

Student Name: Leon Buckney

Student Email: 2404085@Student.UWTSD.ac.uk

Student Number: 2404085

Module: Network Operating Systems and Services (ACCA5023)

Introduction

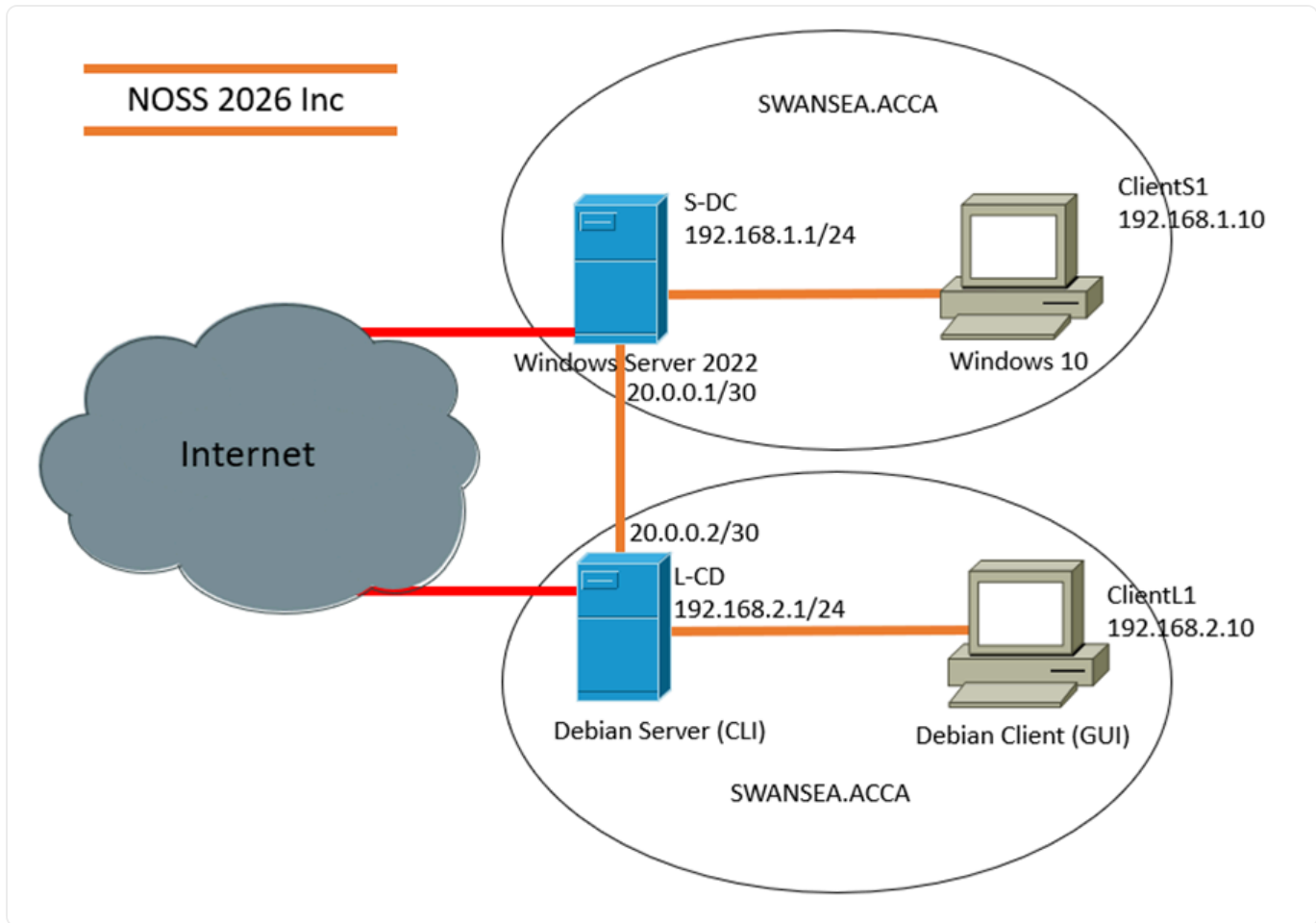
For enterprise networks to increase scalability, security, and resource efficiency, they must leverage crucial components such as virtualisation and network services. Organisations are now heavily deploying virtualized infrastructure to host multiple operating systems and services on a single physical machine. By doing this, companies will be able to reduce hardware costs whilst improving system management and flexibility. This report will cover a scenario in which an organization operates across two locations and requires a virtualized network environment capable of supporting Windows and Debian servers within a domain environment. This report will cover major networking concepts for designing and deploying this infrastructure, including hypervisors, IP addressing, routing, and network services. The research conducted evaluates technologies and approaches, but they must be compatible with the given topology and enterprise requirements. After these concepts have been discussed, the report will then move on to the deployment of the network within a virtualised environment.

Task 1

HyperVisors

In modern-day enterprise IT infrastructures, virtualisation plays a key role by allowing multiple operating systems to run simultaneously on a single physical machine. This is only made possible through a hypervisor, which essentially allocates hardware such as CPUs, memory, and storage to multiple virtual machines. There are two main architectures of hypervisors, these being type 1 (bare metal) and type 2 (hosted). A Type 1 Hypervisor, often referred to as a “bare-metal” hypervisor, operates directly on physical hardware without relying on a host operating system. By directly interacting with the hardware layer, it removes unnecessary overhead and delivers high performance and efficiency. This architecture makes Type 1 Hypervisor ideal for resource-intensive environments such as enterprise data centers and cloud infrastructure.**[1]**

In contrast, a Type 2 Hypervisor, also known as a “hosted” hypervisor, operates on top of an existing host operating system. This type of setup allows the hypervisor to run as an application within the host system, which means that, as mentioned previously, hardware resources must pass through the host system before reaching the virtual machines. While this makes Type 2 Hypervisor easier to install and manage, it can introduce additional performance overhead compared to bare-metal hypervisors. Typically, Type 2 Hypervisors are best used in smaller-scale environments, such as personal workstation, development systems, and educational laboratories.**[2]**



Considering the topology given in the brief, this virtualized environment must be able to support both Windows Server 2022 and Debian Server whilst also operating within the SWANSEA.ACCA domain environment. While Type 1 Hypervisors are commonly used in large enterprise environments due to their performance and scalability, a Type 2 Hypervisor would be the better choice for this implementation. Hosted Hypervisors, such as Oracle VirtualBox or VMware Workstation, allow virtual machines to run on an existing operating system while supporting multiple guest operating systems. This makes them great for building and testing network environments, as they provide flexibility, ease of deployment, and more, all of which are required for this topology.[3]

IP addressing

For communication across a network or the internet to work, data must be transmitted using the Internet Protocol (IP). When data is ready to be sent across a network, it must be broken into smaller units called packets[4]. Without packets, communication between networks would become incredibly slow, inefficient, and brittle. Each packet contains a source IP address and a destination IP address, which can identify the sending and receiving devices[4]. By having these IP addresses in place, the network can predetermine the correct route for the data to follow. One key component for sending data across networks is the router, which analyzes the destination IP address in the packet header to select the quickest, most appropriate path[5]. However, switches operate within local area networks (LANs) and use MAC addresses to direct data to the correct device connected to the network. With all of this said, IP addressing and Network devices enable reliable communication between systems across complex interconnected networks.

Referring back to IP addresses, they are structured according to two main standards, IPv4 and IPv6, both of which provide a unique identifier for devices connected to a network. First, IPv4 uses a 32-bit numerical format and is most often written in dotted-decimal notation, e.g., “192.168.1.1”[6]. This address is broken into network and host portions, allowing the router to determine which network and which specific device within that network it belongs to. But as the number of devices connected to the internet grows rapidly, IPv4 addresses are becoming scarce. To address the declining number of IPv4 addresses, IPv6 was introduced and provides a much larger address space due to its 128-bit address format[6], written in hexadecimal, e.g., “2001:0db8:85a3::8a2e:0370:7334”. By using IPv6, the number of possible addresses has expanded, improving routing efficiency and network communication in modern internet infrastructure.

Another key concept in IP addressing that cannot be forgotten is subnetting. It divides large networks into smaller subnetworks, thereby improving network organisation and efficiency[4]. It also allows administrators to allocate a group of IP addresses [4] to a specific network segment, which makes it easier to manage large networks and control traffic flow. Also, by breaking the network into smaller segments, subnetting can reduce broadcast traffic and improve general network performance. Another great use of subnetting is that it can efficiently use available IP addresses by distributing address ranges according to the needs of different network departments or devices. This can really help organisations manage and optimise their network design, avoiding wasted space, especially in large network infrastructures with many connected devices.

Routing

In order for data to be able to travel between networks, one key process must be in place for it to happen, that being routing. As mentioned previously, data is broken down into packets, and each packet includes a destination IP address. Routers examine this address and determine the best route to reach its destination[8]. To efficiently deliver data, routers use a database called a routing table, which stores information about network paths, including destination networks, subnet masks, and next-hop addresses[9], to forward packets to their destinations. Routing protocols that are set in place allow routers to exchange information and maintain accurate routing tables. Ensuring efficient, reliable communication between devices on different networks.

There are two main methods for implementing routing, the first is static routing, which involves manually configuring routes on routers[10], meaning that administrators have full control and can define which path a packet must follow. This method provides more control over network traffic and greatly improves security, but whenever a change occurs, it must be manually updated. Moving on to the second routing method, dynamic routing (RIP or OSPF), it automatically exchanges routing information between routers and updates the routing table when network conditions change[10]. Whilst dynamic routing is more scalable, especially in large networks, it can also introduce overhead and configuration complexity.

Given that this topology has a small, predictable network structure, it is best to use static routing[10], as it is a simple and reliable way to maintain network connectivity between segments, such as communication between the Windows and Debian servers within the network.

Network services

Large enterprise networks need secure communication, efficient device management, and reliable connectivity, and to achieve this, they use a range of network services. One common service is a Virtual Private Network (VPN), which allows a user to securely access a private network over the internet[11]. VPNs encrypt network traffic and create a tunnel to establish a secure connection between the user and the organization's network[11], protecting sensitive information from unauthorized access. This is extremely important for large companies that support remote access or operate across multiple locations.

Another great service that can automatically assign IP addresses and other network configuration details to devices on a network is the Dynamic Host Configuration Protocol (DHCP)[12]. With its automated IP allocation process, DHCP can reduce

the risk of IP address conflicts and the need for manual configuration. Many enterprises don't just use DHCP, they also use other great services, such as Active Directory Domain Services (AD DS), which can manage users, devices, and permissions within a single network domain[13]. It also provides centralized authentication and access control, enabling administrators to manage network resources more efficiently.

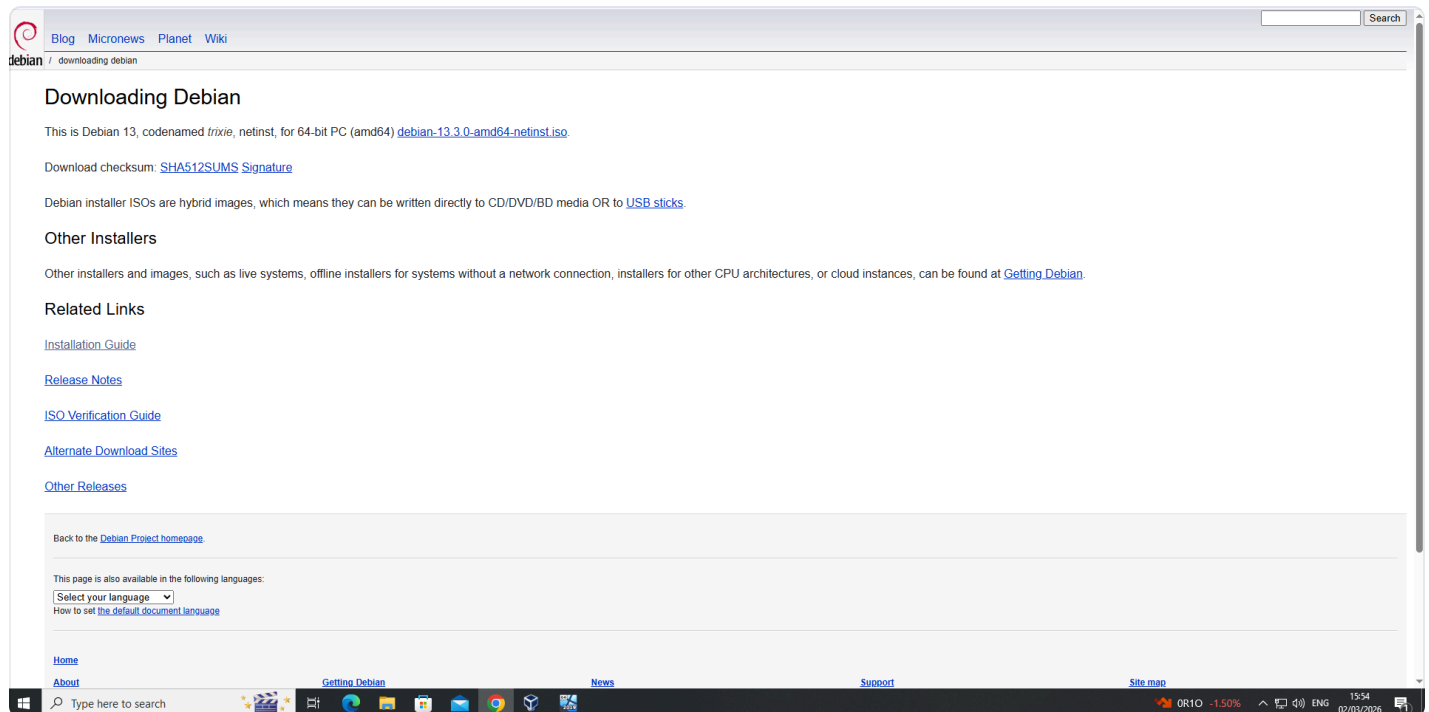
If organisations want to strengthen their network security even more, they can implement Certificate Authorities (CA) as part of a public key infrastructure, they issue digital certificates that verify users, devices, or services, enabling encrypted communication and secure authentication[14]. When combined, these network services play a critical role in maintaining well-managed, scalable, and secure network environments.

