

Research, build, and deploy a Virtualised Enterprise network

2421062

Hasan Sivas

Task 1 - Research	1
Task 2 - Technical Documentation	4
Task 3 - Reflection	20
Bibliography	20

Task 1 - Research

This paper will aim to document the implementation of a prototype network for a hypothetical multinational corporation called NOSS2026 which has two networks based in Swansea and London and cover the basic concepts necessary to develop a suitable network.

Hypervisors

Type I hypervisors, also known as bare-metal hypervisors, are installed directly onto physical hardware and manage virtual machines without the need for a host operating system. Type II hypervisors are software-based and run on top of an existing operating system, relying on it to manage hardware resources[8].

For the purposes of this report and subsequent demonstrations, it would be most suitable to use Type II Hypervisors to prototype the network.

There are numerous reasons why this would be advantageous: First of all, because Type II hypervisors are virtualised OSes that run on top of a host operating system, they are significantly easier to install, configure, delete, reset and manage compared to bare-metal (Type I) solutions, which are implemented directly on an actual machine. This makes Type II hypervisors ideal for rapid prototyping and iterative testing, where changes may need to be made frequently without impacting physical hardware.

The main disadvantage of using Type II hypervisors, however, is that the physical distance between the two networks (London to Swansea) is not accurately simulated and the requirement of an additional software layer will place more strain on the server machine.

IP

IP addresses are unique identifiers assigned to every device connecting to a network[1]- the IP address is included in the TCP/IP protocol packet[3]. Combined with the router's address table, this allows packets to know which device they need to be delivered to on a network, forming the fundamental backbone of information exchange on networks.

IP addresses are divided into two portions, the host portion and the network portion. The network portion of the IP address is the same for all devices on a network/subnet, and allow the network they belong to be immediately identified. The host portion of an IP address is unique for each device on its network[2].

An IPv4 address is made up of 32bits, often formatted as 4bytes or octets e.g. *168.24.10.122*. This should theoretically allow for 4,294,967,296 addresses[3], however some ranges of IP Addresses are reserved as 'private ip addresses' for business use.

FROM	TO
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

Table showing the range of private IPv4 Address spaces

Because the pool of available IPv4 addresses began to shrink due to the increased dependence of society on the internet (IoT, more devices & networks), IPv6 was developed as the solution [2]. IPv6 addresses are 128bits long and have 3.403×10^{38} available addresses [3], making the pool of ipv6 addresses effectively inexhaustible. IPv6 addresses are divided into 8 hexadecimal numbers (hextets). Due to the length of these addresses, there are established rules for shortening them; leading 0s can be hidden and one row only of hextets containing only 0s can be omitted (omitting more than one would shift the frame and make the original address unrestorable) e.g. *2600:CAFE:d:5b3::5*.

IPv6 also boasts greater built-in security in its protocol, more efficient routing (due to simpler fixed headers) and Stateless Autoconfiguration (SLAAC) which allows each device on the network to configure its own address and eliminates the need for DHCP [3]. Therefore, depending on the nature and purpose of the network, it may be worth implementing IPv6 addresses and routing for additional security.

Routing

A router performs the task of forwarding messages between different network subnets, determining the best path forward for packets to be delivered, allocating traffic and building routing tables [5]. Routers build a table which maps IP addresses to each device on the network's MAC address, which they use to determine the most efficient and reliable path for data transmission, resulting in more efficient communication on the network.

Effective routing and subnetting strategies, particularly through techniques such as Variable Length Subnet Masking (VLSM) enhances not just network efficiency but also security[4]. By dividing networks into logically structured subnets, administrators can reduce broadcast domains, improve traffic segmentation, and minimise unnecessary routing overhead. Segmentation supports more secure data transmission by limiting lateral movement across subnets and improving traffic control policies as access to specific networks can be gated to users based on access level.

In addition to static routing, dynamic routing (such as RIPV2 or OSPF) can also be used to ensure that larger networks are kept optimized even as devices are added and removed.

For NOSS2026's needs, it may be best to implement a mix of both static and dynamic routing protocols, with certain static routes being given higher priority based on their usage. The dynamic routing's automatic updates, combined with prioritised static routes, may allow for greater scalability and resilience as the company's network expands.

Network Services

In a network, the machine playing the role of 'server' may provide a variety of services to the other hosts on the network that enable smooth working by automating configuration, enforcing identity control, and providing services such as VPN(Virtual Private Networks)s , secure remote access or issue certificate authority to validate that communications are safe.

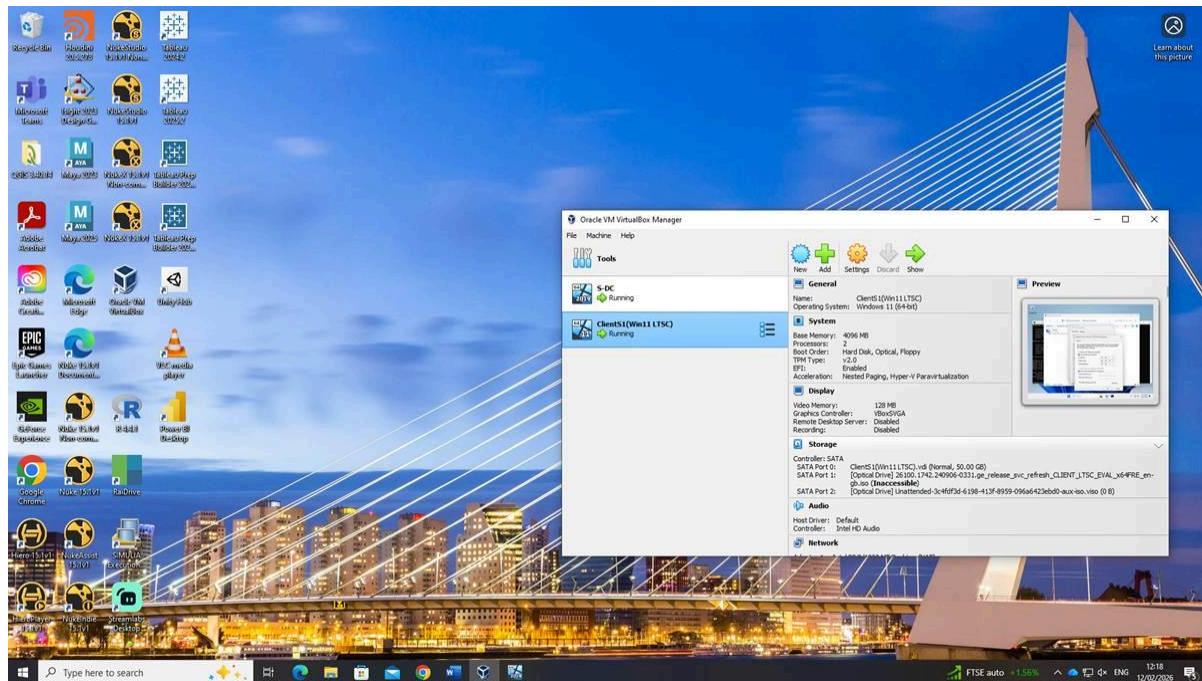
One particularly useful service that may be used on the Swansea network is Microsoft Active Directory Domain Services. This is a GUI-based tool that allows an administrator to configure users, computers, groups, and security policies from a single program[6]. Management can be made quick and easy by sorting users and machines into Organisation Units which policies and changes are applied to in bulk.

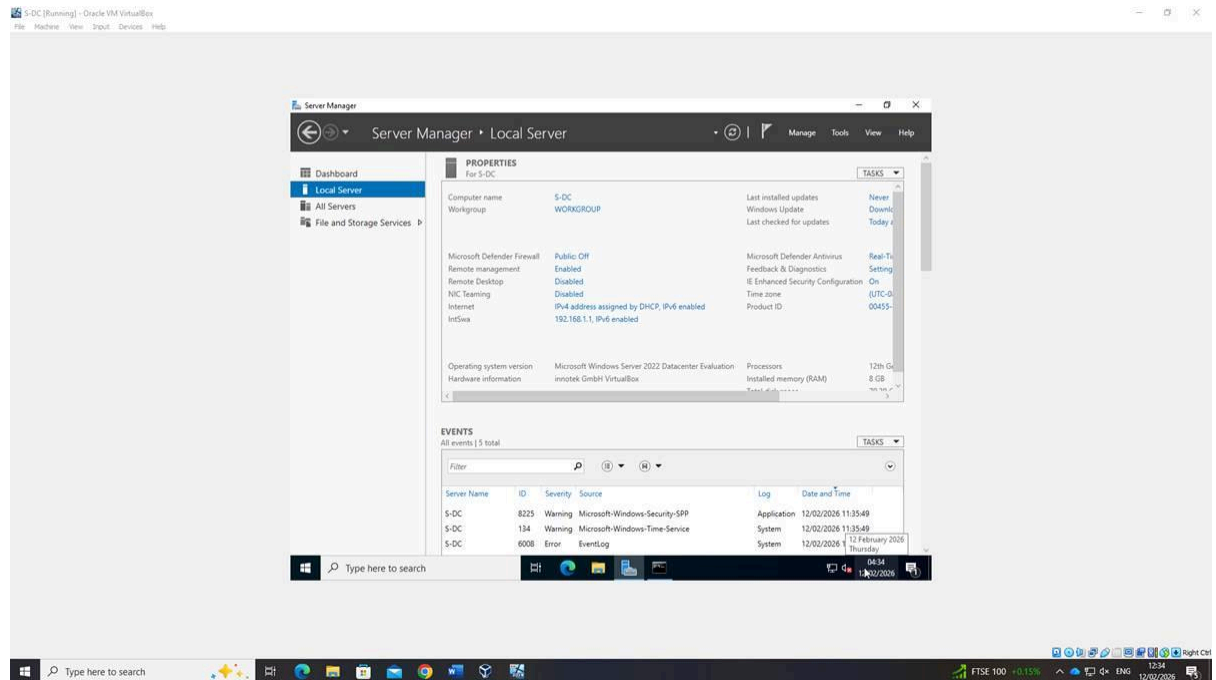
Integrating NOSS2's long-distance networks would most certainly necessitate the use of a VPN service. These can create an encrypted 'tunnel' to remotely access a server without severe risk data being compromised[9]. In particular, traffic between the London and Swansea servers can be routed through monitoring tools to protect from threats such as Man in the Middle attacks[7].

On the NOSS2 network, the server can also serve as the DHCP (dynamic host configuration protocol) provider. This means that IP addresses will be automatically assigned to joining devices, reducing room for human error.

