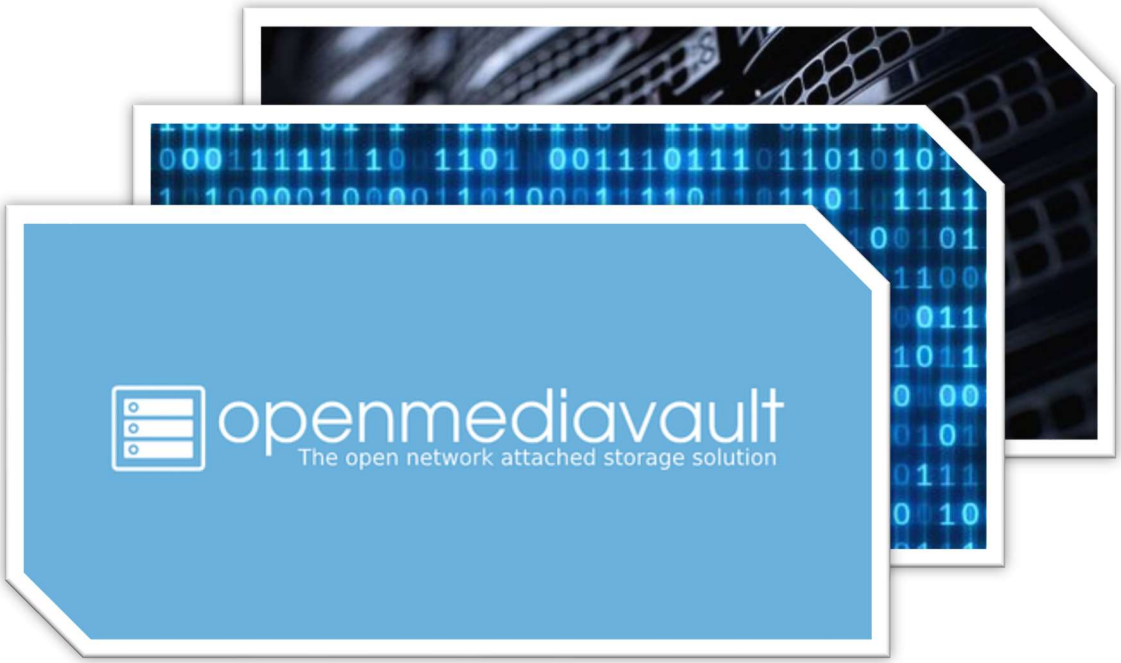
NAS Server Setup & Configuration

Documentation



Lycée Guillaume Kroll

Course / Module: CLOIF2
School: Lycée Guillaume Kroll
Class: BCLC25
Members Team 2:
Moshe Sarkis Marios
Christophe Thill
Teachers:
Mrs. RODESCH Christian
Mrs. SPAGNUOLO Maurizio

Table of Contents

1. Introduction

This report documents the full setup of a Network Attached Storage server completed as part of the Cloud Infrastructure 2 course. The objective of the project was to take a real 19-inch rack-mount server with no operating system and no configured storage and transform it into a fully operational NAS solution ready for deployment in a lab rack environment.

Starting from hardware, our team assembled and configured server MR3S01 step by step. This involved physically installing the RAM modules, native SATA hard drives, and an Areca ARC-1880 hardware RAID controller with its SAS drives, before verifying every component through the server BIOS. From there we configured a RAID 5 array on the hardware controller, set up a second software RAID volume using OpenMediaVault's built-in BTRFS RAID1 feature, and installed OpenMediaVault 8 onto a rear USB stick to keep all internal drives free for storage.

Once the software side was running, we moved on to creating shared folders, user accounts, and access permissions designed around a realistic family home storage scenario with private folders per user and a common shared media space, all accessible over SMB, NFS, and SFTP from Windows, Linux, and macOS clients. The server was then physically mounted in the lab rack and connected to the network, including the out-of-band Intel RMM3 management interface and the dedicated RAID controller port.

The report follows the natural order of the work: background research, hardware staging, RAID configuration, OS installation, rack mounting, NAS configuration, and testing. A separate section documents the day-by-day progress of the project, including the technical problems we ran into and how each one was resolved. Not everything went according to plan the RAID initialisation, the OS boot issues, and the network configuration each required troubleshooting and those experiences are documented honestly alongside the technical content.

