



Armand OCTEAU – Student 2422790

Started on Friday 28 February 2025

NETWORK OPERATING SYSTEMS AND SERVICES (NOSS)

- Research, build, and deploy, a Virtualized network –

Summary

<i>TASK 1</i>	1
Introduction	2
Types of hypervisors	3
IP Addressing	4
Routing	6
Network Services	7
<i>TASK 2</i>	9
Choice of hypervisor	10
Installing Proxmox	10
Creating VMs and installing OSs	12
Building the typology	14
Configuring Swansea's LAN	16
Configuring Cardiff's LAN	20
<i>TASK 3</i>	24
<i>Resources</i>	26

TASK 1

- Research -

Introduction

What is Virtualization?

Virtualization is a rather old technology first brought by IBM in 1967. IBM aimed to run multiple operating systems simultaneously on one single mainframe to reduce cost and enhance efficiency. To do so, they developed virtualization technology. As in its name: virtualization allows the emulation of a physical environment within a virtual one, similar to a sandbox.

This technology can go from emulating entire operating systems, from storage devices or to an entire network infrastructure. To do so, virtualization involves a software called a hypervisor (which we will talk about later) that will aim to simulate hardware functionality.

The arrival of this technology gave us enhanced flexibility and efficiency for IT teams as it provided us a better utilisation of physical resources, easier maintenance and cost savings on hardware equipment.

For example, IT teams use virtualization to segment a physical dedicated rackserver into multiple virtual instances (using a type 1 hypervisor like Proxmox). This ensures that if one system fails such as the website's frontend- the backends, databases and API will continue to run on the same machine. This allows the team to use only a powerful server instead of multiple smaller ones.

Similarly, hosting companies will use virtualization to partition their servers for different clients.

Another example is in cybersecurity, where researchers use virtualization to emulate operating systems (e.g. Windows) to test and analyse viruses without risking damaging their host machine (they'll typically use a type 2 hypervisor like VMWare or Parallels).

Types of hypervisors

As talked above, the key component of virtualization is the hypervisor. This software is assimilated to a software layer which act as an agent between the hardware and the emulated software. It'll efficiently distribute the available hardware resources such as the CPU, the memory or storage among the host and the virtual machine (or multiples).

Currently, there are two types of hypervisors available on the market.

A type 1 Hypervisor, known as Bare-Metal supervisors, runs directly on the hardware machine called the kernel (like an OS). Its design provides high scalability (servers can operate as cluster) and security. That makes it ideal for data centers or production environments. This type requires a powerful hardware (with sometimes multiple CPUs), large RAM and SSD (for high-speed storage).

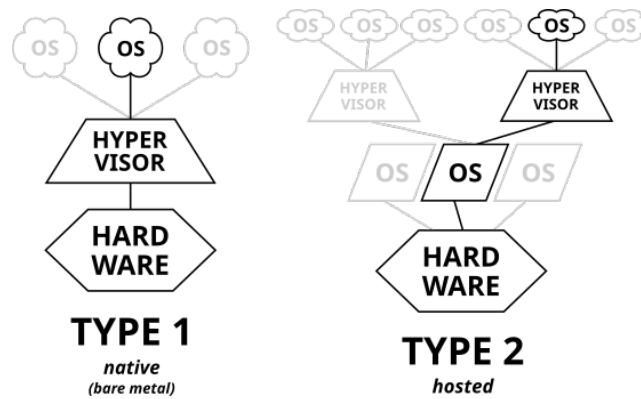
Examples of such hypervisor include Proxmox VE (previously mentioned), VMWare ESXi, Microsoft Hyper-V or Xen.

The advantages of a Type 1 hypervisor include direct hardware access that will ensure high performance for the virtualized machines, that will allow them to run smoothly as if they're natively running on the hardware (even if it's not). It gives an efficient resource management for large-scale deployments like hosting providers or others. VMs are also easily movable from one machine to another.

On the other hand, a Type 2 hypervisor will run on top of an existing operating system (e.g. Windows or Macintosh) like an application. This makes this type more user-friendly, easy to install, and suitable for software testing for developers, cybersecurity research or even personal use (like playing a Windows game on a Macintosh).

Examples will include VMWare Workstation (or Player for a free use), VirtualBox or Parallels Desktop. There's also QEMU (that is opensource) or UTM which is based on QEMU.

A Type 2 hypervisor will support a lot of others functionality that a Type 1 will not. For example, it can support snapshot features that allows the user to save and restore a VM state easily. It's less scalable than a Type 1 but very flexible for multiple OS testing (for app developers for example)



1- Diagram of a Type 1 and a Type 2 hypervisor

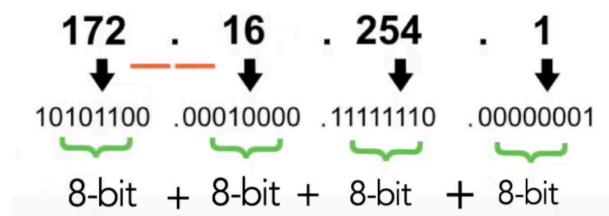
In fact, a Type 1 hypervisor is more like dividing a single machine into multiple independent instances, each acting as its own system with its own behaviours. Whereas a Type 2 hypervisor functions like a sandbox, with a virtual environment within an existing OS like a test subject in a lab.

IP Addressing

To send a letter or postcard to someone, we need their address to ensure it is delivered to them. For both computers to communicate with each other, the same principle applies with IP addresses (Internet Protocol). On the internet, all connected devices are assigned a unique IP address.

Nowadays, two types of IP addresses were normalized: IPv4 and IPv6. IPv4 was designed in the early 80's when the internet first emerged. Its structure allows up to **4.3** billions of unique addresses, which is obviously insufficient for today's needs, given that the global population exceeds 8 billions, with most individuals owning multiple devices.

IPv4 uses a 32-bit address format and is represented as four decimal numbers separated by dots (e.g., 192.168.1.1). Each decimal can go from 0 to 255, because each one is a 8-bit size ($2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 255$ in binary).



2 - Representation of an IPv4 address

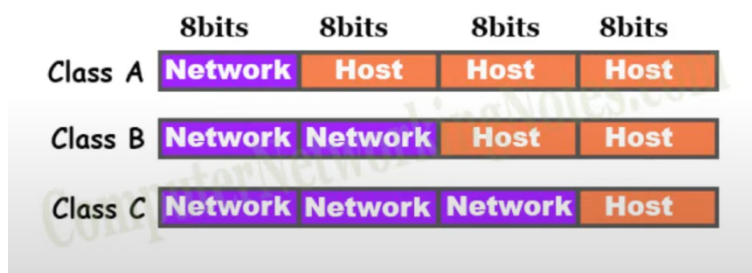
The IPv4 address consists of basically two parts: the network portion and the host portion. Each one is defined by a different class going from A to E.

Class	Starting Address	Ending Address	Purpose
A	0.0.0.0	127.255.255.255	Large networks (e.g. WAN as Wide Area Network)
B	128.0.0.0	191.255.255.255	Medium networks (e.g. MAN)
C	192.0.0.0	223.255.255.255	Smaller networks (e.g. LAN, PAN)
D	224.0.0.0	239.255.255.255	Multicast
E	240.0.0.0	255.255.255.255	Experimental

Those classes were created because of subnetting, and since IPv4 is limited, conservation is essential. Subnetting consists to divide a large network into smaller and more manageable sub-networks (also called subnets). This process helps for efficient IP address allocation (with static or dynamic addressing with a DHCP server).

A subnet mask is used to determine how many bits are allocated for the network and host portions in an IPv4 IP. Subnetting notation follows the format /X with X the numbers of bits assigned for the network. For example, /24 means 24 bits for the network, /16 16 bits and so on. In its numerical form, the subnet mask can be noted as followed: for 24 bits, 255.255.255.0.

That's how IP address classes were designed, with each class having a predefined subnet mask as shown in the following diagram;



Class A addresses have a /8 subnet mask (8 bits), Class B a /16 and class C a /24.

3 - Diagram showing Network / Host repartitions in IPv4 classes

However, since IPv4 addressing is running out of available addresses, the IETF (Internet Engineering Task Force) developed the IP version 6 addressing format in december 1995 that uses a 128-bit address format.

Its represented as eight groups of hexadecimal (HEX) numbers (ranging 1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G). It is also possible to abbreviate its format to have a much readable address with for example 4 zeros being abbreviated from :0000: to ::.



4 - Representation of an IPv6 address

This new format will, at term, replace the old IPv4 format because of its capacity of being able to deliver $3,4 * 10^{38}$ unique addresses.

Routing

Routing is defined by the process of sending packets from one host to another located in another network. This process is usually exerted by a router, which is a device that connects two or more networks together.

It can be compared as a postal delivery system where letters are data packets. Routers act in a way like postal workers that ensure each letter reaches the correct IP address using the best possible route. They exchange information between each other's to update what's called "routing tables" and to determine the best path for the packets based on metrics like bandwidth and latency (also called ping, measured in milliseconds). Routing is here to automate path selection and congesting, all while optimizing traffic flow.

Nowadays, it exists two types of domestic router: a wired one like the Cisco 1941 and a wireless one like the Livebox 5 from Orange (France Telecom) or the Cisco 880 ISR. Beyond domestic use, we can also talk about edge routers, which serve as an intermediary between local networks and core routers. Unlike edge routers, they do not interact directly with a LAN or WAN network but instead form the backbone of the internet infrastructure. They are primarily used by service providers to interconnect all their edge routers.