Task 2

Hypervisor choice

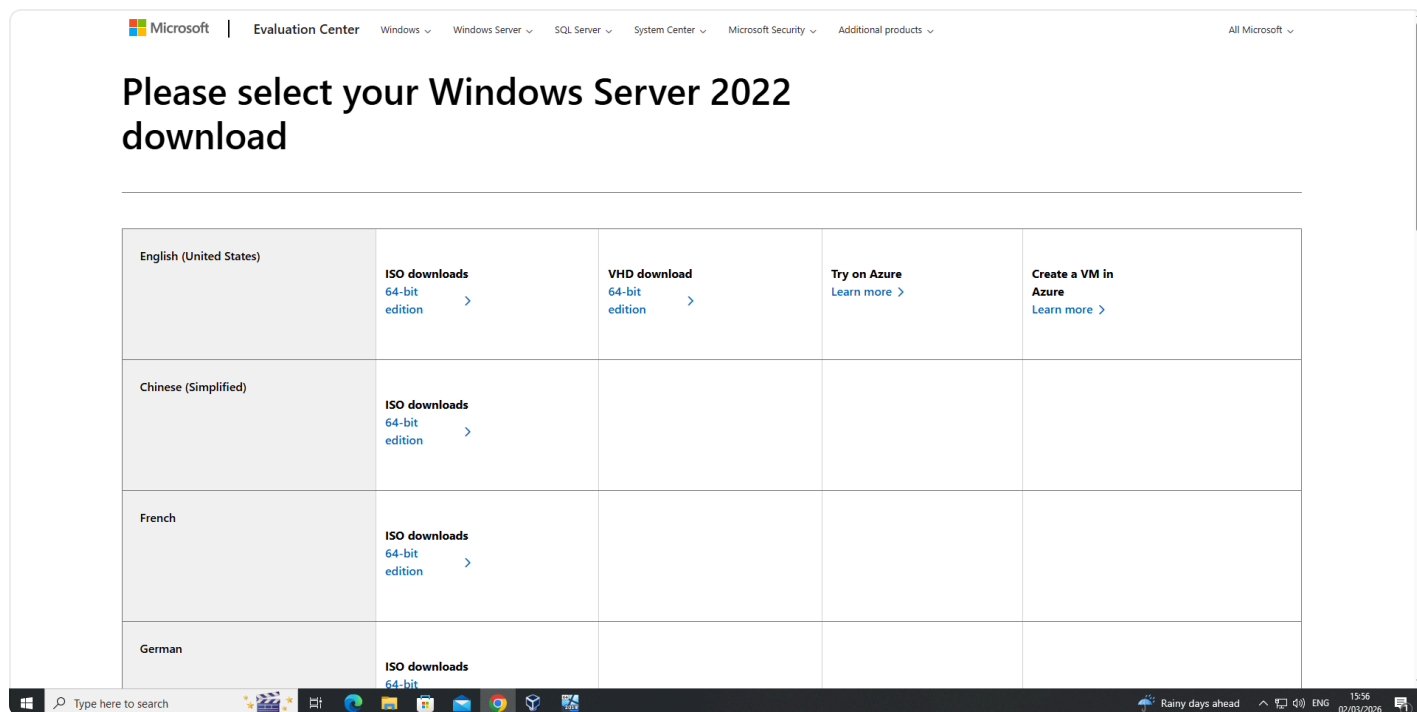
To create these virtual environments, a hypervisor had to be chosen before starting. Earlier in the report, it covers two types of hypervisors, their pros and cons, and, after careful consideration, Oracle VM VirtualBox, a type 2 hypervisor, was chosen. As mentioned previously, this hypervisor allows you to run multiple virtual machines, or even an entire network environment, without additional devices. Using a hypervisor reduces hardware costs, saves space, and allows systems to be easily configured, modified, and reset if something goes wrong. In this project, the chosen hypervisor enabled multiple operating systems, including Windows Server, Windows Client, and Debian systems, to run simultaneously and communicate via virtual networks. This made it possible to test IP addressing as well as routing and network connectivity within a controlled and safe environment.

Debian Download website



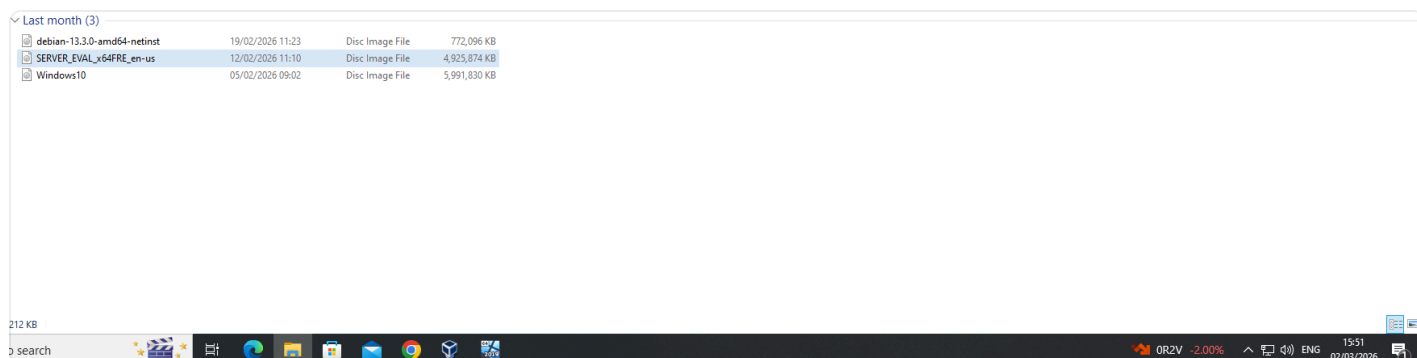
This screenshot shows the official Debian website, where the Debian operating system ISO file was downloaded.

Windows Server Download Website



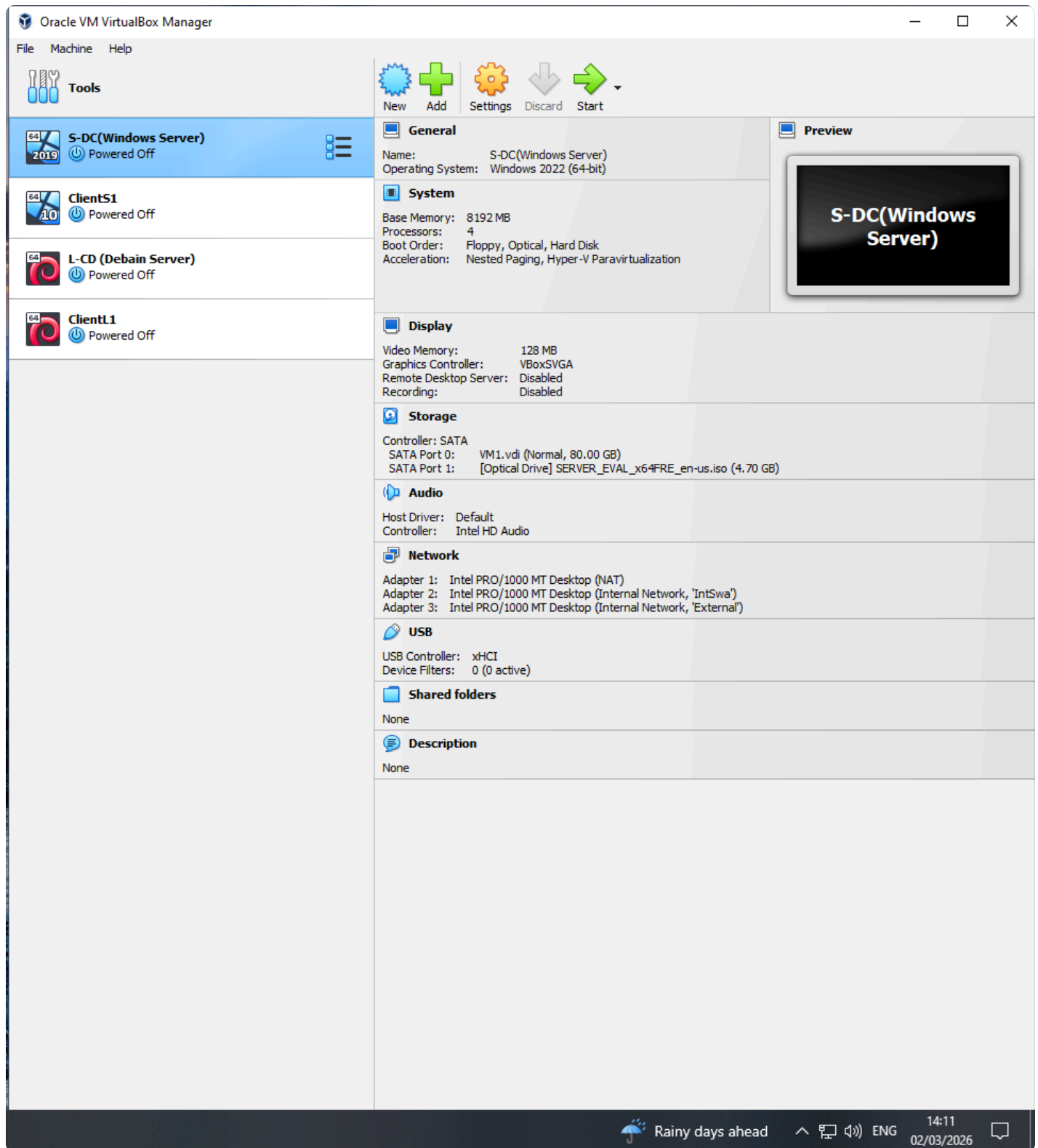
This screenshot shows the official Windows Server webpage, where the Windows operating system ISO file was downloaded.

ISO files in Downloads



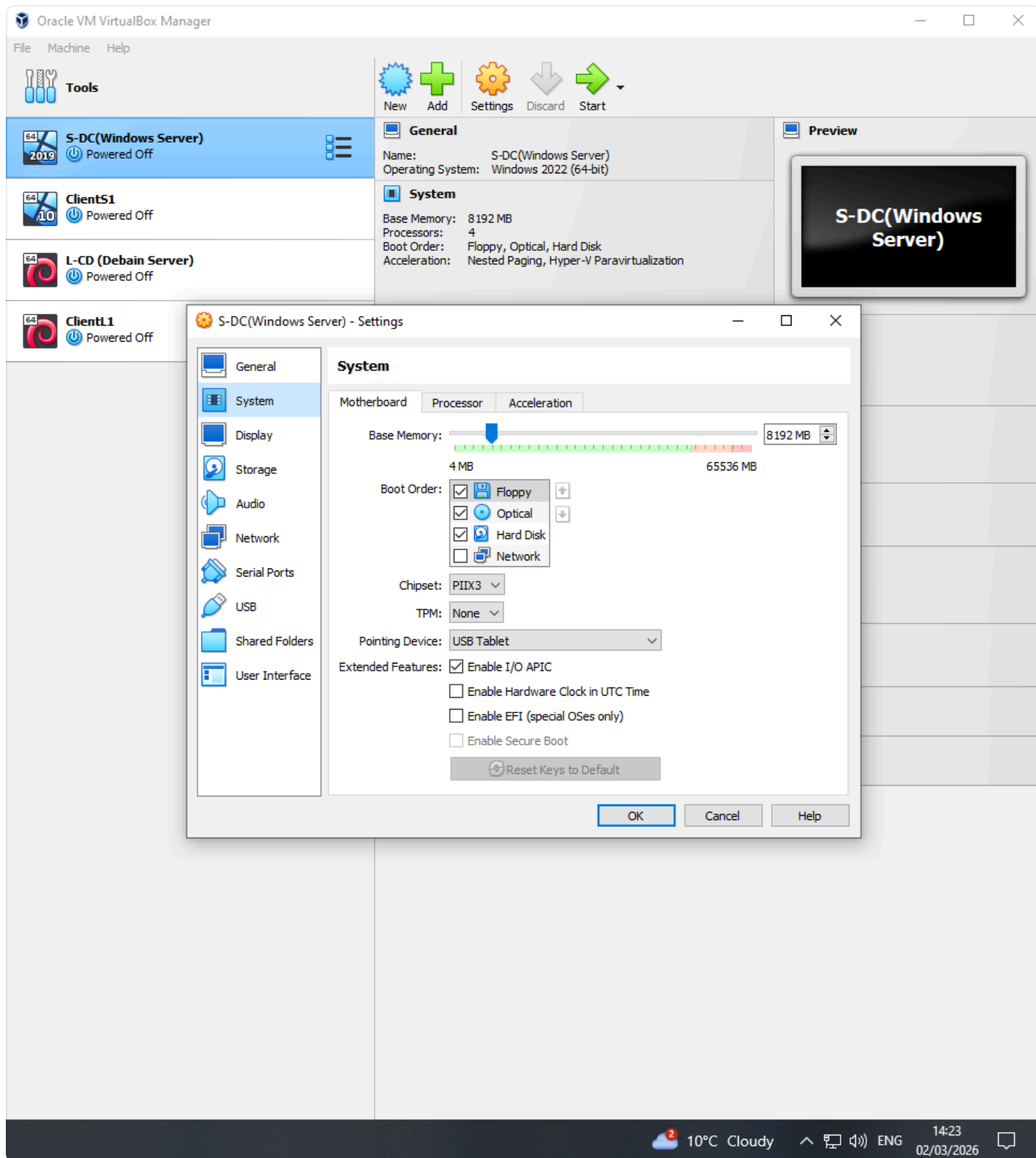
This screenshot shows the ISO files downloaded onto the host machine.

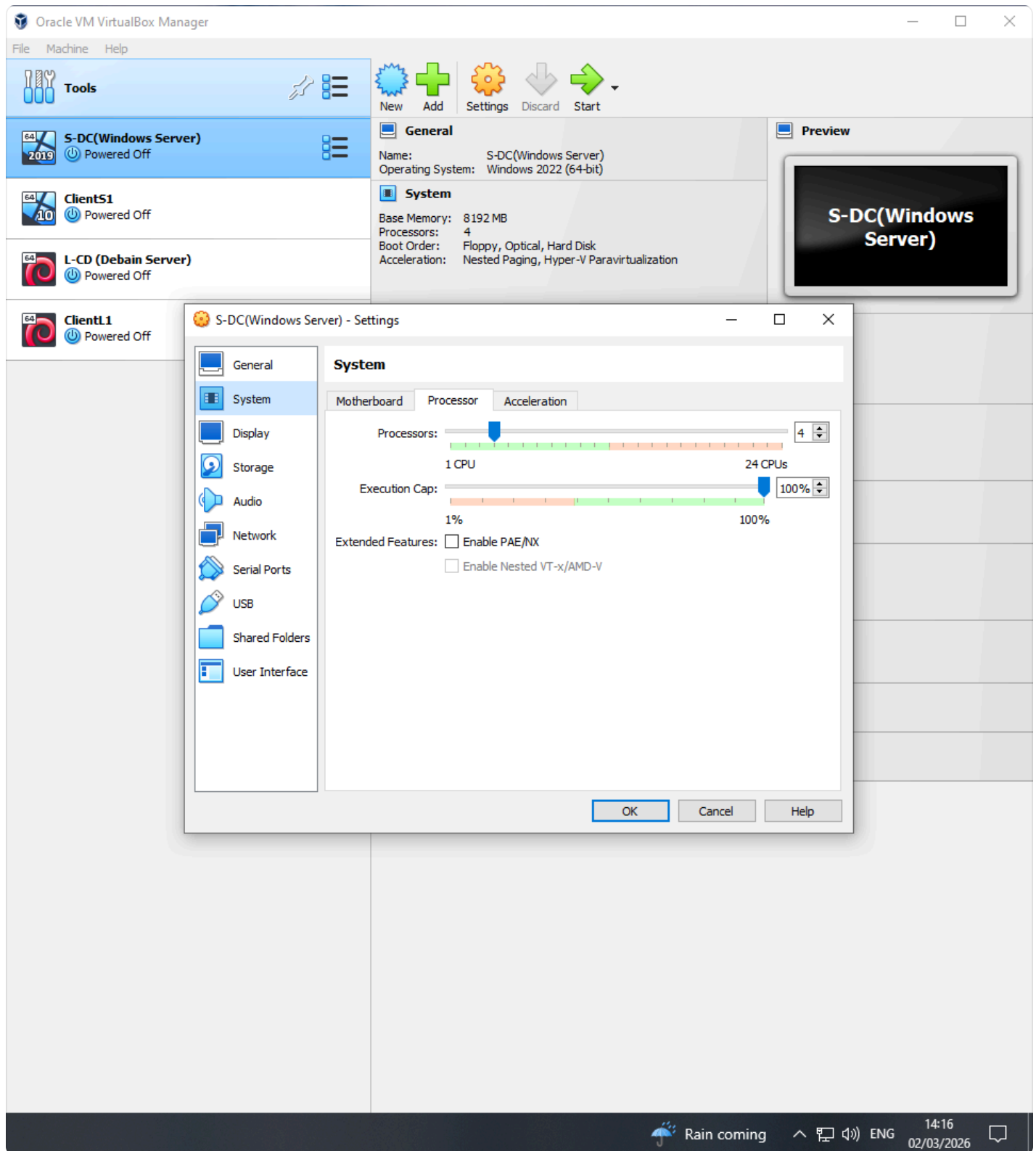
VirtualBox Manager



This screenshot shows the VirtualBox Manager interface displaying all four virtual machines made for this task.

