

Assignment – Network Operating Systems & Services

Ben Roberts (2205382)

Network Operating Systems & Services (ACCA5023)

University of Wales Trinity Saint David – School of Applied Computing

Table of Contents

Introduction.....	4
Task 1.....	5
Hypervisors.....	5
IP Addressing.....	7
Routing.....	9
Network Services.....	10
Task 2.....	11
Choosing a Hypervisor.....	11
Download and Setup.....	12
Configure IP Addressing.....	17
Testing and Troubleshooting.....	25
Task 3.....	27
Technical Reflection.....	27
Personal Reflection.....	27
Conclusion.....	28
References.....	29

Table of Figures

Figure 1: Diagram of a hypervisor architecture example [1].....	5
Figure 2: Example of an IPv4 address [2].....	7
Figure 3: Diagram of the representation of an IPv6 address in the preferred form [3].....	8
Figure 4: Initial configuration of a VirtualBox machine.....	12
Figure 5: Partitioning a Windows server.....	13
Figure 6: Installing the Virtual Box Guest Extensions.....	14
Figure 7: Installing Windows 10 Enterprise in another VM.....	15
Figure 8: Installing the Debian server.....	16
Figure 9: Network topology diagram [4].....	17
Figure 10: IP addressing table for network.....	17
Figure 11: Windows Server IP configuration.....	18
Figure 12: Windows Server ipconfig.....	19
Figure 13: Windows client IP address configuration.....	20
Figure 14: Configuring an IP address on a Debian server.....	21
Figure 15: All IP addresses on the Debian server.....	22
Figure 16: Setting the Debian Client's IP address while installing it.....	23
Figure 17: Windows Server screenshot showing VLSM.....	24
Figure 18: Successful ping between the Windows client and the Windows server.....	25
Figure 19: Successful ping between the Debian client and Debian server.....	26
Figure 20: Unsuccessful ping between Debian and Windows clients.....	26

Introduction

The purpose of this report is to explore the process of setting up a network of four virtualised systems on a simulated network. It also discusses a variety of topics related to networking and virtualisation.

This report is separated into three tasks, with each task focusing on a different topic:

- Task 1 is focused on technical discussion
- Task 2 focuses on the process of setting up the virtual network
- Task 3 serves as a reflection for the second task.

Task 1

Hypervisors

Hypervisors are a method of using one computer system to simulate another via a process called virtualisation. There are two kinds of hypervisors: Type 2 hypervisors, also known as hosted hypervisors [1], are software based and rely on a host operating system for resource allocation while Type 1 (or “bare metal”) hypervisors run the simulated system with direct access to hardware resources. [5]

The architecture of a hypervisor involves a host system managing a number of guest systems via software. Each virtual device has components simulated by what is referred to as a processing element, which can either be a physical device that is passed through to the virtual guest, or an emulated component handled by the VM. While memory and CPU cores are often physical components assigned to the VM, other devices such as hard disks or I/O controllers are emulated and controlled mainly by the host system. [1]

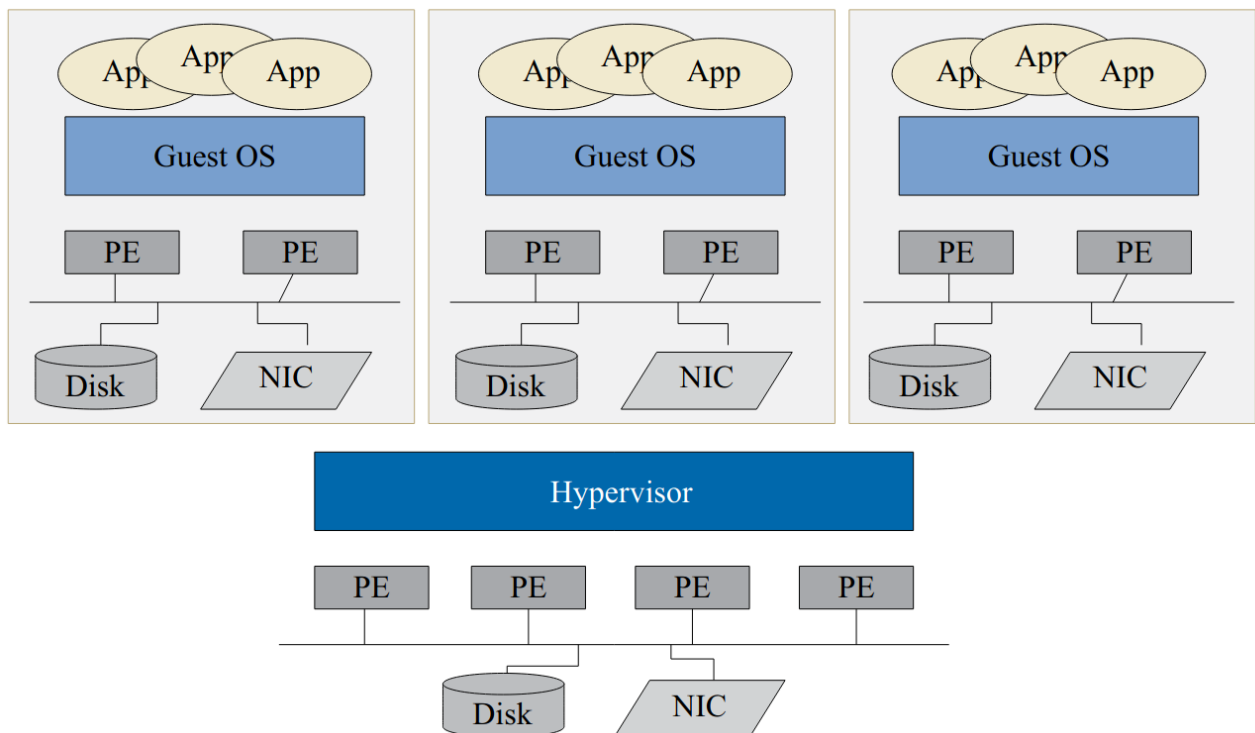


Figure 1: Diagram of a hypervisor architecture example [1]

Virtual systems within a hypervisor are isolated from other guest systems as well as the host. This benefits security because it means security breaches will only affect the targeted guest system, reducing the overall surface area for attacks [6]. Because systems running within a hypervisor are virtual, this also means that they can be backed up and restored via software snapshots. [7]

Running a hypervisor, especially one that is managing multiple VMs at once, is very intensive on system resources. Many modern hypervisor systems also rely on hardware acceleration provided by CPU architectures to improve performance. [8]

Since hypervisors provide a complete virtual computer system that the guest operating system is running on, this usually provides better software support than compatibility layers such as Wine, although strictly performance-reliant applications may not run correctly due to the speed deficiency, especially in instances where components have to be accurately emulated at a low level. [1]

Because hypervisors are virtual, they are easily scalable and do not require adding extra hardware to provide specific requirements, especially for use cases that do not require that many system resources. [9]

While Type 1 hypervisors have better performance [5] and have better security as well as less points of failure as they do not rely on a host operating system to provide low-level functionality, Type 2 hypervisors are often significantly easier to set up and can be installed as a program on a standard computer instead of needing specific device configuration in order to run [10].

IP Addressing

Internet Protocol Addresses are virtual addresses assigned to devices on a network for the purposes of uniquely identifying a network device for the purposes of electronic communication. IP addresses can be statically assigned to a device via manual configuration, or automatically assigned via a network service called DHCP. Two forms of IP addresses are currently in use: IPv4 which is 32-bit, and IPv6 which is 128-bit. [2]

IPv4 is commonly represented in a dotted notation of four 8-bit numbers (or octets). These numbers can be in a range between 0 and 255. An example of a valid IPv4 address would be 192.168.0.1. The first and final addresses of a network are usually reserved as the network and broadcast addresses respectively. [2]

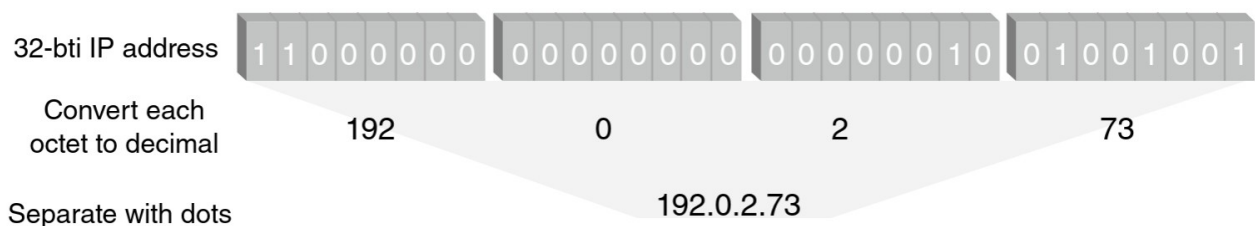


Figure 2: Example of an IPv4 address [2]

IP addresses are generally suffixed with a subnet mask. These are used for subnetting and specify the number of bits used to specify the network. The remaining bits are used to specify the host within a subnet [11]. Smaller subnet masks are able to support more hosts. Subnet masks are represented on a network device similarly to an IP address, with numeric values denoting the number of bits per octet that are used for the network address. For example, a subnet mask of 255.255.255.0 represents a /24 subnet with 254 usable hosts [12]. Subnetting is used to break up larger networks into a number in order to manage traffic and improve security [13].