It exists two types of routing: static and dynamic. Static routing concerned manually configured routes that do not change unless they are modified. They're used in small network, IoT (internet of things) or secure environments (like government infrastructure). Static routes are not scalable and requires manual updates when the topology changes. But they are indeed more secure and predictable while having no bandwidth consumption for updates (route updating). On the other hand, dynamic routing is when routers automatically adjust routes based on the network changes (e.g., when a router is disconnected or when there is outages or link failures).

Two types of routing protocols exist: the Interior Gateway Protocols (IGP) used within an organization and the Exterior Gateway Protocols (EGP) used for global internet routing.

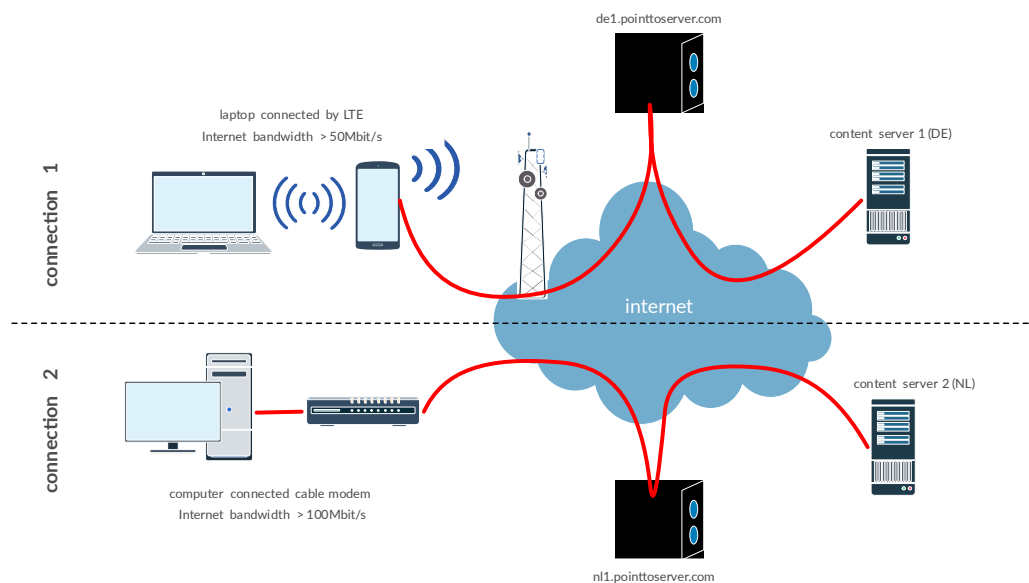
	Protocol	Description
IGP	RIP (Routing Information Protocol)	<i>Simple protocol that uses hop count but that is limited to 15.</i>
	OSPF (Open Shortest Path First)	<i>Protocol that uses link-state for precise routing. They analyse the entire network topology rather than just counting hops.</i>
	EIGRP (Enhanced Interior Gateway Routing Protocol)	<i>This protocol is Cisco proprietary. It's a hybrid approach between the two above.</i>
EGP	BGP (Border Gateway Protocol)	<i>It's the Internet backbone protocol used by ISPs (Internet Service Provider) and cloud providers.</i>

For example, large hospitals use OSPF for dynamic routing when managing multiple data centers. A large hospital may operate more than three data centers if necessary. Each data center

has typically a pair of core routers connected via dark fiber links that are then linked to edge routers located in the others hospital facilities.¹

Network Services

Like in every new infrastructure, services are provided for clients to use. For example, the most used and well-known is a VPN, standing for Virtual Private Network. While it's commonly known for helping us access the U.S. Netflix catalogue, VPNs also allow the creation of encrypted tunnels over a public network and are commonly used to protect devices on public hotspots or to connect to a LAN from external locations. VPNs use protocols like IPsec or SSL/TLS, which ensure secure data transmission.



5 - Diagram of a VPN Network

Another widely used network service is the Dynamic Host Configuration Protocol (DHCP) which was created to overcome the limitations of IPv4 (since there's only 4.3 billion IP address in IPv4, it needed to be efficiently managed). A DHCP server is present on every domestic router and functions to dynamically assign IP addresses to newly connected devices on a network.

On Windows Server, Active Directory Domain Services (AD DS) exists as a Microsoft-developed services. It was created to manage domains, users, computers, and securities policies in a Windows Network.

To explain how it works, let's take the example of UWTSD. On the local network, for a student to log in to a computer, they need to enter their credentials (e.g.,

¹ Answer of jeremiahfelt, "Real world scenerio of ospf use" *Reddit*. Accessed: Jan. 31, 2025. [Online]. Available: https://www.reddit.com/r/ccnp/comments/i6ils8/real_world_scenerio_of_ospf_use/

2422790@student.uwtsd.ac.uk). Their login request is sent to an AD DS Domain Controller, which verifies their username and password. Once access is granted, the student gains access to their files, university applications, and shared drives based on his Active Directory permissions.

Certificate Authorities (CA) are trusted organizations that issues and manages digital certificates to verify the identity of websites or other. We can view it like the blue check mark for certified accounts on social medias. Without these certificates, the website's server can't enable a secure or encrypted communication with SSL/TLS over the internet. The client's web browser will usually block communication if enabled without a certificate. Well-known CAs are for example DigiCert or Let's Encrypt.

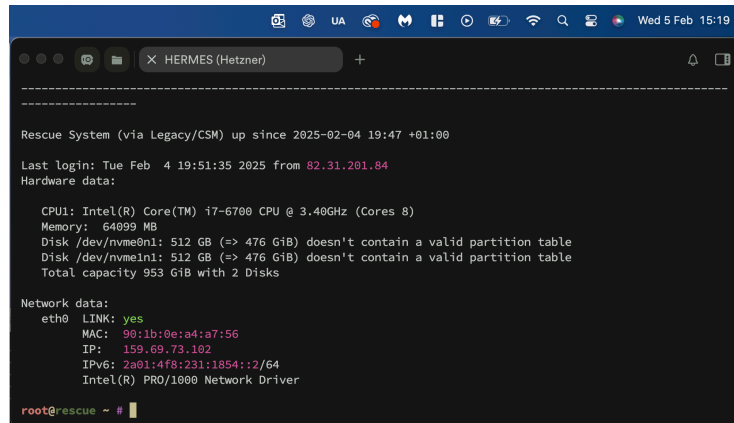
TASK 2

- Report -

Choice of hypervisor

Proxmox, a type 1 hypervisor, was chosen for numerous reasons. Firstly, one already had previous experience with Proxmox, VMs, and servers since one already has a server at his home for production and development purposes. Since one is also working on an M4 MacBook Pro, certain OS versions like Windows Server 2022 were not available for this CPU architecture (arm64 instead of amd64). Furthermore, servers were available at a fair price on Hetzner, a Dutch hosting company that was recommended by one's friend.

Installing Proxmox



```

Rescue System (via Legacy/CSM) up since 2025-02-04 19:47 +01:00

Last login: Tue Feb  4 19:51:35 2025 from 82.31.201.84
Hardware data:

CPU1: Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz (Cores 8)
Memory: 64000 MB
Disk /dev/nvme0n1: 512 GB (=> 476 GiB) doesn't contain a valid partition table
Disk /dev/nvme1n1: 512 GB (=> 476 GiB) doesn't contain a valid partition table
Total capacity 953 GiB with 2 Disks

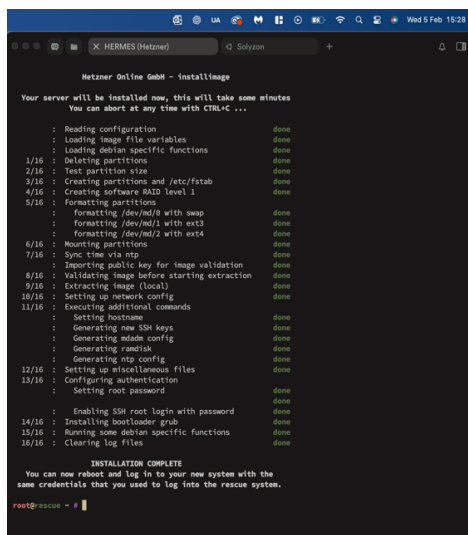
Network data:
eth0  LINK: yes
      MAC: 90:1b:0e:a4:a7:56
      IP:  159.69.73.102
      IPv6: 2a01:4f8:231:1854::2/64
      Intel(R) PRO/1000 Network Driver

root@rescue ~ #