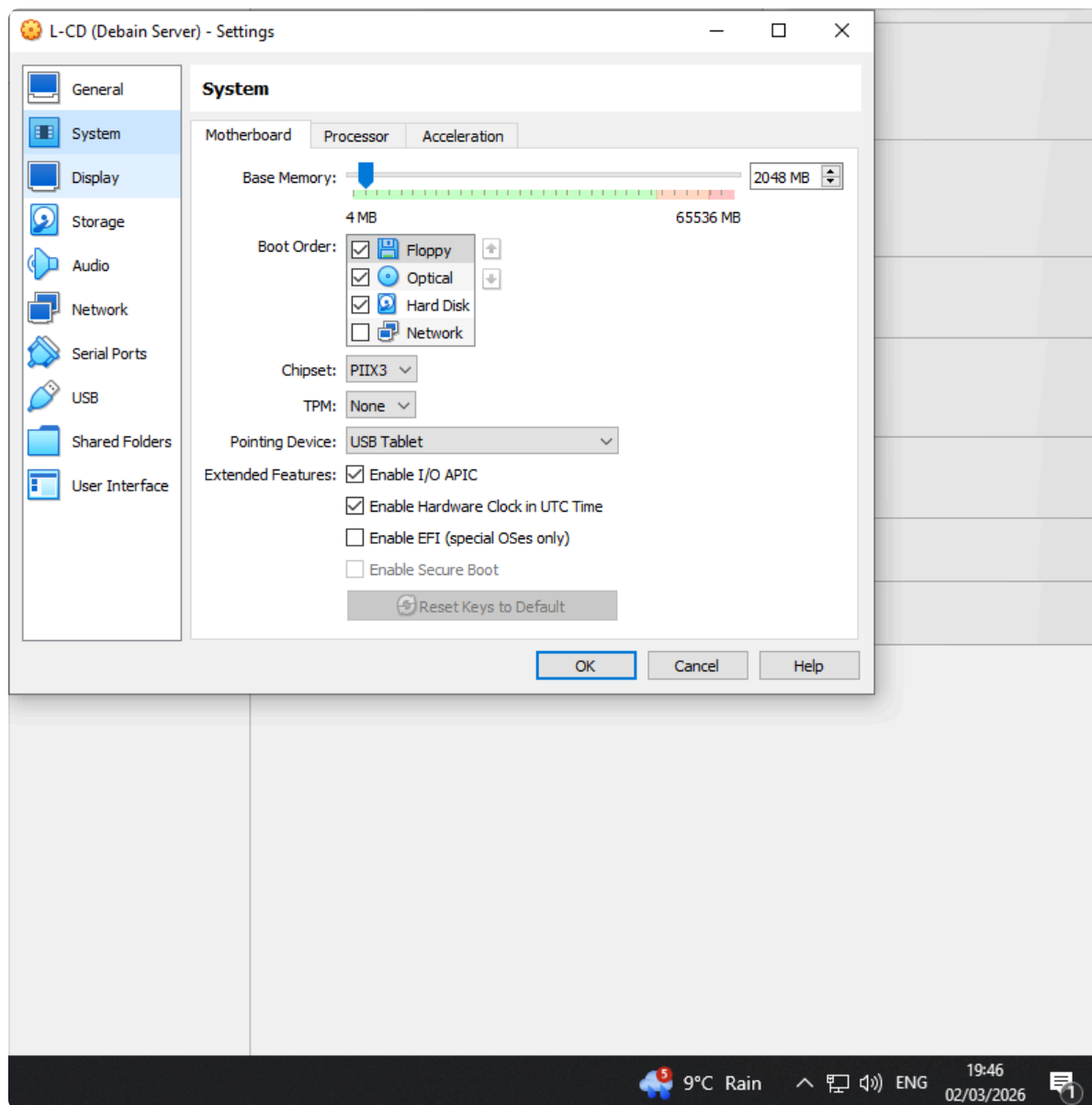
VM Windows System settings

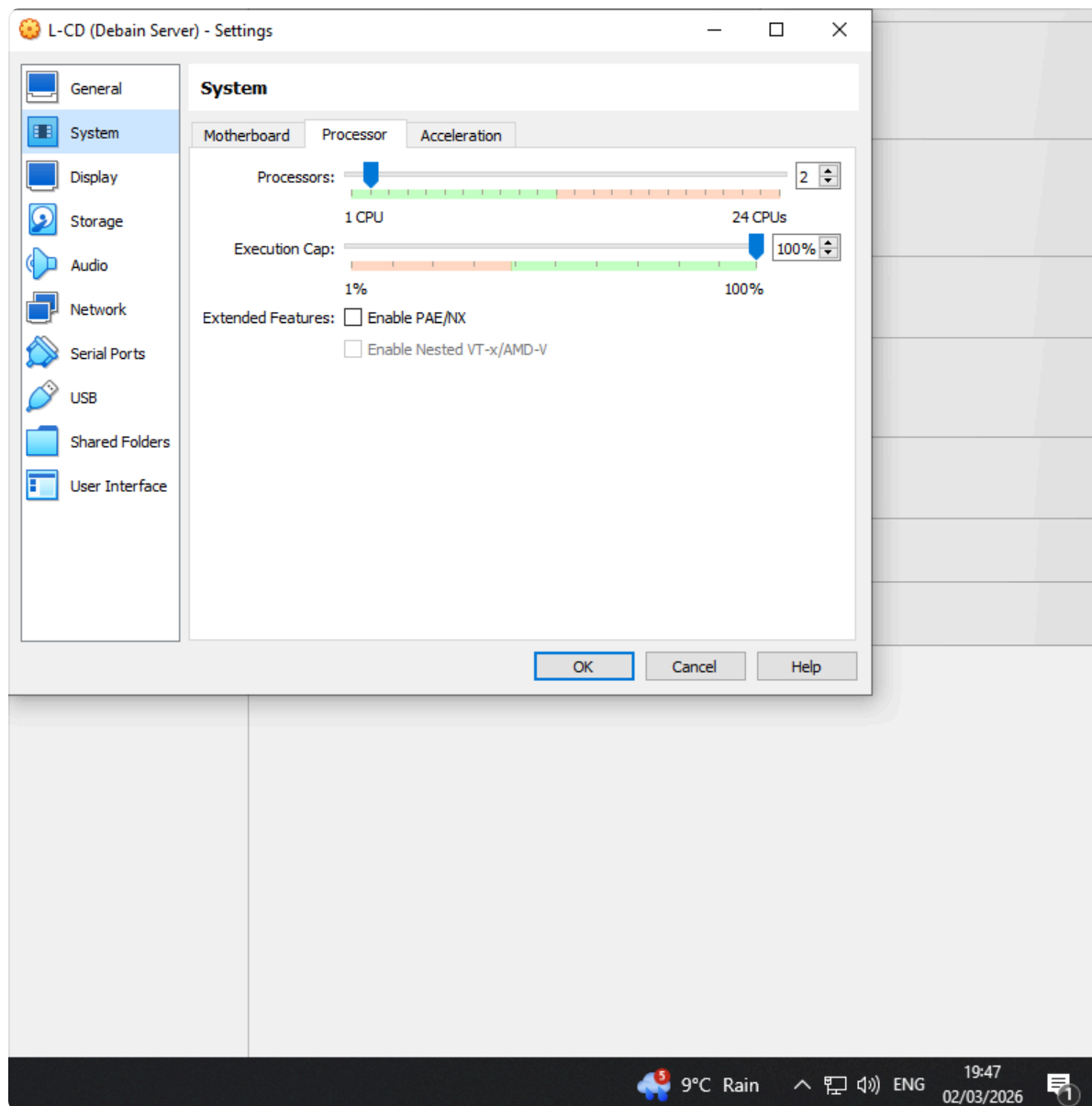




This screenshot shows system settings on the Windows Server Machine, including the amount of RAM allocated, the boot order, and the number of processors.