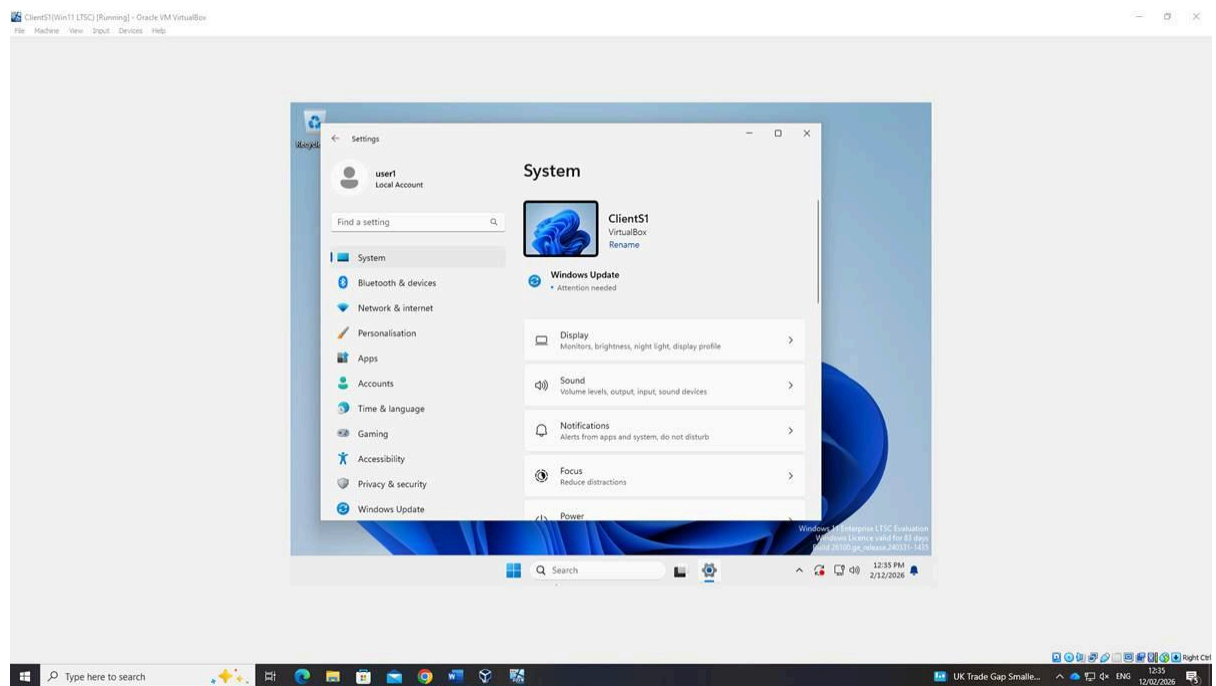
Task 2 - Technical Documentation

Setting up Windows machines

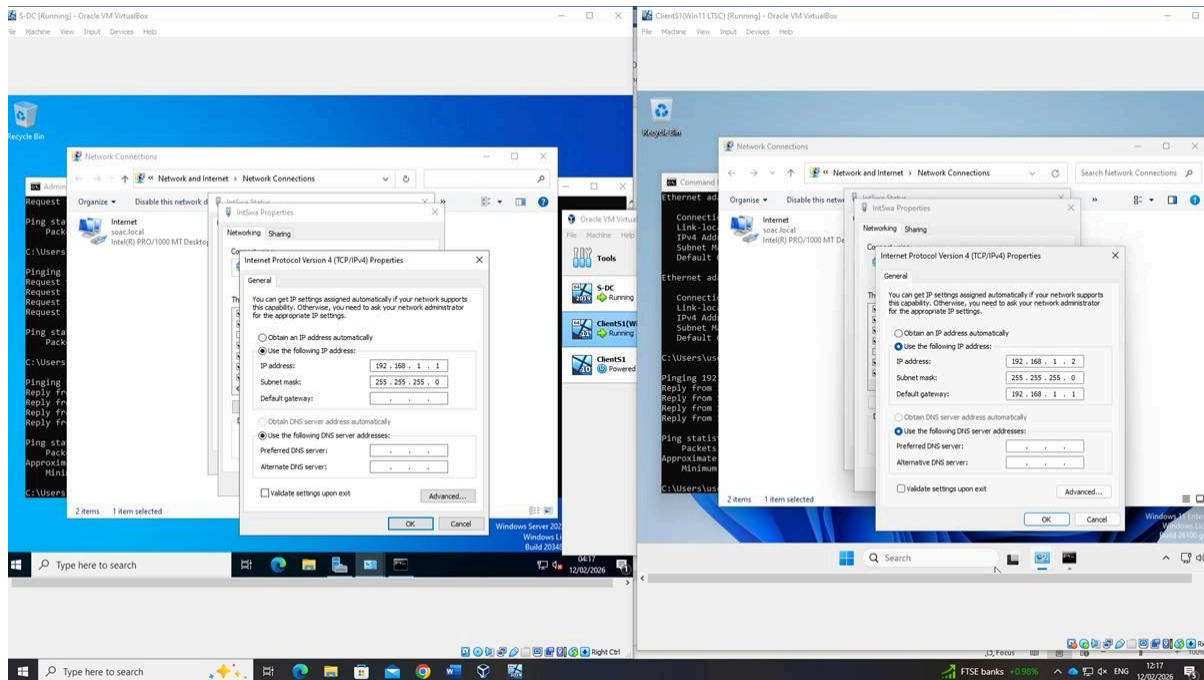




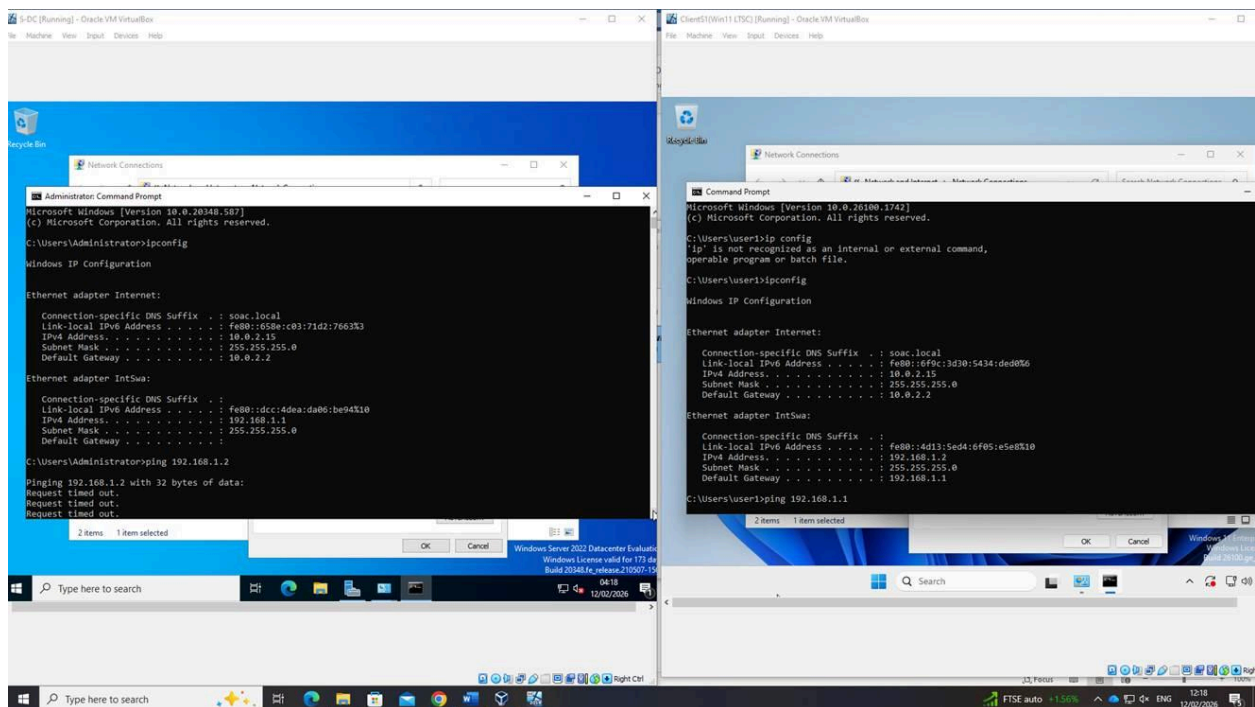
Creating the server machine, system name S-DC



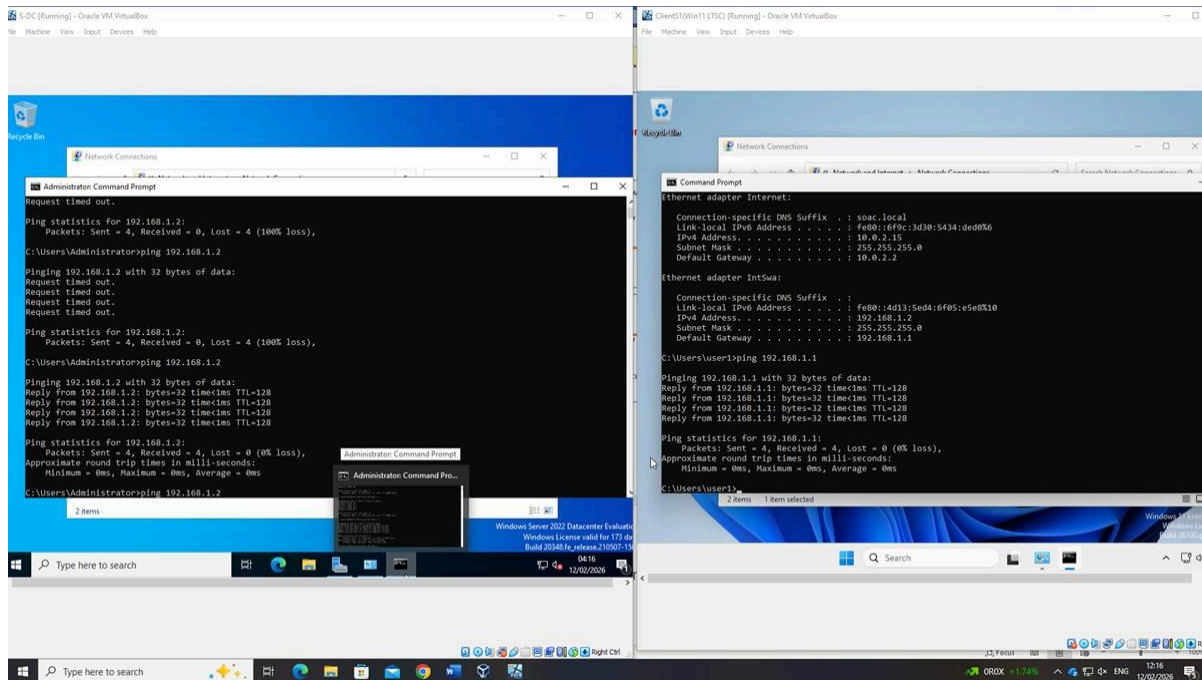
Client system name ClientS1



S-DC and the client were given IP addresses as specified in the topology.

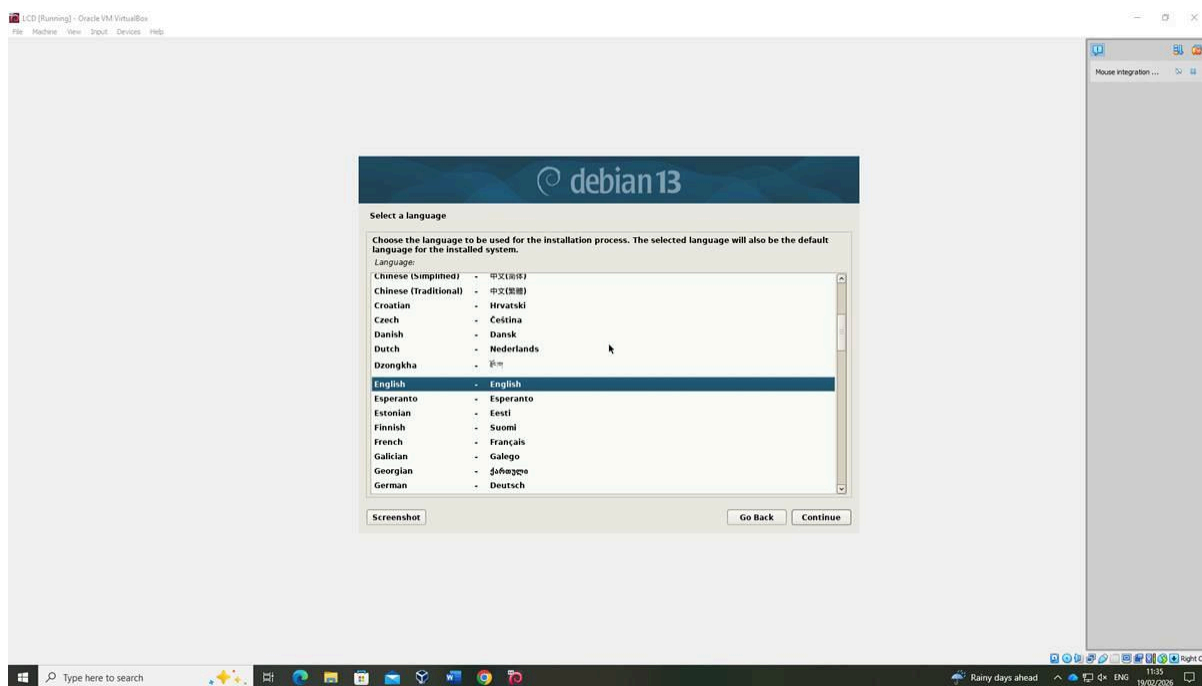


Ip config verifying both machines have the correct configuration.