Some IPv4 ranges are reserved as private, those being 10.0.0.0/8 for large networks, 182.16.0.0/12 for medium-sized networks and 192.168.0.0/16 for small networks. These addresses are not routable on the internet and are instead intended for use in LANs. [2]

As there is a hard limit of roughly 4.2 billion IPv4 addresses due to the use of a 32-bit address space, IPv6 was introduced to counteract IPv4 address depletion. Because IPv6 deals with much larger numbers, it is commonly represented using hexadecimal values. [3]

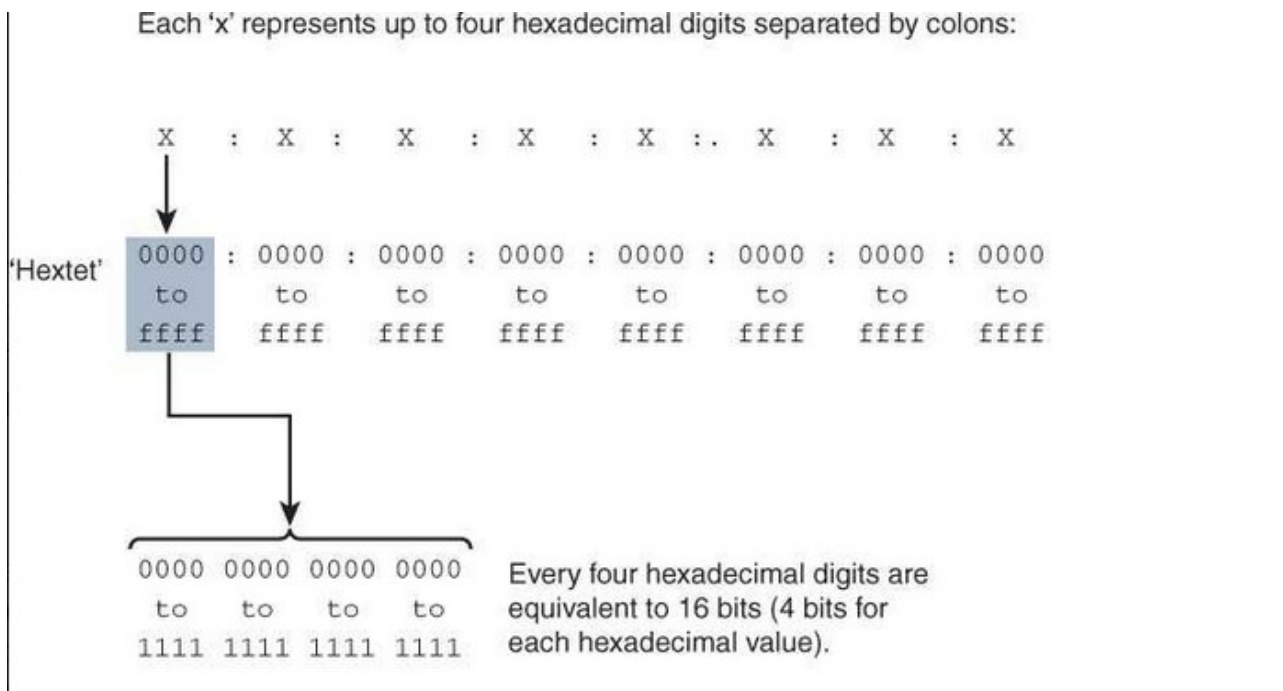


Figure 3: Diagram of the representation of an IPv6 address in the preferred form [3]

Another method of reducing the impact of IPv4 address exhaustion is to utilise classless addressing and variable length subnetting (VLSM). This involves setting a subnet mask that is not divisible by 8 to set a network that supports less hosts. This permits of breaking a network into set groups to only use the exact addresses that are required. For example, /28 subnets can support a maximum of 14 hosts which makes it suited for small networks with only a handful of computers on them. [14]

Routing

Routing is the process of transferring packets across a network using the most efficient route. The two main devices used in networking are switches and routers – switches connect computers together on a network, while routers connect multiple networks together. [15]

Routers will attempt to automatically assign the most efficient route to the target host by use of either static or dynamic routing. Static routes are manually configured on a router and require less bandwidth due to not needing to be calculated. They are also not advertised over a network and thus provide better security. They are, however, cumbersome to implement and maintain on large and/or constantly changing networks. [15]

Dynamic routing is automatically computed on a router using a variety of protocols such as RIP, OSPF and BGP. The main advantage of dynamic routing is that it has better scalability than static routing on account of being automatically assigned, and is also significantly easier to configure. Dynamic routing is also resistant to changes in an existing network. However, it lacks the speed and security of static routing. [15]

One example of how dynamic protocols can be used is in internet connected vehicles. Because vehicles do not remain in one place, the use of static routing would not be possible. Instead, a dynamic protocol for wireless ad-hoc communication designed specifically for moving vehicles could be used to detect the most optimal path to transfer data. [16]

The Routing Information Protocol (RIP) is a legacy protocol for dynamic routing based on a distance-vector format that attempts to determine the route that will take the least number of hops to get to its destination [17]. It comes in two versions, with version 2 adding support for VLSM [17]. It is easier to configure than other protocols [15] and is best suited for small networks [18].

The Open Shortest Path First (OSPF) protocol is a newer dynamic routing protocol designed for larger networks that uses the Dijkstra algorithm to calculate the shortest possible route to the destination host [18]. It has faster performance than RIP [19] and features a very low rate of packet loss [20].

Border Gateway Protocol (BGP) is another dynamic routing protocol. Unlike RIP and OSPF which are used within a network domain, it is an inter-domain routing protocol used to transfer routing information between autonomous systems [21]. It is used by ISPs to route packets across the internet [22].

Network Services

Dynamic Host Configuration Protocol (DHCP) is a method of automatically assigning IP addresses to computers on a network. It works by setting a computer to function as a DHCP server which will lease IP addresses to any devices that connect to the network. Devices that serve a specific role on a network, such as servers, should not use DHCP as doing so would make their functions less consistent however less essential devices, such as workstations, can utilise DHCP to quickly gain network access. [23]

Virtual Private Networks (VPNs) provide secure, encrypted remote access to network resources. They work by utilising a tunnelling protocol to extend an internal network beyond an organisation's LAN. The main protocol used for VPNs is IPSec, which can run in either transport mode, which encrypts the data within an IP packet for remote access and is easier to configure, and tunnel mode which relies on multiple servers to encrypt all connections over the VPN and is more secure. [24]

Active Directory Domain Services (ADDS) are a feature of Windows Server. ADDS can be used to configure a large number of devices on an enterprise network by connecting them to a local domain. This can be used to apply automatic configuration, streamline administrative processes and enable user authentication via the server using the same credentials for any device in the domain. [25] [26]

Certificate Authorities (CAs) are organisations that provide cryptographic certificates. These certificates provide public and private keys for the purposes of secure encryption. A CA can issue and manage keys within its specified zone of authority and, in the event that a private key is compromised, can revoke a certificate and issue a new one in its place. CAs can also recover lost private keys. [24]

Task 2

Choosing a Hypervisor

VirtualBox, a Type 2 Hypervisor, will be used because it is Free and Open Source Software and is also easy to configure. It is also commonly used by both end users and the industry. While a Type 1 hypervisor would provide better performance, it was considered to be too impractical on account of requiring a dedicated device. While renting a Type 1 hypervisor using a remote service such as Microsoft Azure would be an option, doing so would be expensive and would also require a constant internet connection.

Download and Setup

VirtualBox was already installed on the host computer so the only steps required were configuring the operating system.