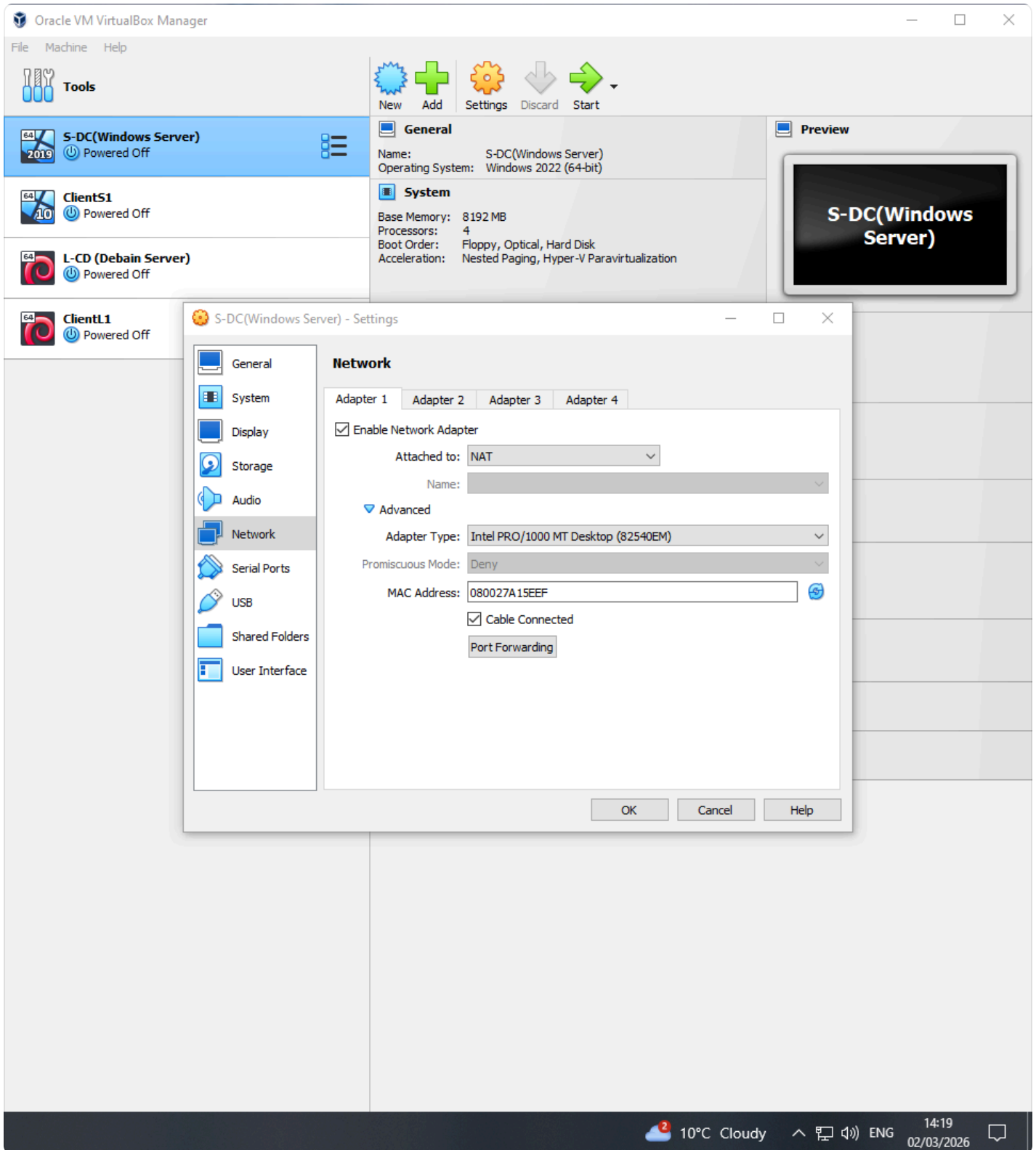
VM Debian System settings

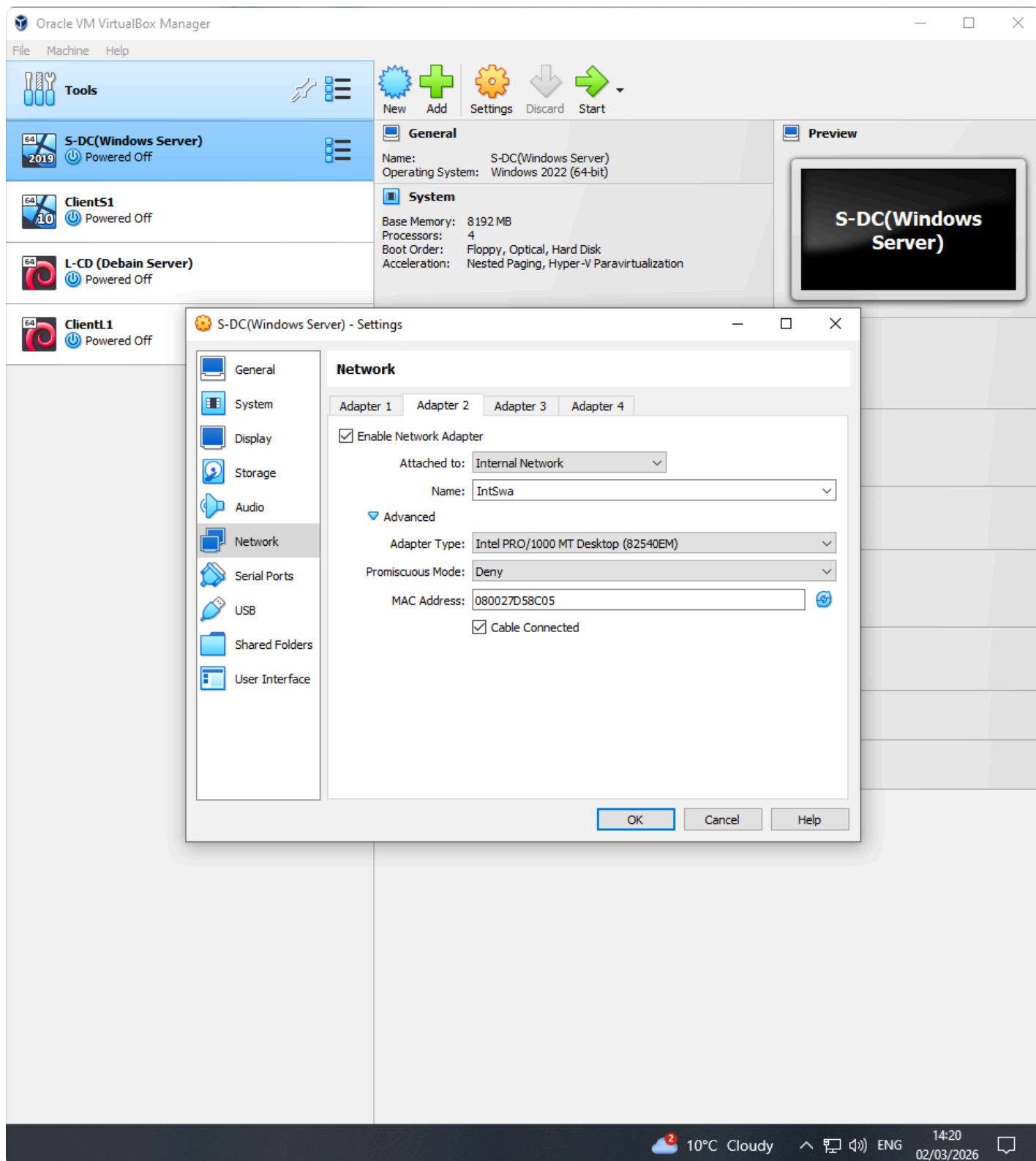


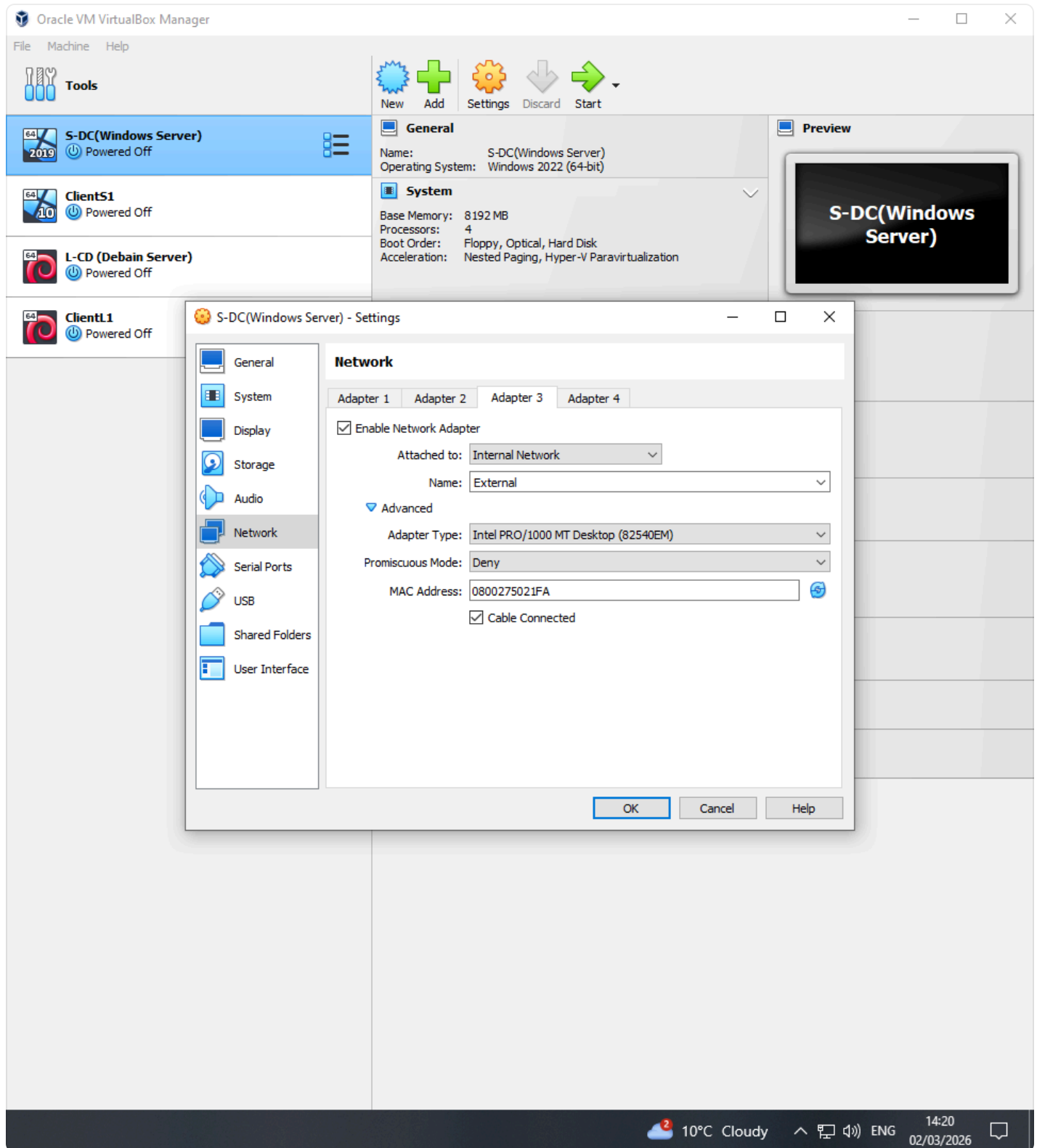


This screenshot shows the system settings on the Debian Server machine, including the amount of RAM allocated, the boot order, and the number of processors.

Windows Server Network Adapter Configuration

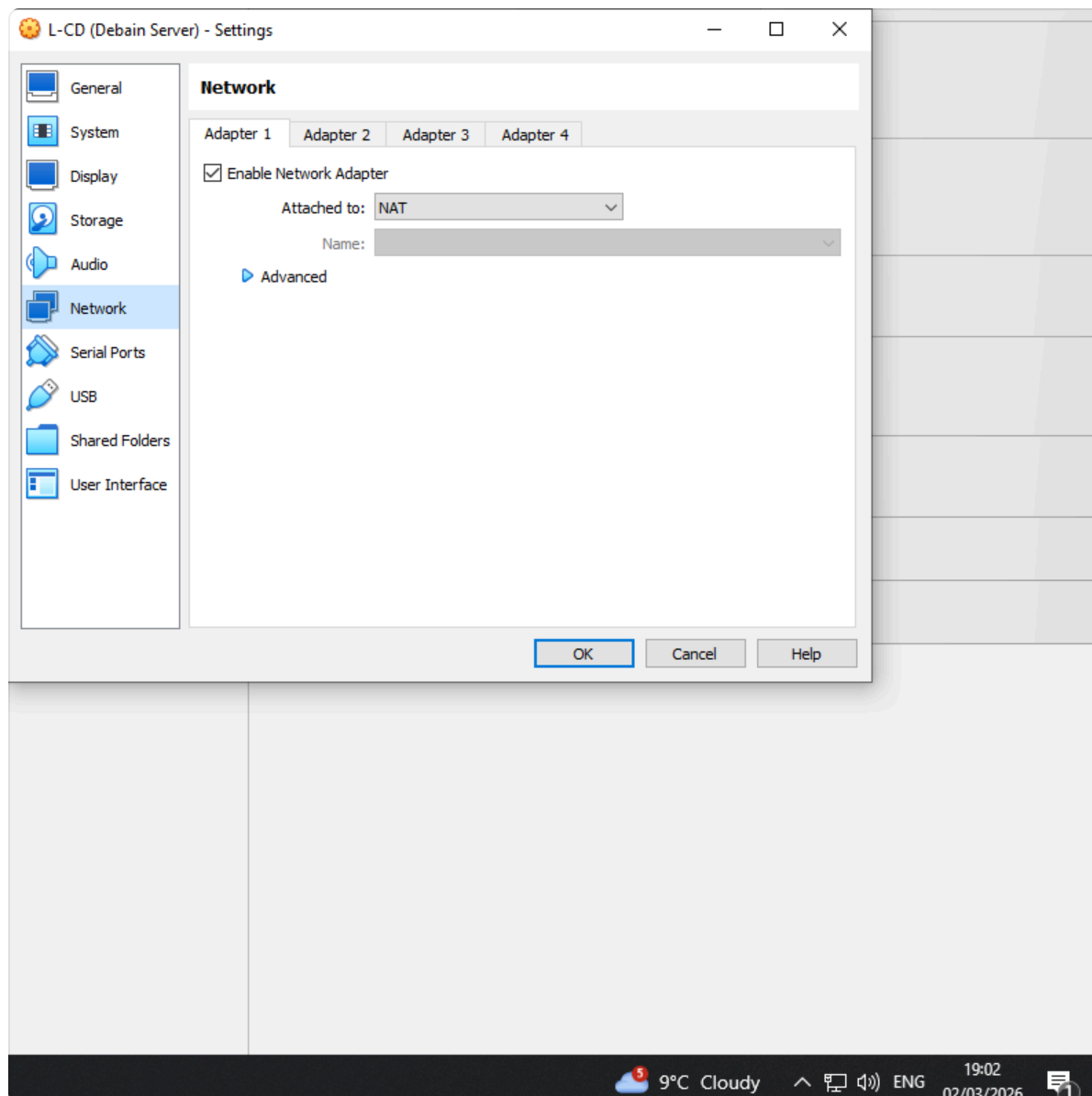


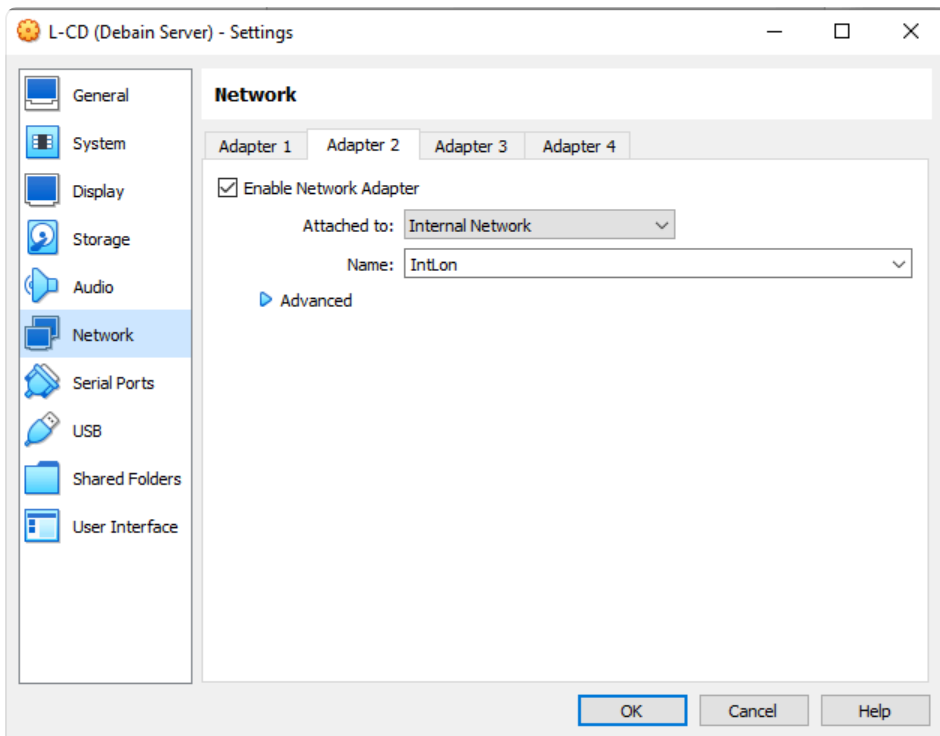


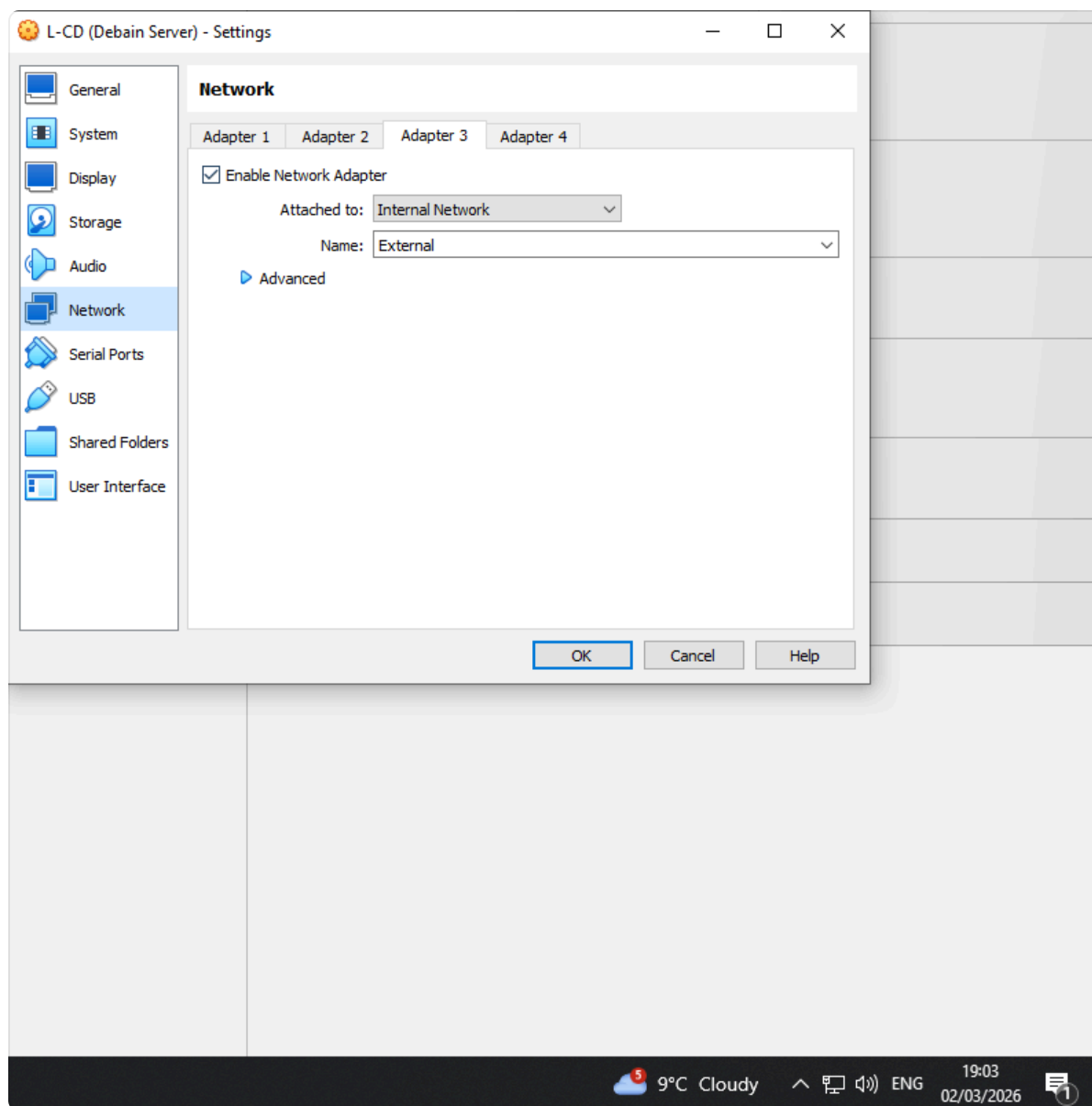


These screenshots show the network adapters for the Windows Server. The second adapter, named IntSwa, allows the Windows server and client to communicate while isolated from other networks, and the third adapter shows the connection between the two servers.

Debian Server Network Adapter Configuration







These screenshots show the network adapters on the Debian Server. As mentioned previously, the second adapter, named IntLon, allows the Debian Server to communicate with the client while remaining isolated from other networks, and the third adapter shows the connection between both servers.