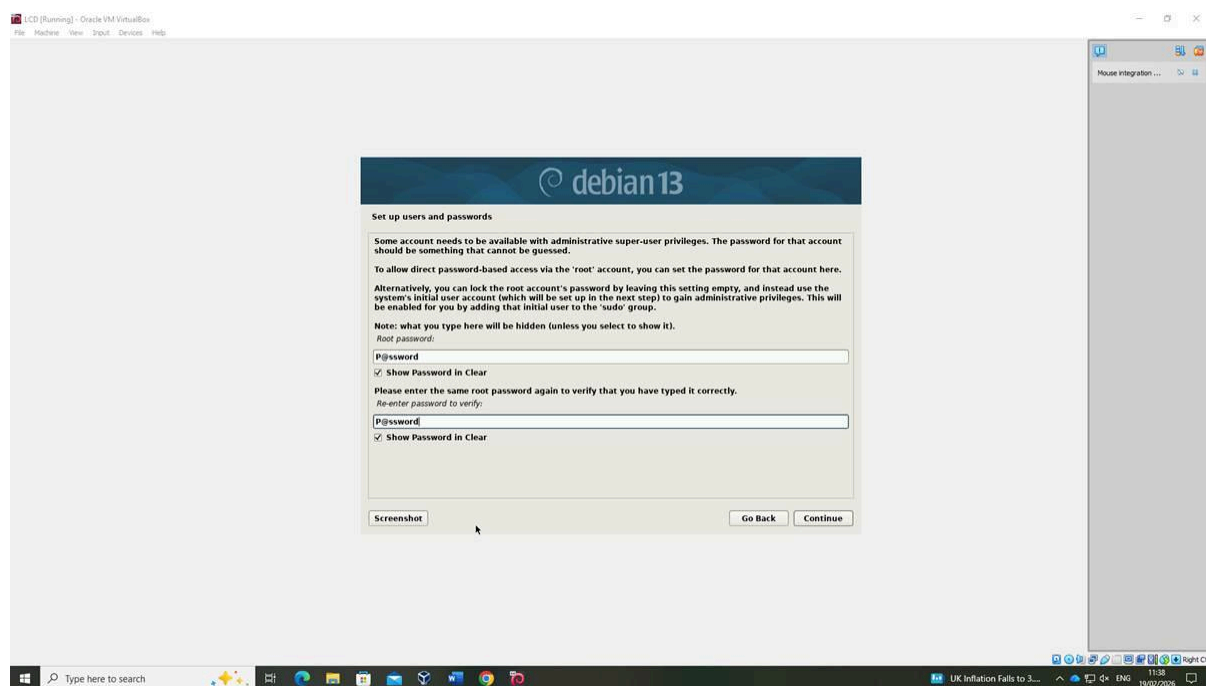
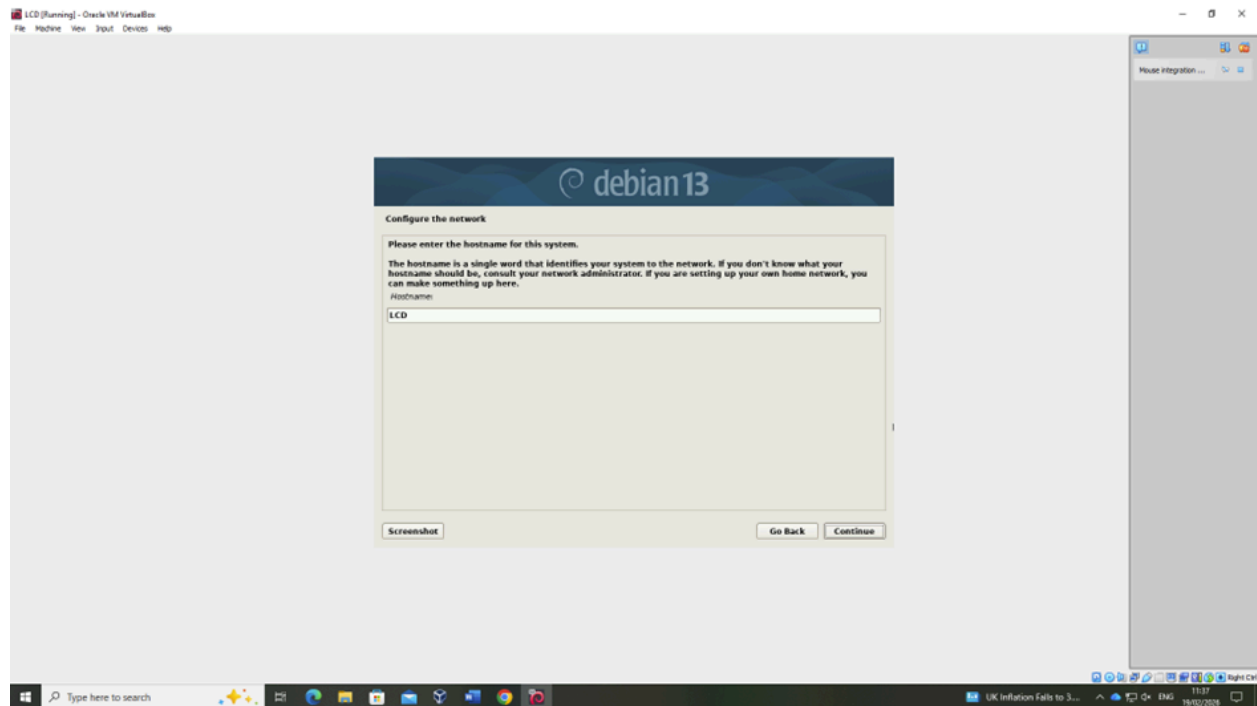


First successful ping between the Windows server and its client (the firewalls had to be disabled to get a successful ping).

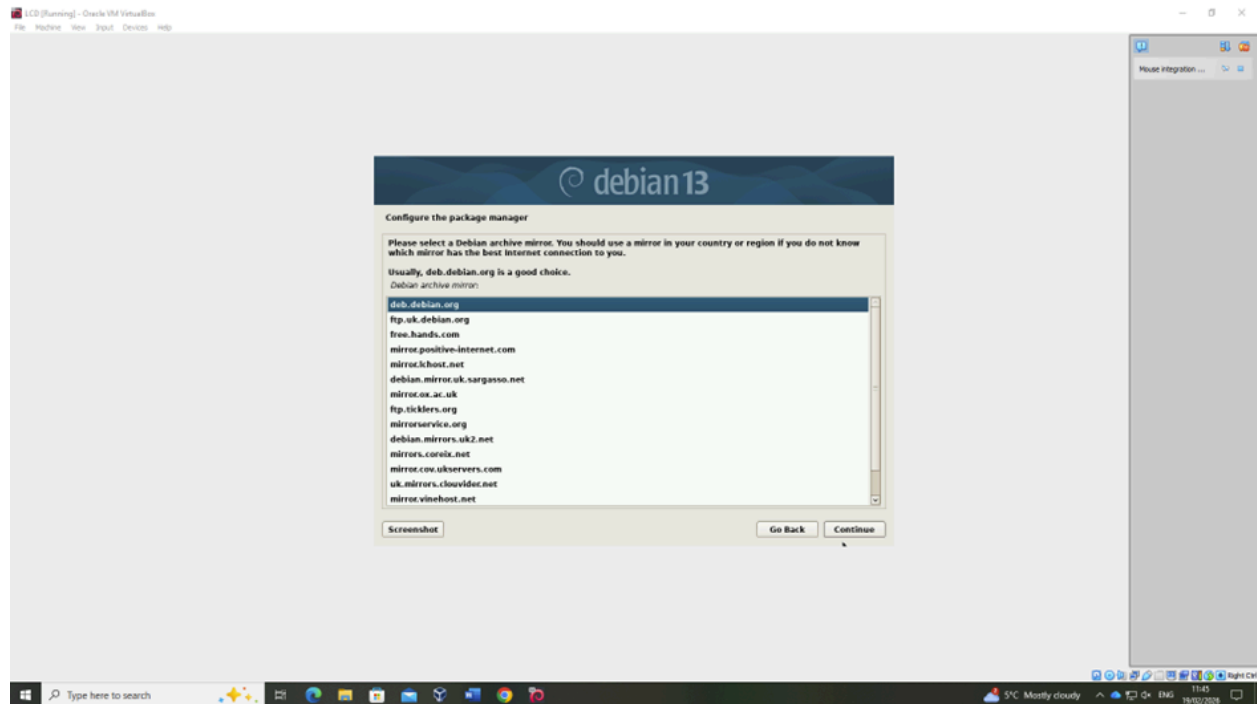
Setting up Linux Machines



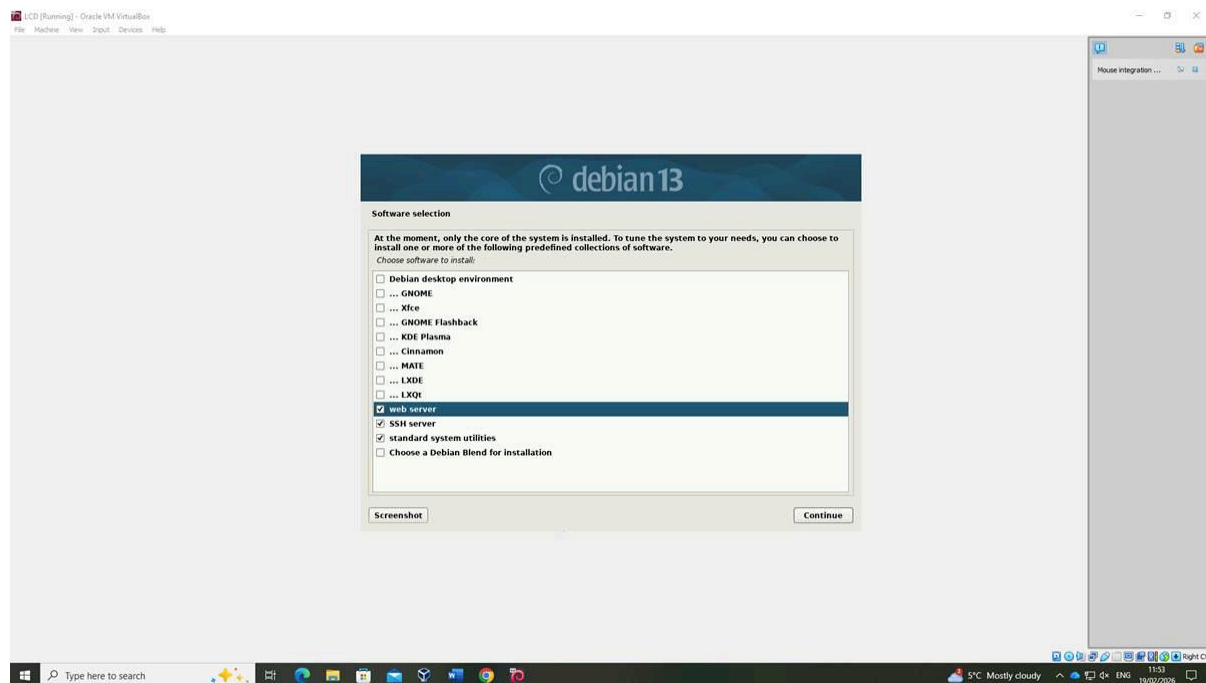
Graphical installation of the Linux server L-CD.