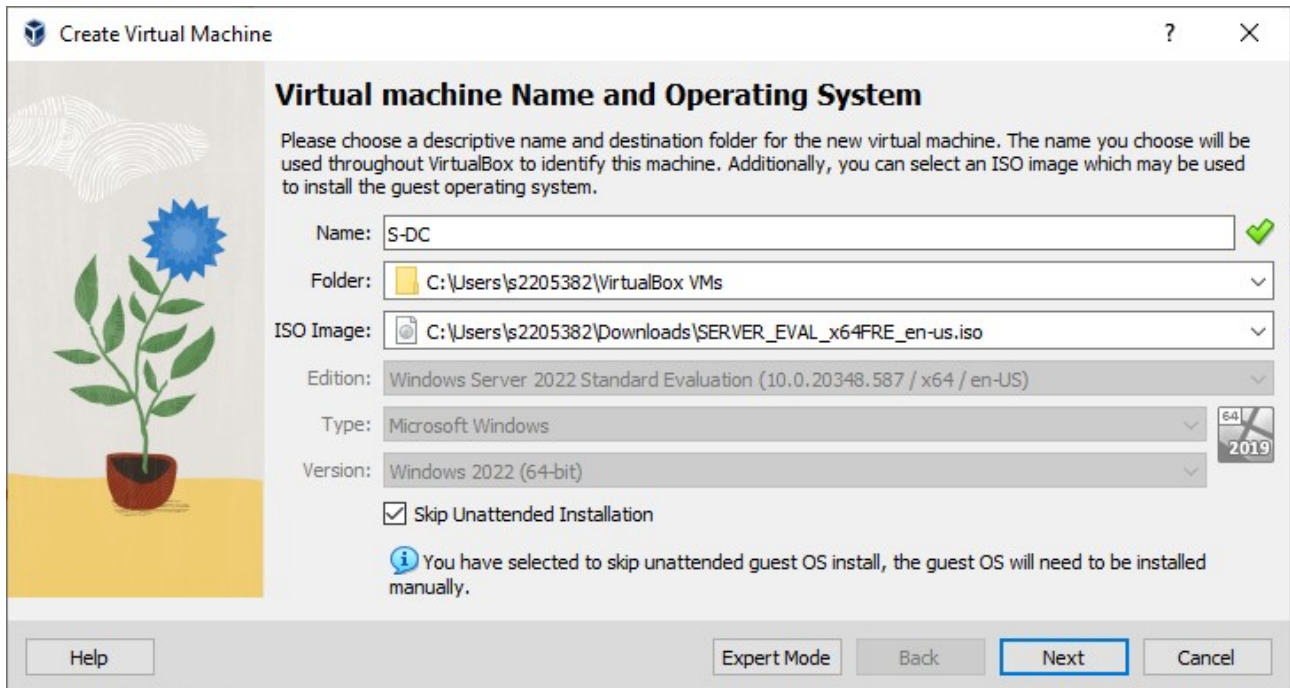


Figure 4: Initial configuration of a VirtualBox machine

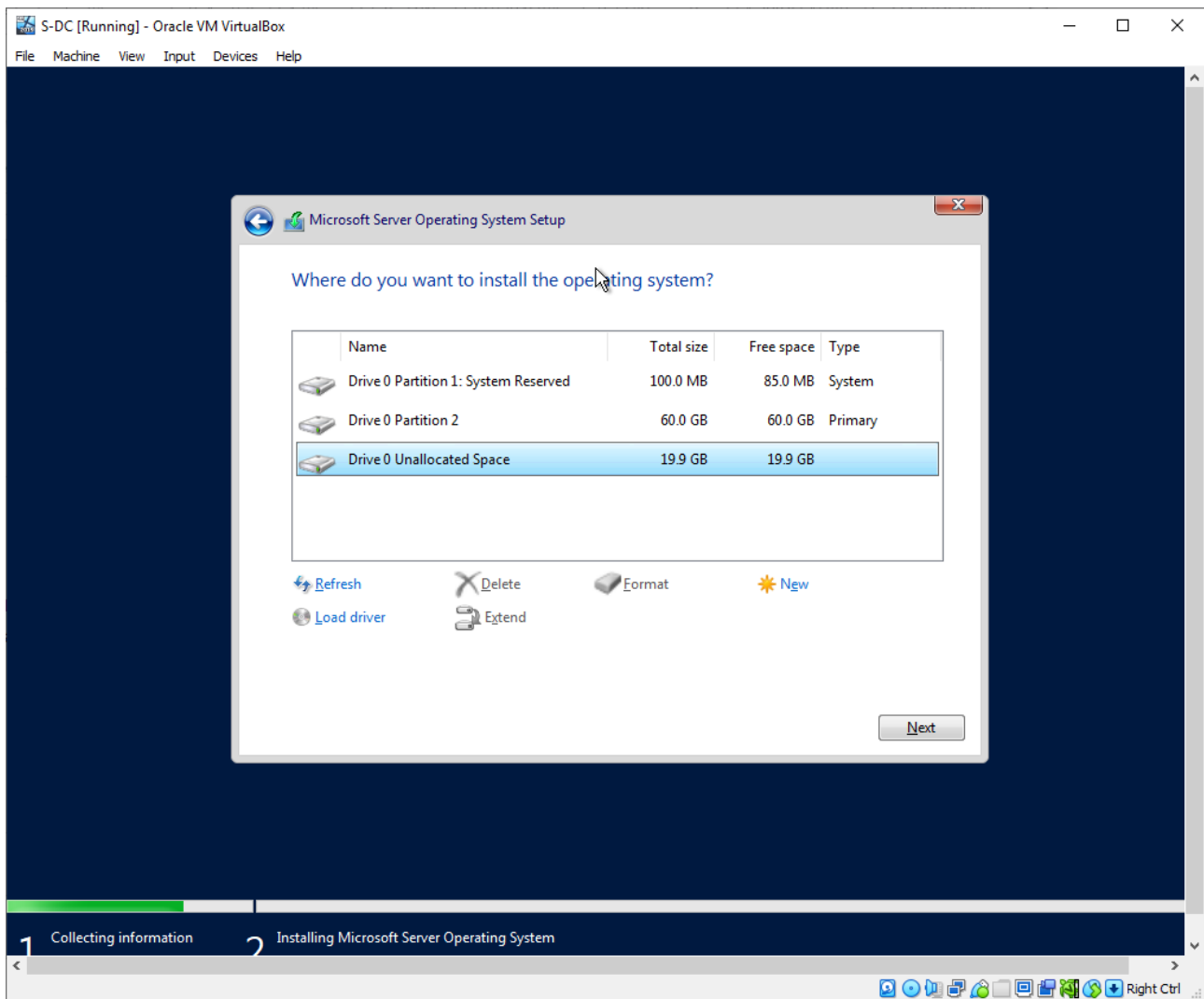


Figure 5: Partitioning a Windows server

The first VM to be installed was Windows Server 2022. It was assigned 16GB of RAM (later lowered to just 8GB due to stability issues) and was given 80GB of storage. A 20GB partition was left unallocated for future use.

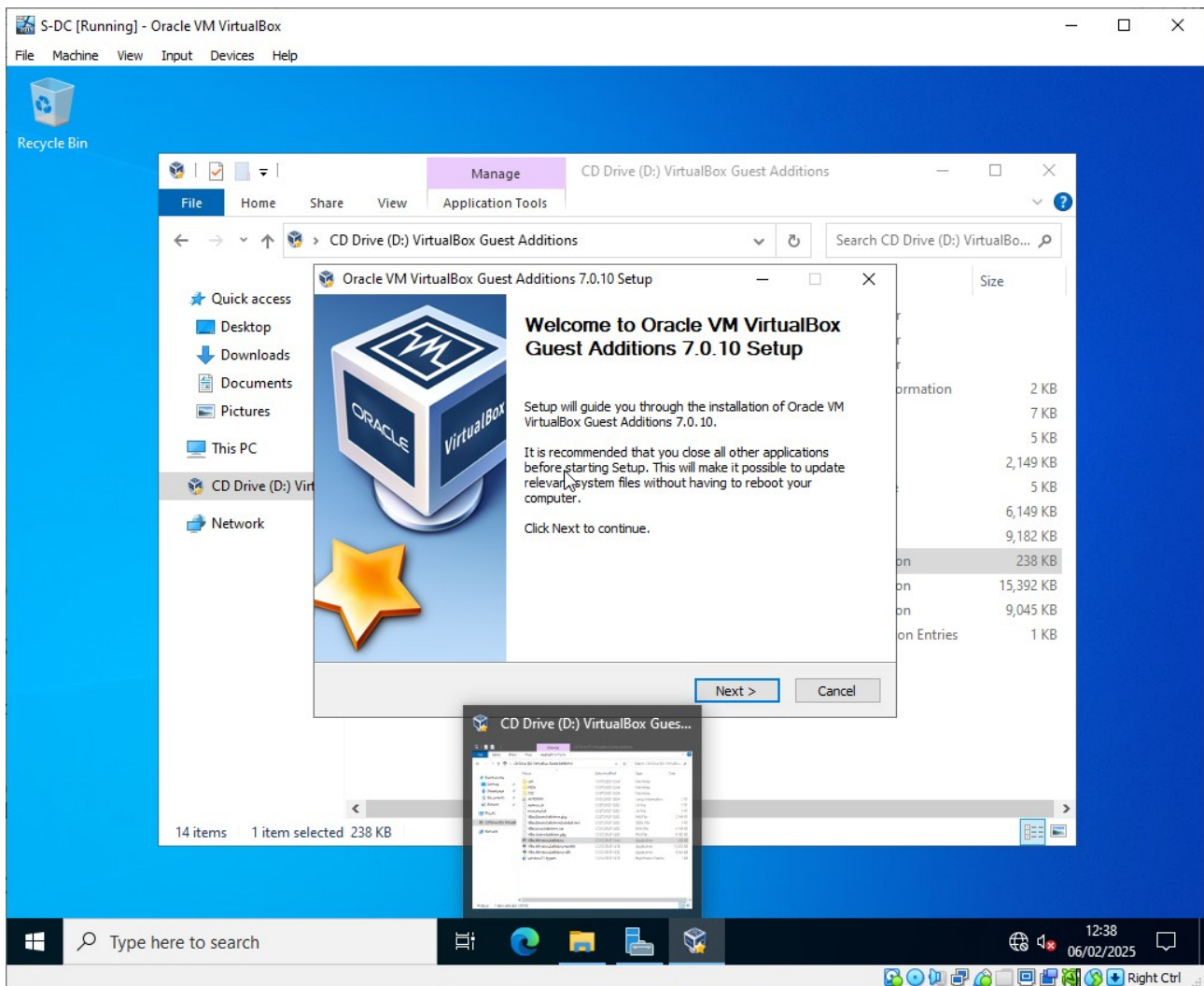


Figure 6: Installing the Virtual Box Guest Extensions

For every virtual machine, the VirtualBox host extensions were installed to enhance performance. This enables a number of features, including but not limited to better mouse support, hardware accelerated graphics, automatic resolution switching based on the size of the host window and synchronisation with the host system's real time clock [27].

Next, a Windows 10 Enterprise guest was installed in another VM.

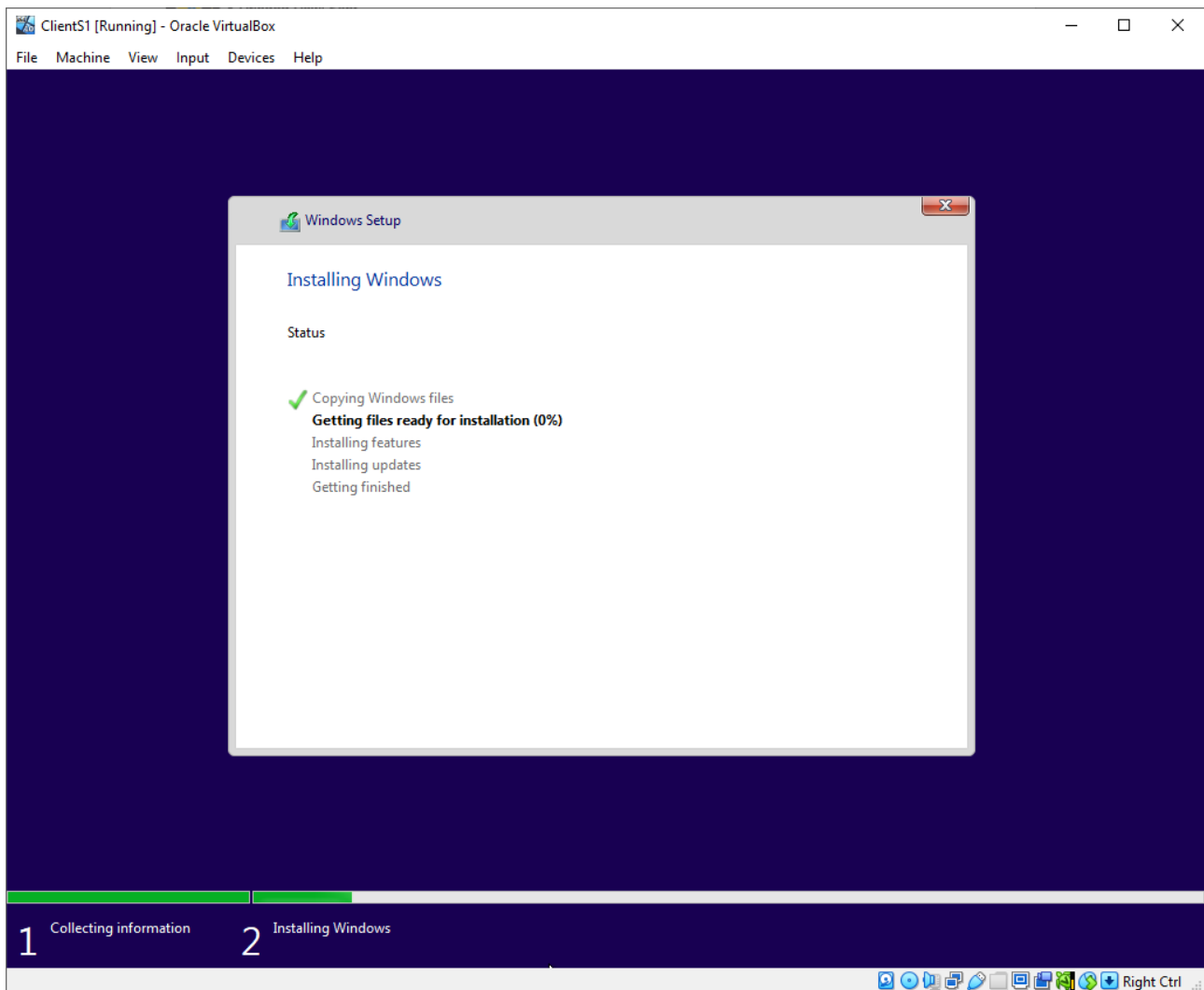


Figure 7: Installing Windows 10 Enterprise in another VM

VMs running Debian Linux were configured. Because Linux has significantly better optimisation than Windows, the Linux VMs did not require as many system resources to configure.

The Debian server did not need a graphical environment, so it was installed in text-only mode. Since the Debian client will most likely be operated by an end user, it was installed with the lightweight XFCE desktop environment.