Windows Server IP configuration

```
Administrator: Command Prompt

Ethernet adapter Internet:

    Connection-specific DNS Suffix  . : soac.local
    Link-local IPv6 Address . . . . . : fe80::3029:38f8:3127:23ef%6
    IPv4 Address. . . . . : 10.0.2.15
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.0.1
                                10.0.2.2

Ethernet adapter InLan:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::195f:6b3d:ad2d:f4e2%8
    IPv4 Address. . . . . : 10.10.10.1
    Subnet Mask . . . . . : 255.255.255.248
    Default Gateway . . . . . : 0.0.0.0

Ethernet adapter External:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::3568:cf22:7e4f:c807%17
    IPv4 Address. . . . . : 10.10.10.17
    Subnet Mask . . . . . : 255.255.255.252
    Default Gateway . . . . . : 

C:\Users\Administrator>
```

Windows Server 2022 Datacenter Evaluation
Windows License valid for 162 days
Build 20348.fe_release.210507-1500

Type here to search

10:18 AM
3/2/2026

```
Administrator: Command Prompt

C:\Users\Administrator>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : s-dc
    Primary Dns Suffix . . . . . : 
    Node Type . . . . . : Hybrid
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No
    DNS Suffix Search List. . . . . : soac.local

Ethernet adapter Internet:

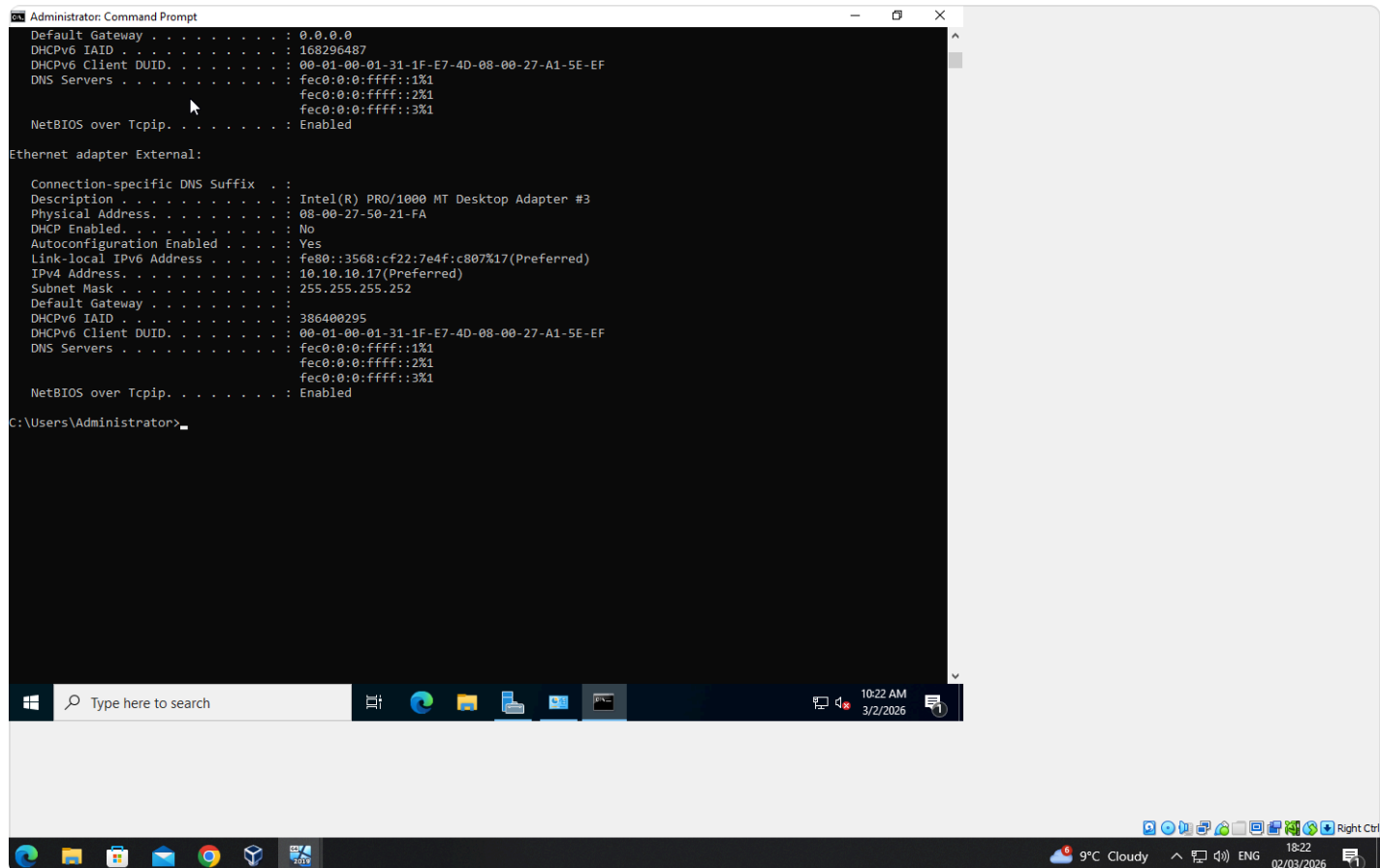
    Connection-specific DNS Suffix  . : soac.local
    Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
    Physical Address. . . . . : 08-00-27-A1-5E-EF
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::3029:38f8:3127:23ef%6(Preferrred)
    IPv4 Address. . . . . : 10.0.2.15(Preferrred)
    Subnet Mask . . . . . : 255.255.255.0
    Lease Obtained. . . . . : Friday, January 24, 1890 7:57:30 AM
    Lease Expires . . . . . : Tuesday, March 3, 2026 6:25:46 AM
    Default Gateway . . . . . : 10.0.0.1
                                10.0.2.2
    DHCP Server . . . . . : 10.0.2.2
    DHCPv6 IAID . . . . . : 101187623
    DHCPv6 Client DUID. . . . . : 00-01-00-01-31-1F-E7-4D-08-00-27-A1-5E-EF
    DNS Servers . . . . . : 192.168.135.24
                                192.168.135.50
    NetBIOS over Tcpip. . . . . : Enabled

Ethernet adapter InLan:

    Connection-specific DNS Suffix  . : 
    Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter #2
    Physical Address. . . . . : 08-00-27-D5-8C-05
    DHCP Enabled. . . . . : No
    Autoconfiguration Enabled . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::195f:6b3d:ad2d:f4e2%8(Preferrred)
    IPv4 Address. . . . . : 10.10.10.1(Preferrred)
    Subnet Mask . . . . . : 255.255.255.248
    Default Gateway . . . . . : 0.0.0.0
    DHCPv6 IAID . . . . . : 168296487
```

Type here to search

10:21 AM
3/2/2026



This screenshot shows the IP configuration for the Windows server using the **ipconfig** command. The server has been assigned a static ip address to accommodate its custom IP addressing scheme designed for this virtual network.

Windows Server Routing Table

Administrator: Command Prompt

```
C:\Users\Administrator>route print
```

Interface List

```
6...08 00 27 a1 5e ef .....Intel(R) PRO/1000 MT Desktop Adapter
8...08 00 27 d5 8c 05 .....Intel(R) PRO/1000 MT Desktop Adapter #2
17...08 00 27 50 21 fa .....Intel(R) PRO/1000 MT Desktop Adapter #3
1.....Software Loopback Interface 1
```

IPv4 Route Table

Active Routes:

Network	Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	0.0.0.0	10.0.0.1	10.0.2.15	281
0.0.0.0	0.0.0.0	0.0.0.0	On-link	192.168.1.1	281
0.0.0.0	0.0.0.0	0.0.0.0	10.0.2.2	10.0.2.15	25
10.0.2.0	255.255.255.0	255.255.255.0	On-link	10.0.2.15	281
10.0.2.15	255.255.255.255	255.255.255.255	On-link	10.0.2.15	281
10.0.2.255	255.255.255.255	255.255.255.255	On-link	10.0.2.15	281
20.0.0.0	255.255.255.252	255.255.255.252	On-link	20.0.0.1	281
20.0.0.1	255.255.255.255	255.255.255.255	On-link	20.0.0.1	281
20.0.0.3	255.255.255.255	255.255.255.255	On-link	20.0.0.1	281
127.0.0.0	255.0.0.0	255.0.0.0	On-link	127.0.0.1	331
127.0.0.1	255.255.255.255	255.255.255.255	On-link	127.0.0.1	331
127.255.255.255	255.255.255.255	255.255.255.255	On-link	127.0.0.1	331
192.168.1.0	255.255.255.0	255.255.255.0	On-link	192.168.1.1	281
192.168.1.1	255.255.255.255	255.255.255.255	On-link	192.168.1.1	281
192.168.1.255	255.255.255.255	255.255.255.255	On-link	192.168.1.1	281
224.0.0.0	240.0.0.0	240.0.0.0	On-link	127.0.0.1	331
224.0.0.0	240.0.0.0	240.0.0.0	On-link	10.0.2.15	281

LEARN MORE

ROLES AND SERVER GROUPS

Roles: 1 | Server groups: 1 | Servers total: 1

Windows Server 2022 Datacenter Evaluation
Windows License valid for 162 days
Build 20348.fe_release.210507-1500

Type here to search

10°C Light rain 15:06 02/03/2026

Administrator: Command Prompt

```
Persistent Routes:
```

Network	Address	Netmask	Gateway	Address	Metric
0.0.0.0	0.0.0.0	0.0.0.0	10.0.0.1	Default	
0.0.0.0	0.0.0.0	0.0.0.0	192.168.1.1	Default	

IPv6 Route Table

Active Routes:

If	Metric	Network	Destination	Gateway
1	331	:::1/128		On-link
6	281	fe80::/64		On-link
8	281	fe80::/64		On-link
17	281	fe80::/64		On-link
8	281	fe80::195f:6b3d:ad2d:f4e2/128		On-link
6	281	fe80::3029:38f8:3127:23ef/128		On-link
17	281	fe80::3568:cf22:7e4f:c807/128		On-link
1	331	ff00::/8		On-link
6	281	ff00::/8		On-link
8	281	ff00::/8		On-link
17	281	ff00::/8		On-link

Persistent Routes:

None

C:\Users\Administrator>

LEARN MORE

ROLES AND SERVER GROUPS

Roles: 1 | Server groups: 1 | Servers total: 1

Windows Server 2022 Datacenter Evaluation
Windows License valid for 162 days
Build 20348.fe_release.210507-1500

Type here to search

10°C Light rain 15:11 02/03/2026

This screenshot shows the routing table for the Windows Server by using the **route print** command. This table contains information about network paths and confirmed network interfaces that the server can identify.

Debian Server IP configuration

```
leon@LCD:~$ sudo systemctl restart networking
leon@LCD:~$
leon@LCD:~$ ip a
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,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:77:97:9a brd ff:ff:ff:ff:ff:ff
    altname enx08002777979a
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 86381sec preferred_lft 75581sec
    inet6 fe80::5bf:50a8:27d:e2d1/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d7:11:b0 brd ff:ff:ff:ff:ff:ff
    altname enx080027d711b0
    inet 10.10.10.9/29 brd 10.10.10.15 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed7:11b0/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ea:77:18 brd ff:ff:ff:ff:ff:ff
    altname enx080027ea7718
    inet 10.10.10.18/30 brd 10.10.10.19 scope global enp0s9
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feea:7718/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
leon@LCD:~$
```

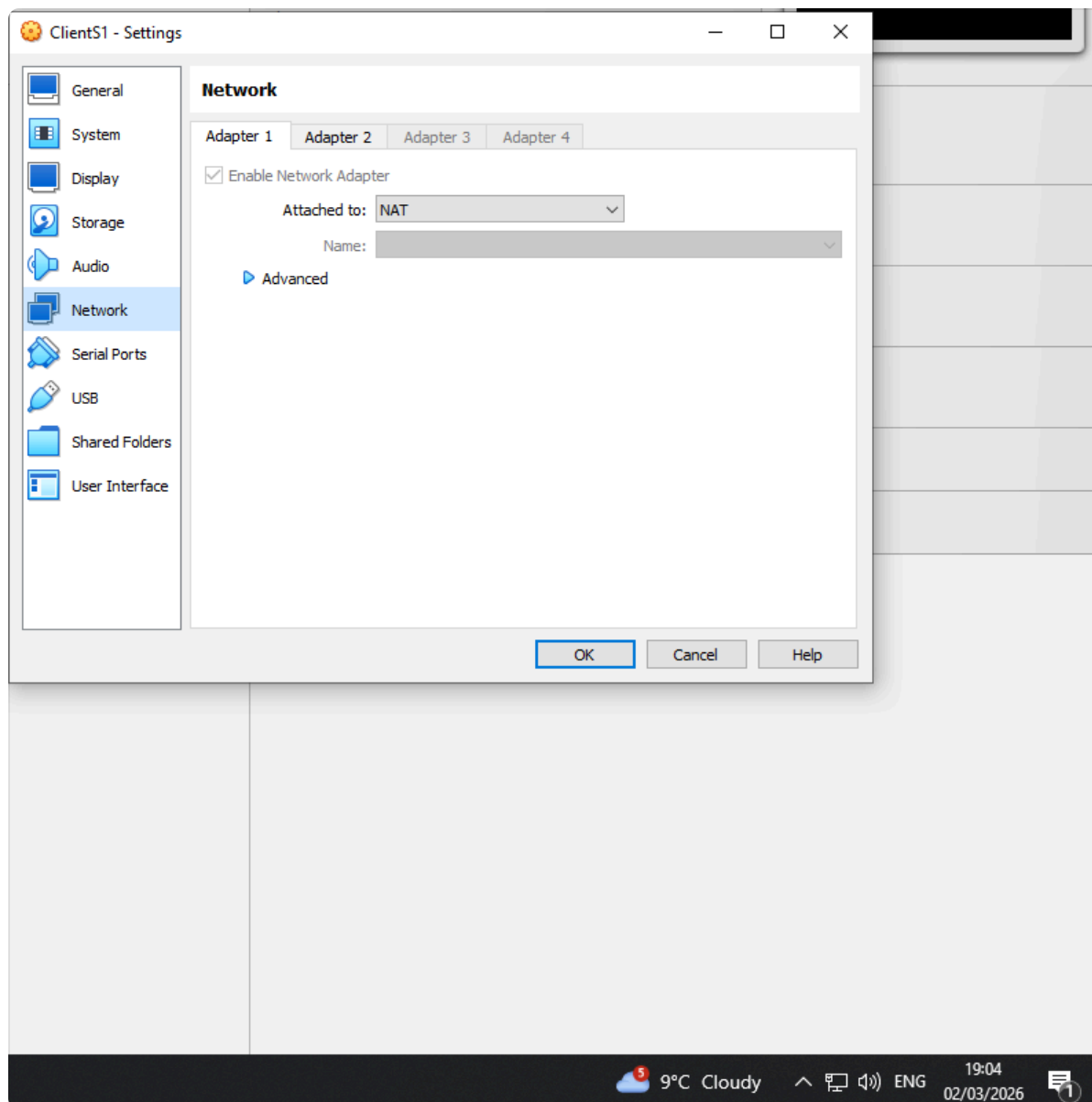
This screenshot shows the network interface configuration of the Debian server using the **ip a** command. This confirms the server has been assigned the correct static IP address with a custom subnet.

