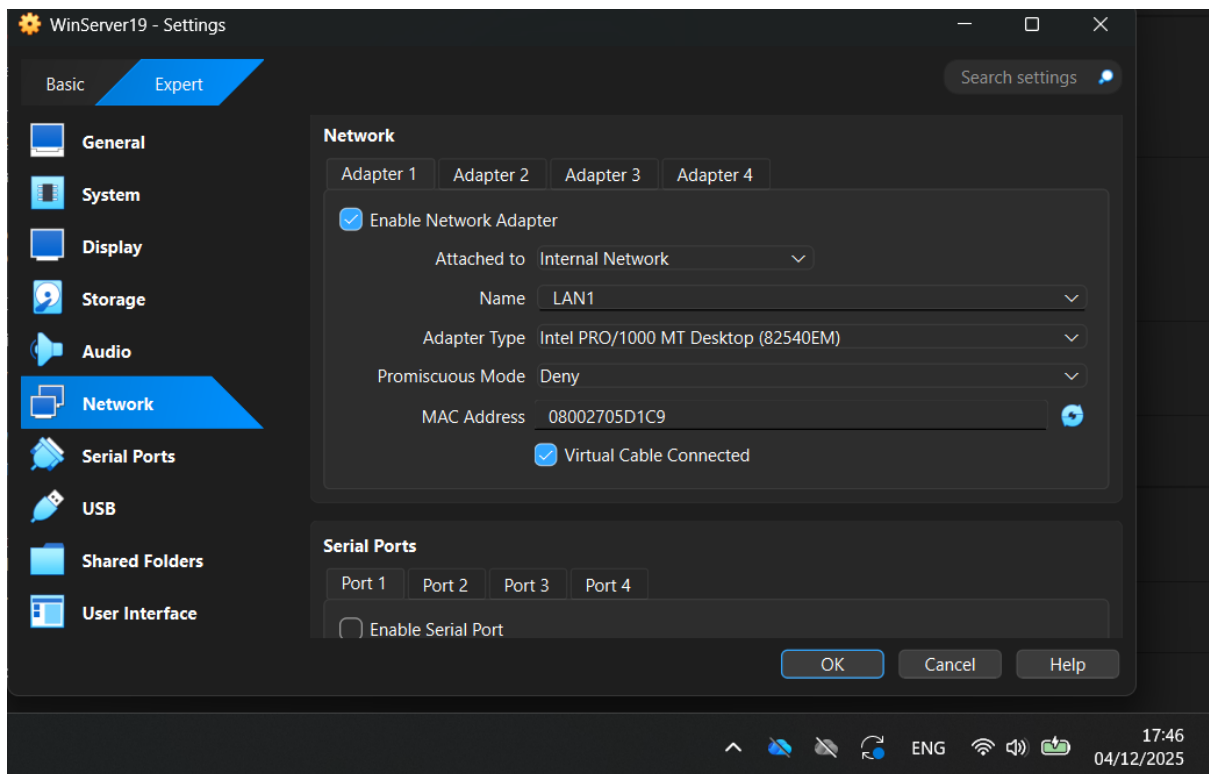
Before moving forward, I need to make sure to add 2 more network adapters in Virtualbox, Adapter 2 and 3.



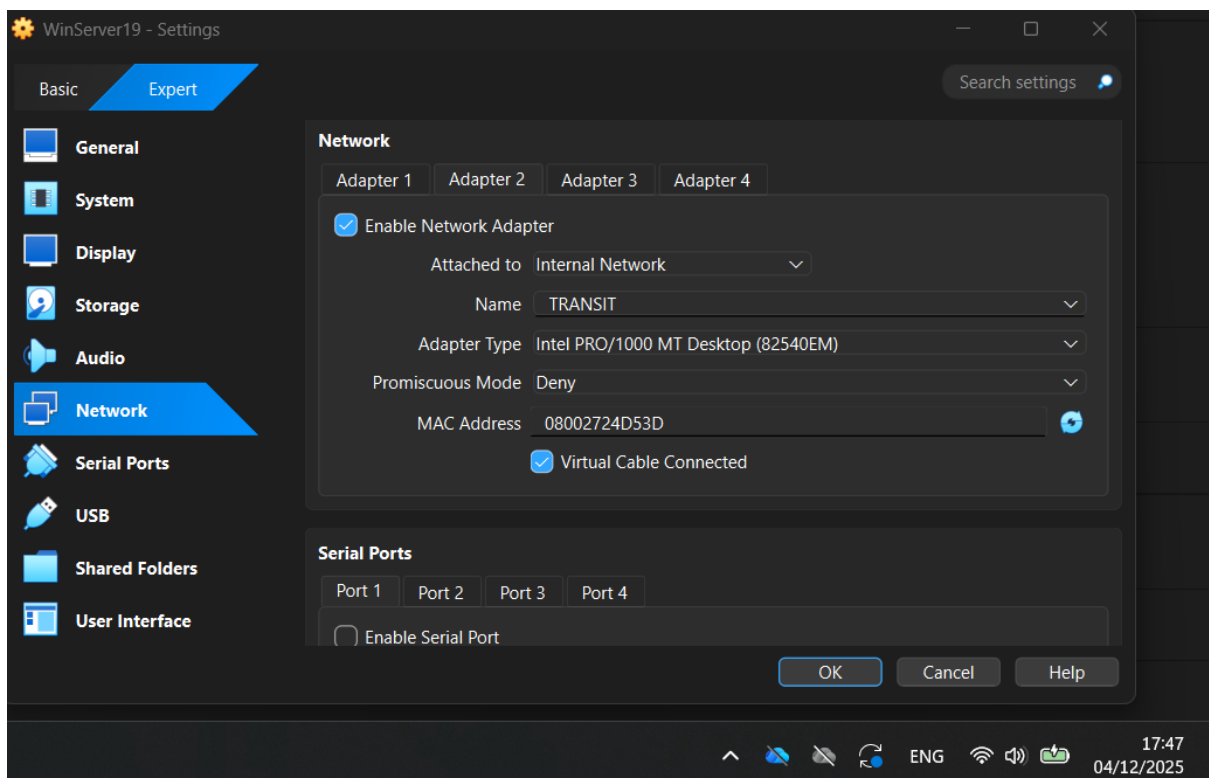
This is the third adapter which will act as the Transit.



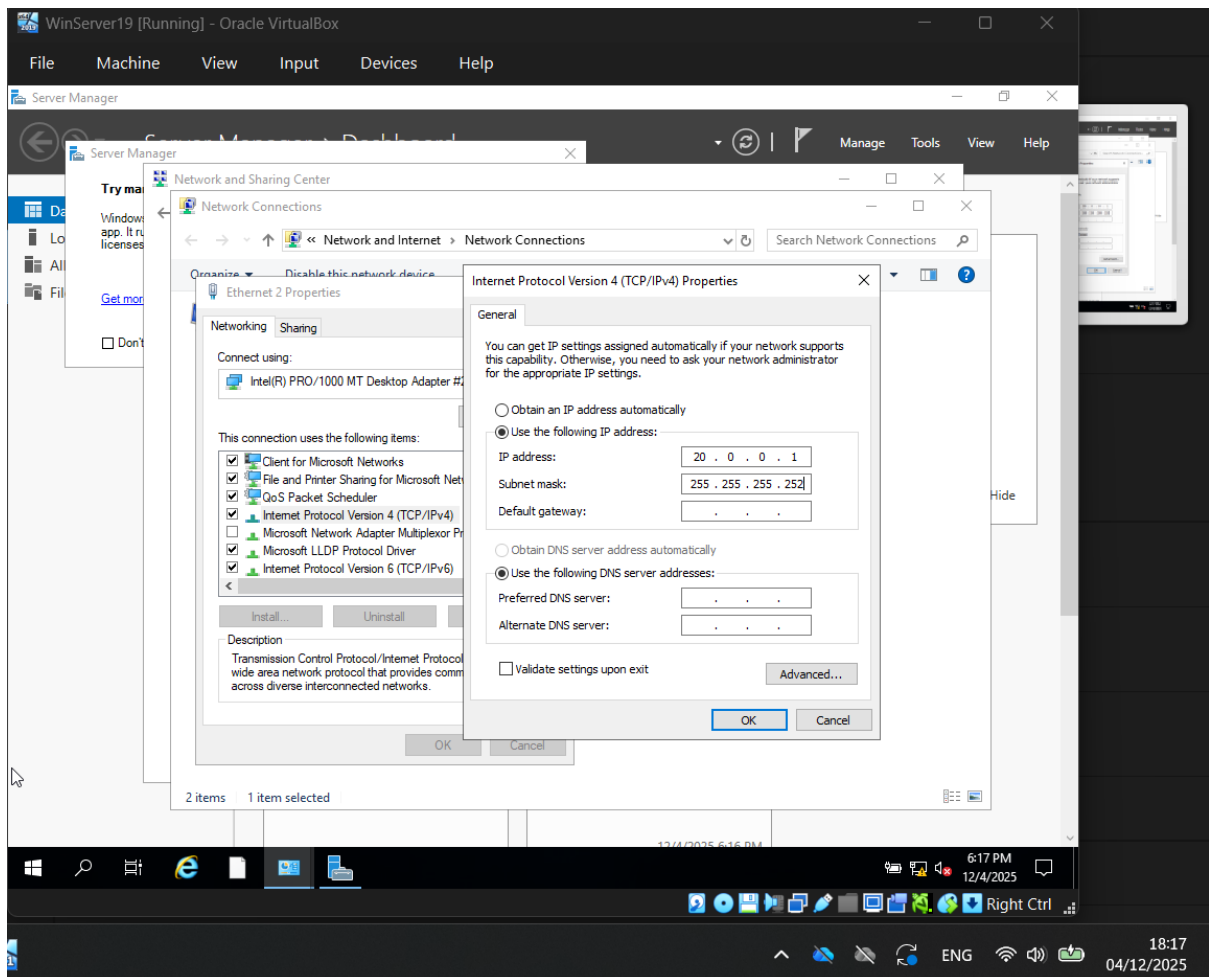
Following, I went to the Debian Server's terminal, and I assigned the LAN IP and the Transit IP.