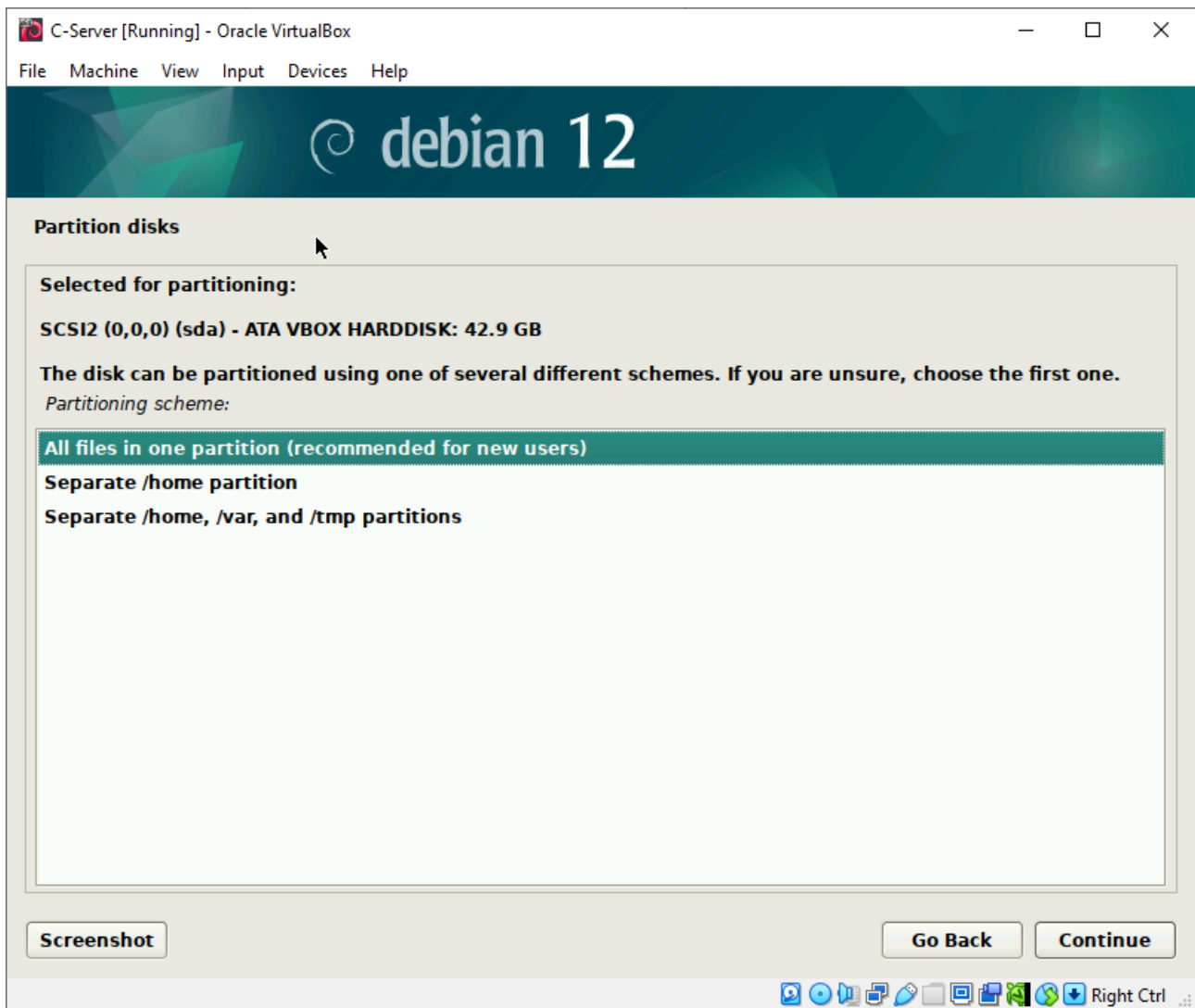


Figure 8: Installing the Debian server

Configure IP Addressing

The topology of the network is that the two servers are connected to each other. Additionally, the Debian client is connected to the Debian server while the Windows client is connected to the Windows server.

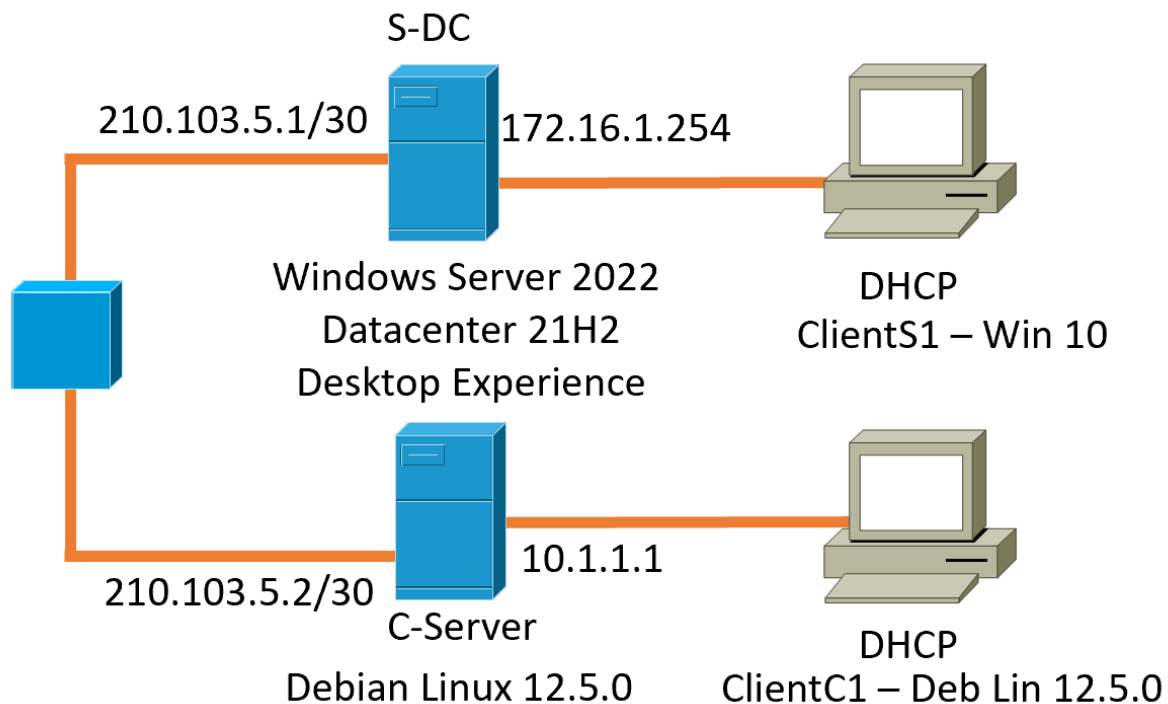


Figure 9: Network topology diagram [4]

An IP addressing table has been put together in order to quickly reference the IP addresses of every device.

Device	IP Address	Subnet Mask
S-DC Windows Server 2022	210.103.5.1/30 (External Network) 172.16.1.1/27 (Internal Swansea Network)	255.255.255.252 255.255.255.224
ClientS1 Windows 11 Enterprise	172.16.1.2/27	255.255.255.224
C-Server Debian 12 Server	210.103.5.1/30 (External Network) 10.1.1.1/27 (Internal Swansea Network)	255.255.255.252 255.255.255.224
ClientC1 Debian 12 Workstation	10.1.1.2/27	255.255.255.224

Figure 10: IP addressing table for network

The Windows Server was the first device to have its IP address configured. It was assigned the first address on both networks that it was connected to as it will likely be the primary administrative device on the network.