```

6 - Screenshot of the rescue OS of the server

At start, the server was booted on rescue mode without any OS. One proceeded to install Debian 12 which is the base OS where Proxmox is built on. Following this guide², Proxmox was then installed. One chose to name the server 'Hermes,' since Hermes is the mythological god of communication and travelers, and the project is about building an entire virtual network.



```

Hetzner Online GmbH - Install/Upgrade

Your server will be installed now, this will take some minutes
You can abort at any time with CTRL+C ...

: Reading configuration                               done
: Loading image file variables                         done
: Loading debian specific functions                    done
1/16 : Deleting partitions                             done
2/16 : Test partition size                             done
3/16 : Creating partitions and /etc/fstab               done
4/16 : Creating software RAID level 1                  done
5/16 : Formatting partitions                           done
: Formatting /dev/md/0 with swap                       done
: Formatting /dev/md/1 with ext3                       done
: Formatting /dev/md/2 with ext4                       done
6/16 : Mounting partitions                             done
7/16 : Sync time via ntp                               done
: Importing public key for image validation            done
8/16 : Validating image before starting extraction     done
9/16 : Extracting image (local)                       done
10/16 : Setting up network config                      done
11/16 : Executing additional commands                  done
: Setting hostname                                    done
: Generating new SSH keys                             done
: Generating sshd config                              done
: Generating rsyslog config                           done
: Generating ntp config                               done
12/16 : Setting up miscellaneous files                 done
13/16 : Configuring authentication                     done
: Setting root password                              done
: Enabling SSH root login with password               done
14/16 : Installing bootloader grub                    done
15/16 : Running some debian specific functions        done
16/16 : Clearing log files                             done

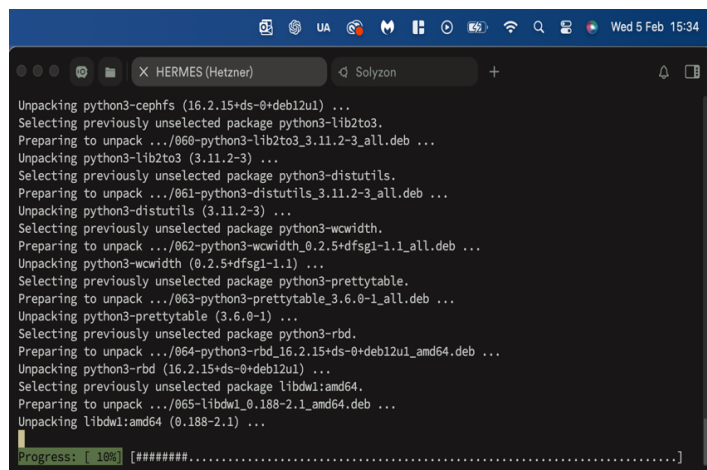
INSTALLATION COMPLETE

You can now reboot and log in to your new system with the
same credentials that you used to log into the rescue system.

root@rescue ~ #

```

7 - Screenshot of the Debian 12 installation



```

Unpacking python3-cephfs (16.2.15+ds-0+deb12u1) ...
Selecting previously unselected package python3-lib2to3.
Preparing to unpack .../060-python3-lib2to3_3.11.2-3_all.deb ...
Unpacking python3-lib2to3 (3.11.2-3) ...
Selecting previously unselected package python3-distutils.
Preparing to unpack .../061-python3-distutils_3.11.2-3_all.deb ...
Unpacking python3-distutils (3.11.2-3) ...
Selecting previously unselected package python3-wcwidth.
Preparing to unpack .../062-python3-wcwidth_0.2.5+dfsg1-1.1_all.deb ...
Unpacking python3-wcwidth (0.2.5+dfsg1-1.1) ...
Selecting previously unselected package python3-pytable.
Preparing to unpack .../063-python3-pytable_3.6.0-1_all.deb ...
Unpacking python3-pytable (3.6.0-1) ...
Selecting previously unselected package python3-rbd.
Preparing to unpack .../064-python3-rbd_16.2.15+ds-0+deb12u1_amd64.deb ...
Unpacking python3-rbd (16.2.15+ds-0+deb12u1) ...
Selecting previously unselected package libdw1:amd64.
Preparing to unpack .../065-libdw1_0.188-2.1_amd64.deb ...
Unpacking libdw1:amd64 (0.188-2.1) ...

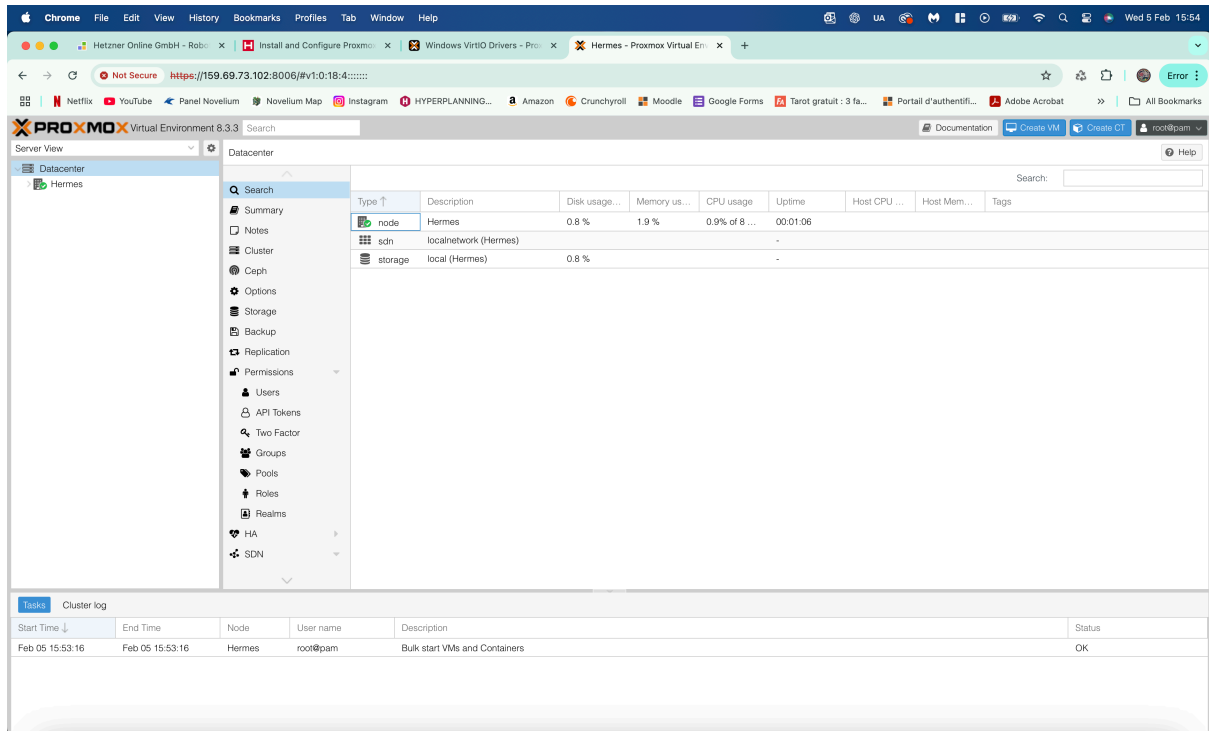
Progress: [ 10%] [#####.....]

```

8 - Screenshot of Proxmox's installation

² https://community.hetzner.com/tutorials/install-and-configure-proxmox_ve

It was then possible to access the Proxmox control panel with the server's IP address and Proxmox assigned port: 159.69.73.103:8006



9 - Screenshot of Proxmox's control panel

Next, it was needed to download the ISOs of the operating systems for each virtual machine that would then be created. As given on the topology, Windows Server 2022³ was downloaded along with Windows 11⁴, Debian 12⁵ and Ubuntu Desktop 20.4⁶.

Since one had already created Windows VMs, one did not forget to download Windows VirtIO Driver⁷ so that the Windows Installation program could access the virtual disk created for the VM.

Then, the ISOs were uploaded on Hermes (the rented server, hereafter referred to as 'Hermes') with FileZilla into the folder /var/lib/vz/template/iso, which is where Proxmox stores ISOs for VMs.

³ <https://www.microsoft.com/en-us/evalcenter/download-windows-server-2022>

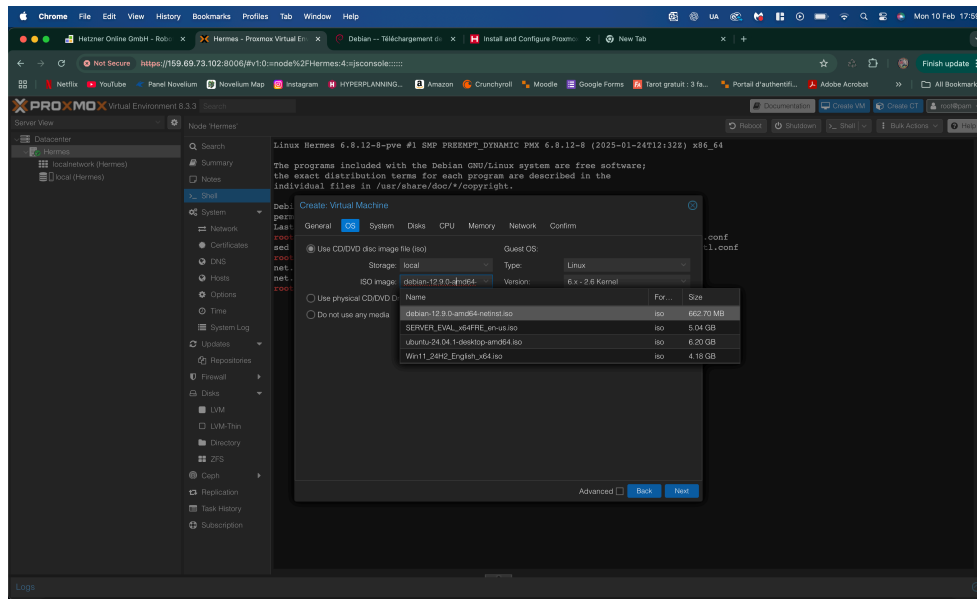
⁴ <https://www.microsoft.com/fr-fr/software-download/windows11>