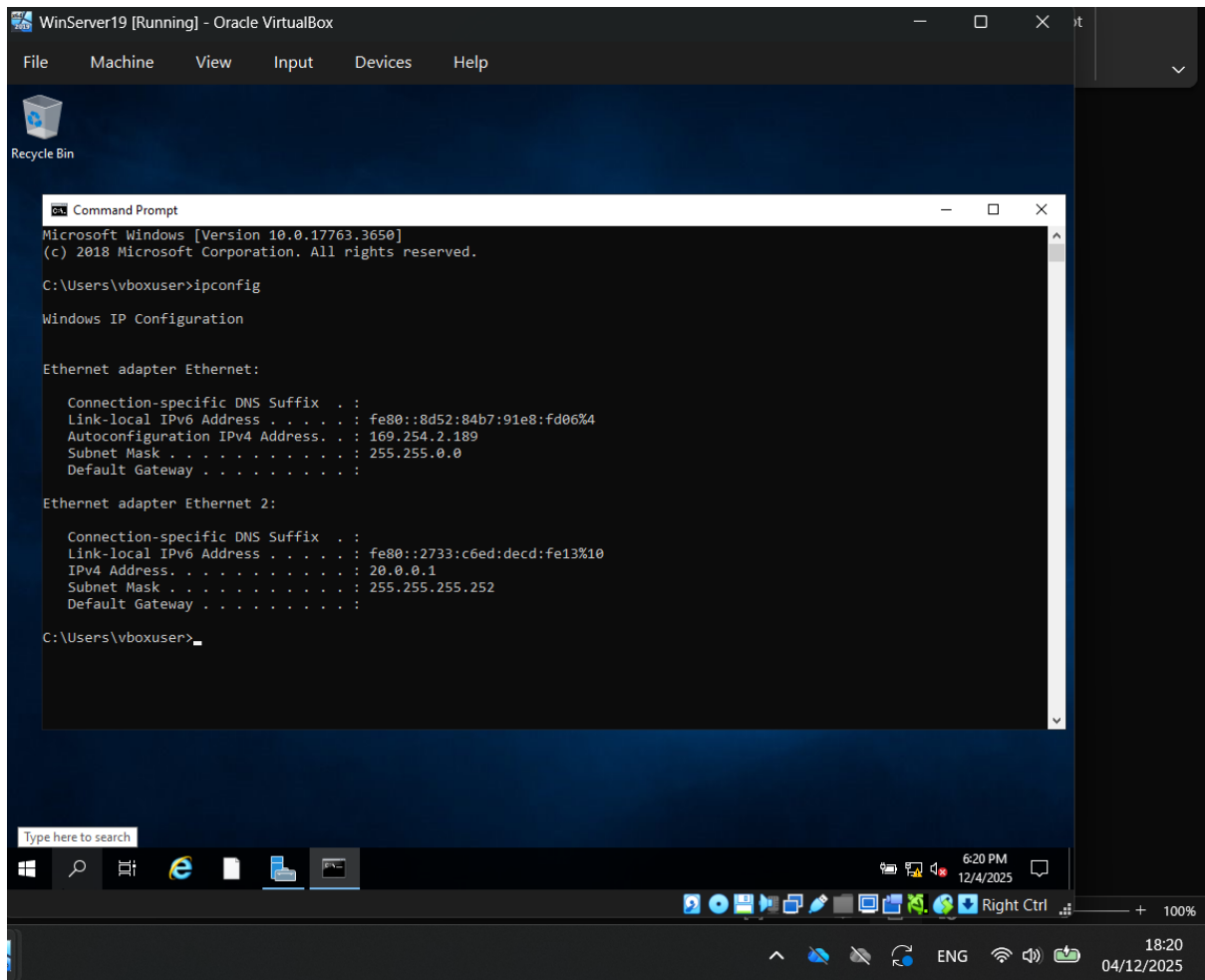
Moving forward, I need to configure the Network settings for the Windows Server, so the ping command works properly. I set up adapter 1 as Internal Network and gave it the name LAN1.



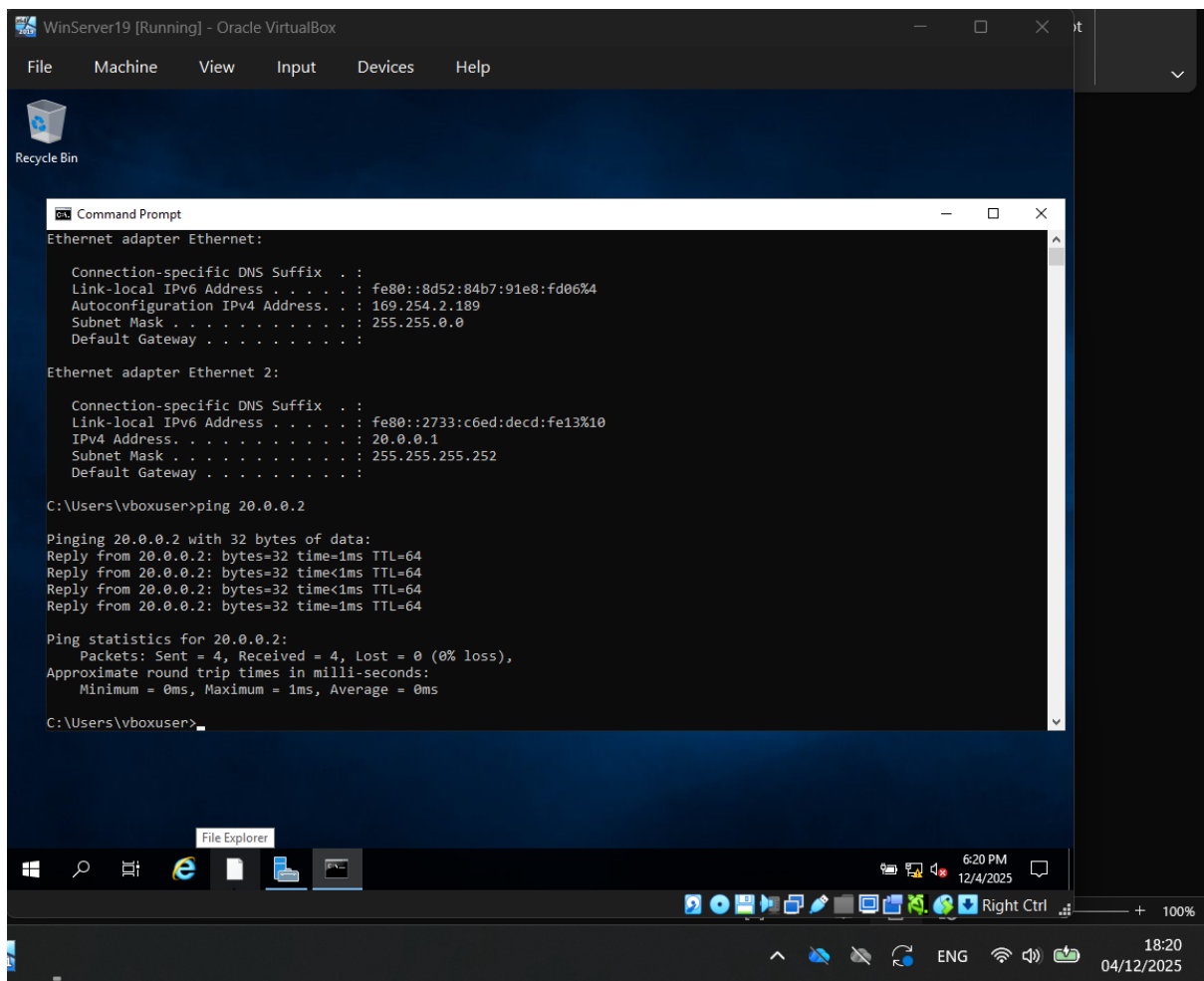
I also set up the Adapter 2 as Internal Network and gave it the same name of TRANSIT found in the Debian adapter's name. Names must match otherwise they will not be connected.