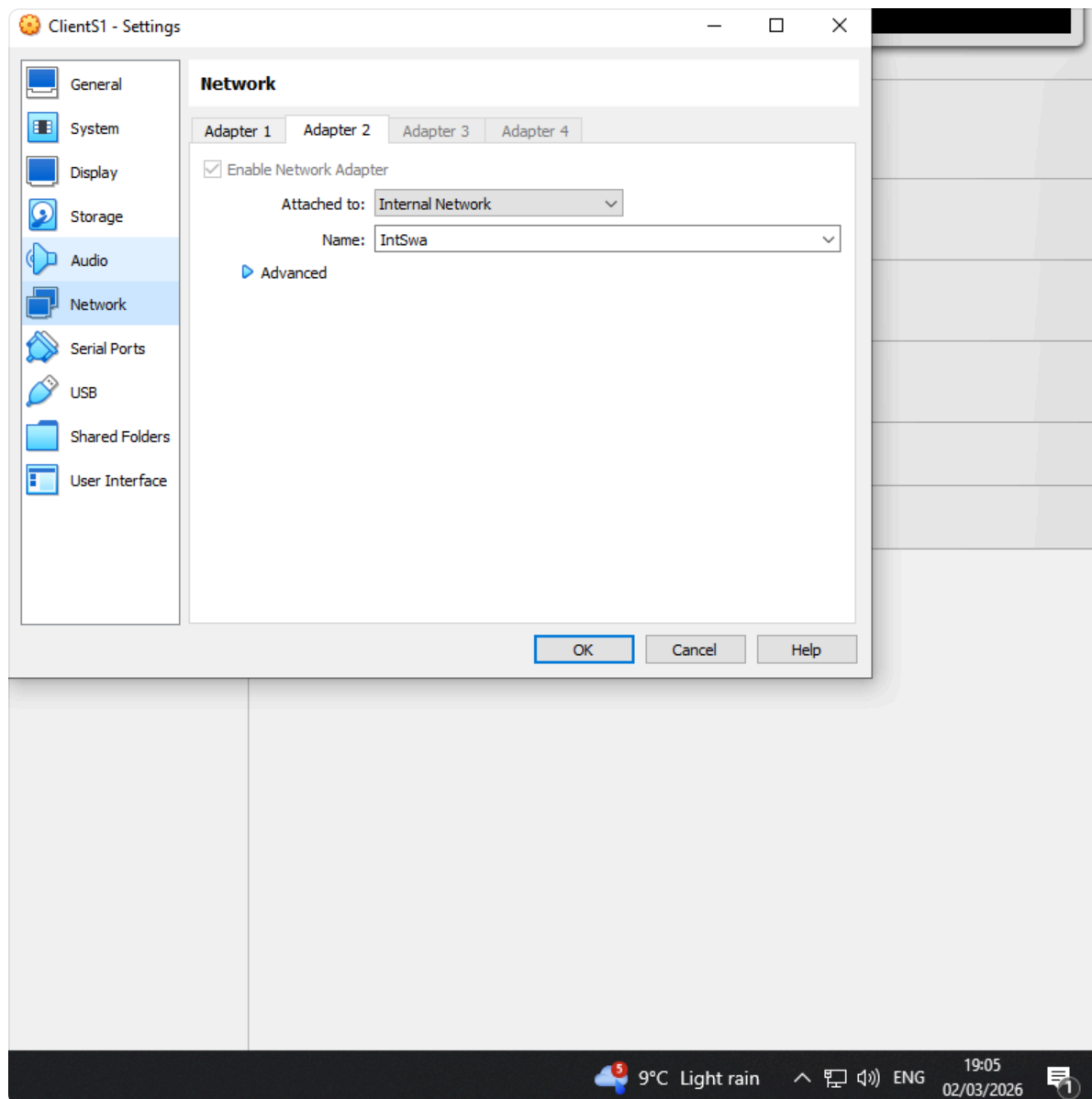
Debian Server Routing Table

```
leon@LCD:~$ ip route
default via 10.10.10.1 dev enp0s9 onlink
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 1002
10.0.2.0/24 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 1002
10.10.10.8/29 dev enp0s8 proto kernel scope link src 10.10.10.9
10.10.10.16/30 dev enp0s9 proto kernel scope link src 10.10.10.18
leon@LCD:~$
```

This screenshot shows the routing table of the Debian server using the **IP route** command. This routing table confirms that the server can communicate with the client Linux machine.

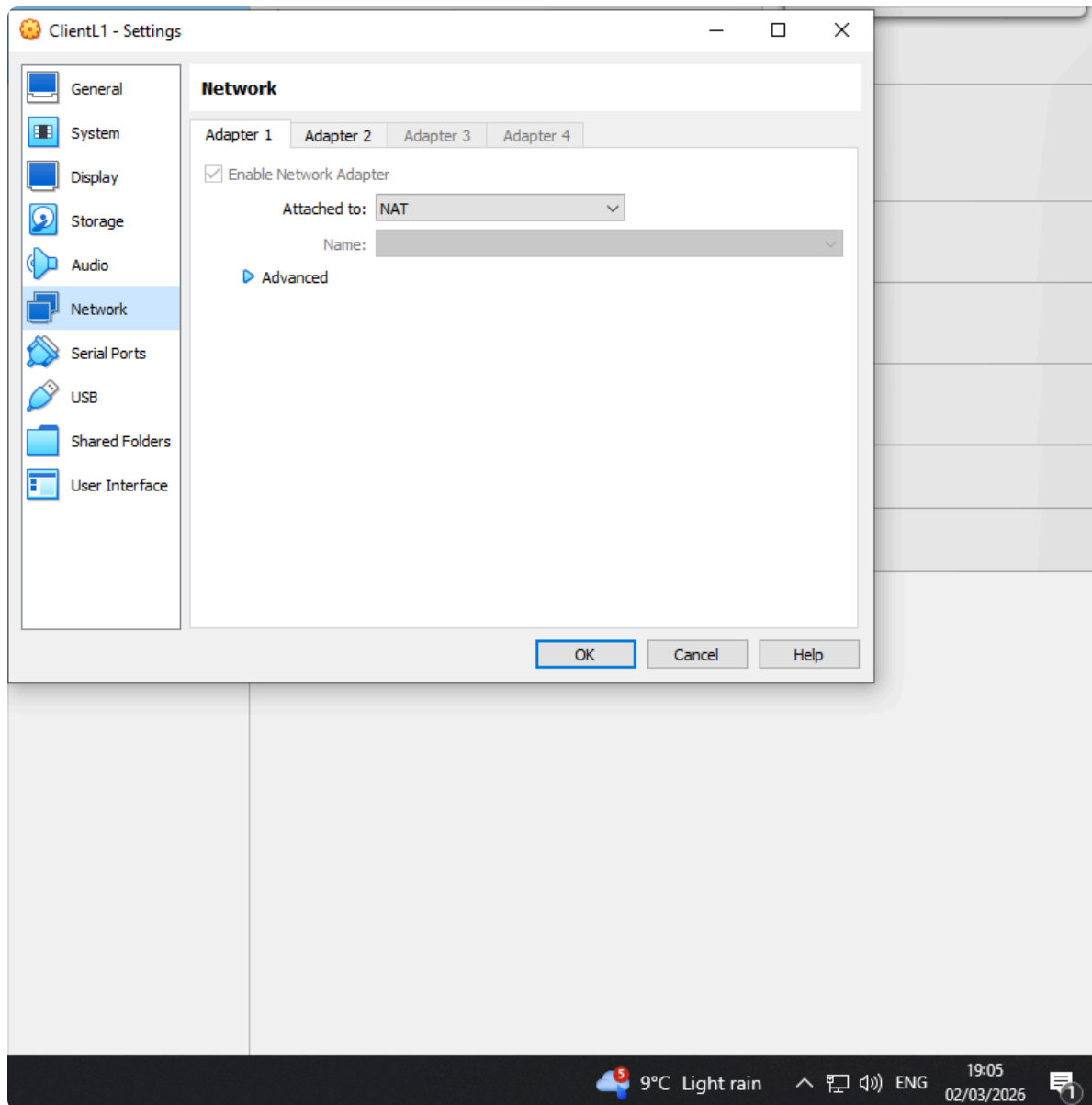
Windows Client Network Adapter Configuration

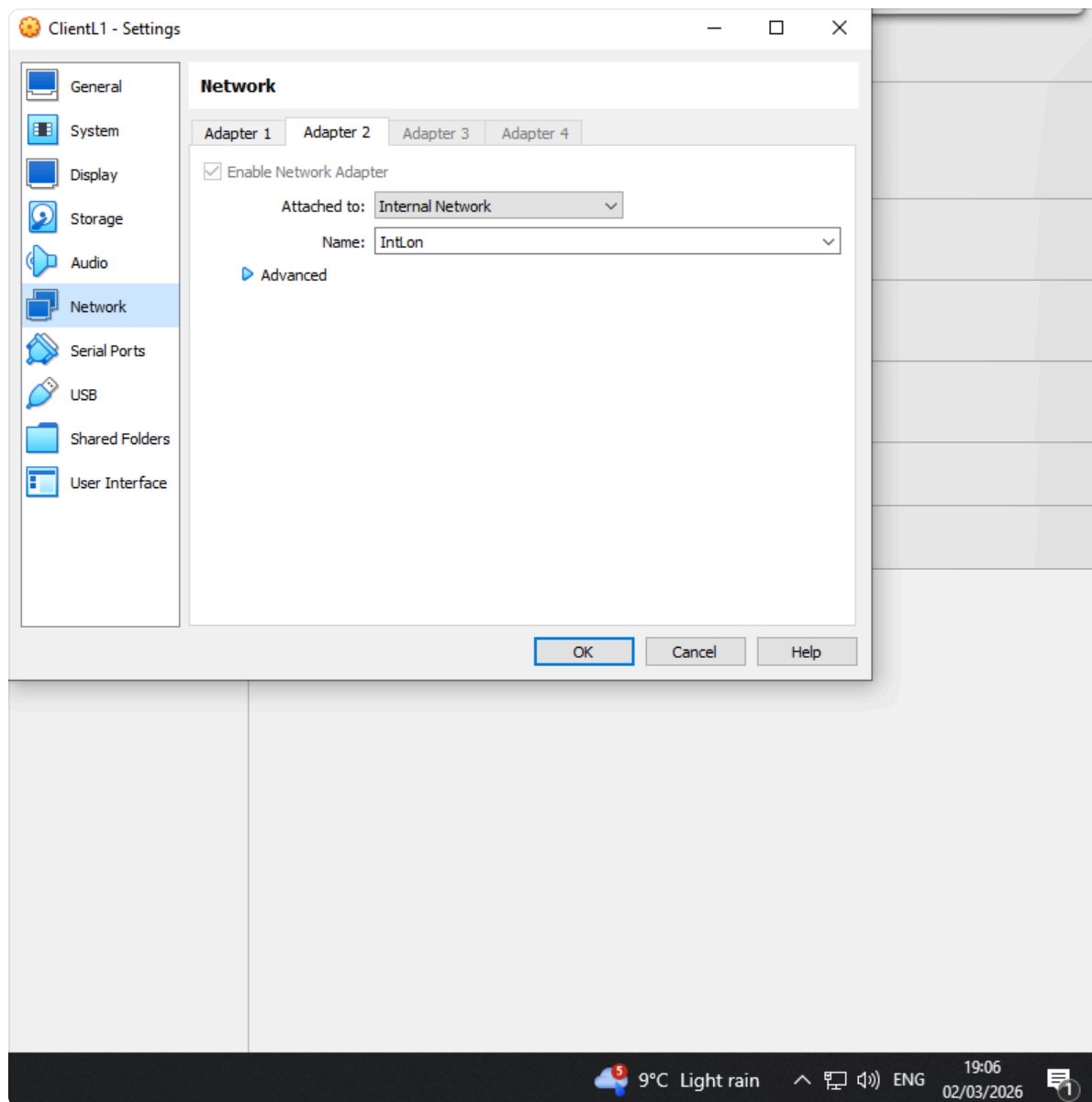




These screenshots show the network adapters for the Windows Client. The second adapter, named IntSwa, allows the Windows client and server to communicate while isolated from other networks.

Debian Client Network Adapter Configuration





These screenshots show the network adapters on the Debian Client. The second adapter, named IntLon, allows the Debian Client to communicate with the server while remaining isolated from other networks.

Windows Client IP configuration

```
Command Prompt
C:\Users\Client1>ipconfigall
'ipconfigall' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Client1>ipconfig

Windows IP Configuration

Ethernet adapter Internet:

    Connection-specific DNS Suffix  . : soac.local
    Link-local IPv6 Address . . . . . : fe80::9f76:4fd0:dbc7:be7a%5
    IPv4 Address. . . . . : 10.0.2.15
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.2.2

Ethernet adapter Solan:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::58ba:95a8:edba:d6ac%11
    IPv4 Address. . . . . : 10.10.10.2
    Subnet Mask . . . . . : 255.255.255.248
    Default Gateway . . . . . : 10.10.10.1

C:\Users\Client1>
```

Updating network adapter or driver
Fixing Ethernet connection issues
Run an internet speed test
Finding my IP address

Activate Windows
Go to Settings to activate Windows.

Type here to search

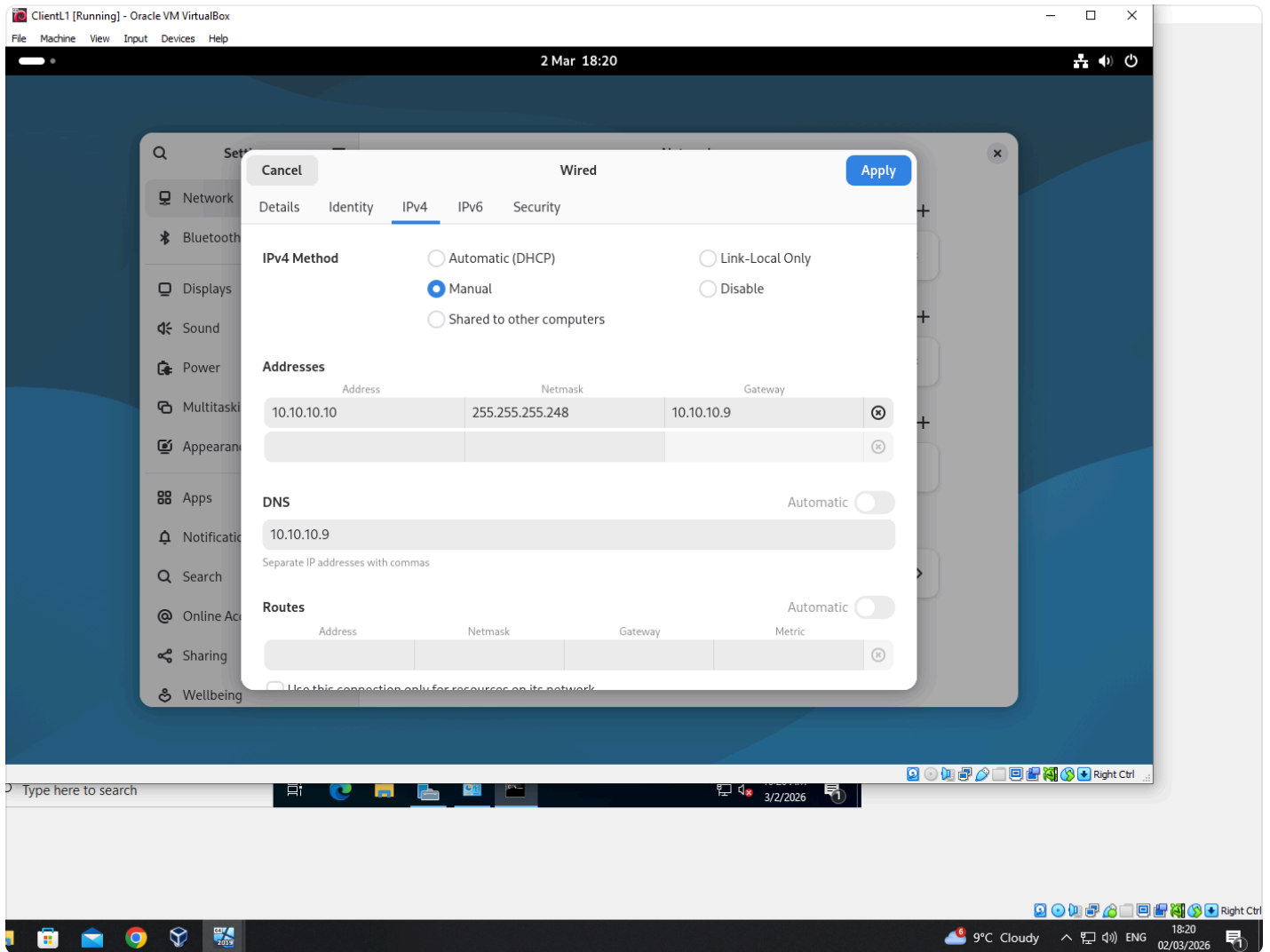
Unit...