⁵ <https://www.debian.org/download>

⁶ <https://ubuntu.com/download/desktop>

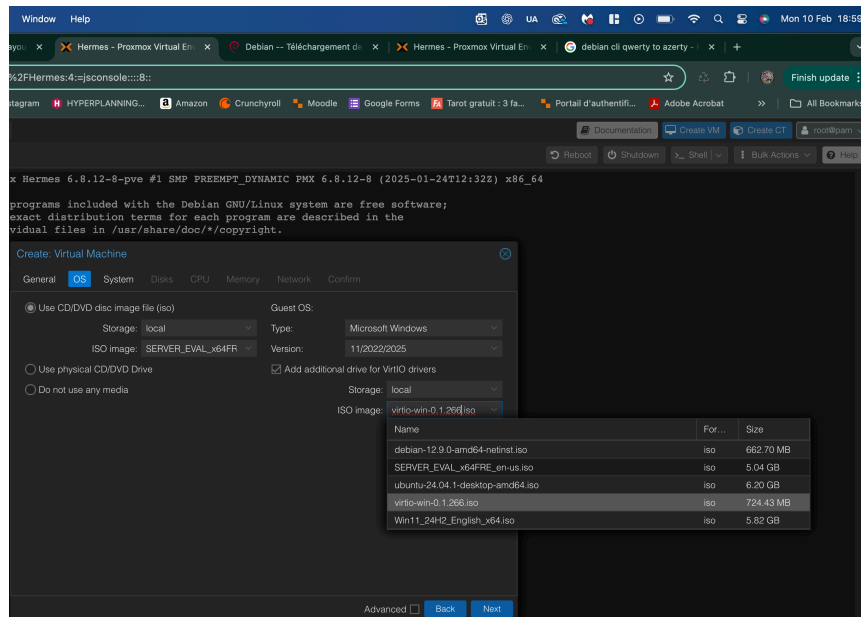
⁷ https://pve.proxmox.com/wiki/Windows_VirtIO_Drivers

Creating VMs and installing OSs



10 - Screenshot of the VM creation process

Each VMs are then being created one by one and were assigned 4 cores of the CPU, 8202 Mb of RAM and a 80Gb SSD disk. The network devices were not mounted at first as they'll be configured later. The installation process was completed without internet access.



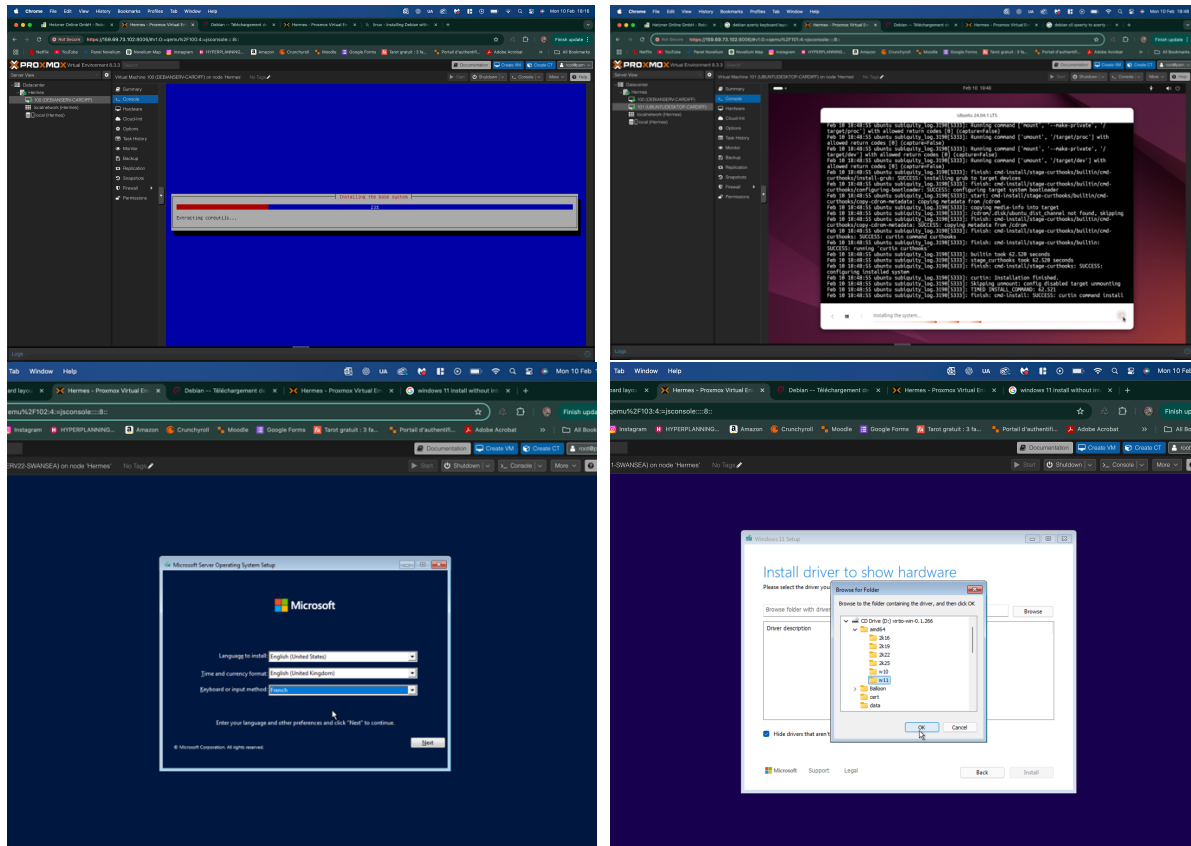
11- Screenshot of the VIRTIO driver being mounted alongside the ISO

The VIRTIO driver was not forgotten and was mounted alongside the system's ISO for the two Windows VMs as shown above.

Each OS was given a name according to its role and location in the course topology:

- DEBIANSERV-CARDIFF
- UBUNTUDESKTOP-CARDIFF
- WINSERV22-SWANSEA
- WIN11-SWANSEA

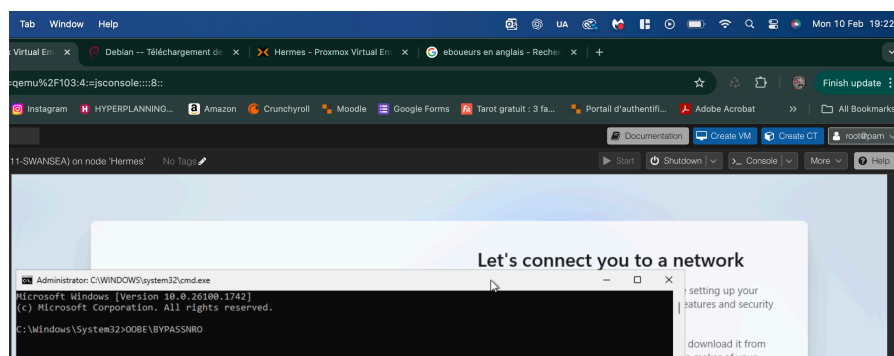
Their primary user was set as ‘administrator’ and password to ‘Allaan2’. Their primary language was set to EN-US, location to EN-UK and keyboard to AZERTY-FRENCH (since one is on a french Macbook with an azerty layout).



12 - Four screenshots of the installation of each VMs

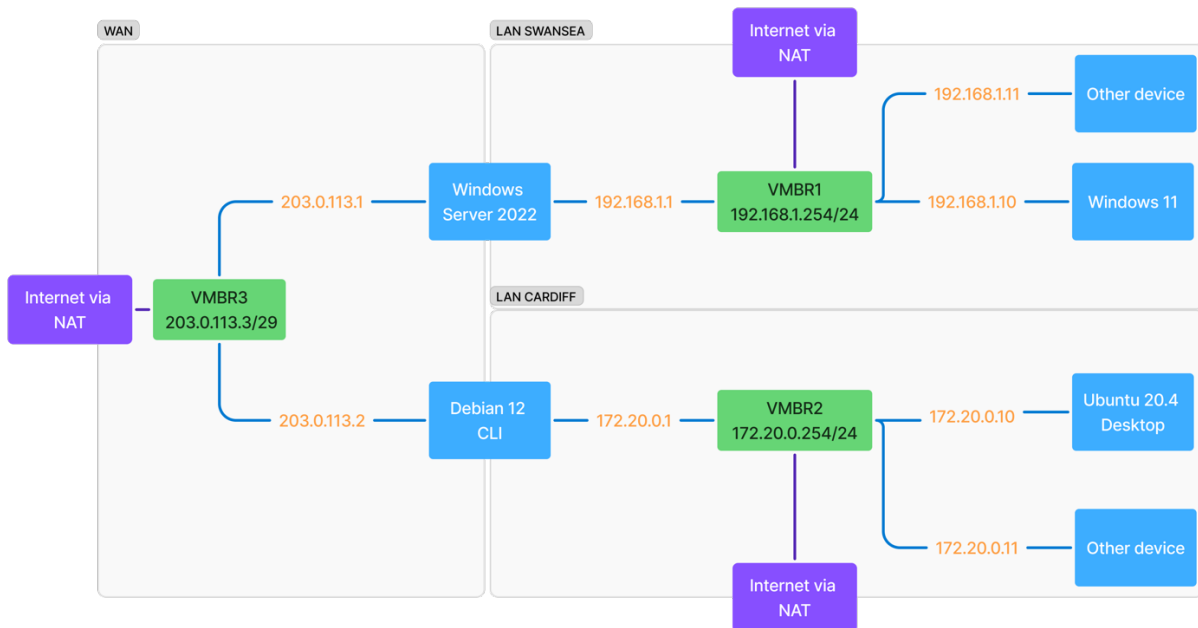
Debian was installed as a CLI os (without any GUI) and Windows Server 2022 as the Datacenter Desktop Experience (to have a GUI).