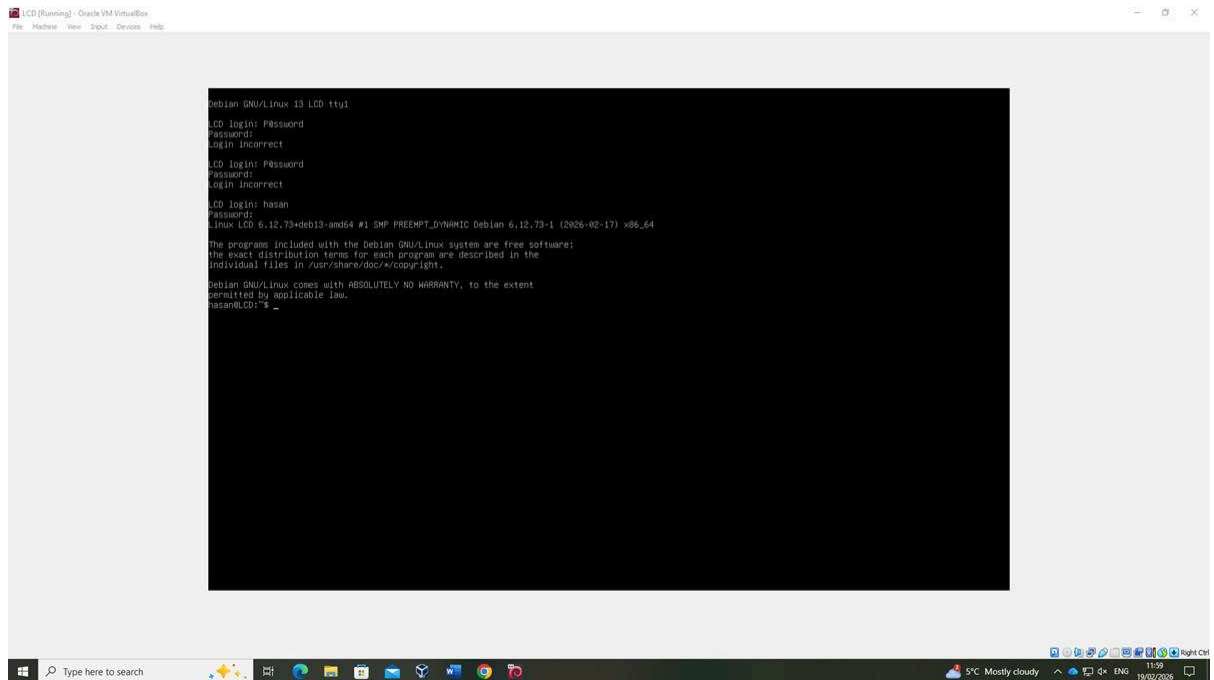
Using the same password as the Windows server (this would be bad practise in a real implementation but is more convenient for testing).



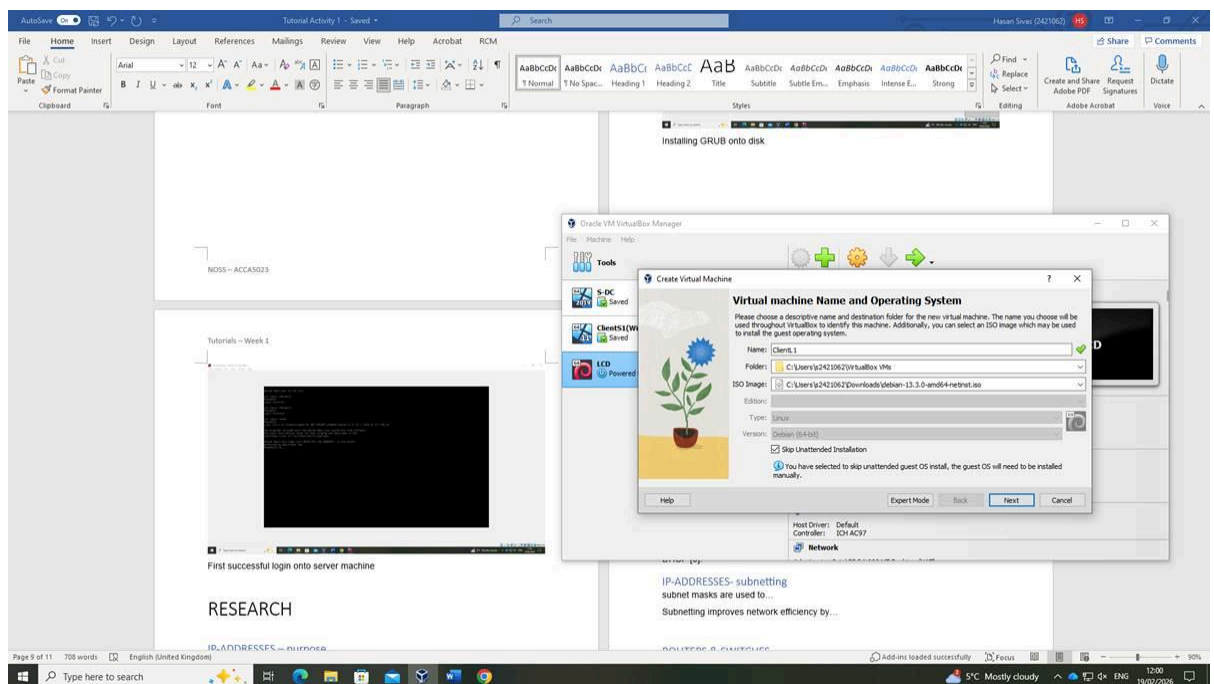
Selecting the provider from which the server gets its Debian installation from.



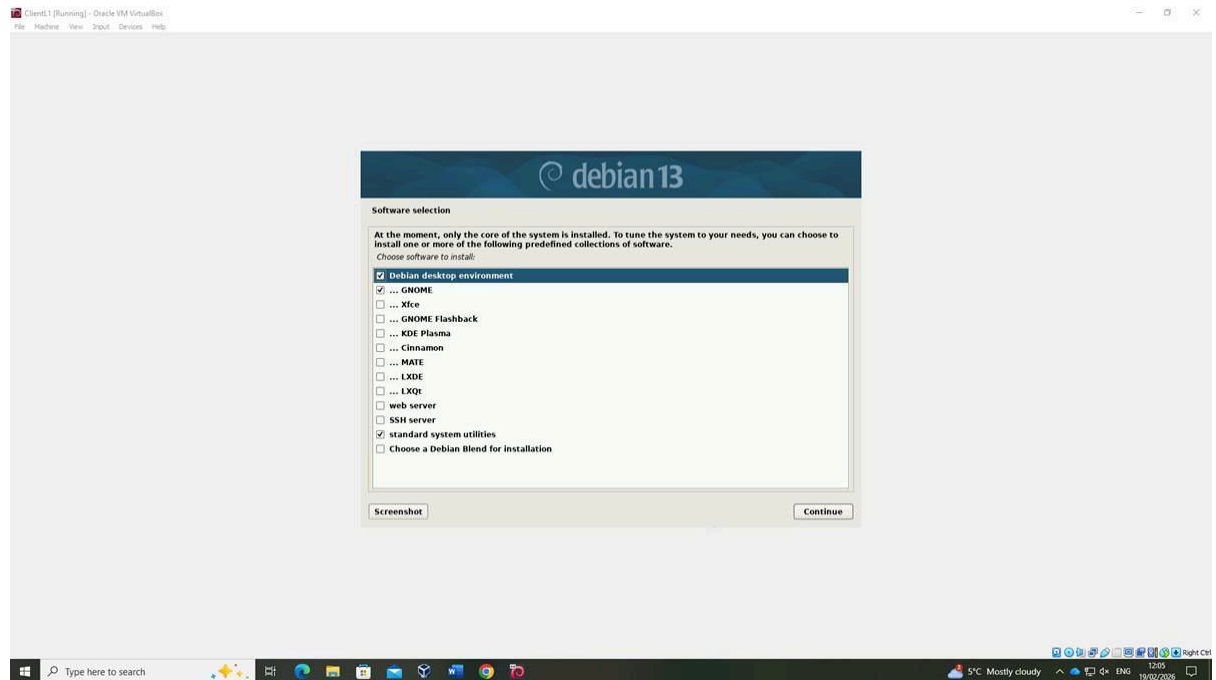
Installing the server as CLI-based to keep it lightweight.



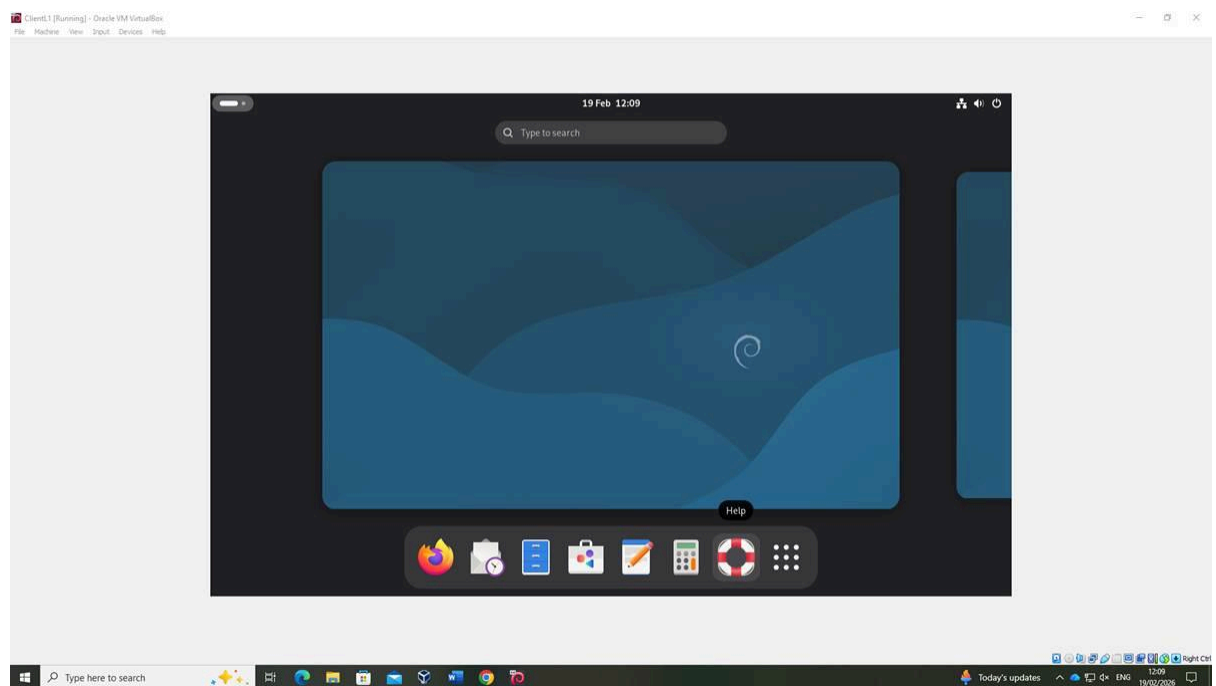
First successful login onto the server machine.