18:20
02/03/2026

9°C Cloudy
18:32
02/03/2026

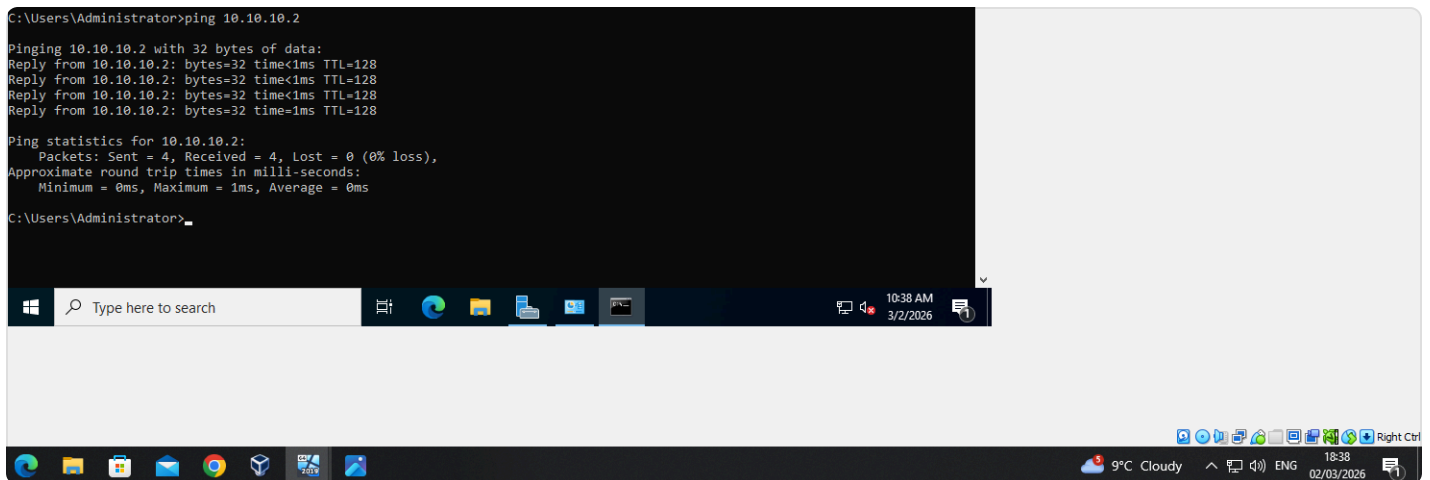
This screenshot shows the IP configuration for the Windows server using the **ipconfig** command. The client device has been assigned a static IP address and a default gateway to communicate with the Windows Server.

Debian Client IP configuration



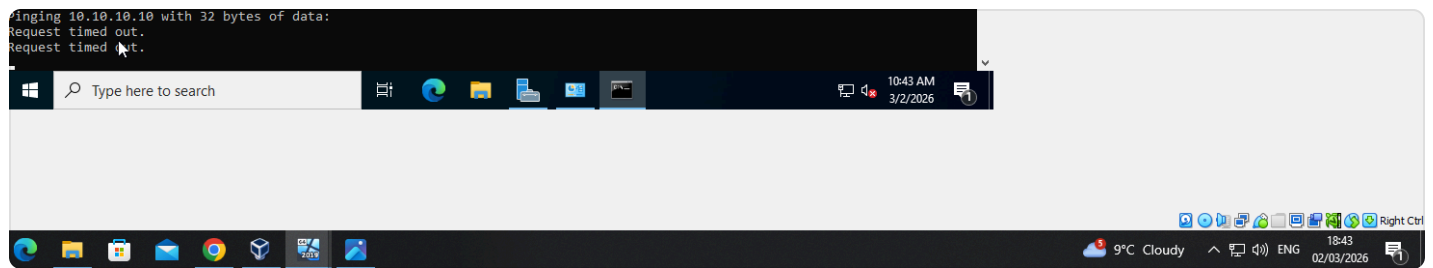
This screenshot shows the IP configuration of the Debian Client using the built-in network GUI. The client device has been assigned a static IP address and a default gateway to ensure communication with the Debian Server.

Windows Server Successful Ping test



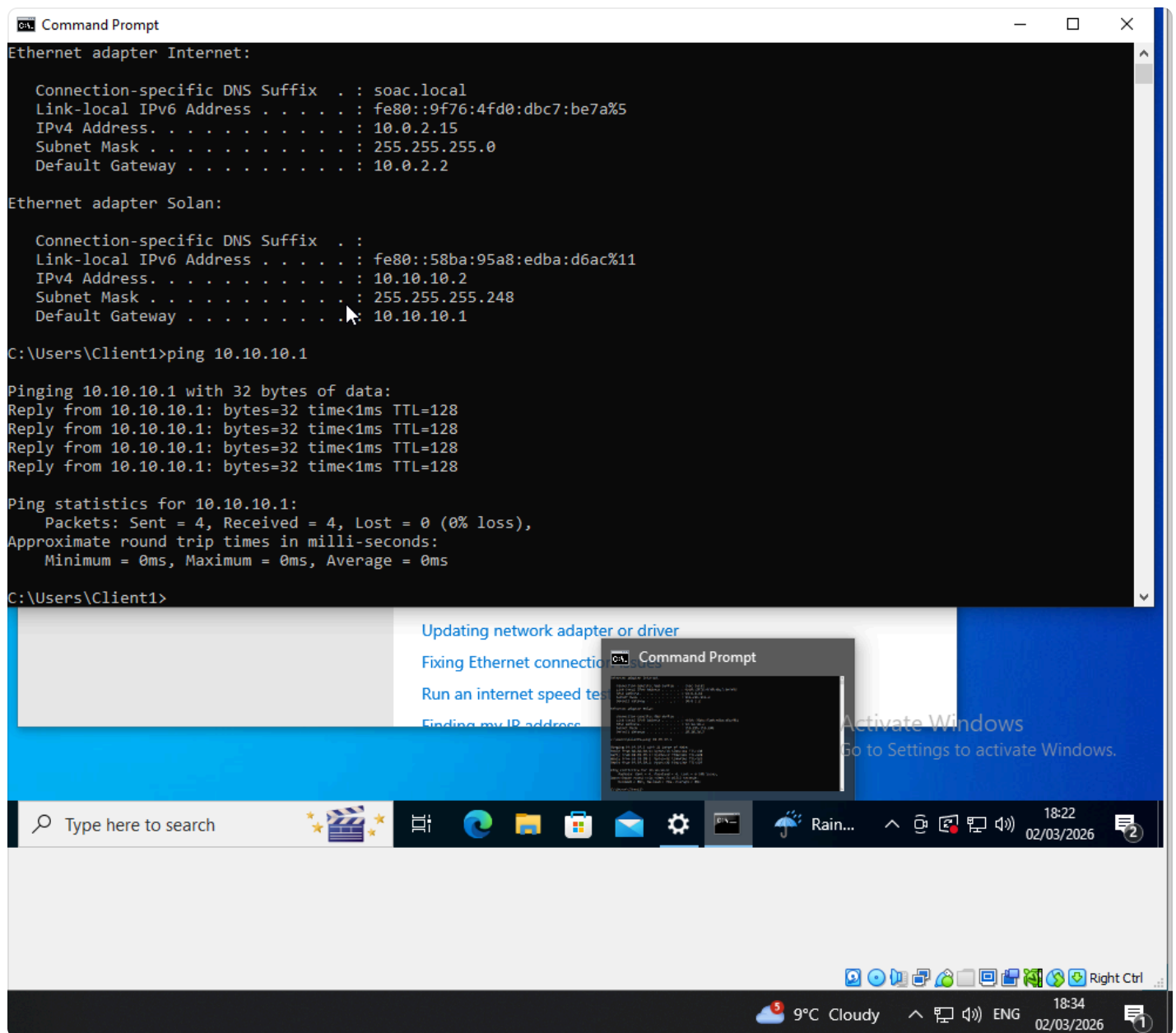
This screenshot shows a successful ping test performed by the Windows server. The replies received confirm that the target device is within reach of the server and that the IP addressing and network configuration for the Windows segment have been correctly implemented.

Windows Server Unsuccessful Ping test



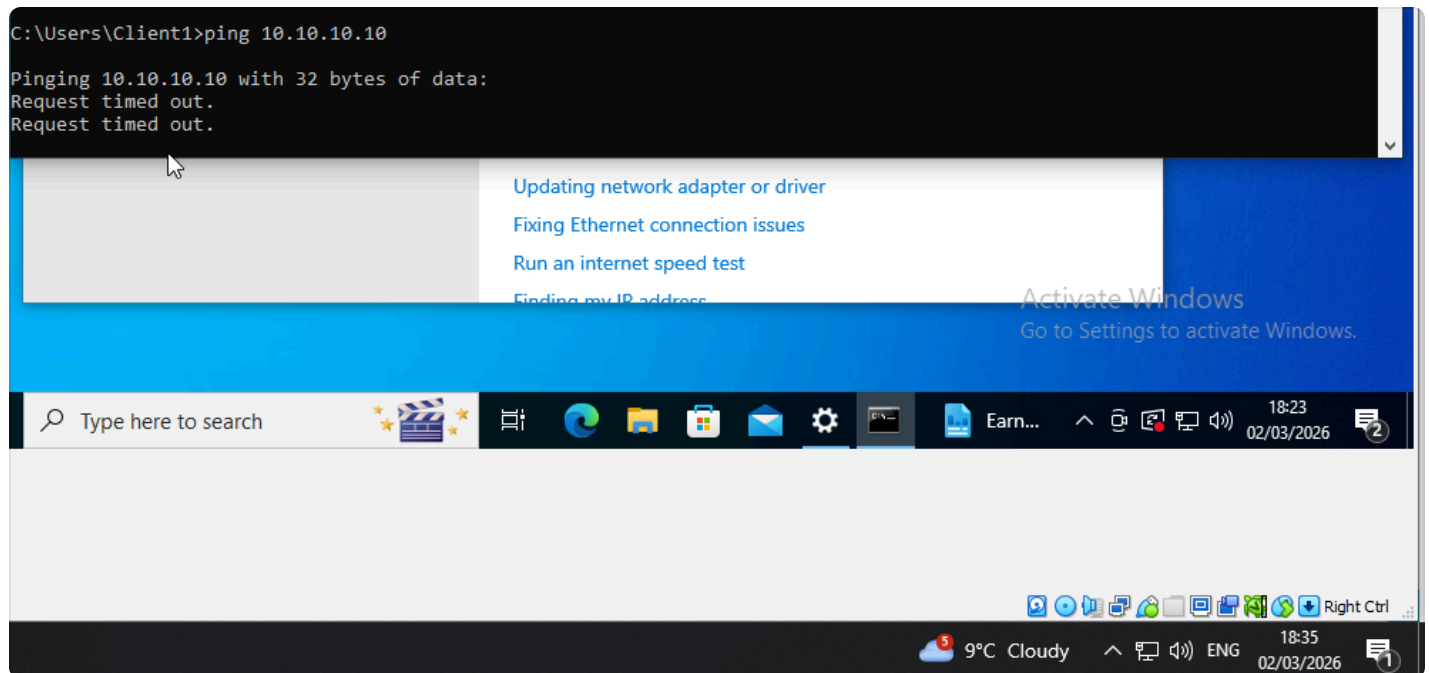
This screenshot shows an unsuccessful ping test from Windows Server to a device on a different network. The reply "request timed out" indicates that the destination IP address could not be reached.

Windows Client Successful Ping test



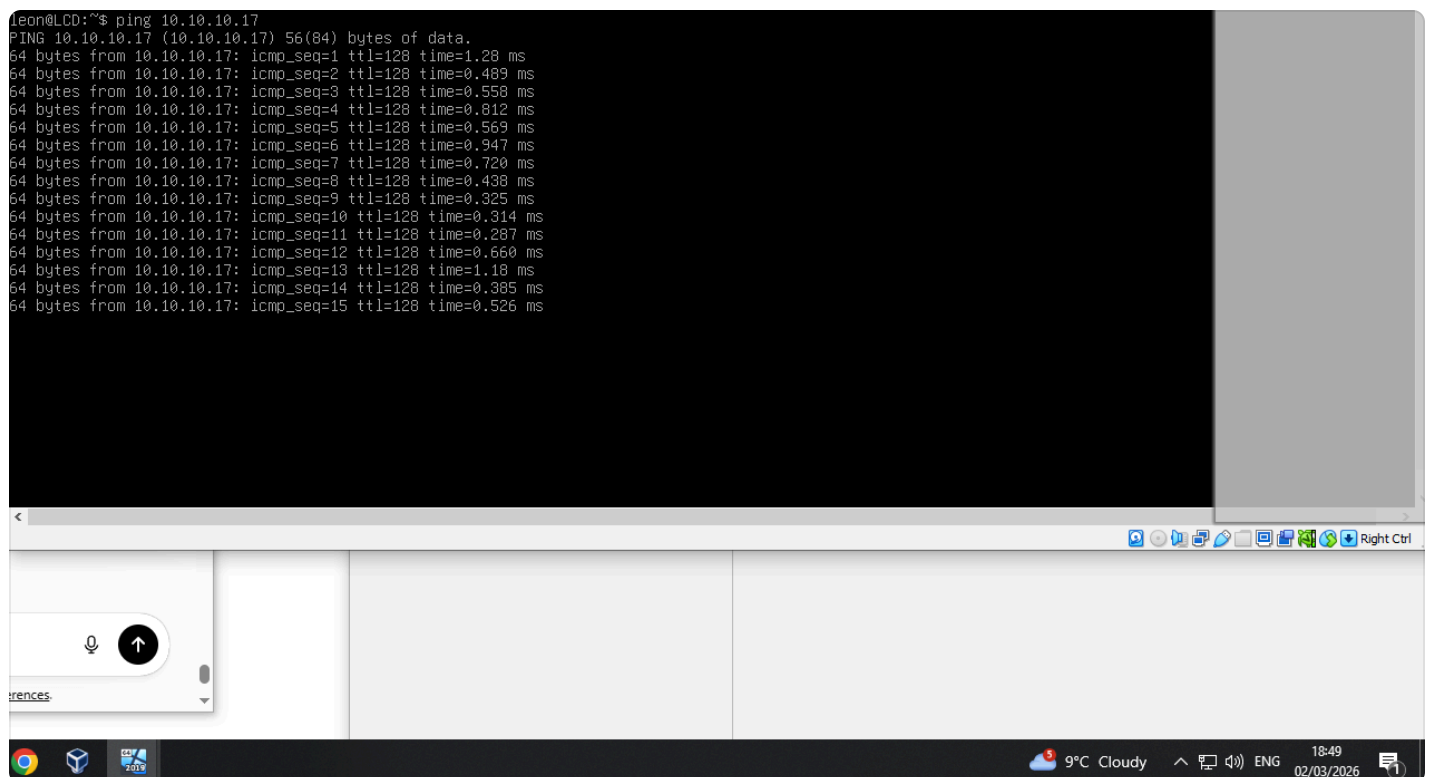
This screenshot shows a successful ping test between the Windows client and server devices. The replies mean that communication within the local subnet is functioning correctly.