Regarding the Windows 11 Home installation, Microsoft required users to have internet access for it. After further researchs, one had to press SHIFT+F10 to open the command prompt and type “OOBE\BYPASSNRO” to bypass this requirement as shown on the following screenshot.



13 - Screenshot of how to bypass Windows 11 Internet required when installing

Building the typology



14 - Diagram of the network topology

Now, we will begin the network configuration and topology setup. On Proxmox, the network configuration works differently than on VirtualBox, as it is not possible to create a virtual network. Proxmox uses what we call ‘network bridges’ that act as routers redirecting traffic to Proxmox. For this project, three Linux-based network bridges will be used. A Linux network bridge is a virtual device that allows multiple network interfaces (the VMs) to connect to the data link layer (Layer 2) of the OSI model (Open Systems Interconnection model). This setup allows different segments of a network to communicate as if they were on a single network.

Therefore, to simulate the WAN and the Swansea/Cardiff LAN, three network bridges were created, called vmbr1, vmbr2, and vmbr3, as shown in the above topology. Each network bridge will act as edge routers and provide NAT access to its respective network for the connected machines. Since a machine can’t have multiple gateways (routers), the VMs acting as servers will have internet access through the vmbr3 bridge.

The IP addresses that were chosen for this project are as follow:

- 203.0.113.0/29 was chosen for the WAN network. This IPv4 range is reserved by the IETF and IANA (Internet Assigned Numbers Authority) as TEST-NET-3. Since we are setting up a private network for testing purposes in a non-production environment, and this bridge is intended to handle public IP addresses, these specific IP addresses were selected for this use-case. A /29 subnet was chosen instead of a /30 to allow more than two devices to connect to the WAN. With a /29 subnet, we have the necessary IP addresses to support the two VMs and the network bridge, while a /30 subnet would only allow for two usable IP addresses.

- 192.168.1.0/24 was chosen for the Swansea LAN. This ip scope is assigned for “local communications within a private network”, so its usage was perfect for this LAN which is a private network.
- 172.20.0.0/24 was chosen for the Cardiff LAN. This ip scope is also assigned for “local communications within a private network”, so its usage was also ideal.

```

GNU nano 7.2 /etc/network/interfaces
auto lo
iface lo inet loopback

iface lo inet6 loopback

auto enp0s31f6
iface enp0s31f6 inet static
    address 159.69.73.102/26
    gateway 159.69.73.65
    up route add -net 159.69.73.64 netmask 255.255.255.192 gw 159.69.73.65 dev enp0s31f6
# route 159.69.73.64/26 via 159.69.73.65

iface enp0s31f6 inet6 static
    address 2a01:4f8:231:1854::2/64
    gateway fe80::1

auto vibr1
iface vibr1 inet static
    address 192.168.1.254/24
    bridge-ports none
    bridge-stp off
    bridge-fd 0

auto vibr2
iface vibr2 inet static
    address 172.20.0.254/24
    bridge-ports none
    bridge-stp off
    bridge-fd 0

auto vibr3
iface vibr3 inet static
    address 203.0.113.3/29
    bridge-ports none
    bridge-stp off
    bridge-fd 0
    post-up echo 1 > /proc/sys/net/ipv4/ip_forward
  
```

15 - Screenshot of the configuration of the 3 network bridges

Then, all the VMs were assigned their network interfaces. The servers such as Windows Server 2022 and Debian 12 were assigned two interfaces: one to connect to the WAN and one to connect to the LAN (vibr3 and vibr1 or 2).

IPv4 forwarding was also enabled on Proxmox to allow the virtual machines (VMs) to access the internet. By enabling IPv4 forwarding, Proxmox can route traffic between the VMs and the external network, allowing them to communicate with the internet through the host machine.

The next part was to isolate the network on Proxmox's side to prevent the machines on each networks to access each other. To do so, one had to use the iptables rules.

```

root@Hermes ~ # sudo iptables -L INPUT -v --line-numbers
Chain INPUT (policy ACCEPT 109 packets, 16841 bytes)
num  pkts  bytes target     prot opt in     out     source     destination
1      0      0 DROP       all  --  !vmbr2 any    anywhere  172.20.0.0/24
2      0      0 DROP       all  --  !vmbr3 any    anywhere  203.0.113.0/29
3      0      0 DROP       all  --  !vmbr1 any    anywhere  192.168.1.0/24
root@Hermes ~ # sudo iptables -t nat -L POSTROUTING -v --line-numbers
Chain POSTROUTING (policy ACCEPT 3 packets, 209 bytes)
num  pkts  bytes target     prot opt in     out     source     destination
1      0      0 MASQUERADE all  --  any    enp0s31f6 192.168.1.0/24  anywhere
2      0      0 MASQUERADE all  --  any    enp0s31f6 172.20.0.0/24  anywhere
3      0      0 MASQUERADE all  --  any    enp0s31f6 203.0.113.0/29  anywhere
root@Hermes ~ # iptables -L FORWARD -v --line-numbers
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
num  pkts  bytes target     prot opt in     out     source     destination
1      0      0 DROP       all  --  vmbr1  vmbr2    anywhere  anywhere
2      0      0 DROP       all  --  vmbr1  vmbr3    anywhere  anywhere
3      0      0 DROP       all  --  vmbr2  vmbr1    anywhere  anywhere
4      0      0 DROP       all  --  vmbr2  vmbr3    anywhere  anywhere
5      0      0 DROP       all  --  vmbr3  vmbr1    anywhere  anywhere
6      0      0 DROP       all  --  vmbr3  vmbr2    anywhere  anywhere
7      0      0 ACCEPT     all  --  vmbr1  vmbr1    anywhere  anywhere
8      0      0 ACCEPT     all  --  vmbr2  vmbr2    anywhere  anywhere
root@Hermes ~ # date
Tue Feb 11 12:07:14 PM CET 2025
root@Hermes ~ #

```

16 - Screenshot of the IPTables rules on Proxmox

As shown above, the INPUT chain is used to filter incoming traffic so that each interface (vmbr1, vmbr2, vmbr3) only receives traffic from its designated source network (in this case, their respective LAN IPv4 address).

The POSTROUTING chain is used for NAT purposes. It redirects traffic from the three bridges to the physical network interface of Hermes to provide internet access to the connected VMs on those networks.

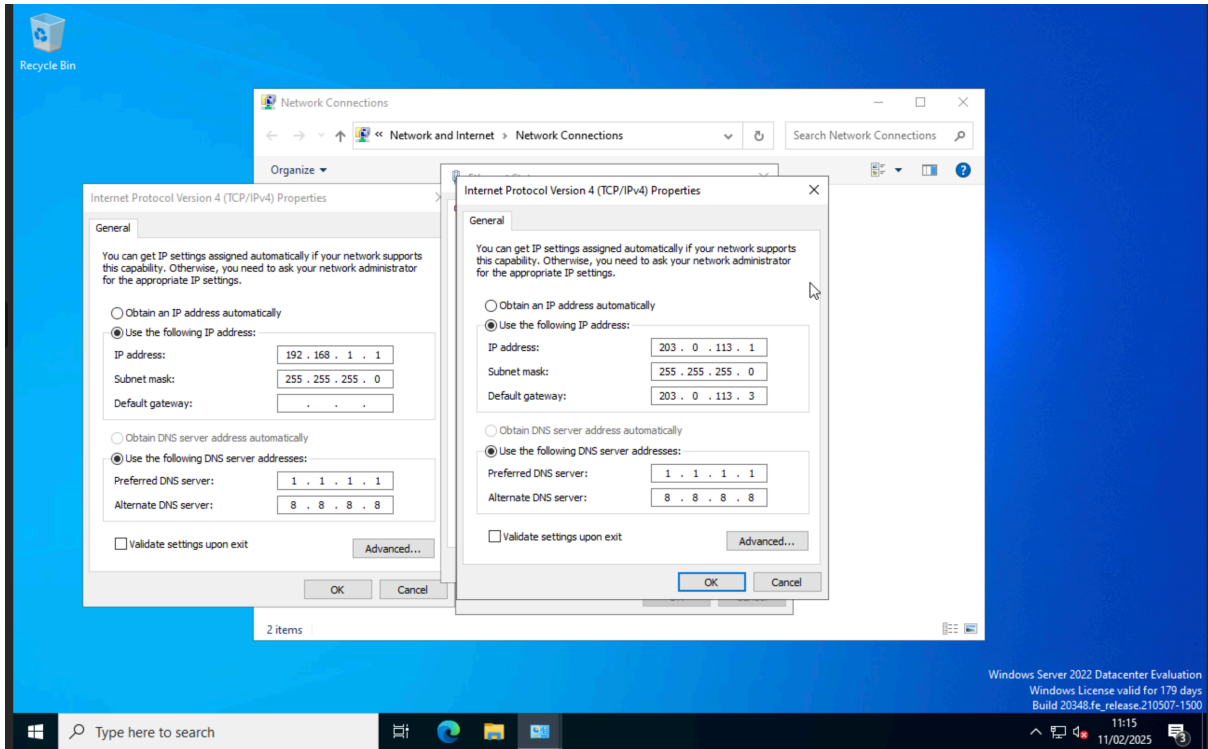
The FORWARD chain was created to enhance isolation. While the INPUT chain handles most of it, this one was created to lock traffic between the three bridges after debugging.