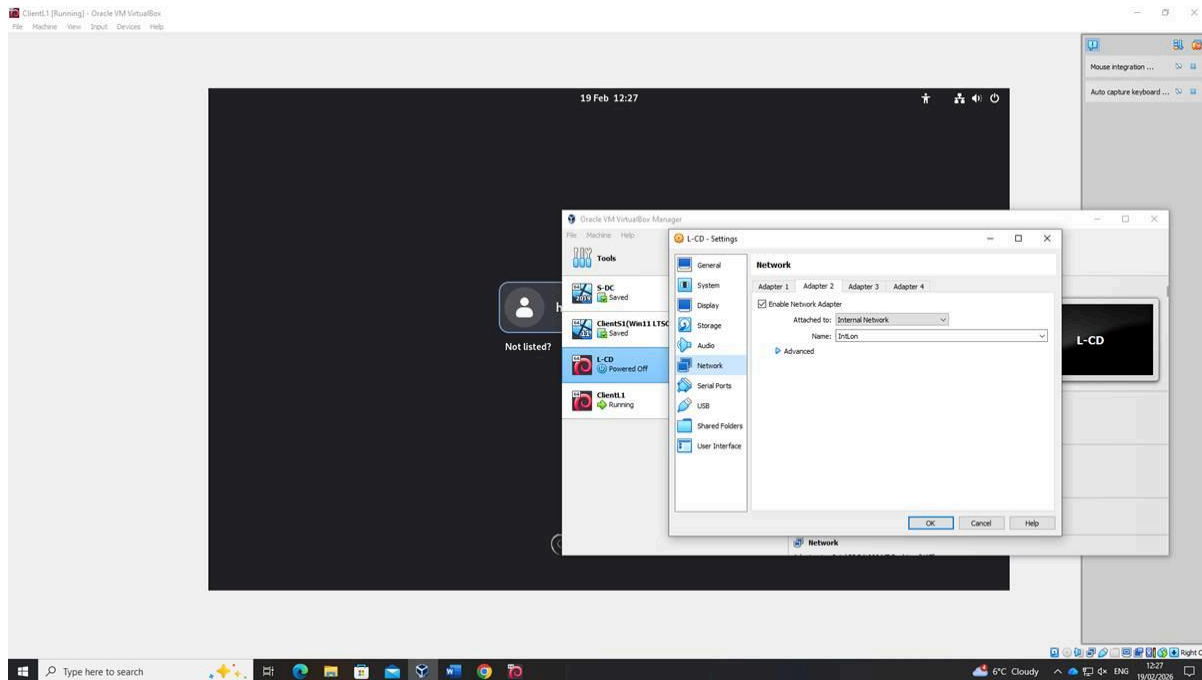
Beginning ClientL 1's installation.



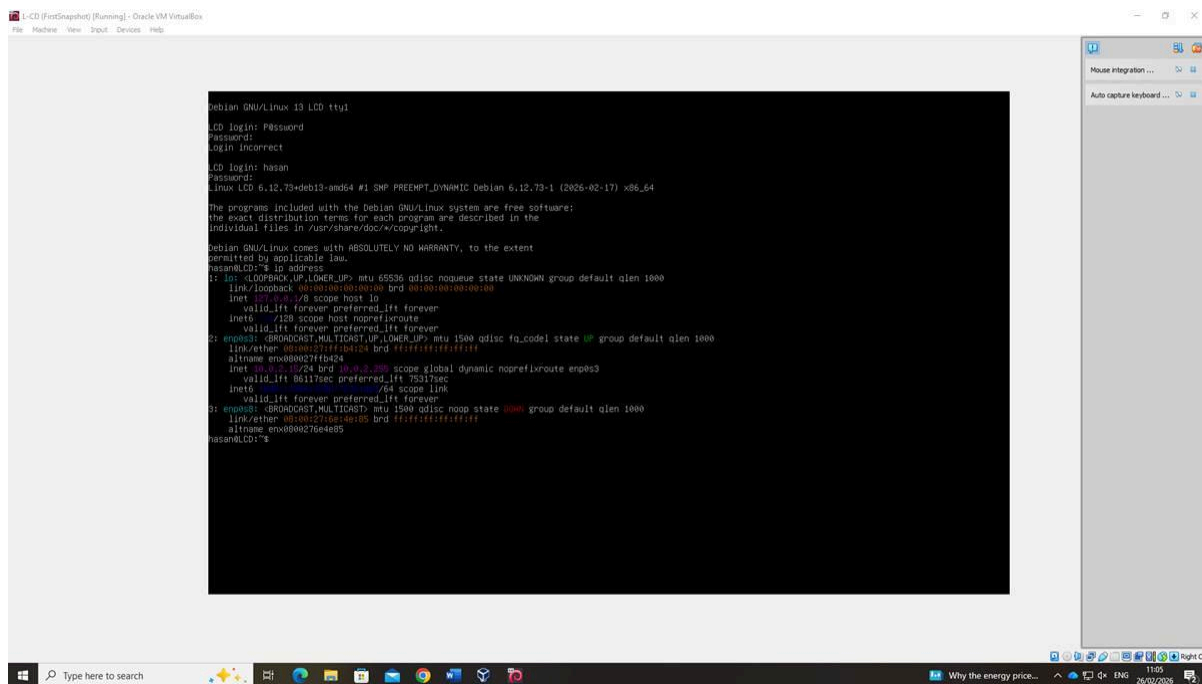
The main difference, unlike the server, the Client Machine requires a graphical user interface.

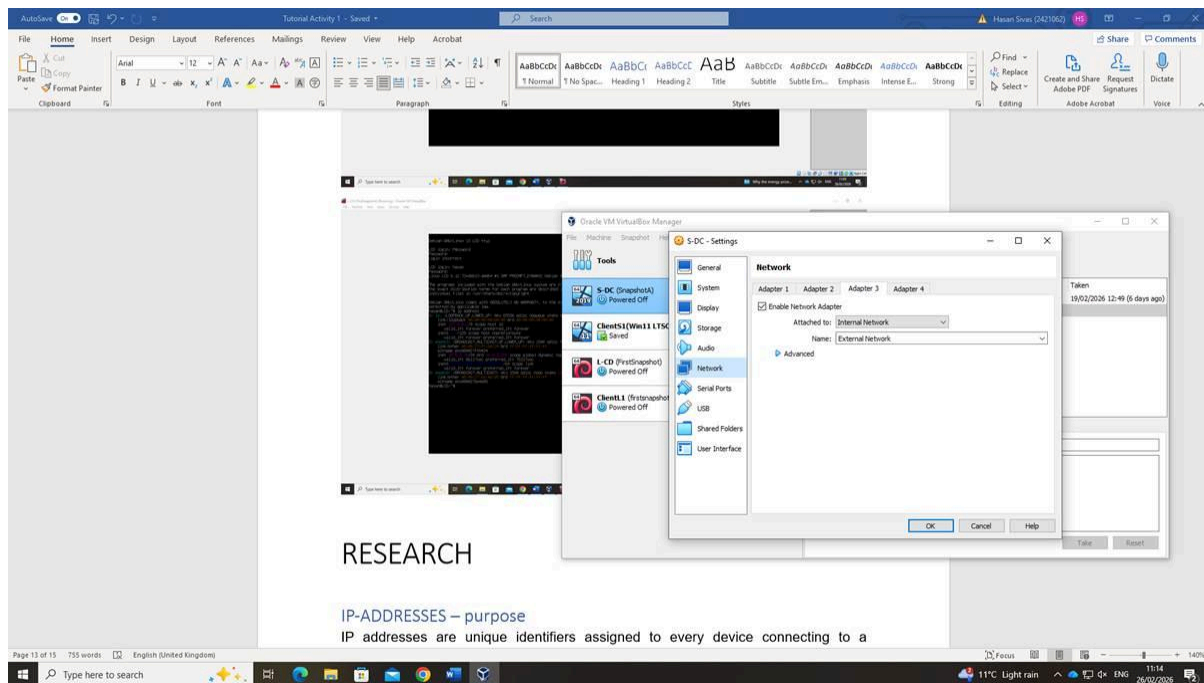
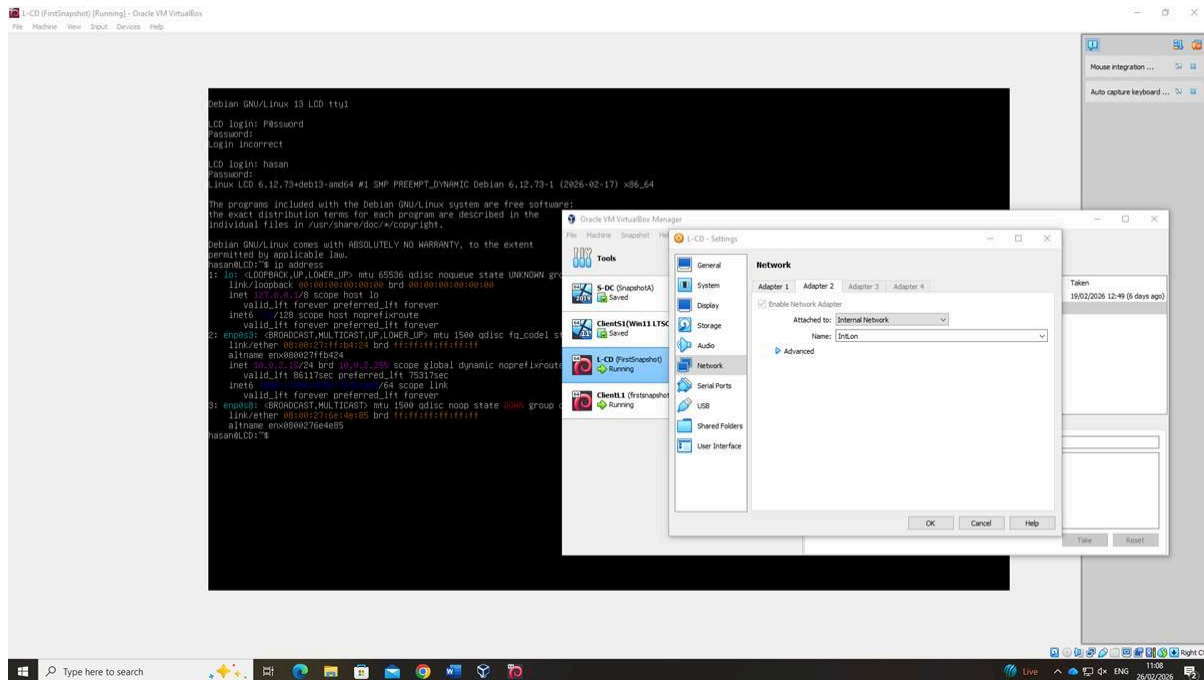


The client machine ClientL1 is successfully running.

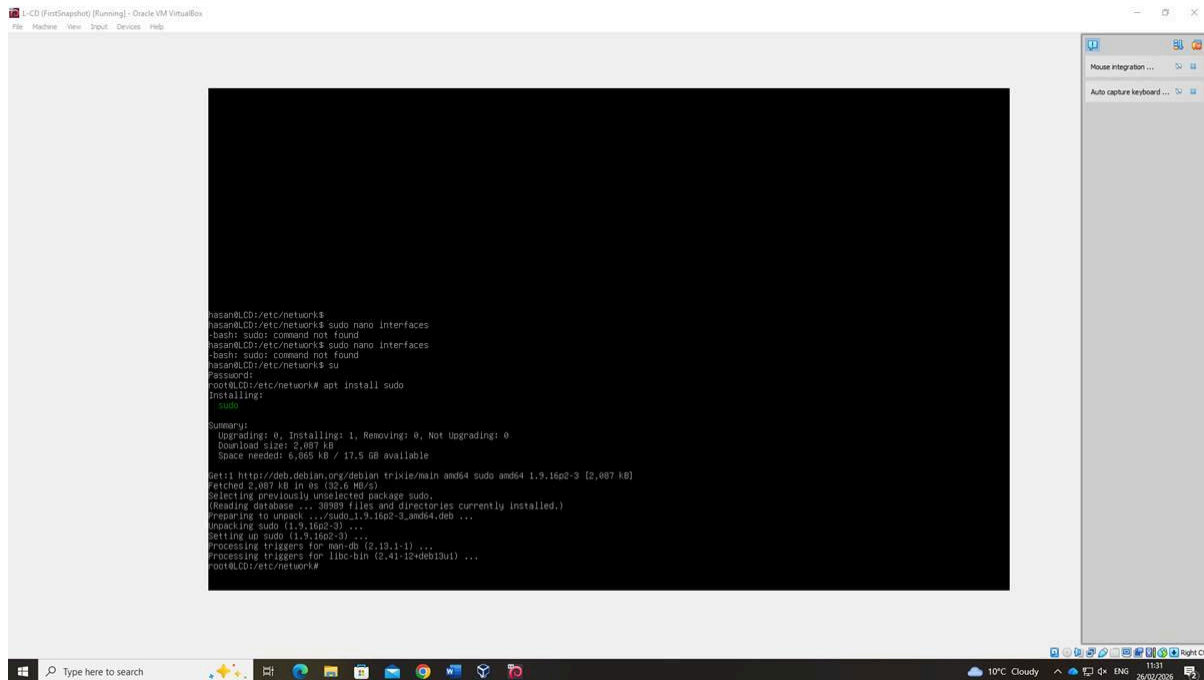


Creating the London internal network named 'IntLon'.