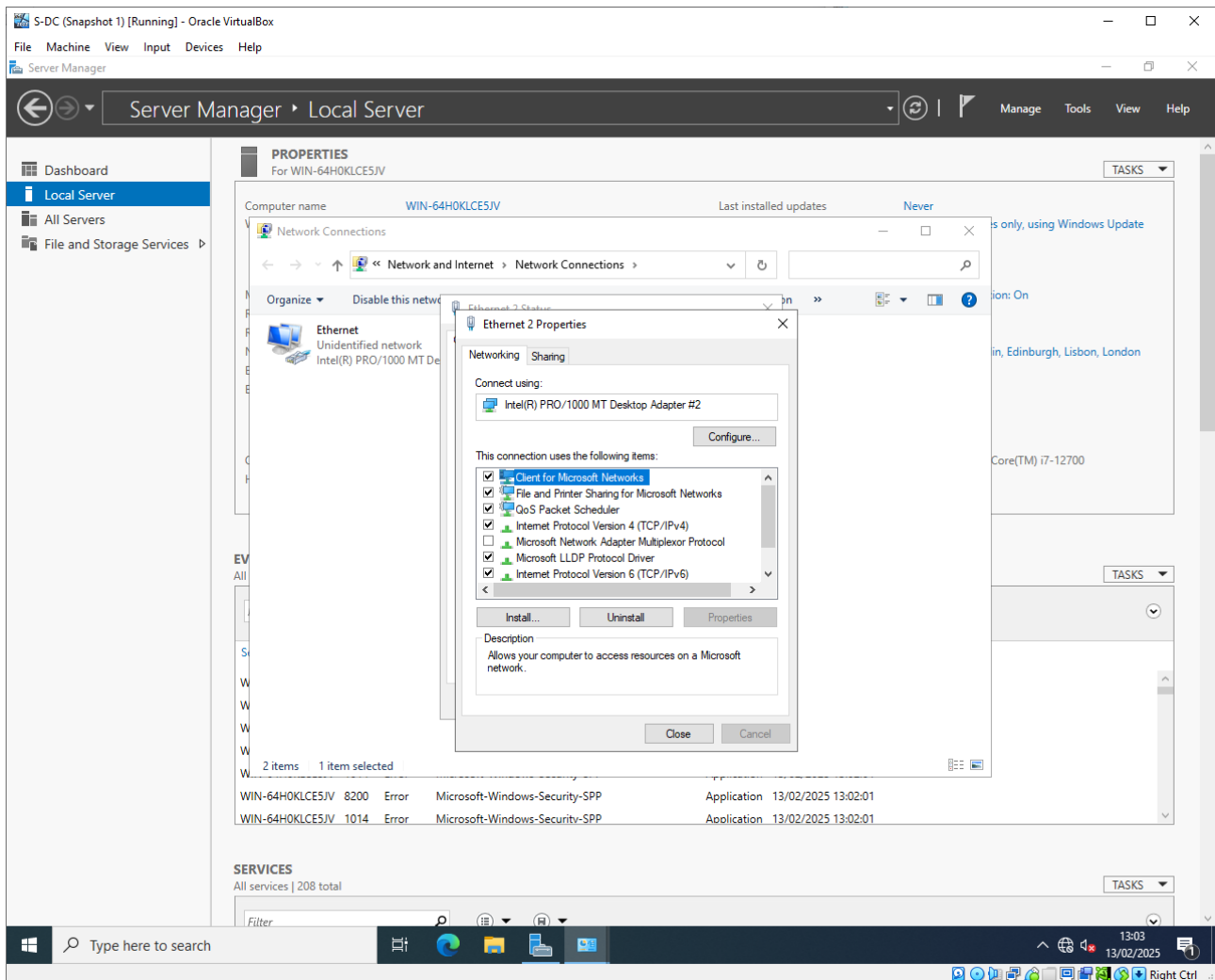
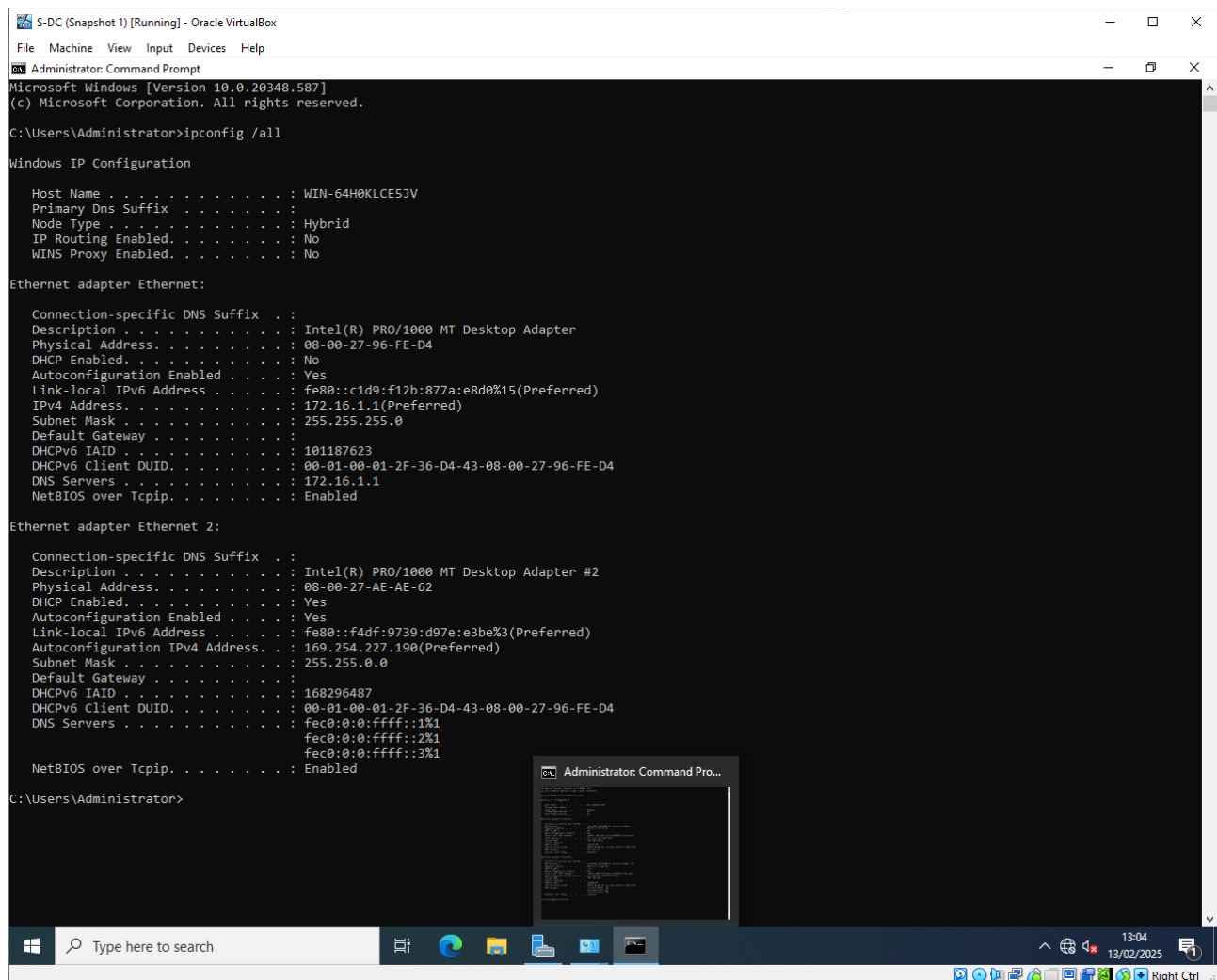


Figure 11: Windows Server IP configuration

To confirm that the IP address was configured correctly, the ipconfig command was used to get an overview of the system's network information.



```
S-DC (Snapshot 1) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Administrator: Command Prompt
Microsoft Windows [Version 10.0.20348.587]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ipconfig /all

Windows IP Configuration

Host Name . . . . . : WIN-64H0KLCESJV
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . . :
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-96-FE-D4
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::c1d9:f12b:877a:e8d0%15(Preferred)
IPv4 Address. . . . . : 172.16.1.1(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
DHCPv6 IAID . . . . . : 101187623
DHCPv6 Client DUID. . . . . : 00-01-00-01-2F-36-D4-43-08-00-27-96-FE-D4
DNS Servers . . . . . : 172.16.1.1
NetBIOS over Tcpip. . . . . : Enabled

Ethernet adapter Ethernet 2:

Connection-specific DNS Suffix . . :
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter #2
Physical Address. . . . . : 08-00-27-AE-AE-62
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::f4df:9739:d97e:e3be%3(Preferred)
Autoconfiguration IPv4 Address. . . : 169.254.227.190(Preferred)
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
DHCPv6 IAID . . . . . : 168296487
DHCPv6 Client DUID. . . . . : 00-01-00-01-2F-36-D4-43-08-00-27-96-FE-D4
DNS Servers . . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . : Enabled

C:\Users\Administrator>
```

Figure 12: Windows Server ipconfig

When the server was confirmed to have been configured correctly, the client was also configured using the exact same method as the server.

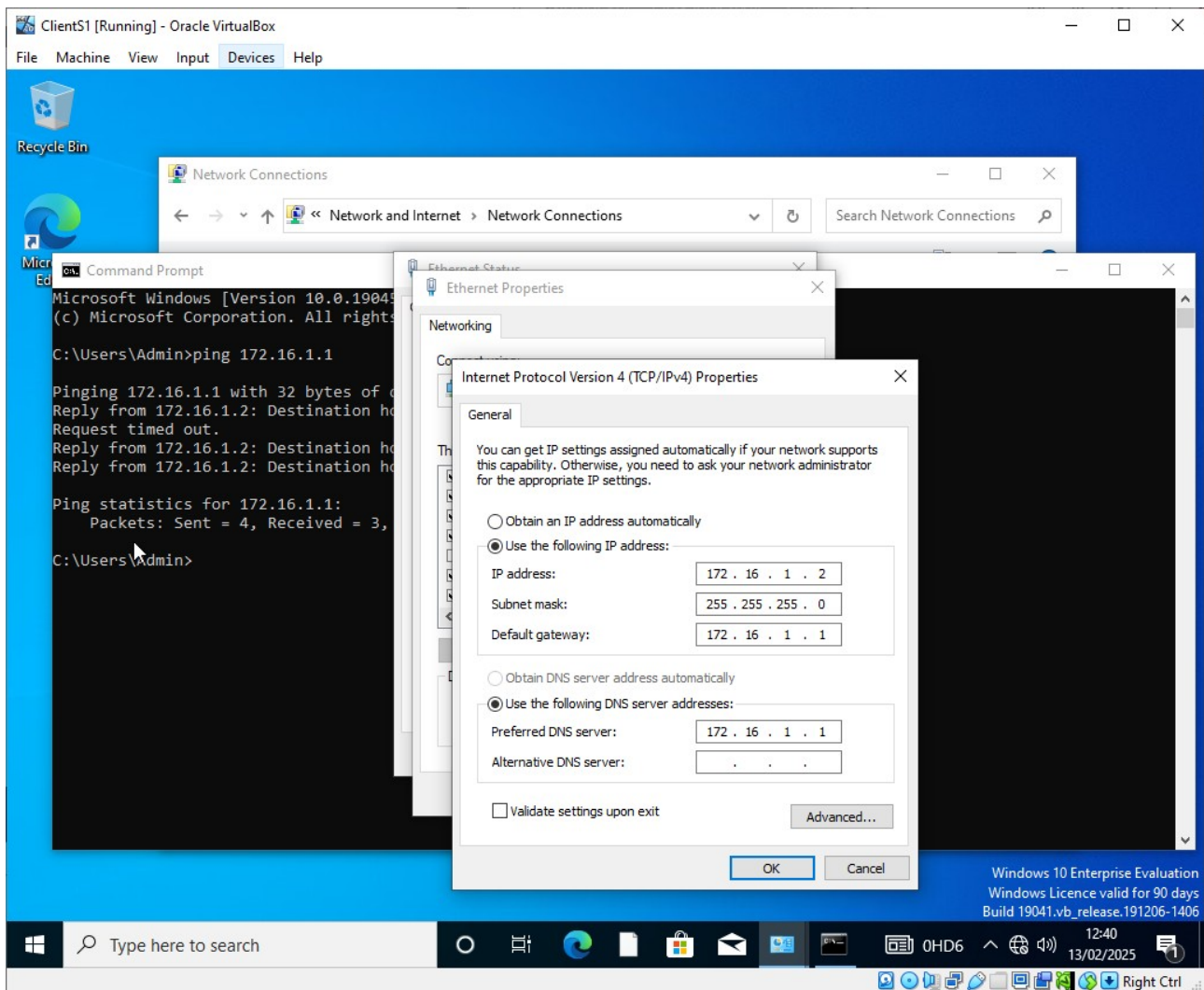
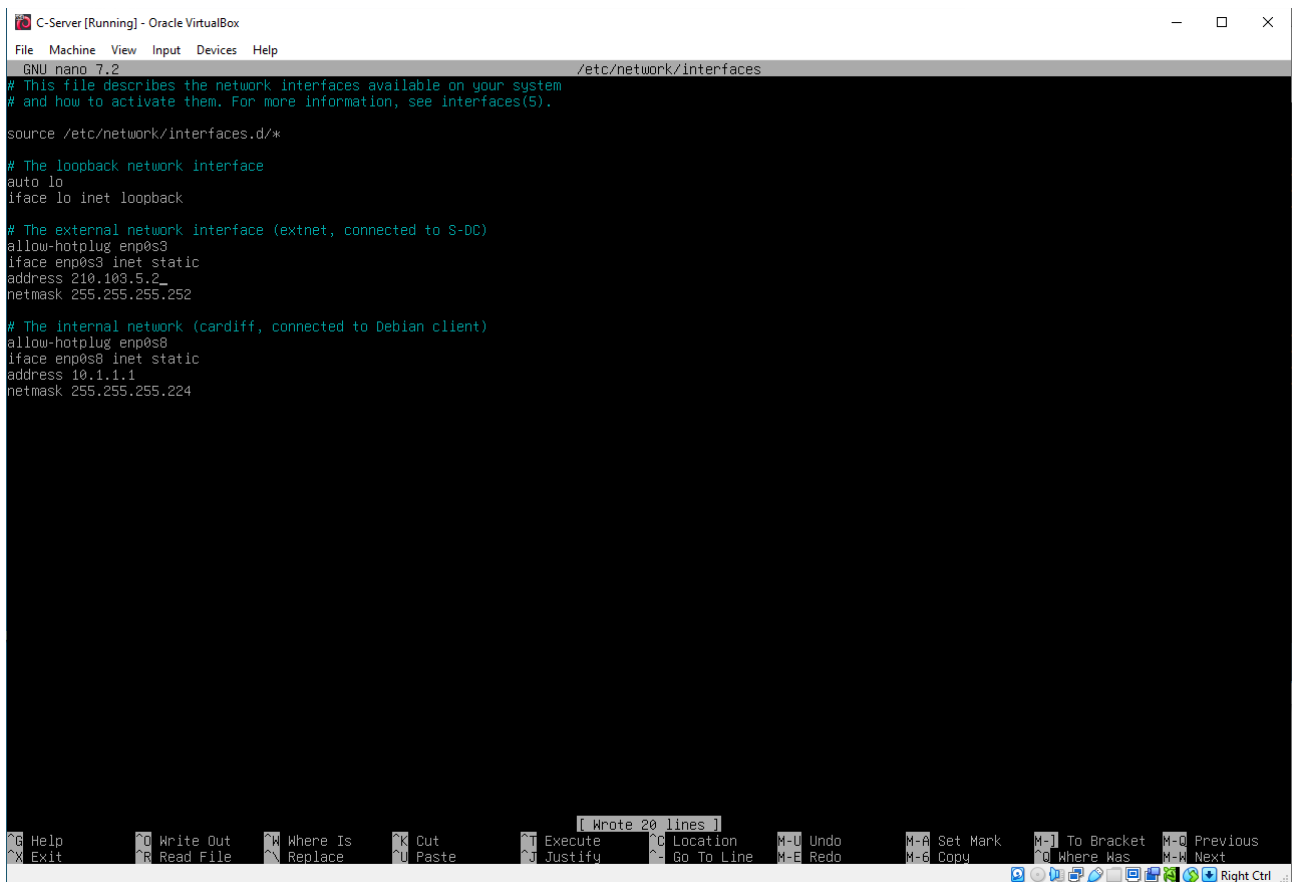