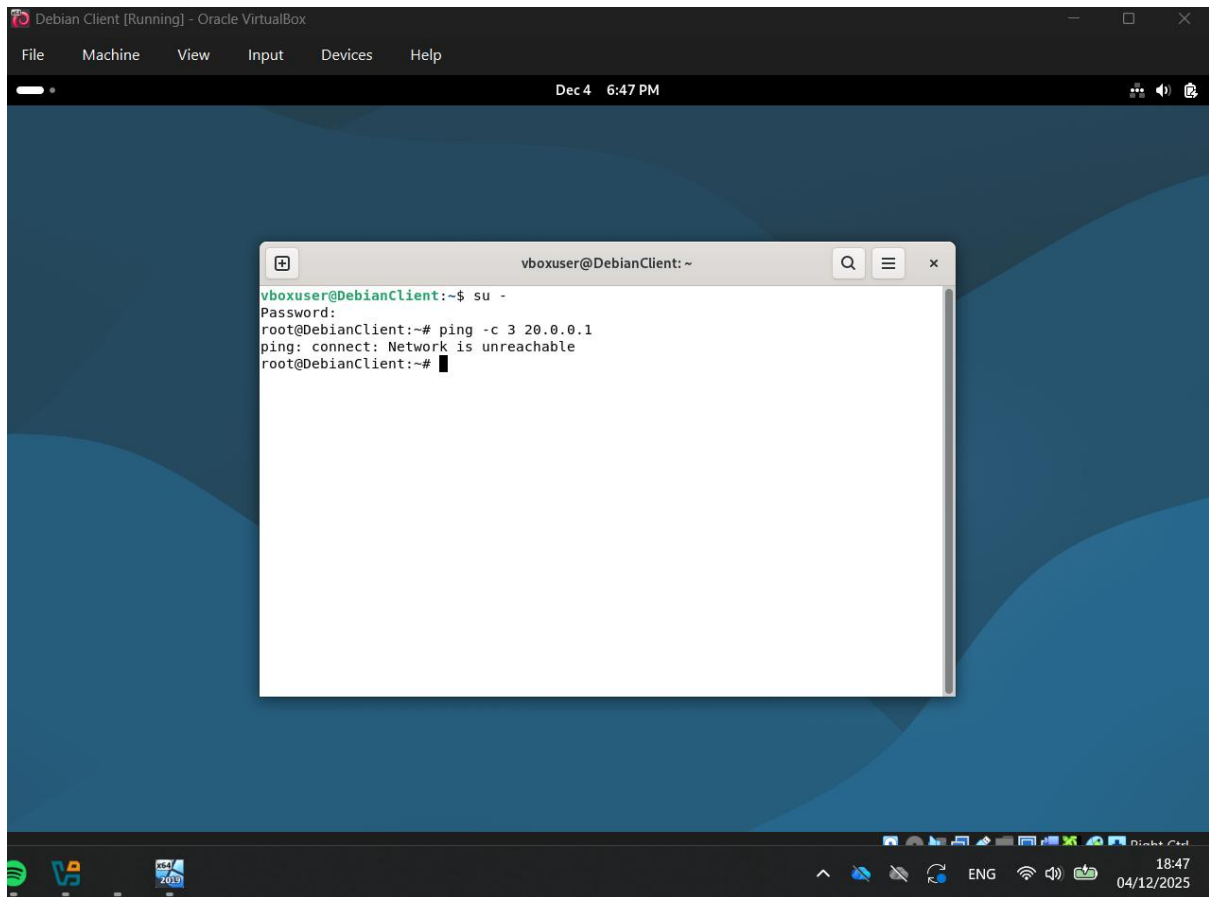
Next, I need to manually set the IP address under Internet Protocol Version 4 (TCP/IPv4) found in Network and Sharing Center.



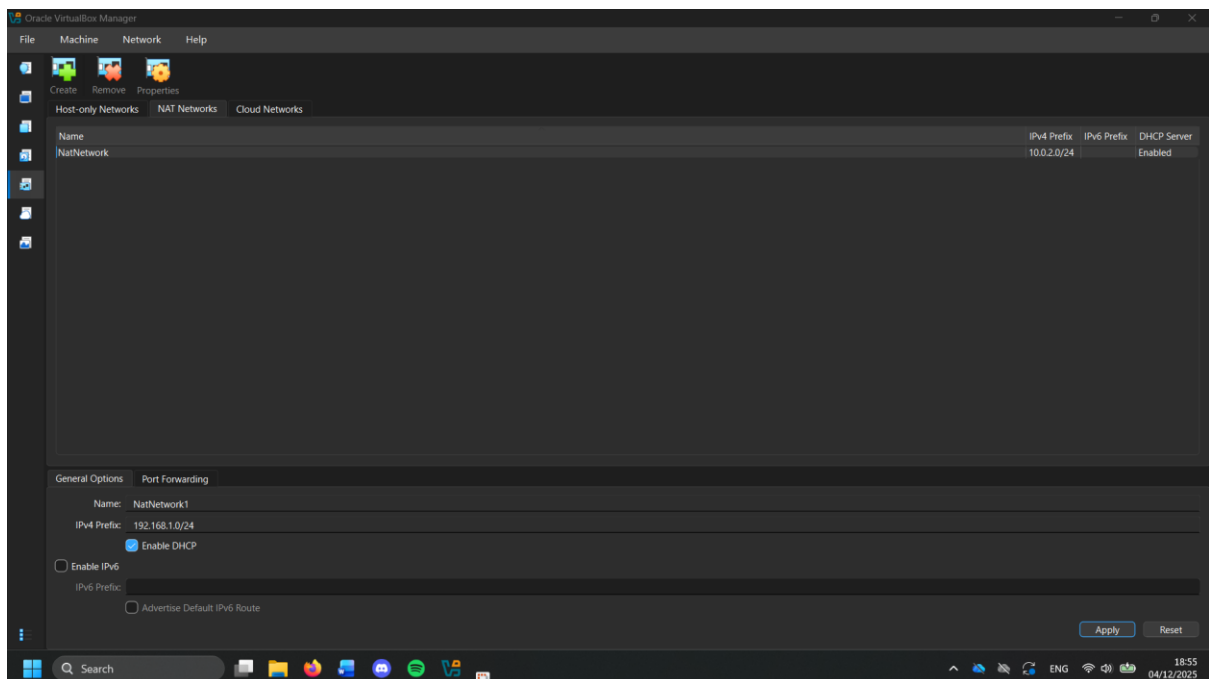
Checking that everything is set up correctly. All appears to be in order.



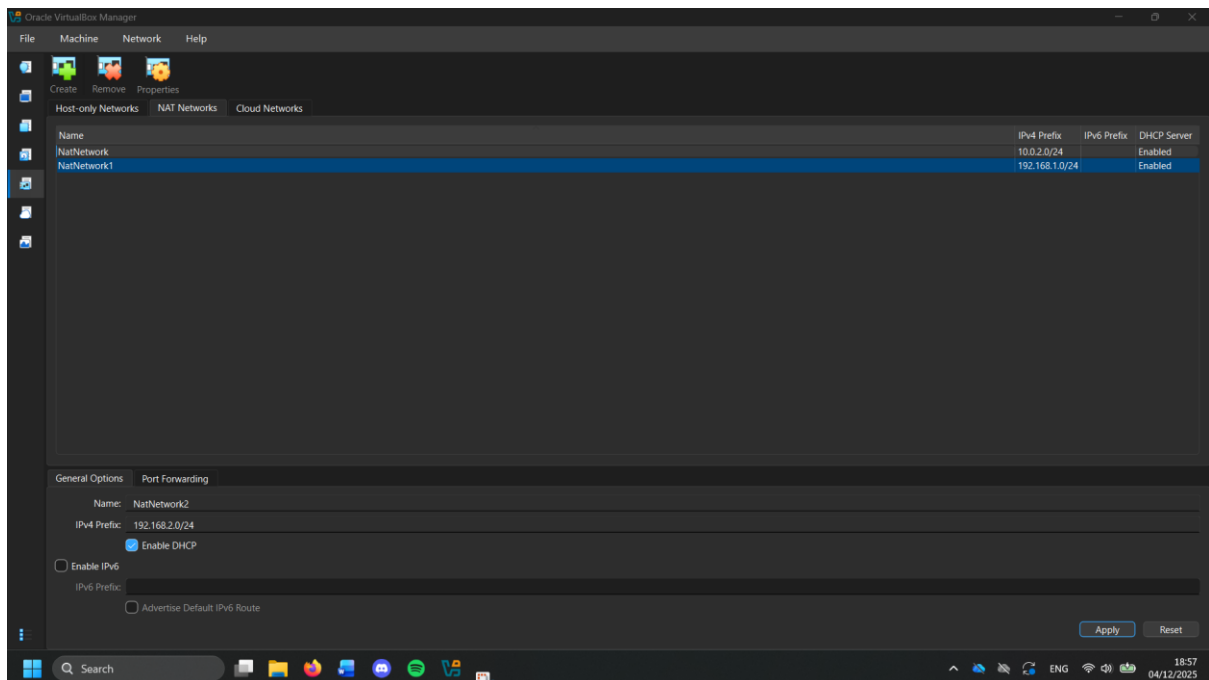
After that, I tested the connectivity in Command Prompt. It works perfectly.