Creation of External Network (connection between L-DC and S-DC).

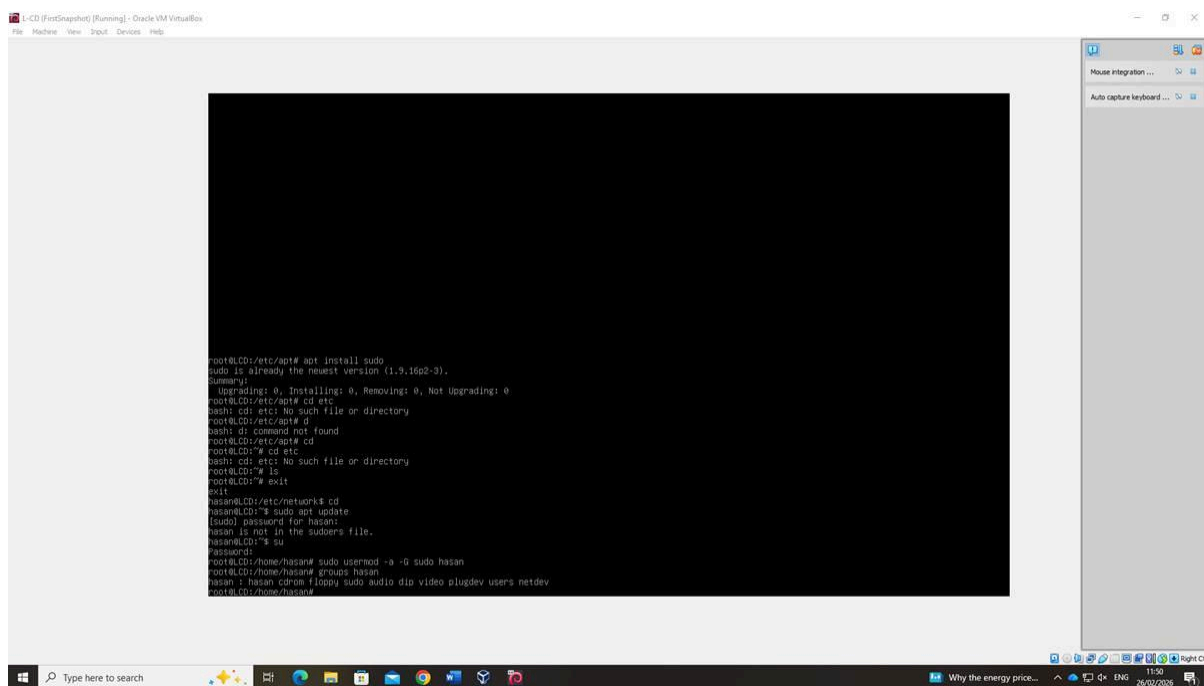


```
hasan@LCD:/etc/network$
hasan@LCD:/etc/network$ sudo nano interfaces
bash: sudo: command not found
hasan@LCD:/etc/network$ sudo nano interfaces
bash: sudo: command not found
hasan@LCD:/etc/network$ su
Password:
root@LCD:/etc/network# apt install sudo
Installing:
  sudo

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 0
  Download size: 2,007 kB
  Space needed: 6,065 kB / 17.5 GB available

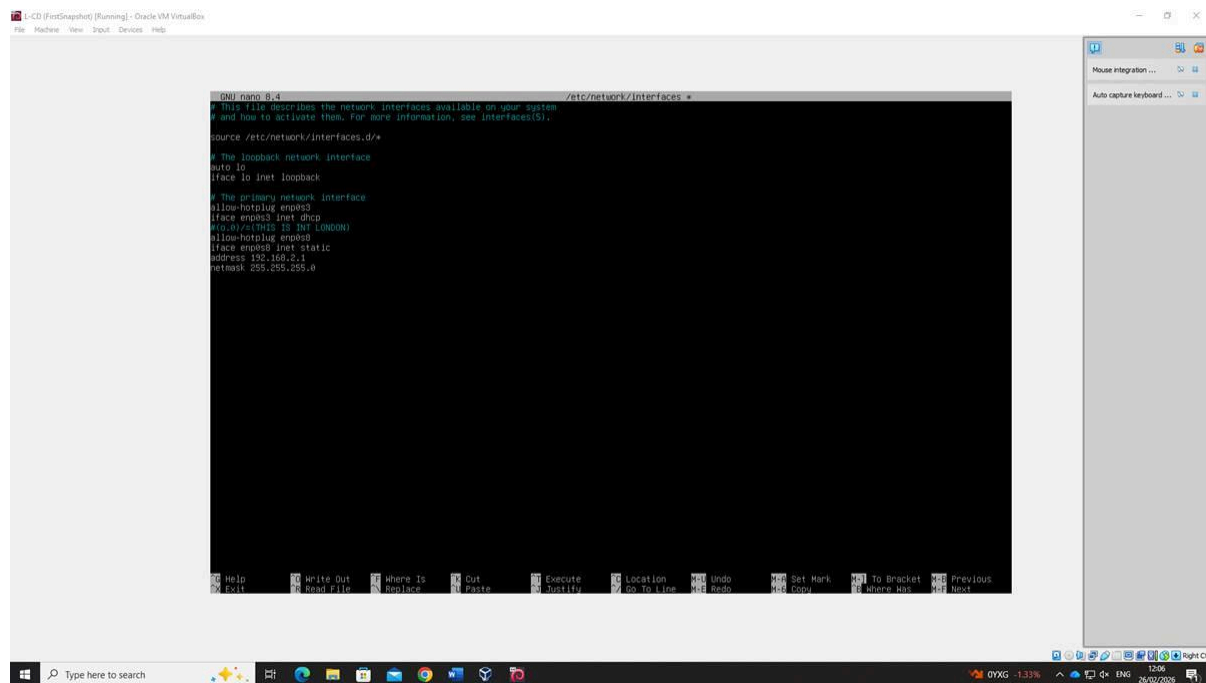
Get:1 http://deb.debian.org/debian trixie/main amd64 sudo amd64 1.9.16p2-3 [2,007 kB]
Fetched 2,007 kB in 0s (32.6 MB/s)
Selecting previously unselected package sudo.
(Reading database ... 38939 files and directories currently installed.)
Preparing to unpack .../sudo.1.9.16p2-3_amd64.deb ...
Unpacking sudo (1.9.16p2-3) ...
Setting up sudo (1.9.16p2-3) ...
Processing triggers for man-db (2.12.1-1) ...
Processing triggers for libc-bin (2.41-12deb1bu1) ...
root@LCD:/etc/network#
```

Installed sudo (necessary to edit the server inventory files).