Figure 13: Windows client IP address configuration

Once the configurations were set on the Windows devices, the next step was to configure the Debian systems which use the 10.1.1.0/27 range. Unlike Windows, the Debian server does not have a graphical user interface so the IP address had to be configured from the command line by editing a configuration file.



```
C-Server [Running] - Oracle VirtualBox
File Machine View Input Devices Help
GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

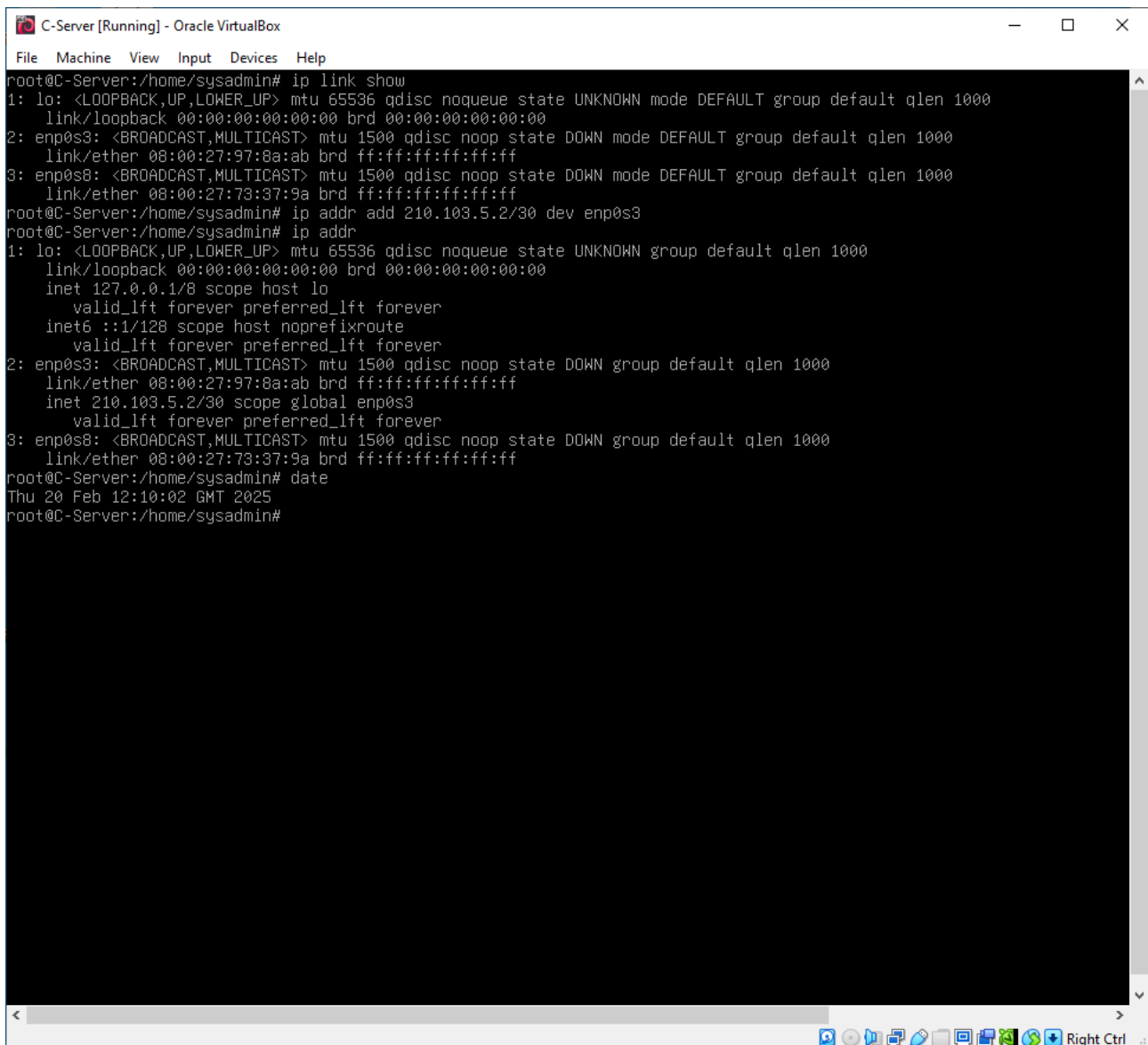
# The external network interface (extnet, connected to S-DC)
allow-hotplug enp0s3
iface enp0s3 inet static
address 210.103.5.2_
netmask 255.255.255.252

# The internal network (cardiff, connected to Debian client)
allow-hotplug enp0s8
iface enp0s8 inet static
address 10.1.1.1
netmask 255.255.255.224
```

The screenshot shows a terminal window titled "C-Server [Running] - Oracle VirtualBox". The window contains the nano text editor editing the file `/etc/network/interfaces`. The editor shows the configuration for three network interfaces: the loopback interface `lo`, the external network interface `enp0s3` (connected to S-DC), and the internal network interface `enp0s8` (connected to Debian client). The `enp0s8` interface is configured with a static IP address of `10.1.1.1` and a netmask of `255.255.255.224`. The bottom of the window shows the nano editor's command palette with various shortcuts like `Ctrl+G` for Help, `Ctrl+O` for Write Out, etc.

Figure 14: Configuring an IP address on a Debian server

Once the configuration was saved, the `ip` command was used to check whether or not the IP links and addresses had been correctly configured.



```
root@C-Server:/home/sysadmin# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 08:00:27:97:8a:ab brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/ether 08:00:27:73:37:9a brd ff:ff:ff:ff:ff:ff
root@C-Server:/home/sysadmin# ip addr add 210.103.5.2/30 dev enp0s3
root@C-Server:/home/sysadmin# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:97:8a:ab brd ff:ff:ff:ff:ff:ff
    inet 210.103.5.2/30 scope global enp0s3
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:73:37:9a brd ff:ff:ff:ff:ff:ff
root@C-Server:/home/sysadmin# date
Thu 20 Feb 12:10:02 GMT 2025
root@C-Server:/home/sysadmin#
```

Figure 15: All IP addresses on the Debian server

Because the Debian server had been configured before the client was installed, this meant that for the Debian client the IP address was able to be set during the installation process.