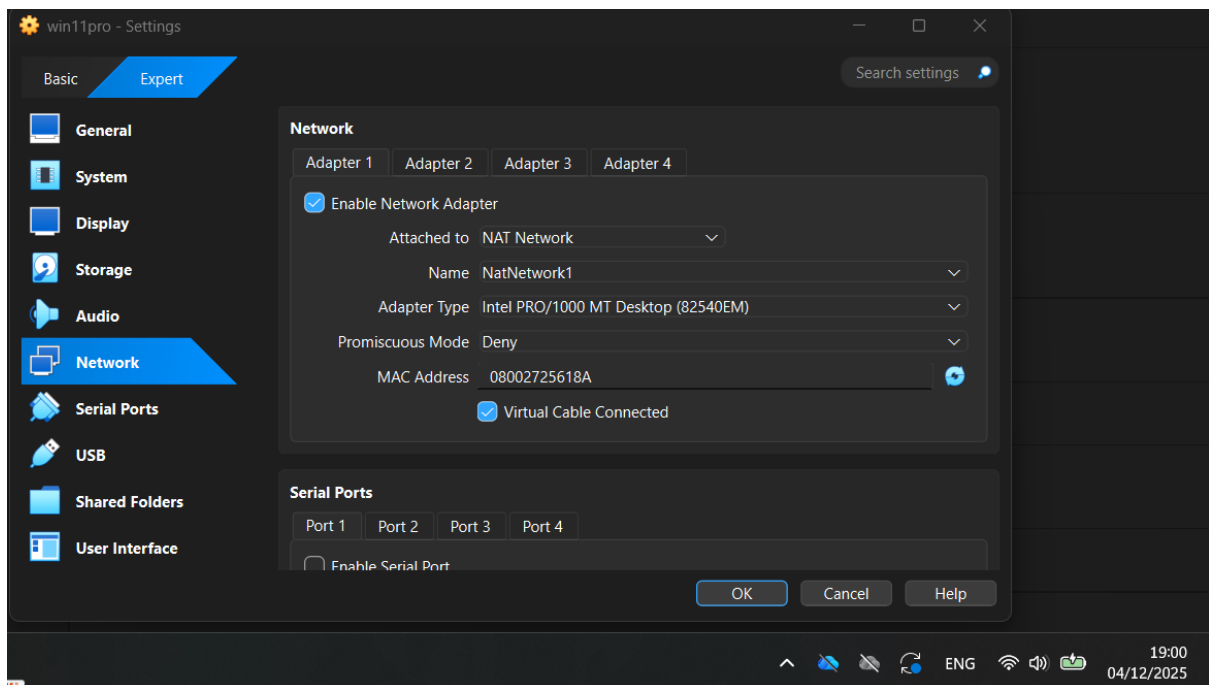
I'm unsure what I did wrong in the last step but the ping from the Debian Server did not work. I will try again using a different approach.

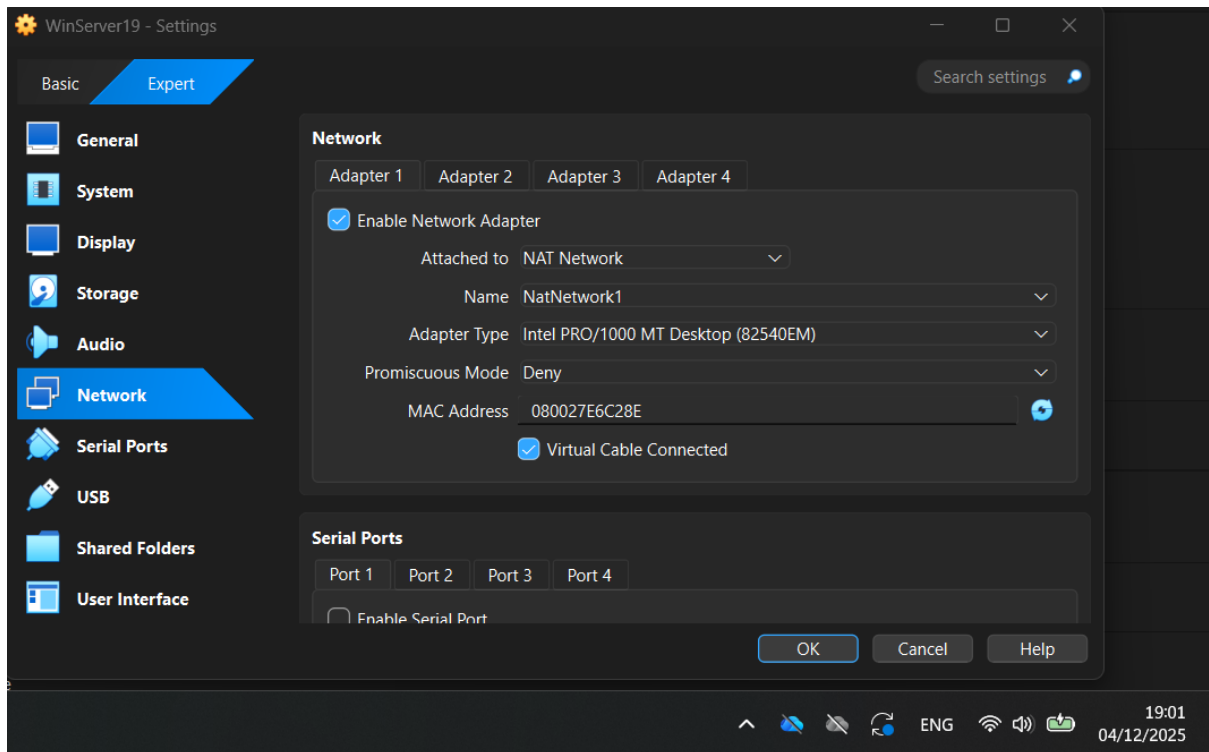


First step is to create a Nat Network 1; this Network will be the Windows one. I set up the appropriate CIDR to the one shown in the topology and enabled DHCP.

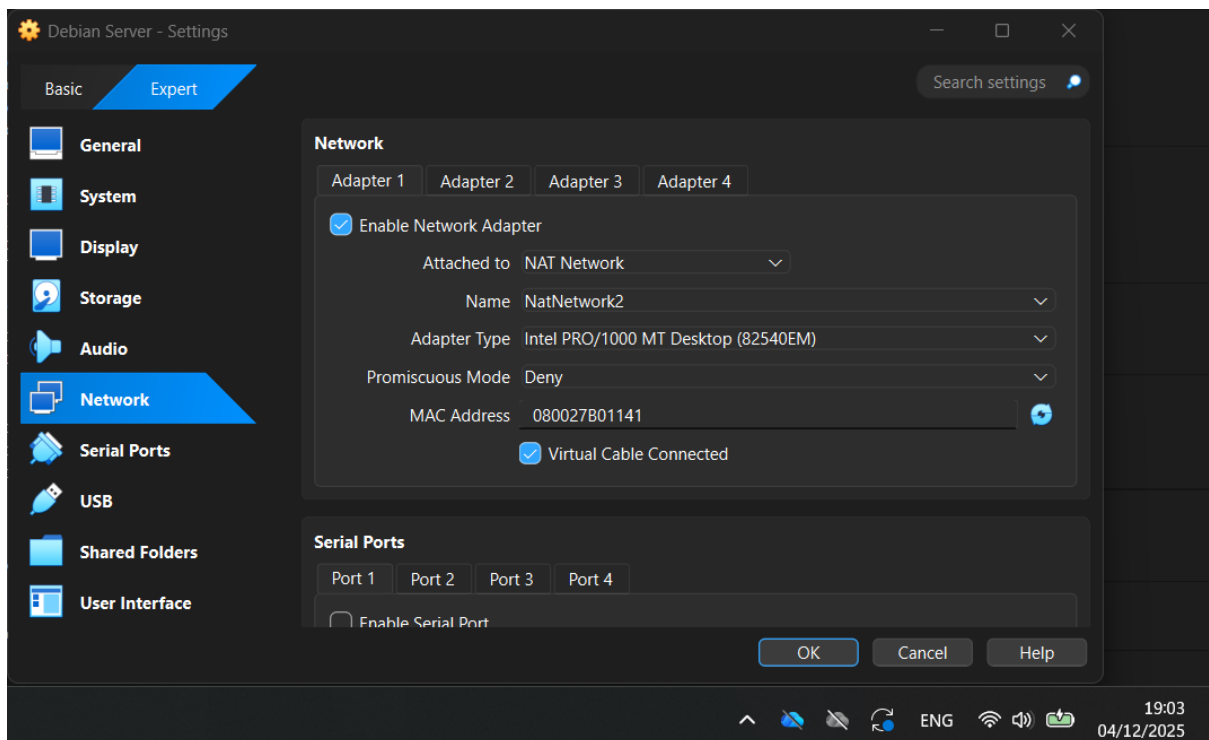


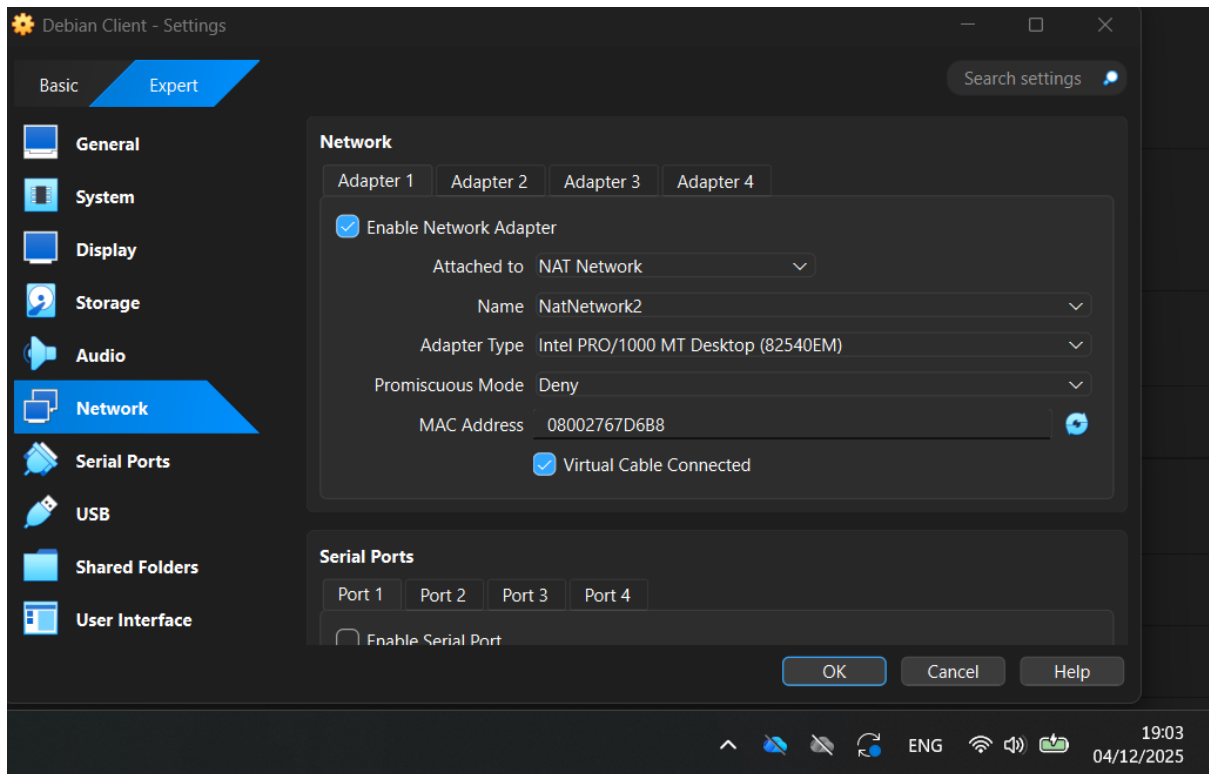
I did the same with a second Nat Network; this one pertaining to the Linux counterpart with the appropriate CIDR shown in the topology.