Windows Client Unsuccessful Ping test



This screenshot shows an unsuccessful ping test from the Windows Client device to a device on a Debian network. This timed-out response indicates that communication between the two networks isn't yet possible.

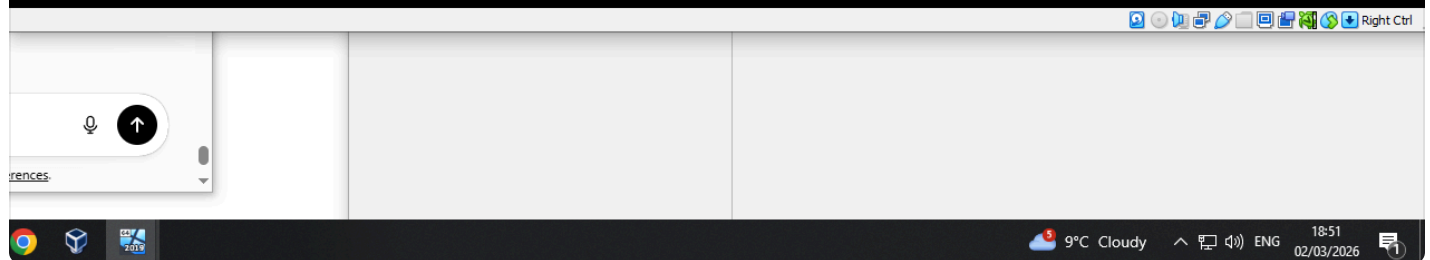
Debian Server Successful Ping test



This screenshot shows a successful ping test from the Debian Server device by using the ping command. The received replies show that the Debian Server can communicate with other devices on the same network.

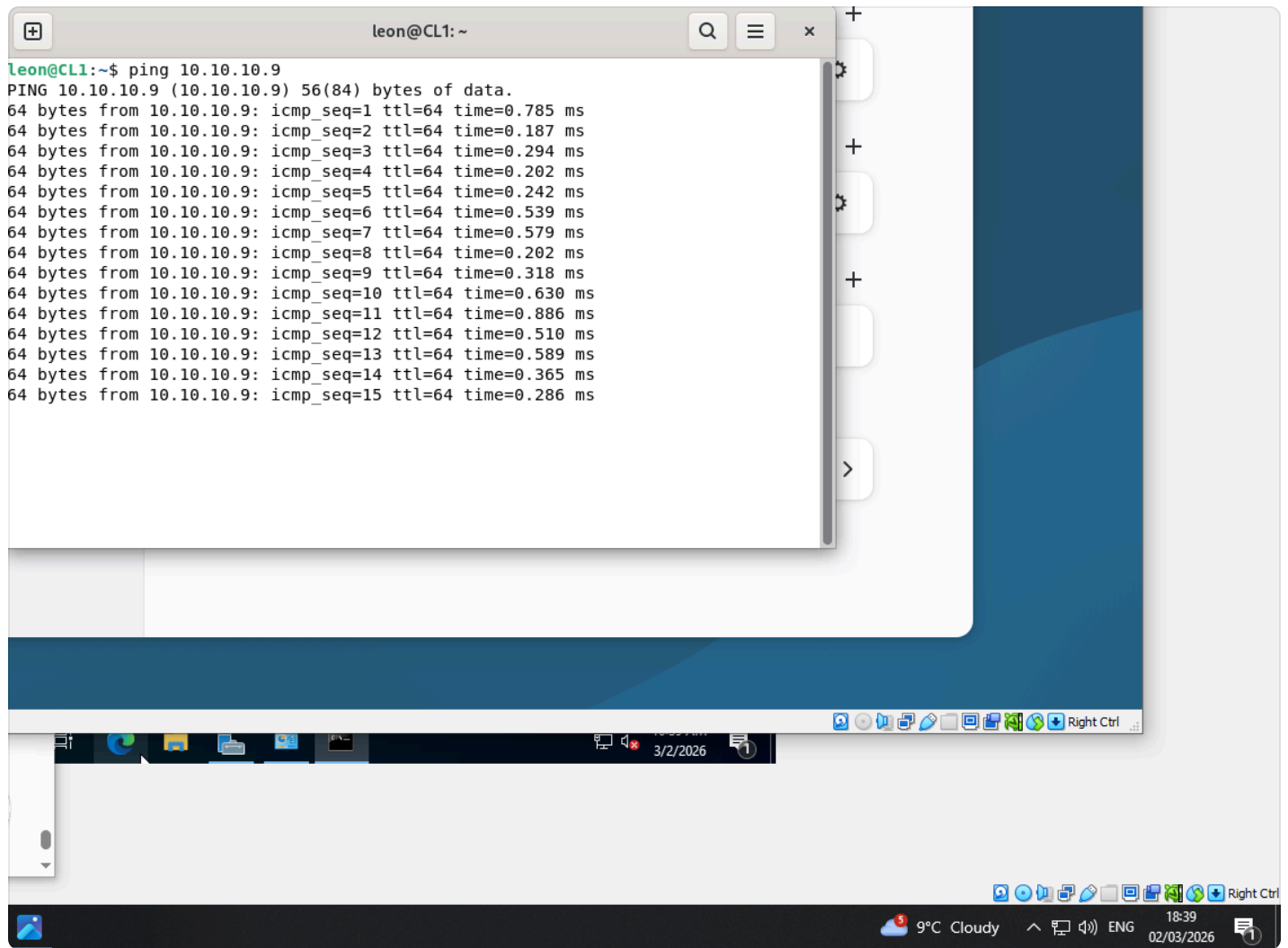
Debian Server Unsuccessful Ping test

```
leon@LCD:~$ ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
From 10.10.10.18 icmp_seq=1 Destination Host Unreachable
From 10.10.10.18 icmp_seq=2 Destination Host Unreachable
From 10.10.10.18 icmp_seq=3 Destination Host Unreachable
From 10.10.10.18 icmp_seq=4 Destination Host Unreachable
From 10.10.10.18 icmp_seq=5 Destination Host Unreachable
From 10.10.10.18 icmp_seq=6 Destination Host Unreachable
From 10.10.10.18 icmp_seq=7 Destination Host Unreachable
```



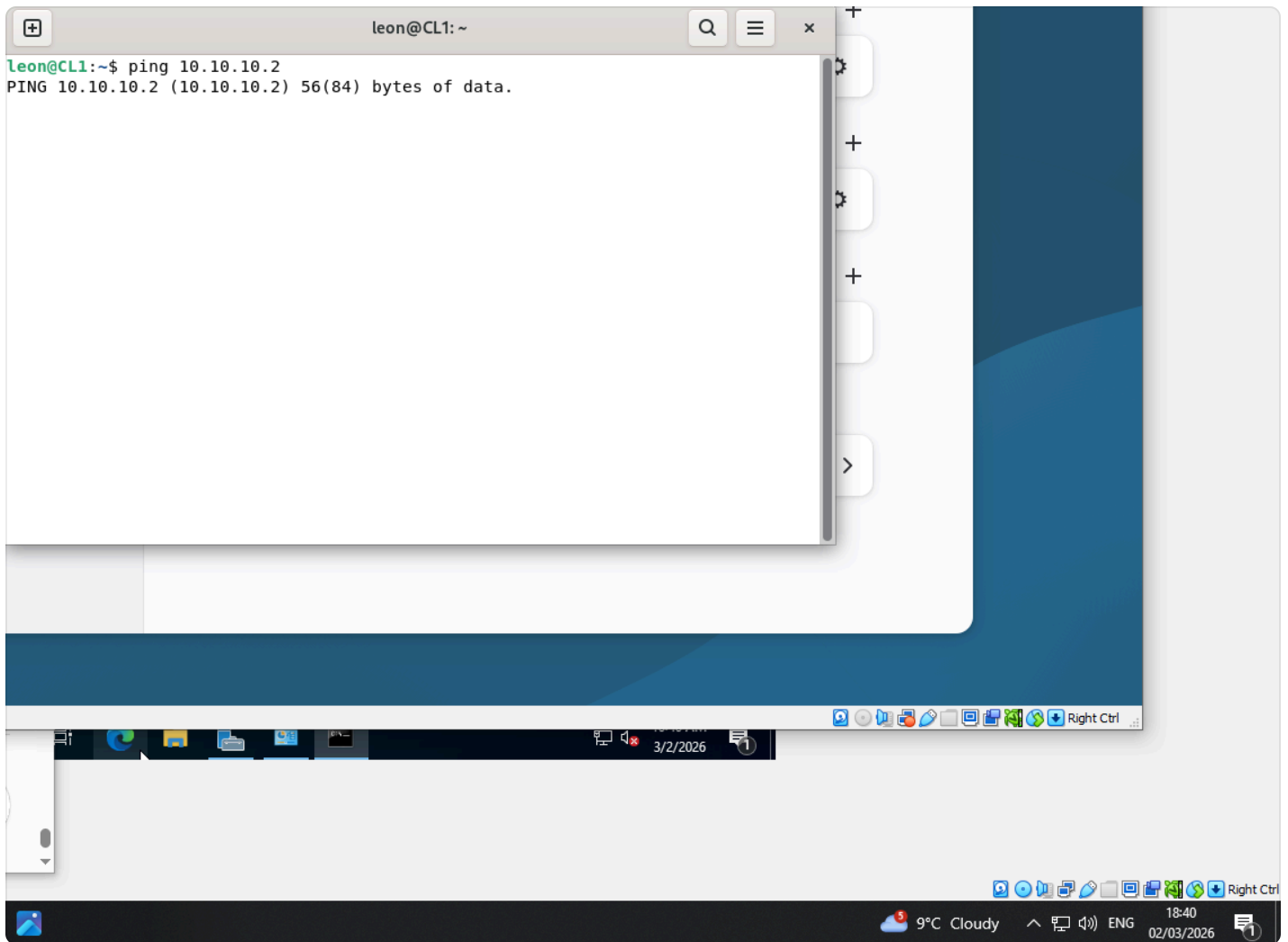
This unsuccessful ping test performed by the Debian Server shows that it isn't possible to communicate with it. The destination host message indicates that it cannot be reached because it resides on a different subnet.

Debian Client Successful Ping test



This screenshot shows a successful ping test between the Debian client and server devices. The replies mean that communication within the local subnet is functioning correctly.

Debian Client Unsuccessful Ping test



This screenshot shows an unsuccessful ping attempt from the Debian Client to a device located on a Windows network. The failed request confirms that it cannot communicate with devices outside its subnet.

Task 3

Personal Reflection

During the initial stage of creating the virtual machines, a few technical difficulties arose. The first was that I tried to create them as quickly as possible, giving the instructions only a glance and then realizing I had missed something important. For instance, I gave too little RAM to the Windows server and too much to the Debian server. Only after creating the machines did I realize I had done so, so I had to destroy them and rebuild them. Another issue I had was assigning the IP addresses to the wrong Debian interface. I was applying the IP addresses to `enp0s8`. Still, after trying a ping test from the Debian client to the Debian server, a message appeared saying "Destination Host Unreachable," so I checked that I had used the correct IP addresses I designed, and they were correct. I ran the command `ip a` and `ip route` everything was correct. Until I looked at the Internal network adapters, I realised I had assigned the client's IP address to the NAT interface instead of the Internal Network interface.

This experience allowed me to gain a better understanding of how to set up Virtual Machines, especially Linux servers and clients, as they were brand new to me and I had never configured one before. I already had experience creating Windows virtual machines, but I did forget a few steps when creating the Server, so it was great to have instructions online and a recap to learn new tricks when implementing them.

References

- [1] Red Hat, "What is a hypervisor?" Red Hat, 2024. [Online]. Available: <https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor>.
- [2] VMware, "What is a hypervisor?" VMware. [Online]. Available: <https://www.vmware.com/topics/hypervisor>.
- [3] N. U. Amin, B. Shadman, F. Mukhtiyar and L. L. Soon, "The Role and Impact of Hypervisor Technology in Modern IT Infrastructure," Preprints, 2025. [Online]. Available: https://www.researchgate.net/publication/391867411_The_Role_and_Impact_of_Hypervisor_Technology_in_Modern_IT_Infrastructure
- [4] Cloudflare, "What is a packet? | Network packet definition," Cloudflare Learning Center. [Online]. Available: <https://www.cloudflare.com/learning/network-layer/what-is-a-packet/>.
- [5] Microchip Technology Inc., "Introduction to TCP/IP (Part 1) – Routers, Switches and Addressing," Microchip Developer Help, Nov. 10, 2023. [Online]. Available: <https://developerhelp.microchip.com/xwiki/bin/view/applications/tcp-ip/routers-switches-addressing/>.
- [6] Juniper Networks, "Understanding IPv4 and IPv6 Protocol Families," Juniper Networks Documentation. [Online]. Available: <https://www.juniper.net/documentation/us/en/software/junos/interfaces-security-devices/topics/topic-map/security-interface-ipv4-ipv6-protocol.html>
- [7] Microsoft, "Understand TCP/IP addressing and subnetting basics," Microsoft Learn, Windows Client Troubleshooting. [Online]. Available: <https://learn.microsoft.com/en-us/troubleshoot/windows-client/networking/tcpip-addressing-and-subnetting>
- [8] Cloudflare, "What is routing? | IP routing," Cloudflare Learning Center. [Online]. Available: <https://www.cloudflare.com/learning/network-layer/what-is-routing/>.
- [9] TechTarget, "What is a routing table?" TechTarget SearchNetworking, Nov. 30, 2022. [Online]. Available: <https://www.techtarget.com/searchnetworking/definition/routing-table>.
- [10] GeeksforGeeks, "Difference between Static and Dynamic Routing," GeeksforGeeks, Oct. 07, 2025. [Online]. Available: <https://www.geeksforgeeks.org/computer-networks/difference-between-static-and-dynamic-routing/>
- [11] Cloudflare, "What is a VPN? | Virtual Private Network," Cloudflare Learning Center. [Online]. Available: <https://www.cloudflare.com/learning/network-layer/what-is-a-vpn/>
. [Accessed: 03-Mar-2026].
- [12] Cisco Systems, "Dynamic Host Configuration Protocol (DHCP)," Cisco. [Online]. Available: <https://www.cisco.com/c/en/us/tech/ip/dynamic-host-configuration-protocol-dhcp/index.html>
. [Accessed: 03-Mar-2026].
- [13] Microsoft, "Active Directory Domain Services Overview," Microsoft Learn. [Online]. Available: <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>
. [Accessed: 03-Mar-2026].
- [14] DigiCert, "What is a Certificate Authority (CA)?," DigiCert. [Online]. Available: <https://www.digicert.com/faq/public-trust-and-certificates/what-is-a-certificate-authority>
. [Accessed: 03-Mar-2026].