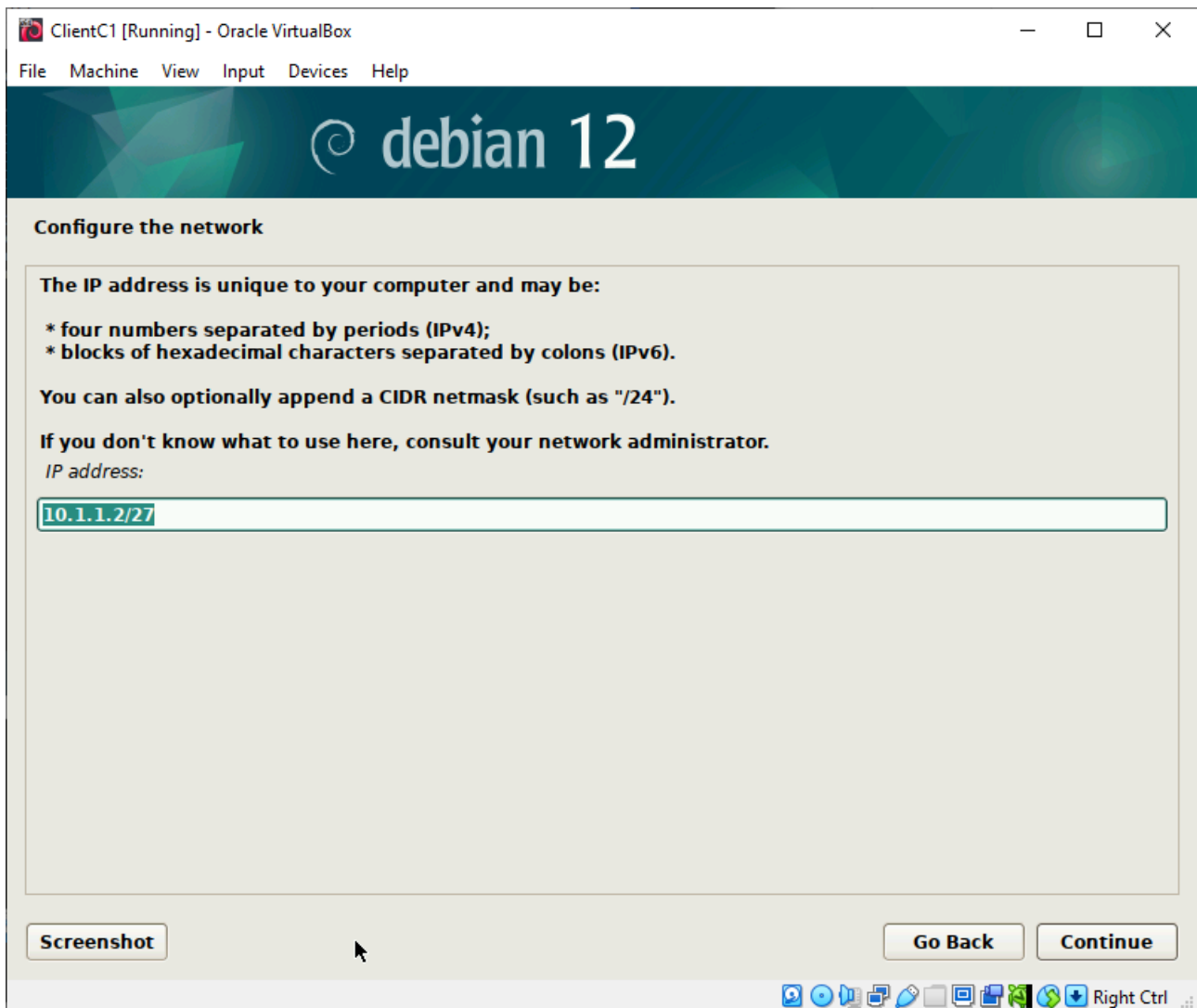


Figure 16: Setting the Debian Client's IP address while installing it

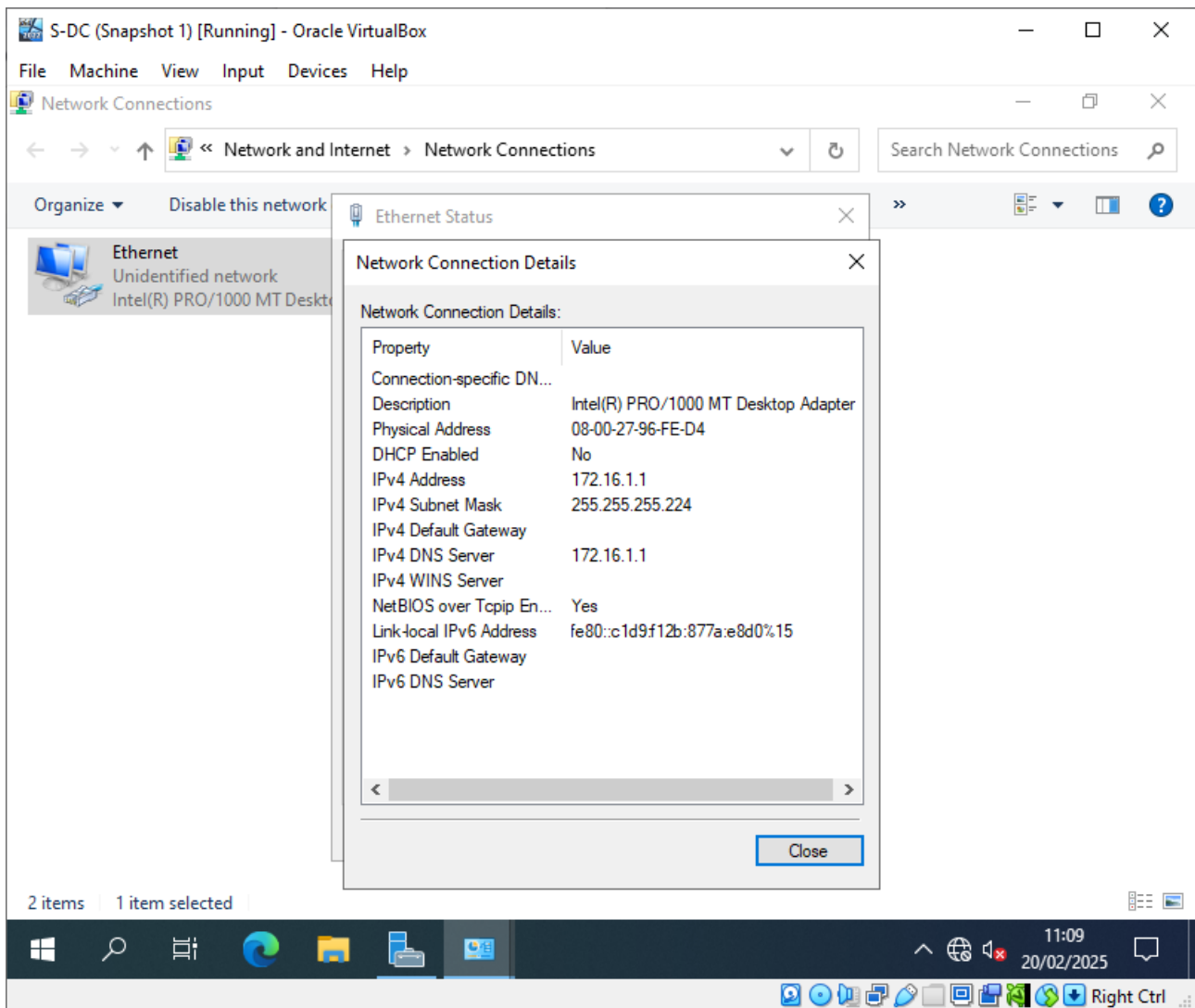


Figure 17: Windows Server screenshot showing VLSM

VLSM was later added to the networks to save on the limited number of address. For the LANs, the /27 subnet supporting 30 hosts was used to be efficient with IP addresses whilst also allowing for a decent number of devices on the network. /30 was used for the server connections as there are only two devices on that particular network.

Testing and Troubleshooting

Once every IP address had been configured, the next step was to ensure that each of the networks could communicate with each other.

For the connection tests, the ping command was used. This is a command available on most operating systems and sends packets to the specified IP address in order to confirm whether or not a connection is successful [28].

The first connection test was made between the two Windows systems which are on the 172.16.1.2 network, immediately after both devices had been configured. The connection test between both virtual machines was successful.

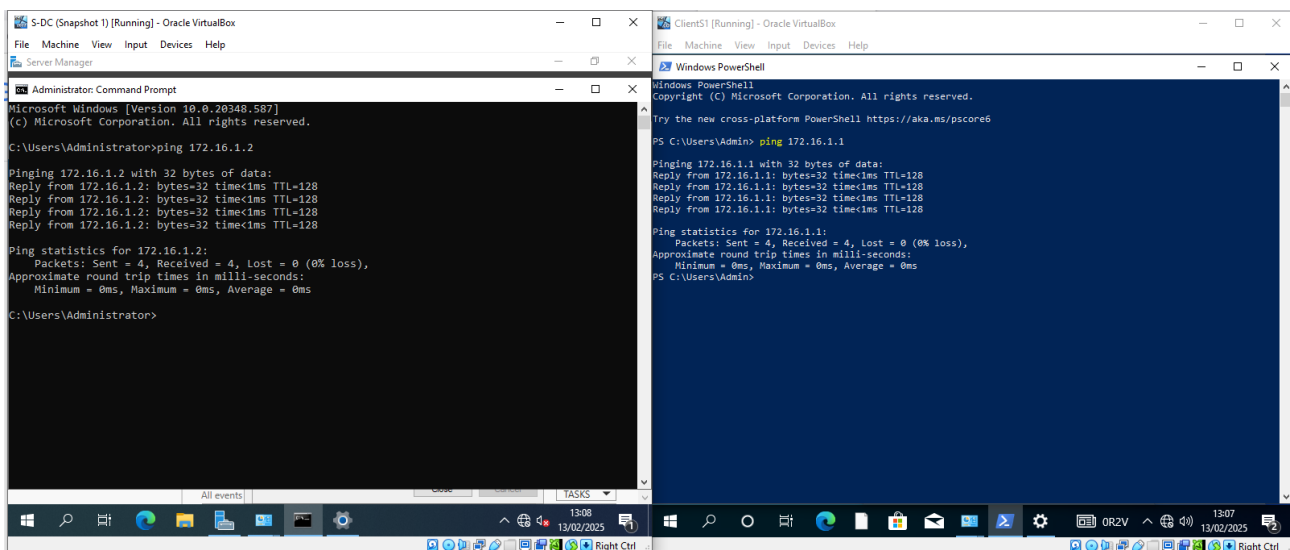


Figure 18: Successful ping between the Windows client and the Windows server

Afterwards, a connection test was performed between the Debian virtual machines was performed once they had also been configured. As with above the connection test between the two Debian systems was successful.

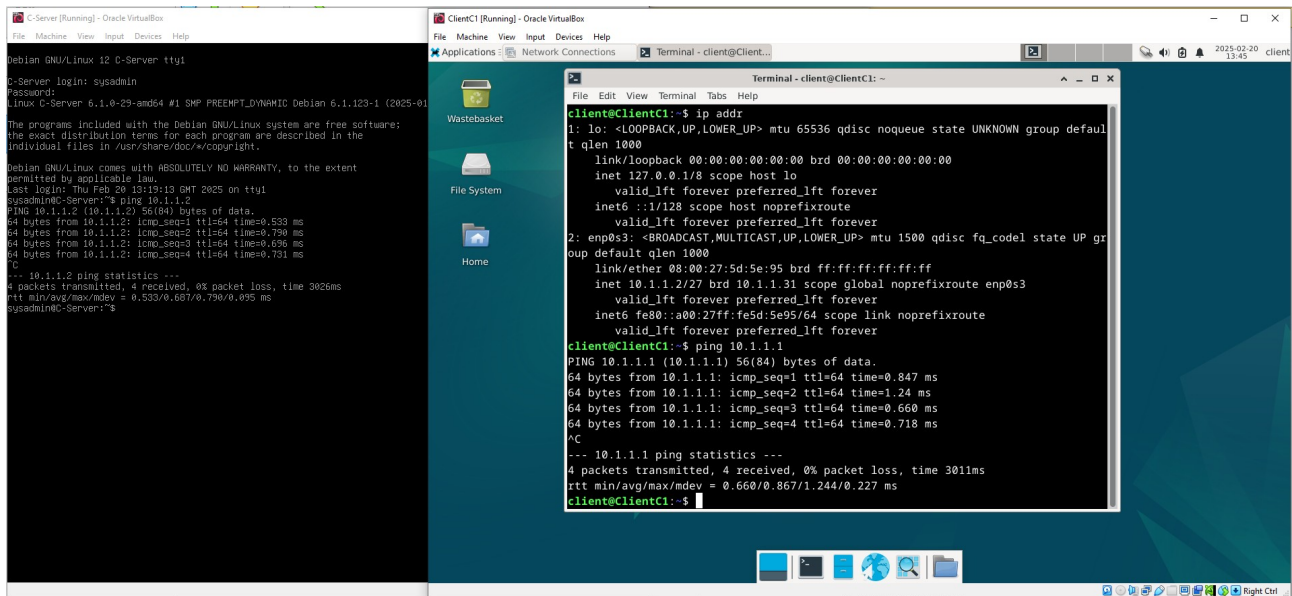


Figure 19: Successful ping between the Debian client and Debian server

It was, however, not possible to ping between the two client systems. This is because the Debian and Windows clients are on different networks (10.1.1.0 and 172.16.1.0) and, unlike the servers, are not directly connected. An alternative connection system, such as a Virtual Private Network, will be required in order for the two systems to communicate.