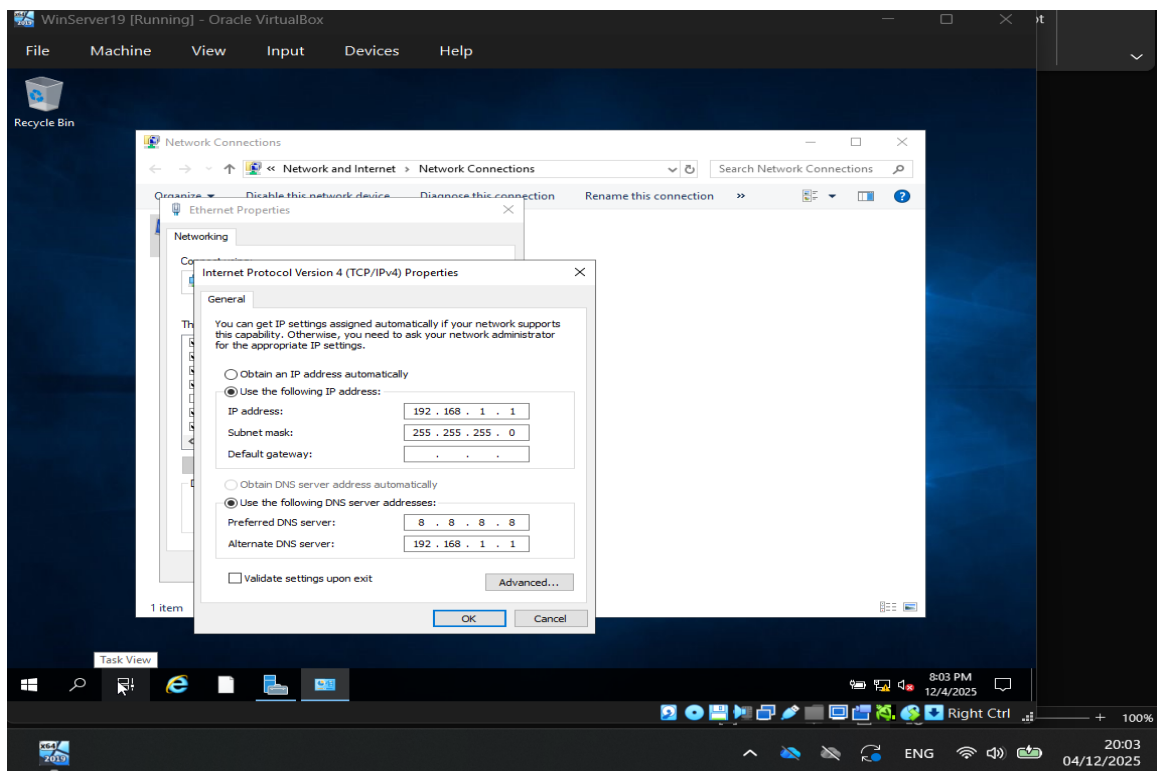


Next, I will connect the Windows VMs to NatNetwork1 for both the Windows Server and the Windows Client as shown in both screenshots above.

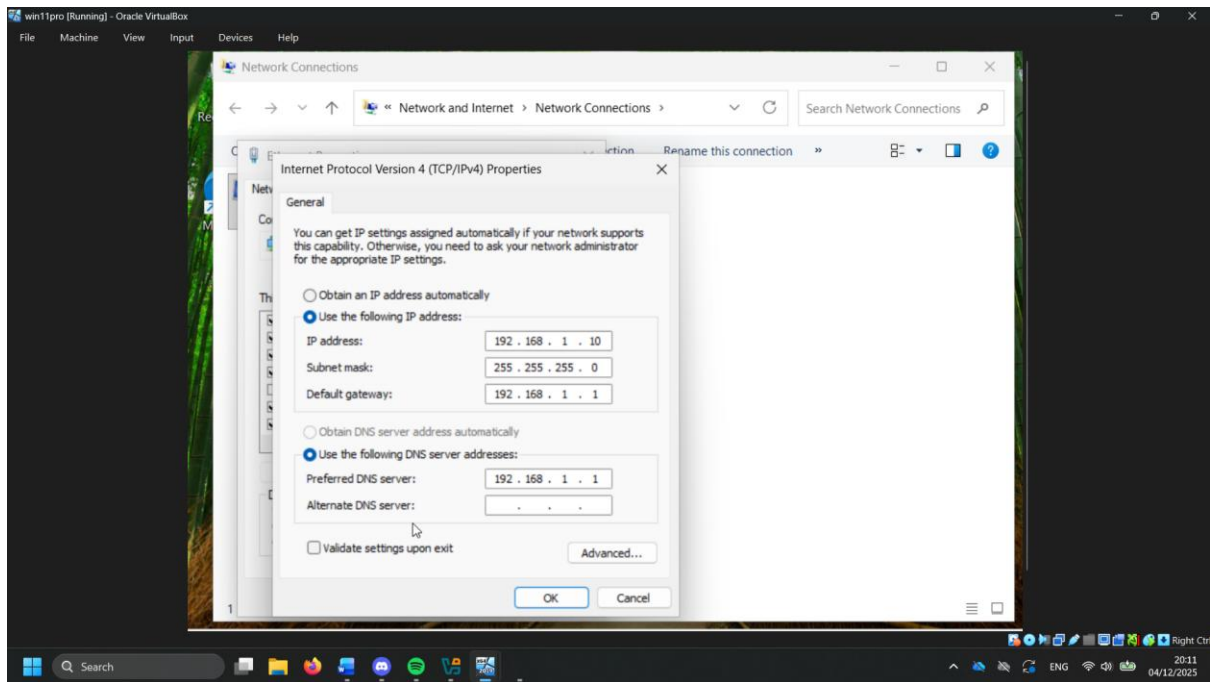




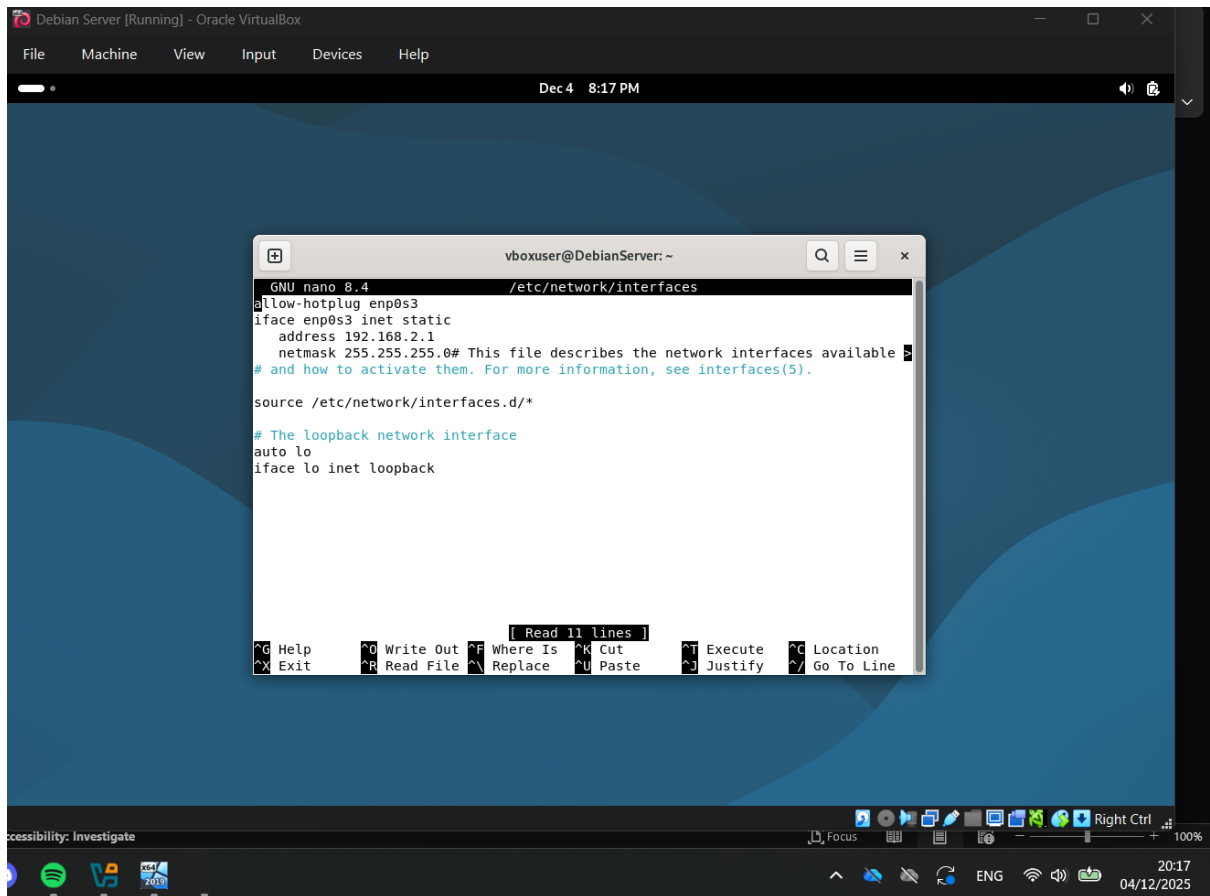
Same process with the Debian Server and Client counterparts but instead of connecting them to NatNetwork1, I connected them to NatNetwork2.