2. Research & Background

2.1 What is Network Attached Storage (NAS)?

A NAS is a dedicated file server that connects to a local network and exposes shared storage to any device on that network. Instead of plugging an external drive directly into one computer, every machine on the LAN or even over the internet can reach the same centrally managed pool of disks.

Technically, a NAS communicates using standard networking protocols over TCP/IP, either through a wired Ethernet connection or Wi-Fi. The most common file-sharing protocols it serves are SMB/CIFS for Windows clients and NFS for Linux/Unix clients, though modern NAS systems support many more.

From a practical standpoint, NAS devices are used for:

- Central file storage and sharing documents, photos, videos, and any unstructured data.
- Automated backups for workstations, laptops, and mobile devices.
- Media streaming at home (to smart TVs, game consoles, etc.) or hosting a personal cloud accessible from the internet.
- In business environments: collaborative file access, long-term archiving, and disaster recovery storage.

2.2 DAS vs NAS vs SAN

The three main storage architectures are often confused. Here is a clear comparison:

	DAS	NAS	SAN
Full name	Direct Attached Storage	Network Attached Storage	Storage Area Network
How it connects	Cable directly into one host (USB, SAS, SATA)	Ethernet / Wi-Fi (LAN)	Dedicated Fibre Channel or iSCSI network
Access type	Block-level, one host only	File-level, shared across many clients	Block-level, shared across many servers
Typical use case	Single workstation or server extension	Home/SMB shared storage, personal cloud	Enterprise: databases, VMware datastores

2.3 Why Compute and Storage Are Separated in Professional Environments

In any sizeable production environment, the compute layer (servers running applications, virtual machines, databases) is kept physically and logically separate from the storage layer. On the surface this might seem inefficient, but there are strong reasons why this is the standard approach:

- Independent scalability: A company might need to double its disk capacity without touching its application servers or vice versa. Separation makes this straightforward.
- Higher availability: If one compute node crashes, the storage array stays online. Other servers can fail over and continue reading from and writing to the same data.
- Centralised management: Having a single storage platform means snapshots, replication, and access control policies are managed in one place rather than scattered across every individual server.
- Performance optimisation: Storage systems can be tuned specifically for I/O throughput and disk access patterns (e.g., SSD caches, RAID tiers), while compute servers are optimised for CPU and RAM. Mixing the two workloads on the same hardware leads to resource contention.
- Maintenance windows: Firmware updates or hardware replacements on the storage side can often be done non-disruptively, without affecting the compute layer.

In our lab scenario, we experienced a small version of this principle: the Areca hardware RAID controller manages our SAS disks independently of the OS, so the NAS software simply sees a pre-built, redundant volume rather than raw disks.

3. Server Hardware

3.1 Form Factor and Casing Dimensions

Our server is a standard 19-inch rack-mount chassis in the 2U form factor. The "U" (rack unit) is a standardised unit of measurement used across the data centre industry; one U equals 44.45 mm in height. A 2U server therefore occupies 88.9 mm of vertical rack space. The 19-inch width is the outer flange width, meaning the actual usable interior is slightly narrower this is what allows multiple servers from different manufacturers to fit in the same rack.

Form Factor	2U (19-inch rackmount)
Height per U	44.45 mm
Server Height	2U = 88.9 mm (3.44 inches)
Server Width	16.93 inches (430 mm)
Server Depth	27.95 inches (710 mm)
Server Name	MR3S01

3.2 External Ports and Connectors

3.2.1 Front Panel

The front panel of the server is where day-to-day drive access and basic connectivity happen. We identified the following ports and components:

- 5 SATA drive bays for hot-swap storage devices
- 1 VGA port for connecting a local monitor
- 1 USB port for installation media or local management

3.2.2 Rear Panel

The rear panel is far more feature-rich and is where all permanent cabling is done:

- 1 VGA port secondary monitor output
- 4 USB ports: 2 x USB 2.0, 2 x USB 3.0 the rear USB is where the OS Samsung USB stick was connected
- 1 Serial console port (IN) for out-of-band serial access
- 2 standard LAN ports (1 per CPU) used for regular network traffic
- 1 RMM (Remote Management Module) network port dedicated out-of-band management interface
- 1 dedicated network port for the Areca RAID controller allows the web-based RAID GUI to be accessed
- 2 PSU (Power Supply Unit) connectors supporting the redundant dual-PSU configuration

3.3 Internal Hardware Components

After opening the chassis, we identified every major internal component by cross-referencing with the server board manual (page 38). The hardware configuration for MR3S01 was:

CPUs	2x Intel processors (dual-socket board)
RAM	8 x 2 GB DDR3 modules = 16 GB total
RAID Controller	1x Areca hardware RAID controller (PCIe), with backup battery unit
Native HDDs	2 x 160 GB SATA (DRIVE_0 and DRIVE_1, connected to mainboard SATA ports)
RAID HDDs	3 x Seagate SAS Cheetah 146 GB (DRIVE_2, DRIVE_3, DRIVE_4, connected to RAID controller ports 1–3)
Fans	3 system cooling fans
PSUs	2 redundant power supply units
Server Board	1x Intel dual-socket server mainboard
Other	Processor air duct, midplane board for backplane connectivity

3.4 Internal Ports and Connectors

The internal connectivity is what ties everything together. We found the following internal interfaces:

- 8 SATA ports located in front of the fan wall used for the native SATA HDDs and available expansion
- 6 SAS/SATA ports on the Areca RAID controller the three Cheetah drives are plugged into ports 1–3
- 1 PCIe (PCI Express) slot occupied by the Areca RAID controller
- 12 RAM slots on the motherboard 8 of which are populated with 2 GB modules

4. Staging — Hardware and Firmware Preparation

Before any software can be installed, the physical hardware needs to be correctly assembled and verified. This stage is called 'staging' and it is one of the most critical phases a mistake here can cause data loss, hardware damage, or subtle stability issues that only surface later under load.

4.1 Installing the RAM Modules

The server uses DDR3 ECC (Error-Correcting Code) memory, which is standard for server-grade hardware. Unlike desktop RAM, server ECC memory can detect and correct single-bit memory errors on the fly, improving overall system reliability. We installed eight 2 GB modules, filling the correct slots as specified in the server manual to ensure dual-channel or quad-channel operation where possible.

During installation, care was taken to:

- Ground ourselves before handling components to avoid electrostatic discharge.
- Insert modules firmly until both retaining clips clicked into place.
- Check that each module was seated in the correct slot for the memory topology.

4.2 Installing the Native HDDs

Two WDC WD1600AAJS 160 GB SATA hard drives were installed into the mainboard-connected bays. DRIVE_0 (/dev/sda) was connected to SATA_0 on the mainboard and DRIVE_1 (/dev/sdc) was connected to SATA_1. These two drives would later be used as the two members of a software BTRFS RAID 1 volume managed by OpenMediaVault.

4.3 Installing the RAID Controller and SAS HDDs

The Areca hardware RAID controller was seated into the PCIe slot on the mainboard. Hardware RAID controllers offload the RAID processing from the main CPU, and they often include a battery backup unit (BBU) which preserves the write cache even during a power failure preventing data corruption.

The three Seagate SAS Cheetah 146 GB drives were then installed in bays DRIVE_2 through DRIVE_4 and connected to RAID controller ports 1, 2, and 3 respectively. SAS (Serial Attached SCSI) drives are enterprise-grade, designed for 24/7 operation with higher IOPS (Input/Output Operations Per Second) than standard SATA disks.

4.4 Power Supply Check and Failure Simulation

The server came equipped with two PSUs for redundancy. We verified both were functional by powering on the server and checking the front panel status LEDs. We then simulated a PSU failure by disconnecting one power cable while the server was running. The server continued operating without interruption, confirming that the hot-swap redundancy was working correctly. The failed PSU showed an alert on the front panel, which is the expected behaviour.