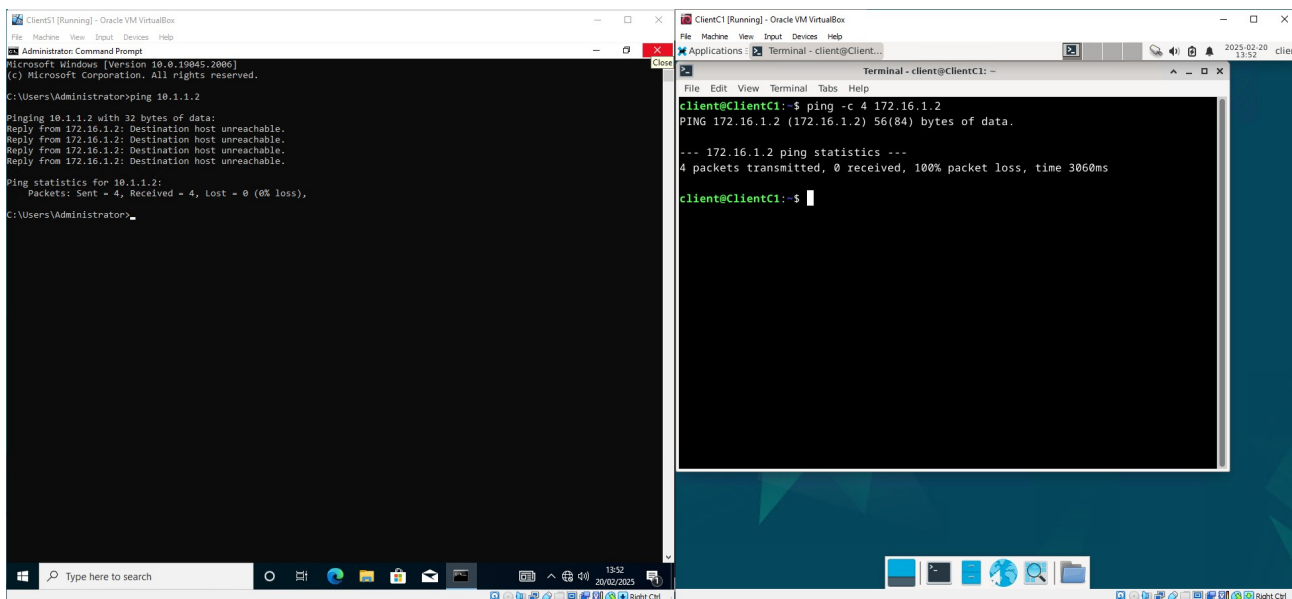


Figure 20: Unsuccessful ping between Debian and Windows clients

Task 3

Technical Reflection

One issue that occurred during the configuration process was that the Windows systems had connection issues due to the firewall. Because of this, the firewall had to be disabled in order to allow the two VMs to communicate. In the future, the firewall should be re-enabled and properly reconfigured.

The client systems could not communicate as they are on different networks. A VPN could be used to enable a connection between them.

Personal Reflection

Given more time, the virtual system could have been better configured with more advanced functionality. If the project was to be conducted again, more insight into technical details would have been considered.

Conclusion

In conclusion, a set of four virtual machines were set up to simulate a computer network. A variety of other topics, such as virtualisation, network services and the internet protocol.

Due to resource restrictions, certain elements of the assessment were considered problematic, especially when on-site access to the computer hosting the virtual machines was not available. Use of remote access software to reach the host computer may be worth considering for future attempts, although doing so would be somewhat inefficient due to hardware restrictions.

However, despite the difficulties and challenges the desired result as specified by the was successfully achieved.

References

- [1] Tsafirir Dan, Nieh Jason, and Bugnion Edouard, *Hardware and software support for virtualization*. in Synthesis Lectures on Computer Architecture, Number 38. San Rafael, California: Morgan & Claypool Publishers, 2017. doi: 10.2200/S00754ED1V01Y201701CAC038.
- [2] Rooney Tim and Dooley Michael Earl, *IP address management*, Second edition. in IEEE press series on networks and service management. Hoboken, New Jersey: Wiley, 2021. doi: 10.1002/9781119692263.
- [3] Graziani Rick, *IPv6 fundamentals: a straightforward approach to understanding IPv6*, Second edition. Indianapolis, IN: Cisco Press, 2017.
- [4] N. Kaliyamurthy, 'Network Operating Systems and Services assignment brief'. UWTSD, 2025.
- [5] 'What is a hypervisor?' Accessed: Feb. 28, 2025. [Online]. Available: <https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor>
- [6] 'What Are Hypervisors? | IBM'. Accessed: Feb. 28, 2025. [Online]. Available: <https://www.ibm.com/think/topics/hypervisors>
- [7] Y. Yang, B. Mao, H. Jiang, Y. Yang, H. Luo, and S. Wu, 'SnapMig: Accelerating VM Live Storage Migration by Leveraging the Existing VM Snapshots in the Cloud', *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 6, pp. 1416–1427, Jun. 2018, doi: 10.1109/TPDS.2018.2790389.
- [8] A. S. Thyagaturu, P. Shantharama, A. Nasrallah, and M. Reisslein, 'Operating Systems and Hypervisors for Network Functions: A Survey of Enabling Technologies and Research Studies', *IEEE Access*, vol. 10, pp. 79825–79873, 2022, doi: 10.1109/ACCESS.2022.3194913.
- [9] A. Bratterud and H. Haugerud, 'Maximizing hypervisor scalability using minimal virtual machines', *IEEE Int. Conf. Cloud Comput. Technol. Sci.*, 2013, Accessed: Mar. 06, 2025. [Online]. Available: <http://hdl.handle.net/10642/2009>
- [10] M. Portnoy, *Virtualization Essentials*. Newark, UNITED STATES: John Wiley & Sons, Incorporated, 2016. Accessed: Mar. 06, 2025. [Online]. Available: <http://ebookcentral.proquest.com/lib/trinity/detail.action?docID=4644086>
- [11] A. B. Butterfield, G. E. N. E. Ngondi, and A. K. Kerr, 'subnet mask', in *A Dictionary of Computer Science*, A. Butterfield, G. E. Ngondi, and A. Kerr, Eds., Oxford University Press, 2016. Accessed: Mar. 14, 2025. [Online]. Available: <https://www.oxfordreference.com/display/10.1093/acref/9780199688975.001.0001/acref-9780199688975-e-6478>
- [12] P. M. Vieira, N. R. Freitas, V. B. Lima, D. Costa, C. Rolanda, and C. S. Lima, 'Multi-pathology detection and lesion localization in WCE videos by using the instance segmentation approach', *Artif. Intell. Med.*, vol. 119, p. 102141, Sep. 2021, doi: 10.1016/j.artmed.2021.102141.
- [13] M. R. Sabir, M. S. Mian, K. Sattar, and M. A. Fahiem, 'IP Address Space Management using Aggregated Fixed Length Subnet Masking', in *2007 International Conference on Electrical Engineering*, Apr. 2007, pp. 1–4. doi: 10.1109/ICEE.2007.4287346.
- [14] C. Okoh, W. N. Theophilus, P. Dawkins, and S. Paheerathan, 'Enhancing Data Security Through VLSM Subnetting and TCP/IP Model in an ENT', *Appl. Sci.*, vol. 14, no. 23, pp. 10968–, 2024, doi: 10.3390/app142310968.
- [15] Cisco Networking Academy Program, *Routing and switching essentials v6. Companion guide*. Indianapolis, Indiana: Cisco Press, 2017.
- [16] M. Karyakarte, A. Agarkar, L. Kulkarni, M. Patil, G. Chavhan, and B. Sule, 'Dynamic opportunistic routing protocol for ad-hoc Internet of Vehicles (IoV)', *Computing*, vol. 106, no. 6, pp. 1707–1728, Jun. 2024, doi: 10.1007/s00607-023-01248-9.

- [17] Q. F. Yang, H. H. Shi, and Z. Y. Zhao, 'Simulation and Analysis of RIPv1 and RIPv2', *Appl. Mech. Mater.*, vol. 336–338, p. 2460, Jul. 2013, doi: 10.4028/www.scientific.net/AMM.336-338.2460.
- [18] 'Performance Evaluation of Wireless Routing Protocols: RIP vs OSPF | IEEE Conference Publication | IEEE Xplore'. Accessed: Mar. 17, 2025. [Online]. Available: <https://ieeexplore-ieee-org.ezproxy.uwtsd.ac.uk/document/9719366>
- [19] N. W. Abdul Majid and S. Fuada, 'RIP VS. OSPF ROUTING PROTOCOLS: WHICH ONE IS THE BEST FOR A REAL-TIME COMPUTER NETWORK?', *Simetris J. Tek. Mesin Elektro Dan Ilmu Komput.*, vol. 11, pp. 249–256, Apr. 2020, doi: 10.24176/simet.v11i1.3796.
- [20] T. M. Diansyah, D. Handoko, I. Faisal, A. Yunianti, K. Chiuloto, and R. Liza, 'Design Analysis of OSPF (Open Shortest Path First) Routing by Calculating Packet Loss Of Network WAN (Wide Area Network)', *J. Phys. Conf. Ser.*, vol. 1361, no. 1, pp. 12087-, 2019, doi: 10.1088/1742-6596/1361/1/012087.
- [21] S. Kent, C. Lynn, and K. Seo, 'Secure Border Gateway Protocol (S-BGP)', *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 582–592, Apr. 2000, doi: 10.1109/49.839934.
- [22] K. Weitz, D. Woos, E. Torlak, M. D. Ernst, A. Krishnamurthy, and Z. Tatlock, 'Scalable verification of border gateway protocol configurations with an SMT solver', *SIGPLAN Not.*, vol. 51, no. 10, pp. 765–780, 2016, doi: 10.1145/3022671.2984012.
- [23] L. Huawei Technologies Co., *Data Communications and Network Technologies*. Springer Nature, 2023. doi: 10.1007/978-981-19-3029-4.
- [24] M. Whitman and H. Mattord, *Principles of Information Security*, 6th edition. Australia Brazil Mexico Singapore United Kingdom United States: Cengage Learning, 2017.
- [25] Dauti Bekim, *Windows Server 2022 administration fundamentals: a beginner's guide to managing and administering Windows Server environments*, Third edition. Birmingham: Packt Publishing, 2022.
- [26] S. Perrott, *Windows Server 2022 & Powershell All-in-One For Dummies*. For Dummies, 2022.
- [27] 'Chapter 4. Guest Additions'. Accessed: Apr. 18, 2025. [Online]. Available: <https://www.virtualbox.org/manual/ch04.html>
- [28] robinharwood, 'ping'. Accessed: Apr. 18, 2025. [Online]. Available: <https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/ping>