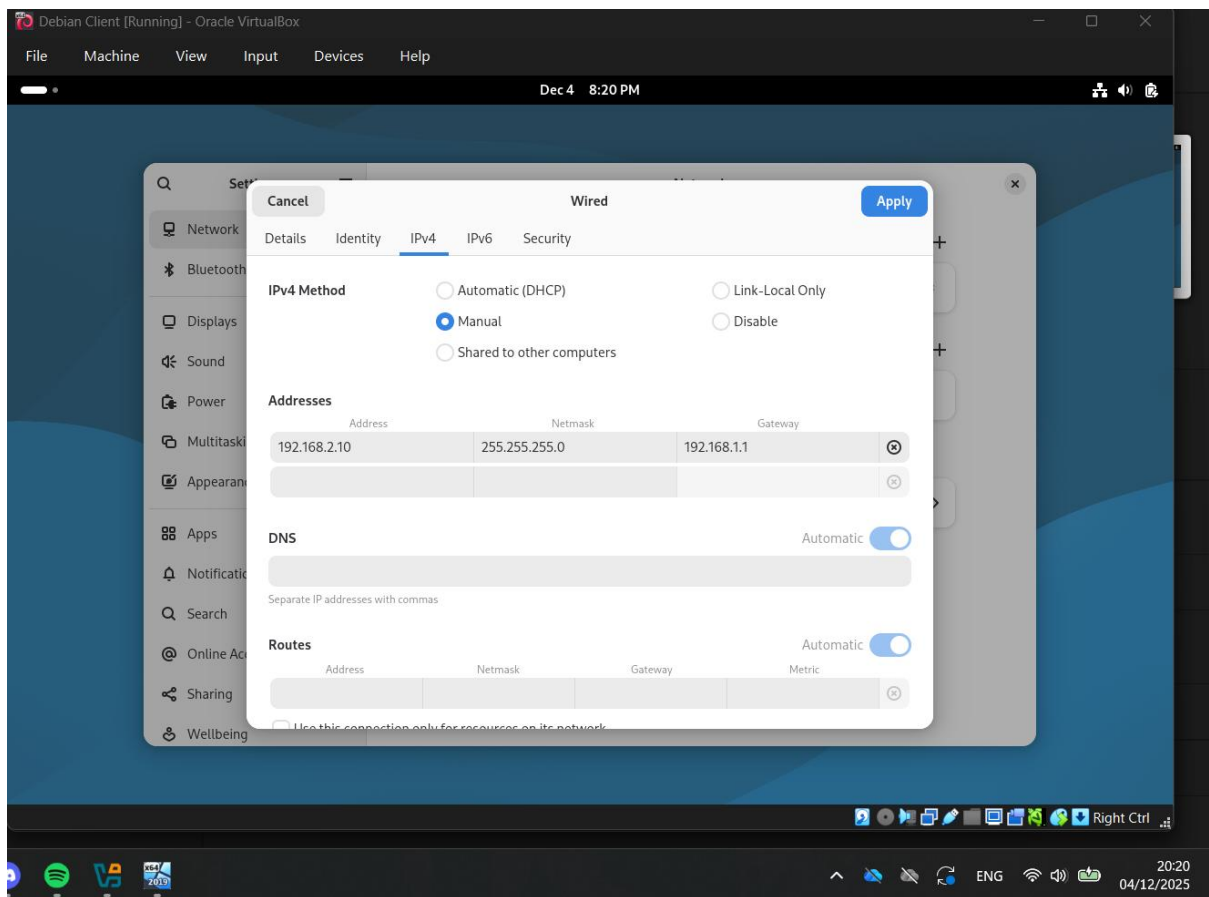
Next step is to set up the Windows Server with the appropriate IP address shown in the topology.



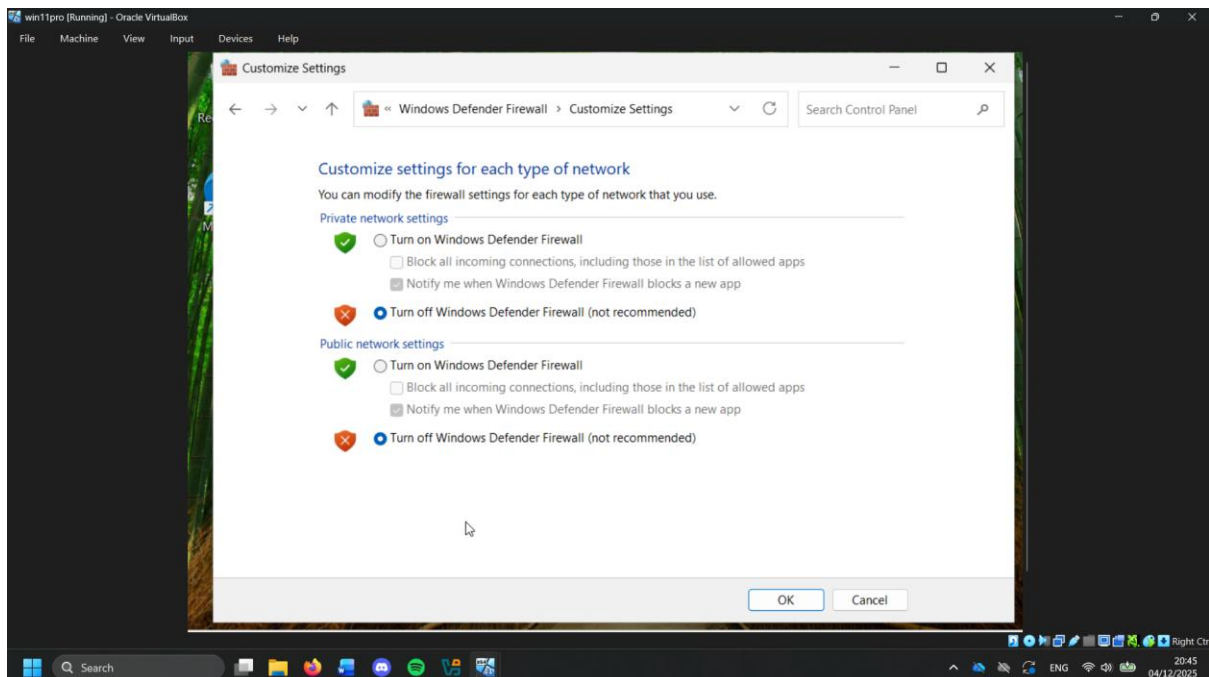
Doing the same for the Windows Client.