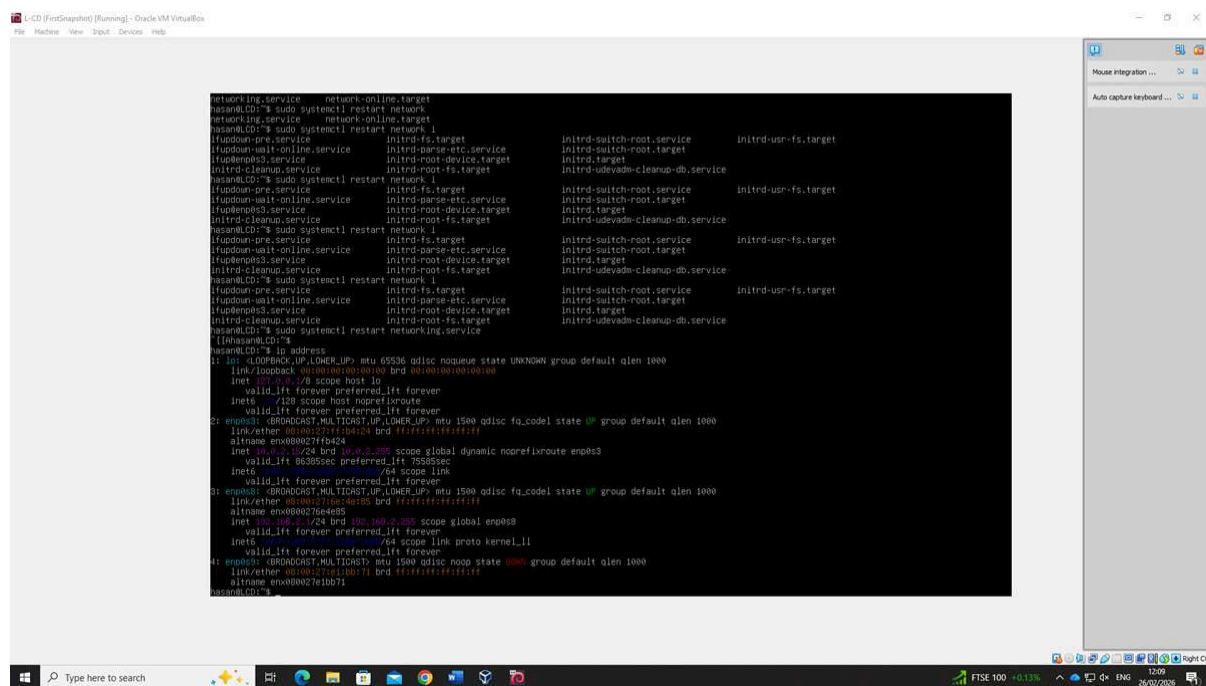


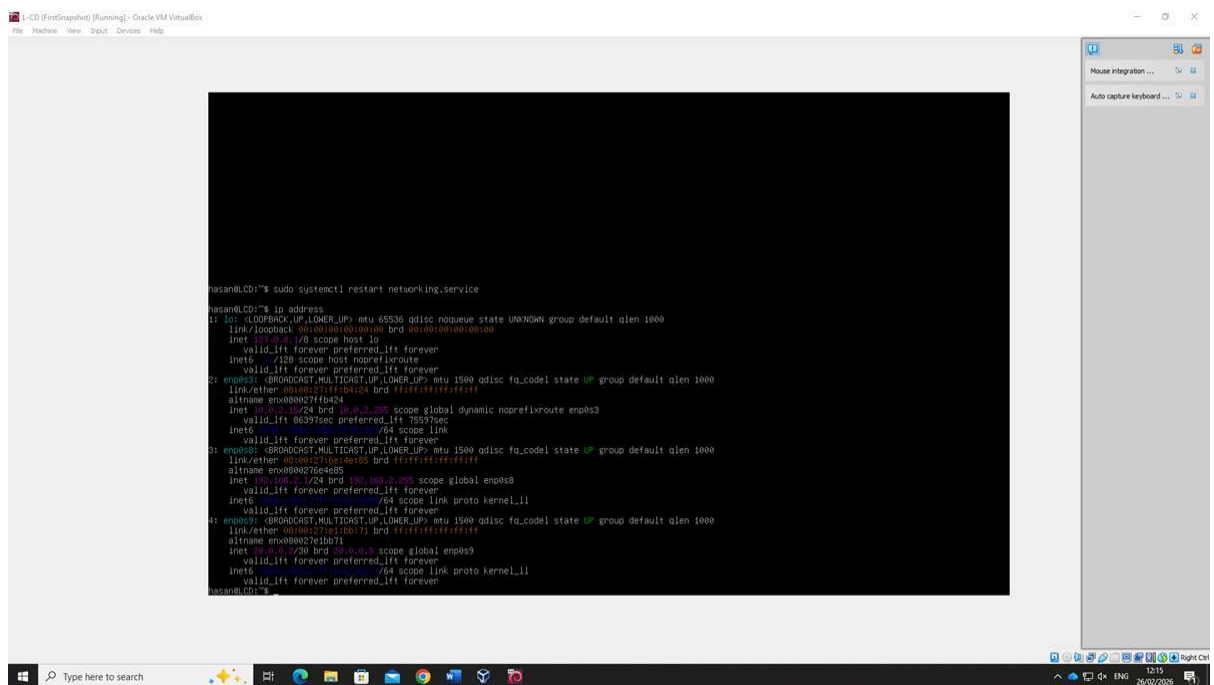
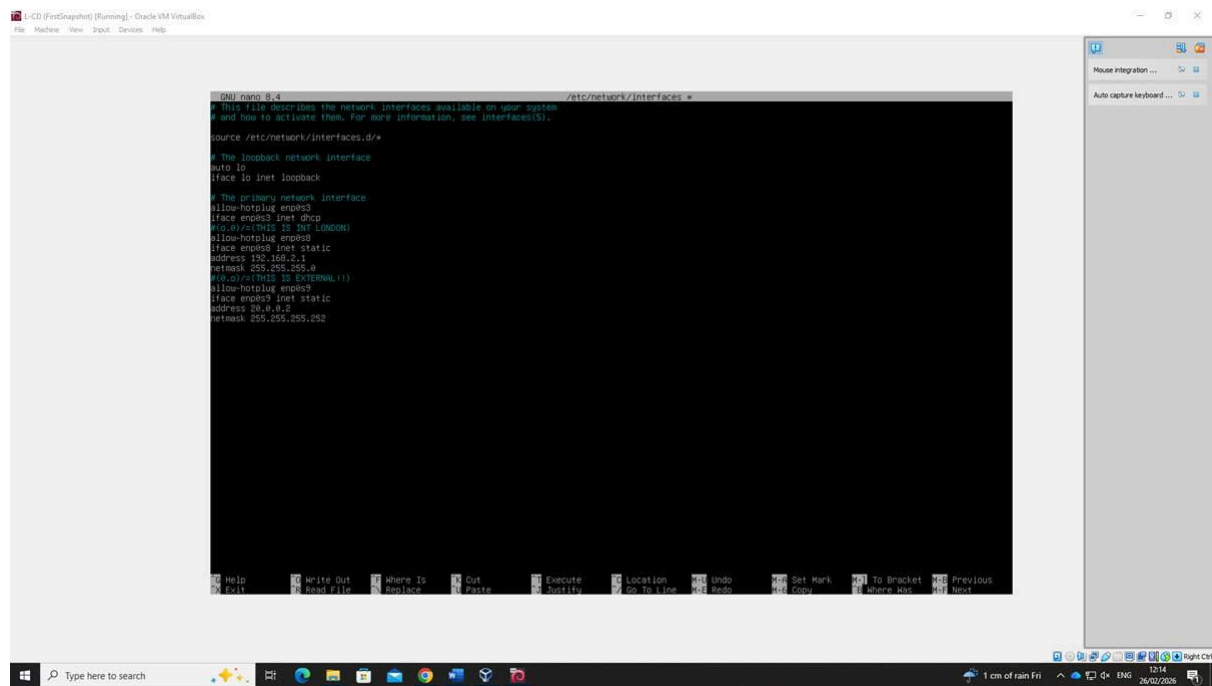
```
root@LCD:/etc/apt# apt install sudo
sudo is already the newest version (1.9.16p2-3).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
root@LCD:/etc/apt# cd etc
bash: cd: etc: No such file or directory
root@LCD:/etc/apt# cd
bash: cd: command not found
root@LCD:/etc/apt# cd
root@LCD:/etc/apt# cd etc
bash: cd: etc: No such file or directory
root@LCD:/etc# ls
root@LCD:/etc# exit
exit
hasan@LCD:/etc/network$ cd
hasan@LCD:~$ sudo apt update
[sudo] password for hasan:
hasan is not in the sudoers file.
hasan@LCD:~$ su
Password:
root@LCD:/home/hasan# sudo usermod -s /bin/bash hasan
root@LCD:/home/hasan# groups hasan
hasan : hasan cdrom floppy sudo audio dialout video plugdev users netdev
root@LCD:/home/hasan#
```

Adding 'hasan' to the sudo users group.

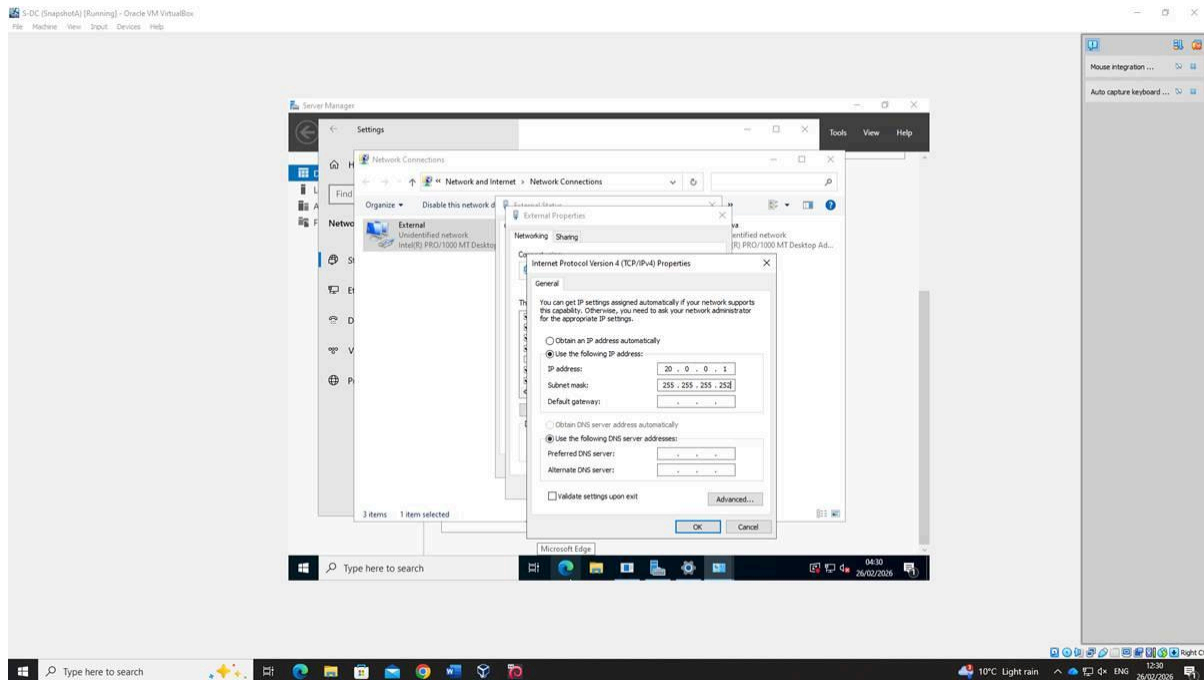


Configuring the internal network for the Linux server.