For those rules to be persistent after reboot, the package iptable-persistent was also installed. Those rules were then saved and applied.

Configuring Swansea's LAN

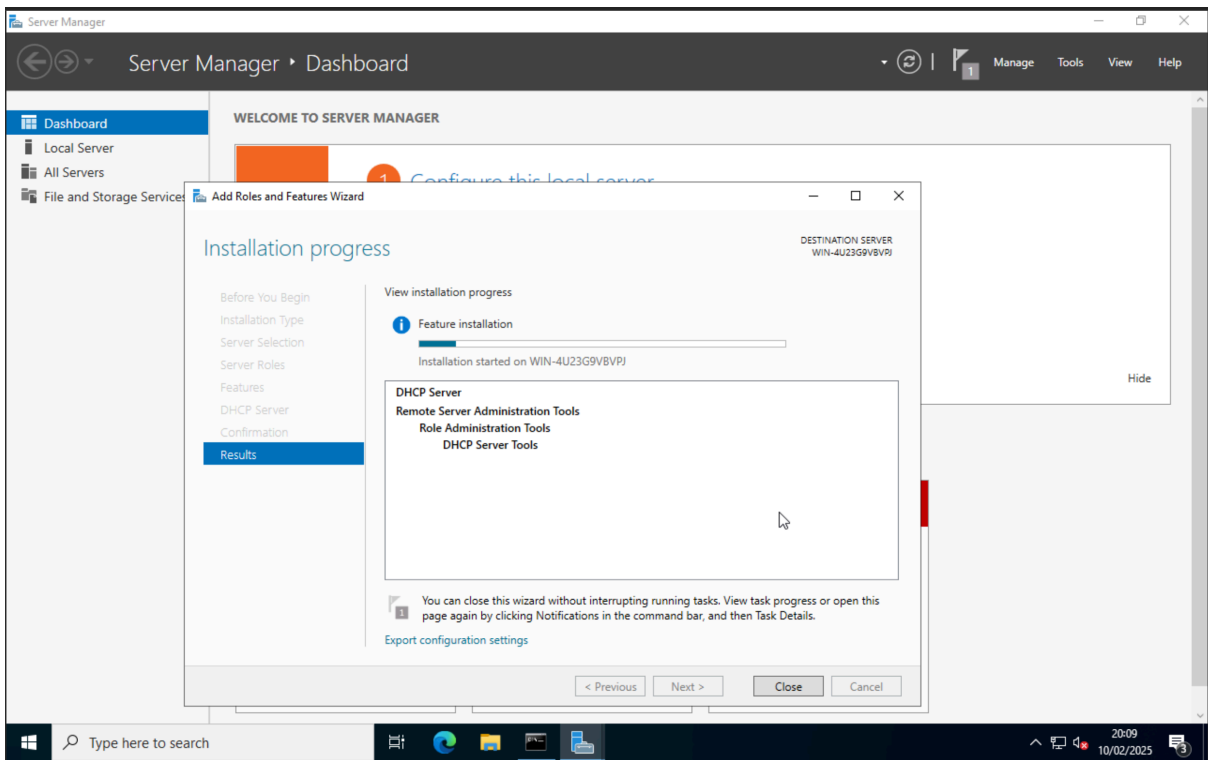
First of, one had to configure Windows Server 2022's network interfaces. The chosen IP addresses were assigned statically to the VM by typing WIN+R and `ncpa.cpl`. No default gateway was assigned for the LAN interface because WS22 will access internet via vmbr3, so via the WAN.

The chosen DNS's servers were 1.1.1.1 (Cloudflare's one) and 8.8.8.8 (Google's one) as an alternate DNS. This choice will be the same across all the other machines.



17 - Screenshot of the IPv4 configuration on Windows Server 2022

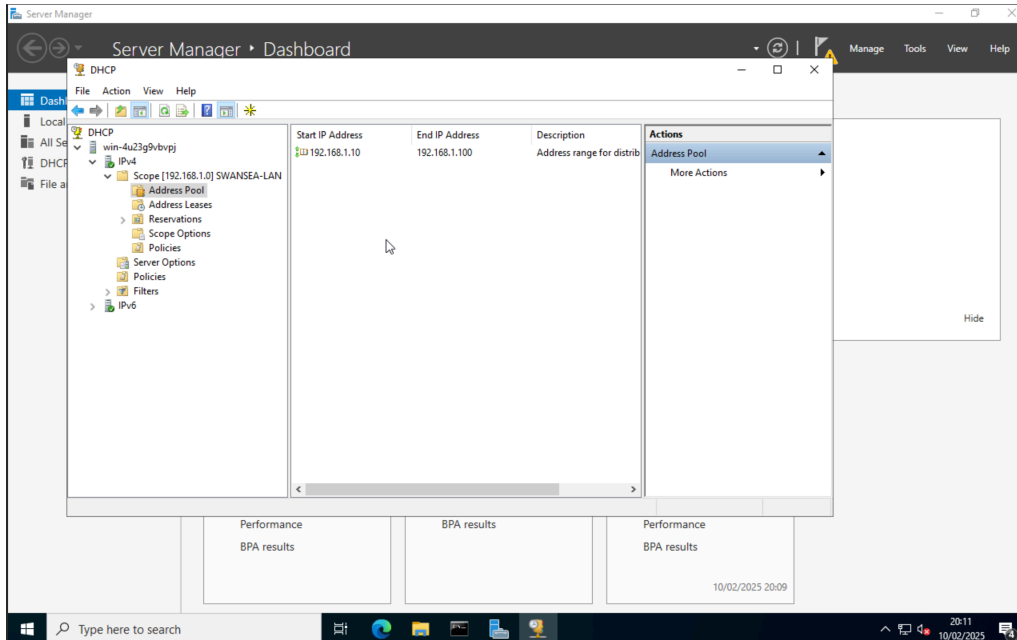
For the client's machine to be assigned an IP address and for an easier configuration (without assigning IP addresses statically), the DHCP server feature was installed too.



18 - Screenshot of the installation of the DHCP server on WS22

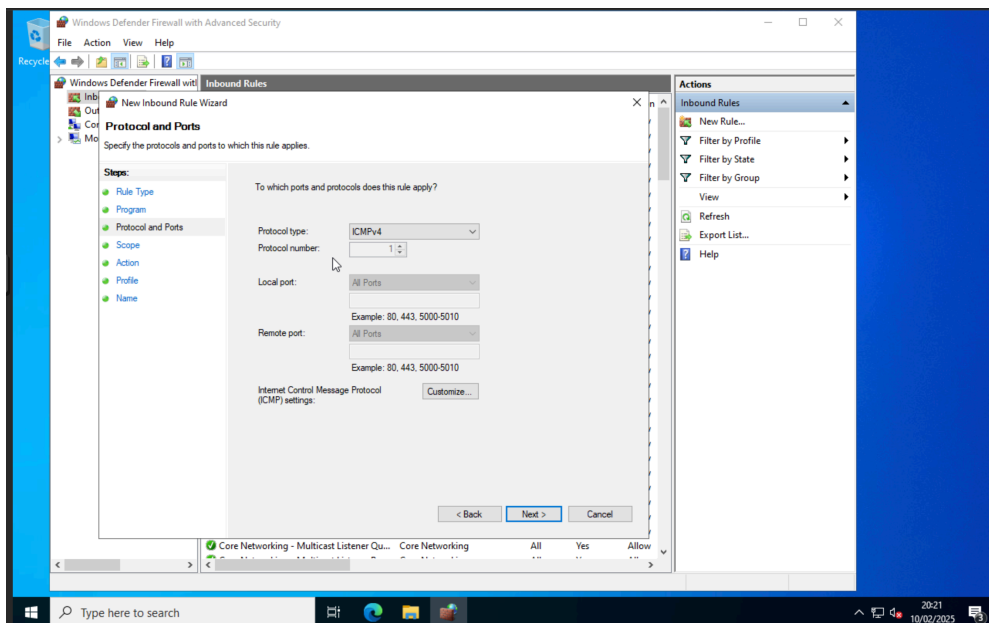
The DHCP server was configured as follows:

- The address pool ranges from x.x.x.10 to x.x.x.100, providing a total of 90 available addresses.
- The gateway was assigned to Proxmox's interface (vmbri1 in this LAN) to provide internet access.