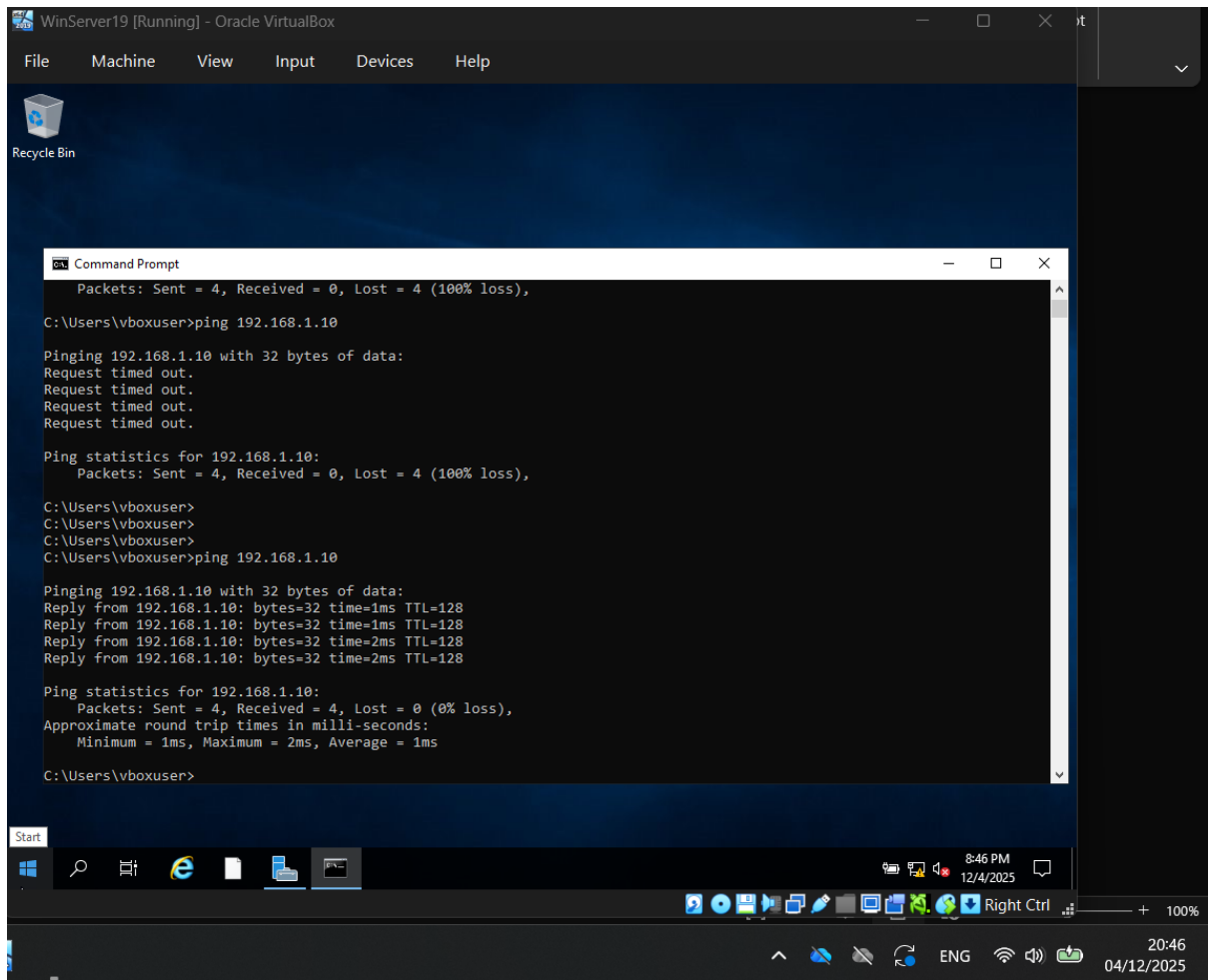
Making use of the text editor nano, I configured the correct IP address on the Debian Server.



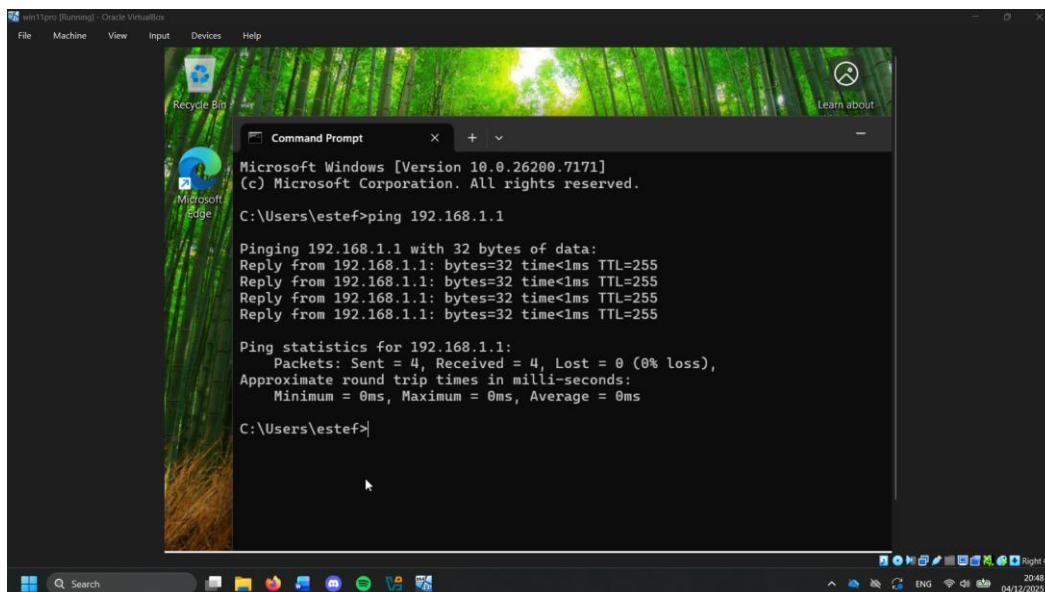
Lastly, I needed to configure the proper IP address for the Debian Client counterpart. I did so by locating the network settings.



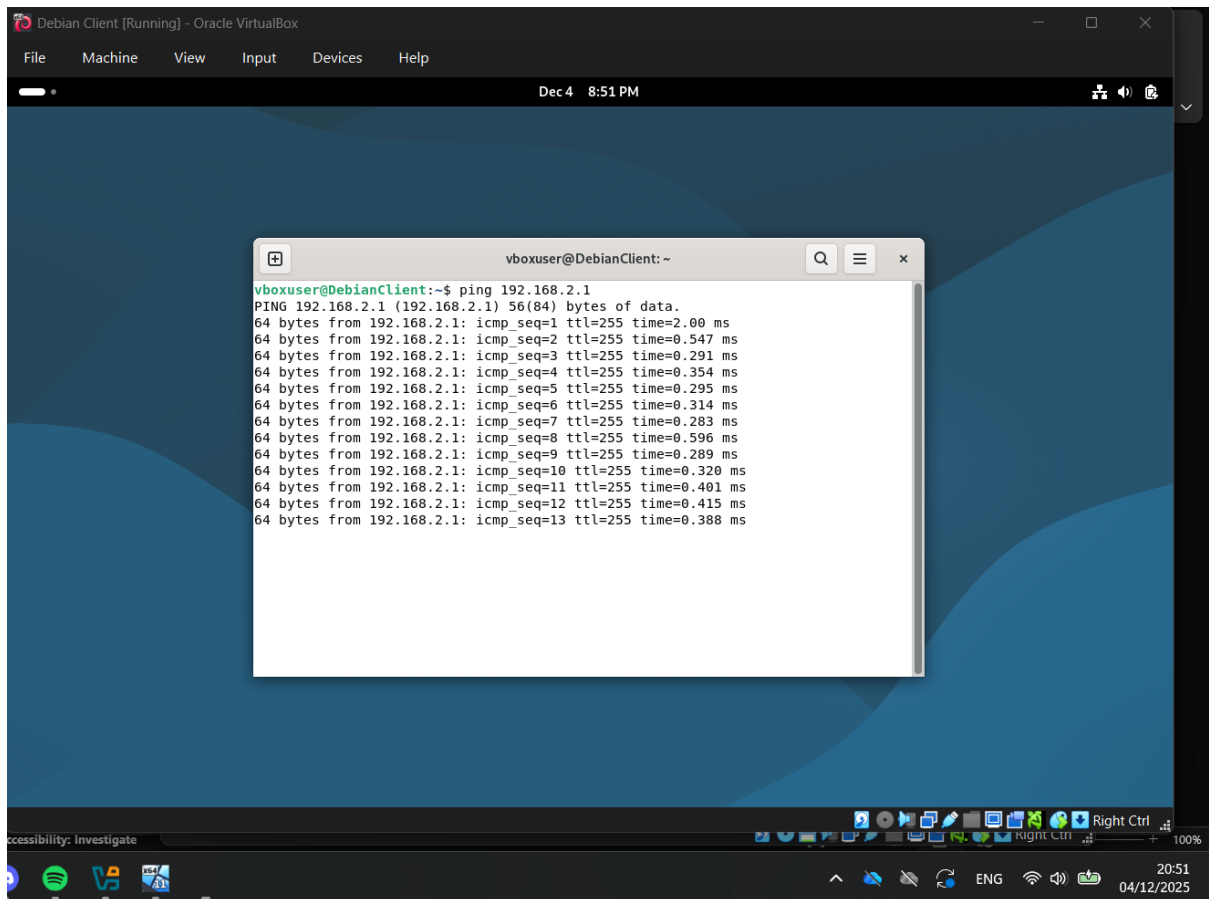
To be able to ping the Windows client, I had to disable Firewall.



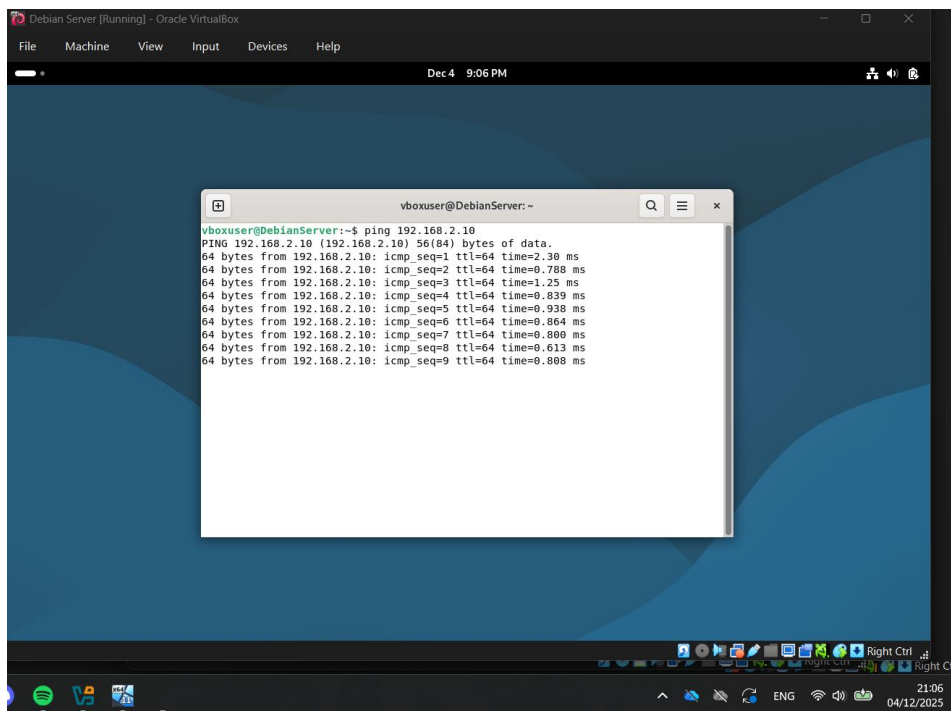
As seen in the picture above, I tried multiple times to reach 192.168.1.10 (Windows client) but it was not possible due to the Windows Firewall protecting the machine.