4.5 Peripheral Connection and BIOS Check

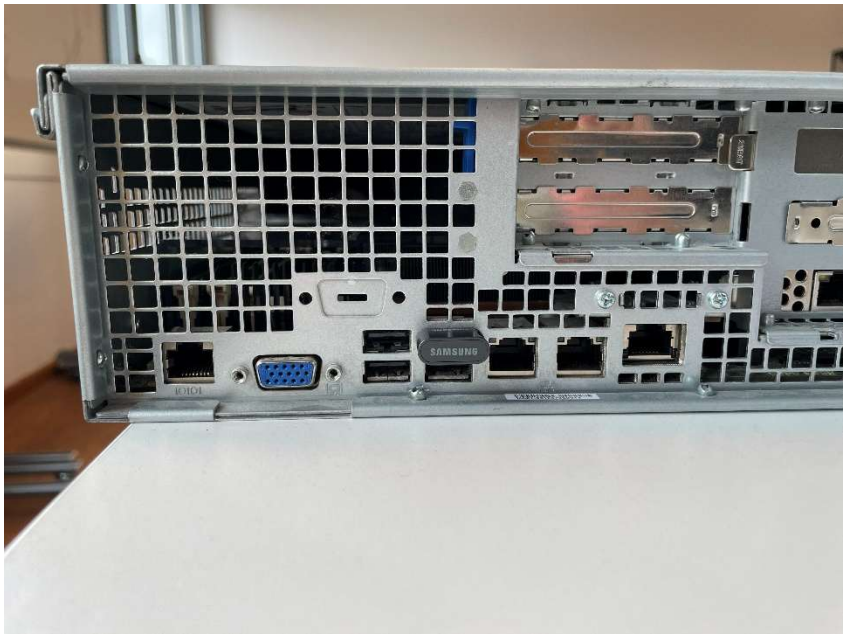
A monitor (via VGA), keyboard, and mouse were connected to the rear panel. On first boot, we entered the BIOS to verify:

- All installed RAM was detected: 8 modules visible, total 16 GB.
- All HDDs were listed: both SATA drives and the controller were detected (the SAS drives are managed by the RAID controller, so they appear as a single virtual disk in the BIOS after RAID configuration).
- No error codes or POST failures were present.
- The boot device order was set correctly USB first, then internal devices.
- The RMM (Remote Management Module) IP address was set to 192.168.0.40 with subnet 255.255.255.0 and gateway 192.168.0.254.

4.6 RAID Controller IP Verification

The Areca RAID controller has its own dedicated network interface for web-based management. We verified the RAID controller IP was configured to 192.168.0.140 (subnet 255.255.255.0, gateway 192.168.0.254), which would allow us to access its configuration interface from any machine on the management subnet.

4.7 Photos



5. RAID Controller Configuration

5.1 What is RAID?

RAID (Redundant Array of Independent Disks) is a method of combining multiple physical drives into a single logical unit. Depending on the RAID level chosen, this can provide redundancy (protection against drive failure), improved read/write performance, or a combination of both.

RAID Level	Technique	Min. Drives	Drive Failure Tolerance
RAID 0	Striping — data split across drives	2	None — one failure loses all data
RAID 1	Mirroring — identical copy on each drive	2	1 drive can fail, data survives
RAID 5	Striping with distributed parity	3	1 drive can fail, data survives
RAID 6	Striping with dual parity	4	2 drives can fail simultaneously
RAID 10	Mirrored pairs, then striped	4	1 drive per mirrored pair can fail