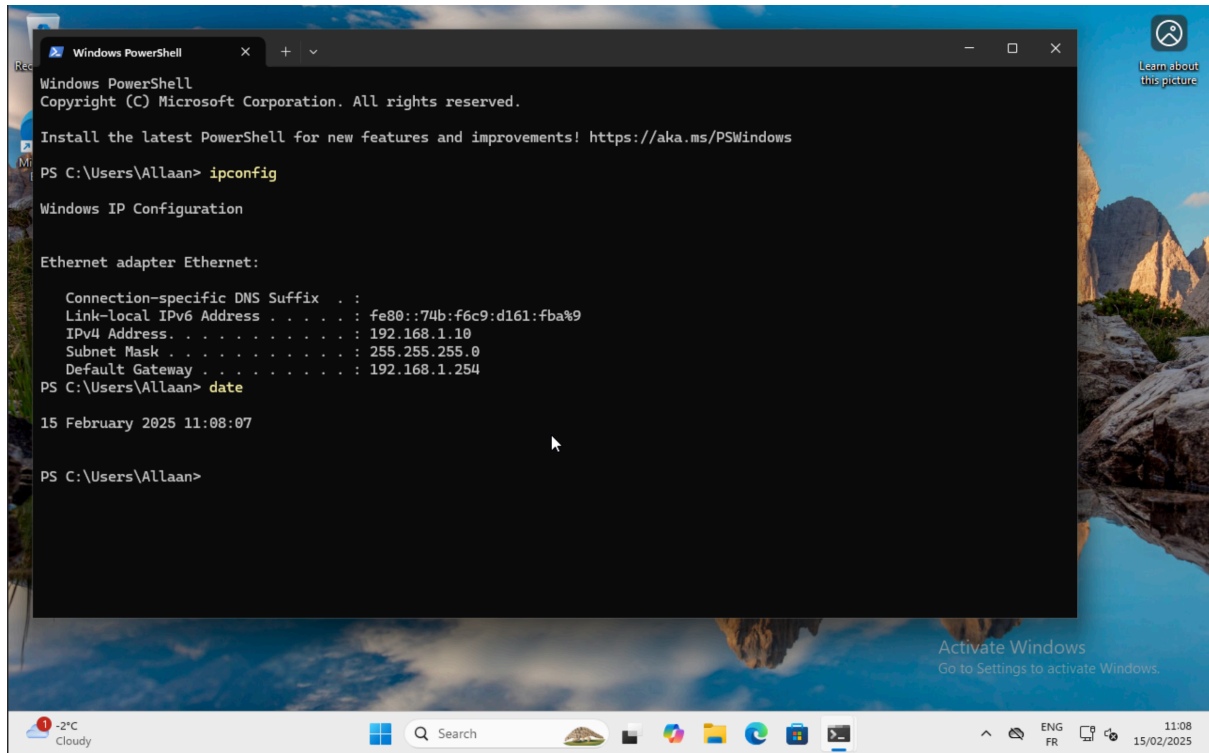
19 - Screenshot of the DHCP server configuration

Now, last but not least, one had to create a new firewall rule on both WS22 and Windows 11 to allow pingging between those machines. (By default, Windows blocks incoming ping requests). Therefore, the protocol ICMPv4 was authorized on the Advanced Firewall settings.



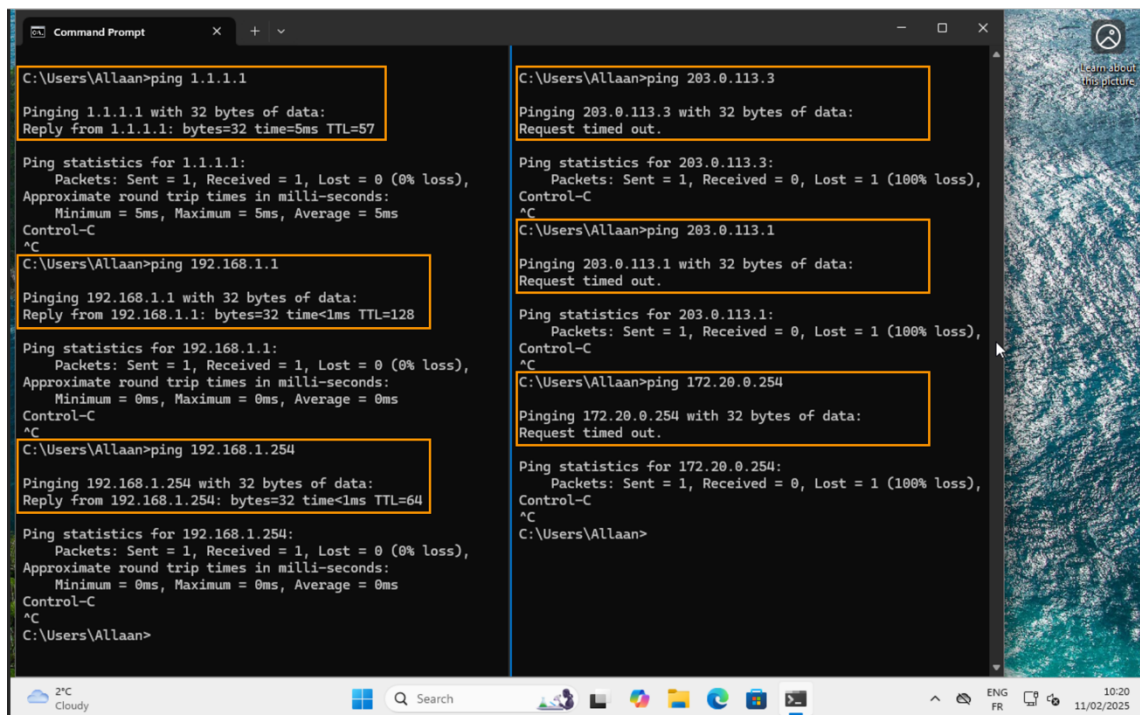
20 - Authorizing ICMPv4 Protocol on WS22's firewall

Finally, the Windows 11 VM can be powered on to check if everything is functioning correctly. As previously configured, this VM is connected to the internet and was automatically assigned its IP address (192.168.1.10), the first address in the configured DHCP pool. The gateway is set to the vmbr1 address. One had nothing to do than power on the VM.



21 - Screenshot of the IP of Win11

Now let's ping.



22 - Pinging tests on Win11

“On the left side, we ping Cloudflare’s DNS to check for internet access (Success), the WS22 VM (Success), and Proxmox’s vmbr1 interface (Success).

On the right side, we ping Proxmox’s vmbr3 (WAN) interface (Timed out), WS22’s WAN IP (Timed out), and Proxmox’s vmbr2 (Cardiff LAN) interface (Timed out).

```

Administrator: Command Prompt
C:\Users\Administrator>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: bytes=32 time<1ms TTL=128
Reply from 192.168.1.10: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.10:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C

C:\Users\Administrator>ping 203.0.113.3

Pinging 203.0.113.3 with 32 bytes of data:
Reply from 203.0.113.3: bytes=32 time<1ms TTL=64

Ping statistics for 203.0.113.3:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C

C:\Users\Administrator>ping 172.20.0.254

Pinging 172.20.0.254 with 32 bytes of data:
Request timed out.

Ping statistics for 172.20.0.254:
    Packets: Sent = 1, Received = 0, Lost = 1 (100% loss),
Control-C
^C
C:\Users\Administrator>
  
```

23 - Pinging tests on WS22

Now, let’s test on WS22. First, we ping Win11 VM’s (Success), Proxmox’s vmbr3 (WAN) interface (Success) and Proxmox’s vmbr2 (Cardiff LAN) interface (Timed out).

Everything is working correctly.

Configuring Cardiff’s LAN

```

GEMU (DEBIANSERV-CARDIFF) - noVNC
https://159.69.73.102:8006/?console=kvm&novnc=1&vmid=100&vmname=DEBIANSERV-CARDIFF&node=Hermes&resize=off&cm
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

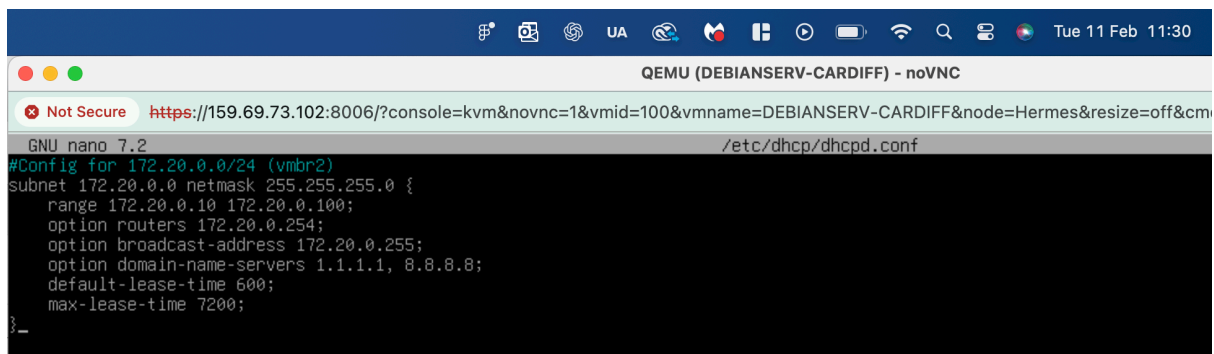
#WAN (vmbr3)
auto ens18
iface ens18 inet static
    address 203.0.113.2/29
    gateway 203.0.113.3

#LAN (vmbr2)
auto ens19
iface ens19 inet static
    address 172.20.0.1/24
    netmask 255.255.255.0
    broadcast 172.20.0.255
  
```

24 - Screenshot of the configuration of Debian's network interfaces

Since Proxmox runs on a Debian 12-based system, the configuration remains largely the same. On Debian 12, network interfaces can be configured in `/etc/network/interfaces`. Using the nano editor, this configuration was then modified. As shown above, both the LAN and WAN interfaces were configured with their respective IP addresses. The gateway remains set only for the WAN interface, and DNS servers were added to `/etc/resolv.conf`.