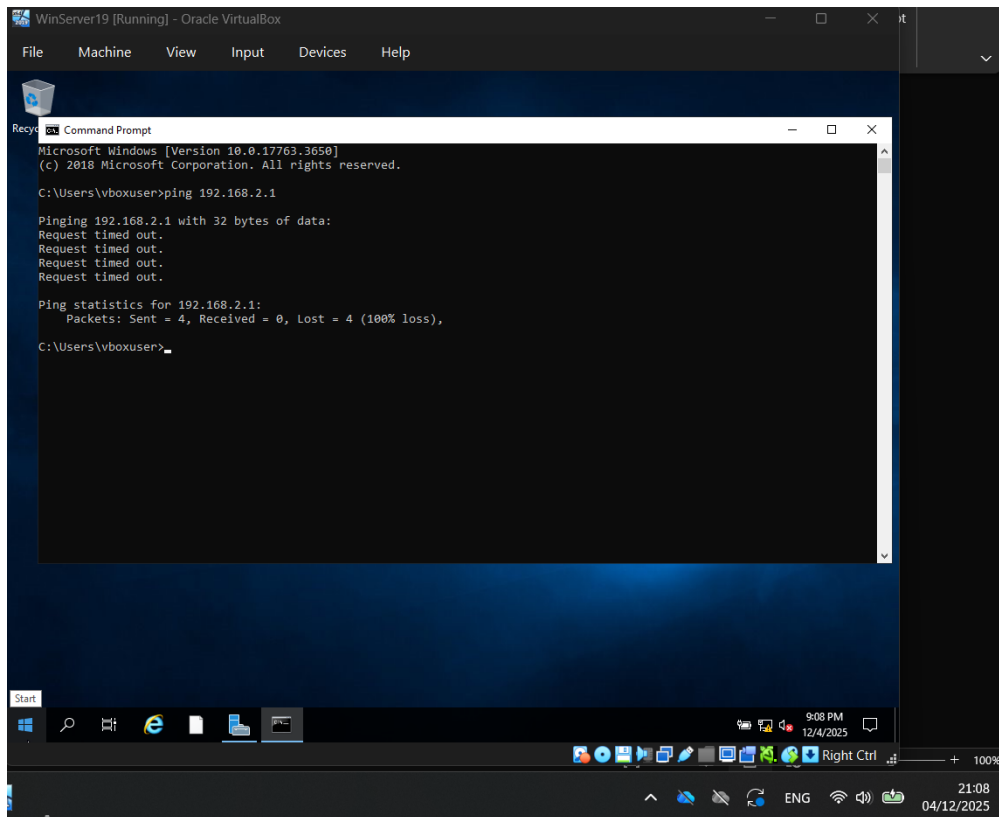
The ping to 192.168.1.1 (Windows server) from the Windows client was successful.



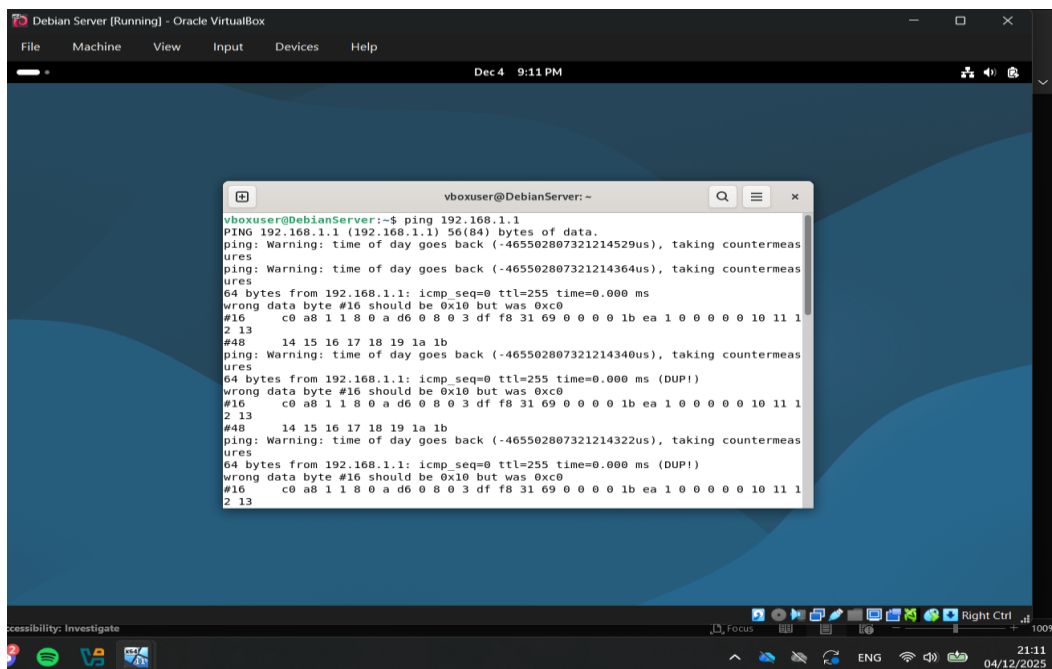
This is the Debian Client side pinging the Debian Server. It is working.



This is the Debian Server pinging the Debian Client successfully.



When trying to ping the Debian Server from the Windows Server the connection is unsuccessful because Nat networks in Oracle are isolated from one another by default and do not automatically route traffic between them.



Same thing occurs when trying to ping the Windows Server from the Debian Server.

4. Reflection

Going into this assignment, I personally thought it was going to be an easy task to complete which would not take a long time. My judgment was somewhat accurate, however, only for the first part of the assignment, the research. When it comes to the second task, which was the practical side, I ran into a plethora of problems which took long to understand and eventually fix. For example, my first idea was to implement the networks via static routing because I thought it would be the simpler way which would also provide an easier path to document my work. Additionally, and for the sake of this assessment, there was no point in using dynamic routing since I was not going to make use of the scalability it offers. However, upon trying and trying and running into problems I could not manage to find a fix to; I opted to ditch that path altogether and decided to employ dynamic routing combined with a bit of static routing to achieve the end goal of completing this assignment. I was thoroughly proud of my work as soon as I managed to make all the pings work and for all the networks to function as intended. I personally also enjoyed the trial and error involved in finishing the second part of this assessment. Overall, it was a great experience which will help me elevate my skills and understanding of Virtual machines and networking.

5. References

- [1] S. Susnjara and I. Smalley, "What are hypervisors?," IBM, <https://www.ibm.com/think/topics/hypervisors> (accessed Dec. 1, 2025).
- [2] J. Ellingwood, "Understanding IP addresses, subnets, and CIDR notation for networking," DigitalOcean, <https://www.digitalocean.com/community/tutorials/understanding-ip-addresses-subnets-and-cidr-notation-for-networking> (accessed Dec. 1, 2025).
- [3] Sameer, "IPv4 address classes explained: A, B, C, D, E with examples," FirewallFlow, <https://www.firewallflow.com/ipv4-address-classes/> (accessed Dec. 1, 2025).
- [4] "What is IPv6?," Cisco, <https://www.cisco.com/site/us/en/learn/topics/networking/what-is-ipv6.html> (accessed Dec. 2, 2025).

- [5] J. Lavaya, "Advantages of dynamic routing & How Dynamic Routing Works," Tutorials, <https://onlinetutorialhub.com/ccna/advantages-of-dynamic-routing/> (accessed Dec. 2, 2025).
- [6] H. Andrea, "Comparison of static VS dynamic routing in TCP/IP Networks," Networks Training, <https://www.networkstraining.com/static-vs-dynamic-routing-in-networks/> (accessed Dec. 2, 2025).
- [7] GeeksforGeeks, "Static and dynamic routing," GeeksforGeeks, <https://www.geeksforgeeks.org/computer-networks/difference-between-static-and-dynamic-routing/> (accessed Dec. 2, 2025).
- [8] "Virtual Private Networks (vpns)," NCSC, <https://www.ncsc.gov.uk/collection/device-security-guidance/infrastructure/virtual-private-networks> (accessed Dec. 2, 2025).
- [9] "Dynamic Host Configuration Protocol (DHCP) overview," Microsoft Learn, [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831825\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831825(v=ws.11)) (accessed Dec. 3, 2025).
- [10] "What Is Dynamic Host Configuration Protocol (DHCP)?," Fortinet, <https://www.fortinet.com/uk/resources/cyberglossary/dynamic-host-configuration-protocol-dhcp> (accessed Dec. 3, 2025).
- [11] "Active directory domain services overview," Microsoft Learn, <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview> (accessed Dec. 3, 2025).
- [12] "What is public key infrastructure (PKI) & how does it work?," Okta, <https://www.okta.com/identity-101/public-key-infrastructure/> (accessed Dec. 3, 2025).
- [13] "The PKI principles," NCSC, <https://www.ncsc.gov.uk/collection/in-house-public-key-infrastructure/pki-principles> (accessed Dec. 4, 2025).