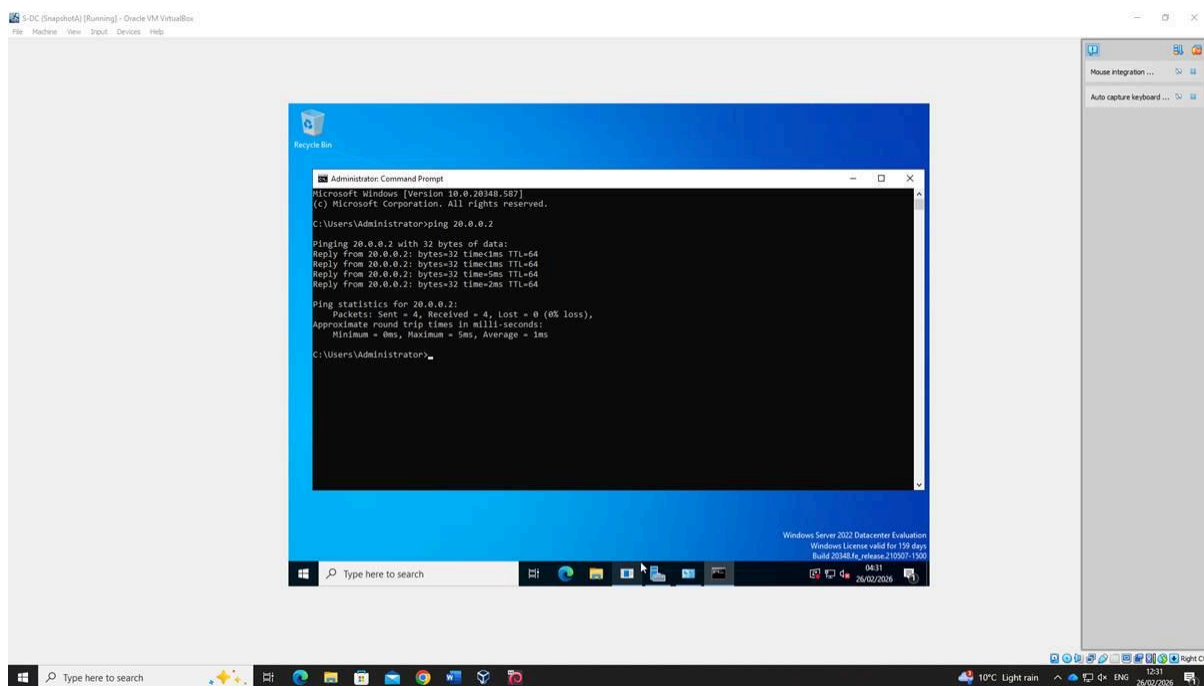


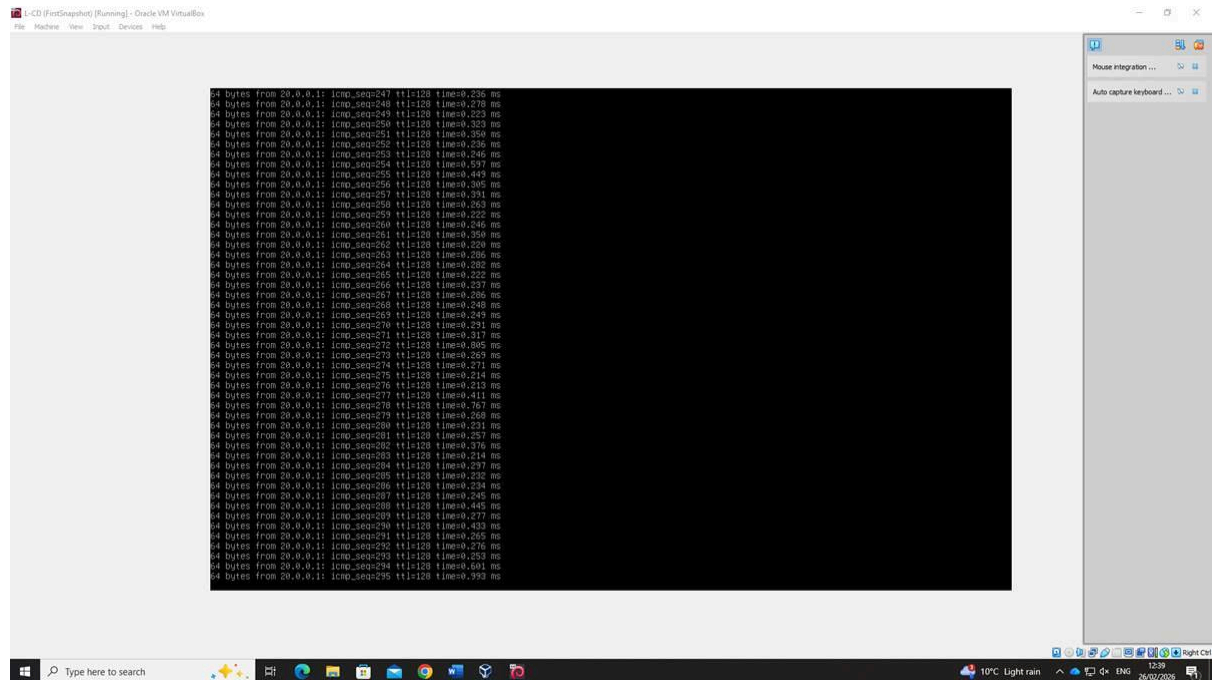


Configuring the external network for L-DC

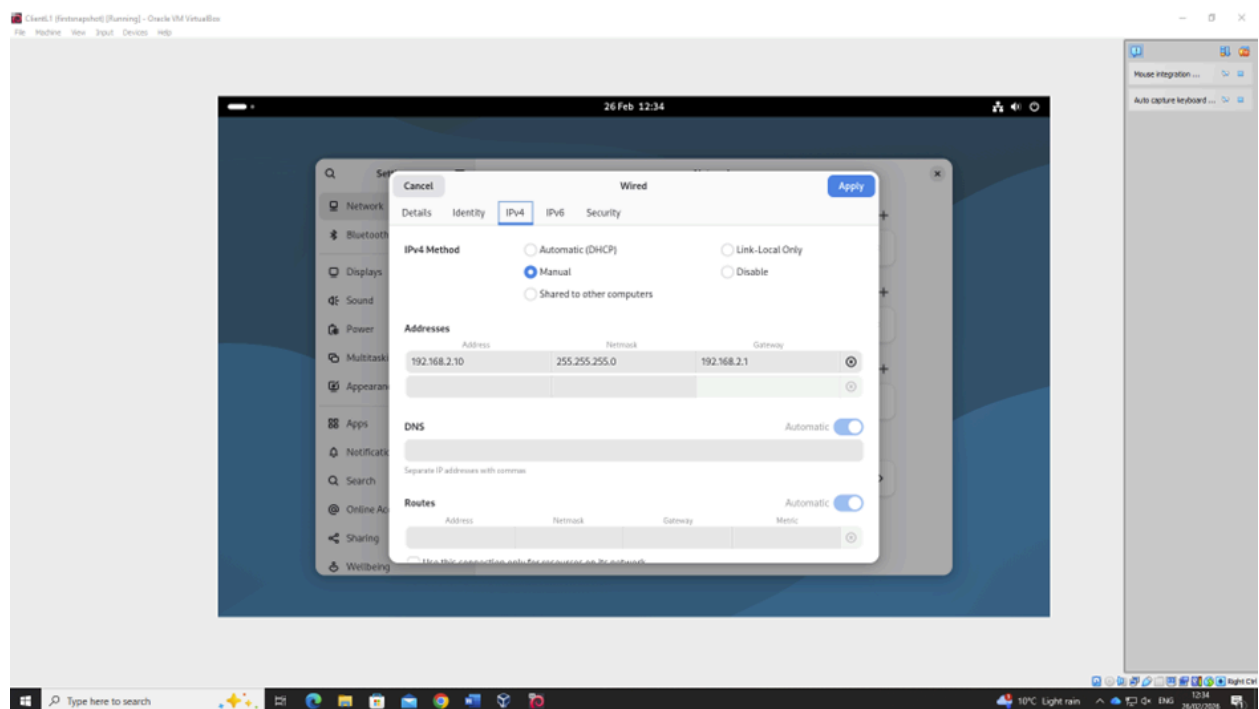


Going back and configuring the External network for the Windows server

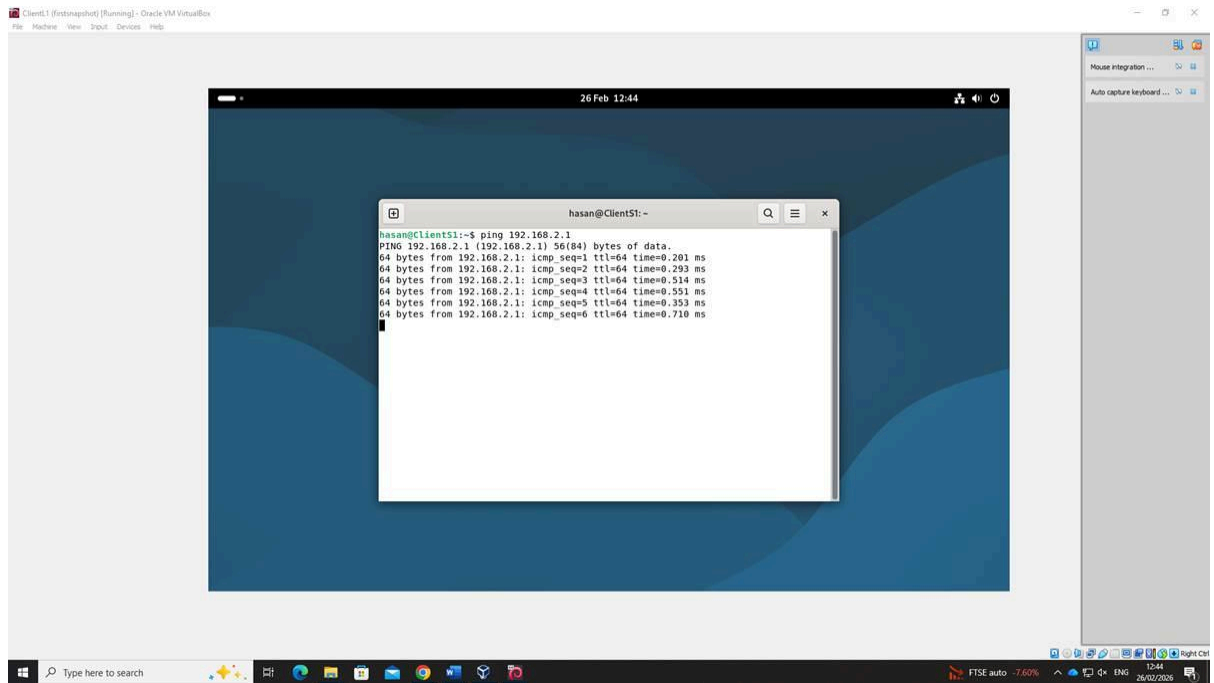
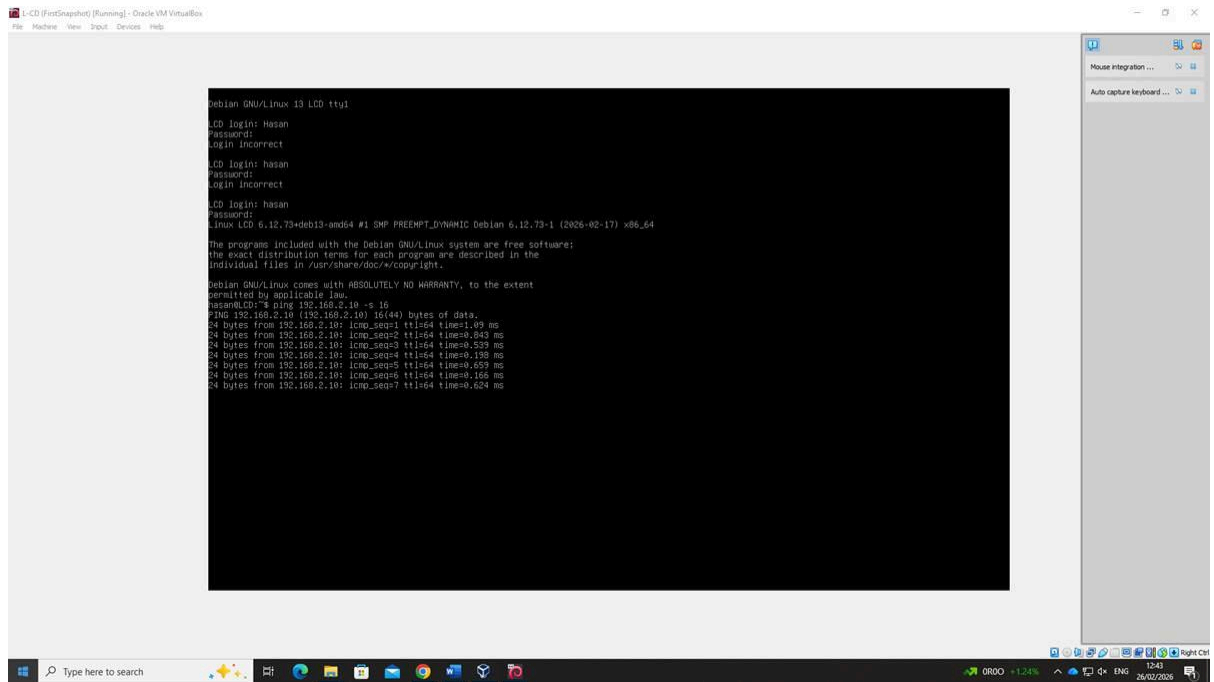




Successful pings between Windows Server (S-DC) and Linux Server (L-DC)



Configuring the Linux client's IP address.



Successful pings between L-DC and ClientL1.

Task 3 - Reflection

In this section, I will reflect on the process of setting up and configuring the prototype for NOSS2026's servers. The process began with obtaining the relevant image files from the official Windows website, which I then installed into Oracle VirtualBox, a type II hypervisor that would allow me to virtualise both the Windows-based Swansea network and Linux(Debian)-based London network. The machines were then all configured with IP addresses and connections according to the given topology. Connectivity was tested using pings between server and host machines as well as between the two servers..

The first difficulty I encountered during deployment was when configuring the internal networks for the Windows machines (S-DC & Client S1). Despite the IP Addresses being configured correctly, I was unable to conduct successful pings between the two. Likely because of one or both of the firewalls blocking ICMP(Internet Control Message Protocol) traffic. As a temporary measure, I disabled the firewalls on both devices, allowing them to ping with no issues.

Another issue I experienced was when I was configuring the ip addresses for the Linux server L-DC. As this was my first time installing Linux from scratch, I was unaware that superuserpermissions were needed to configure server files. The solution to this was to install Sudo and enable the account as a superuser.

Bibliography

- [1] Fortinet, "What is an IP Address? How it works? How to Locate it?," *Fortinet*.
<https://www.fortinet.com/uk/resources/cyberglossary/what-is-ip-address>
- [2] Cisco Netacad,
https://www.netacad.com/courses/networking-essentials?courseLang=en-US&instance_id=c03d7710-0e9a-4f47-b7b9-c57c838107f0
- [3] Amazon Web Services, "IPv4 vs IPv6 - Difference Between Internet Protocol Versions - AWS," *Amazon Web Services, Inc.*, 2024.
<https://aws.amazon.com/compare/the-difference-between-ipv4-and-ipv6/>
- [4] C. Okoh, W. N. Theophilus, P. Dawkins, and Sebamalai Paheerathan, "Enhancing Data Security through VLSM Subnetting and TCP/IP Model in an ENT," *Applied Sciences*, vol. 14, no. 23, pp. 10968–10968, Nov. 2024, doi:
<https://doi.org/10.3390/app142310968>.

- [5] T. Co., *Data Communications and Network Technologies*. Springer Nature, 2022.
- [6] Microsoft, “Active directory domain services overview,” *Microsoft*, Mar. 11, 2025.
<https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>
- [7] NCSC, “Virtual Private Networks (VPNs),” *www.ncsc.gov.uk*, Jun. 29, 2021.
<https://www.ncsc.gov.uk/collection/device-security-guidance/infrastructure/virtual-private-networks>
- [8] A. T. Azzam, R. Munadi, and R. Mayasari, “Performance Analysis Of Firewall As Virtualized Network Function On VMware ESXi Hypervisor,” *JURNAL INFOTEL*, vol. 11, no. 1, p. 29, Mar. 2019, doi: <https://doi.org/10.20895/infotel.v11i1.425>.
- [9] R.-C. Bucşa, “Teleworking and Securing Data with VPN Technology,” *Economy transdisciplinarity cognition journal*, vol. 23, no. 1, pp. 78–85, Jan. 2020.