Although, since this VM was installed and configured without network access, it was necessary to update the APT sources manually. Once updated, network access was established, and apt update & apt upgrade ran successfully.

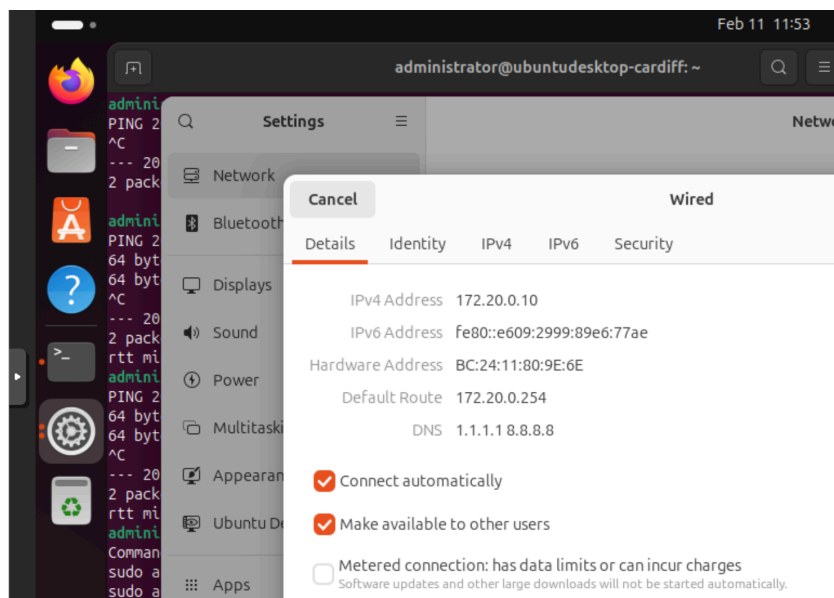


25 - Screenshot of the DHCP server configuration on Debian 12

For the DHCP server, the package `isc-dhcp-server` was then installed for this purpose. It was configured with the same settings as the one on WS22 but with Cardiff's LAN IP addresses as shown above.

Finally, one did not forget to enable IPv4 forwarding.

Let's check on the client's VM (Ubuntu Desktop).



26 - Screenshot of the IP address of Ubuntu Desktop

We can now see that the IP address was configured automatically without anything to do.

```

Feb 11 11:54
administrator@ubuntudesktop-cardiff: ~

administrator@ubuntudesktop-cardiff:~$ ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=57 time=5.75 ms
^C
--- 1.1.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.745/5.745/5.745/0.000 ms
administrator@ubuntudesktop-cardiff:~$ ping 172.20.0.1
PING 172.20.0.1 (172.20.0.1) 56(84) bytes of data.
64 bytes from 172.20.0.1: icmp_seq=1 ttl=64 time=0.398 ms
^C
--- 172.20.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.398/0.398/0.398/0.000 ms
administrator@ubuntudesktop-cardiff:~$ ping 172.20.0.254
PING 172.20.0.254 (172.20.0.254) 56(84) bytes of data.
64 bytes from 172.20.0.254: icmp_seq=1 ttl=64 time=0.107 ms
^C
--- 172.20.0.254 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.107/0.107/0.107/0.000 ms
administrator@ubuntudesktop-cardiff:~$ ping 203.0.113.3
PING 203.0.113.3 (203.0.113.3) 56(84) bytes of data.
^C
--- 203.0.113.3 ping statistics ---
8 packets transmitted, 0 received, 100% packet loss, time 7197ms
^C
administrator@ubuntudesktop-cardiff:~$ ping 203.0.113.2
PING 203.0.113.2 (203.0.113.2) 56(84) bytes of data.
^C
--- 203.0.113.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1018ms
^C
administrator@ubuntudesktop-cardiff:~$ ping 192.168.1.254
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
^C
--- 192.168.1.254 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3073ms
^C
administrator@ubuntudesktop-cardiff:~$ ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
^C
--- 192.168.1.10 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3052ms

```

27 - Screenshot of pinging test on Ubuntu Desktop

First, we ping Cloudflare's DNS to check for internet access (Success). Next, we ping Debian 12's VM (Success), Proxmox's vmbr2 interface (Success), and Proxmox's vmbr3 (WAN) interface (Timed out). The Debian 12 VM's vmbr3 IP address also times out, as does Proxmox's vmbr1 (Swansea LAN) interface and the Windows 11 VM's IP address (Timed out).

Virtual Machine 100 (DEBIANSERVER-CARDIFF) on node 'Hermes' No Tags

```

root@DEBIANSERVER-CARDIFF:~# ping 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=57 time=5.91 ms

--- 1.1.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.908/5.908/5.908/0.000 ms
root@DEBIANSERVER-CARDIFF:~# ping 203.0.113.3
PING 203.0.113.3 (203.0.113.3) 56(84) bytes of data.
64 bytes from 203.0.113.3: icmp_seq=1 ttl=64 time=0.205 ms

--- 203.0.113.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.205/0.205/0.205/0.000 ms
root@DEBIANSERVER-CARDIFF:~# ping 172.20.0.10
PING 172.20.0.10 (172.20.0.10) 56(84) bytes of data.
64 bytes from 172.20.0.10: icmp_seq=1 ttl=64 time=0.504 ms

--- 172.20.0.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.504/0.504/0.504/0.000 ms
root@DEBIANSERVER-CARDIFF:~# ping 172.20.0.254
PING 172.20.0.254 (172.20.0.254) 56(84) bytes of data.
64 bytes from 172.20.0.254: icmp_seq=1 ttl=64 time=0.249 ms

--- 172.20.0.254 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.249/0.249/0.249/0.000 ms
root@DEBIANSERVER-CARDIFF:~# ping 192.168.1.254
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
^C
--- 192.168.1.254 ping statistics ---
62 packets transmitted, 0 received, 100% packet loss, time 62463ms

root@DEBIANSERVER-CARDIFF:~# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
^C
--- 192.168.1.1 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1012ms

root@DEBIANSERVER-CARDIFF:~# date
Tue 11 Feb 11:44:05 GMT 2025
root@DEBIANSERVER-CARDIFF:~#

```

28 - Screenshot of ping test on Debian 12

Now, we test pinging on the Debian 12 VM. As usual, we first ping Cloudflare's DNS to check for internet access (Success). Then, we ping Proxmox's vmbr3 (WAN) interface (Success), Ubuntu Desktop's IP address (Success), and Proxmox's vmbr2 (Cardiff LAN interface) (Success).

Finally, we ping Proxmox's vmbr1 (Swansea LAN interface) (Timed out) and WS22's Swansea LAN IP address (Timed out).

Servers are now connected together and can successfully ping each other. Clients are also connected to their respective locations & servers and automatically assigned an IP address via DHCP. The topology is completed.

TASK 3

- Reflection -

During the conception of this coursework, numerous problems were encountered. Almost all of them are already described in Task 2, but some are not. Aside from the Windows 11 installation, which was complicated without an internet access, the most obvious issue was working on a remote machine (the rented server called Hermes). Configuring the network on a physically distant machine can result in a loss of contact with the remote server if improperly configured. And that's exactly what happened—halfway through the project, the network configuration of the server was broken, rendering it inaccessible. As a result, everything had to be reinstalled and reconfigured from scratch, going from Debian to Proxmox, and the virtual machines.

This project wasn't as difficult as initially expected: the network is now properly configured, just as it would be in a real-world setup, and everything functions as it should.

Now, let's move on to Coursework 2 for furthermore configuration.

Resources

- (1) Diagram of a Type 1 and a Type 2 hypervisor. "Hypervisor" *Wikipedia*. Accessed: Jan. 30, 2025. [Online]. Available: <https://fr.wikipedia.org/wiki/Hyperviseur>
- (2) Representation of an IPv4 address. NETWORKING WITH H, *what is IP address ? | CISCO CCNA 200-301* (Nov. 9, 2020). Accessed: Jan. 31, 2025. [Online Video]. Available: <https://youtu.be/GPPFASALbQo?si=10HuHy9zTNpze5aC>
- (3) Diagram showing Network / Host repartitions in IPv4 classes. NETWORKING WITH H, *IP address classes explained | class A , B , C , D , E | Free CCNA 200-301* (Dec. 10, 2020). Accessed: Jan. 31, 2025. [Online Video]. Available: <https://youtu.be/0dFNpNgiTAA?si=kGcLVhMmwsIYoDHN>
- (4) Representation of an IPv6 address. D. ASTURIAS, "What is IPv6?" *RapidSeedbox*. Accessed: Jan. 31, 2025. [Online]. Available: <https://www.rapidseedbox.com/blog/what-is-ipv6>
- (5) Diagram of a VPN Network. P. LUIDOLT, "VPN Network Diagram [classic]" *creately*. Accessed: Feb. 4, 2025. [Online]. Available: <https://creately.com/diagram/example/ipv0yyob1/vpn-network-diagram-classic>