5.2 Creating the RAID 5 Array (areca_raid5)

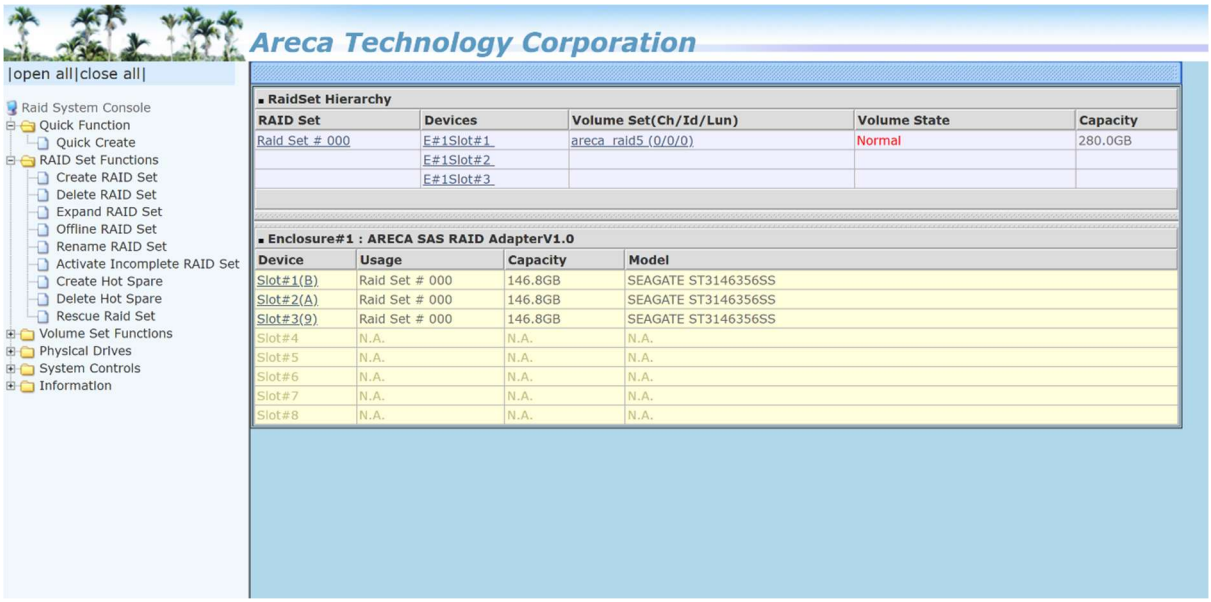
To access the Areca RAID controller interface, we connected a laptop to the dedicated RAID management network port and navigated to 192.168.0.140 in a browser. The web GUI loaded correctly and displayed the three connected SAS drives.

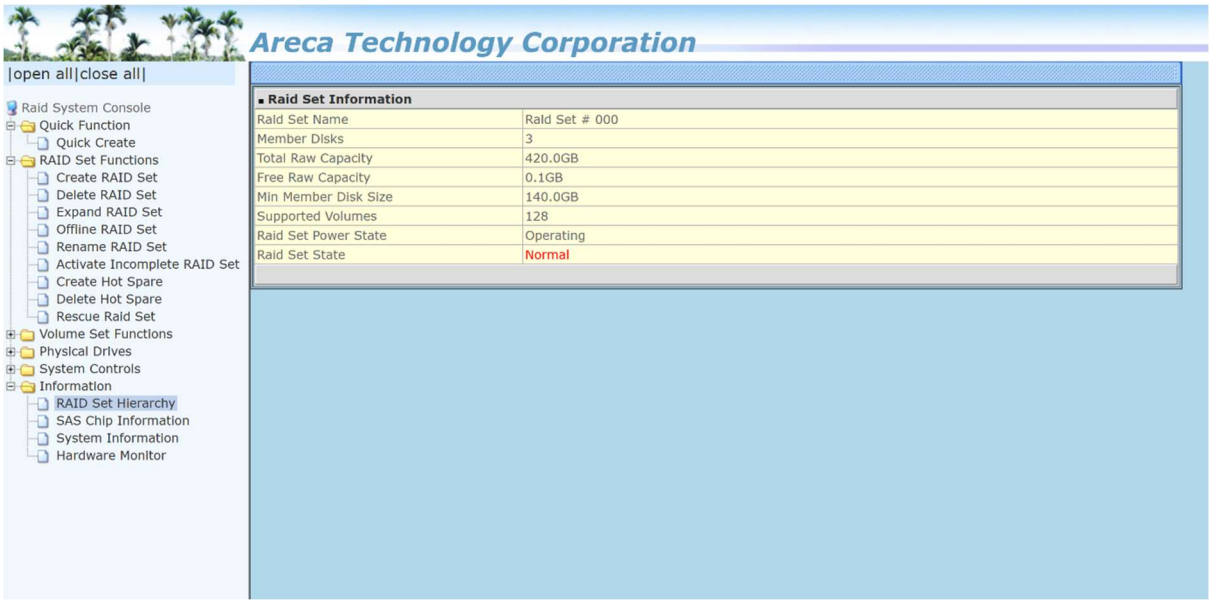
We then created a new RAID 5 array using all three Seagate SAS Cheetah drives. RAID 5 was the appropriate choice here because:

- It offers fault tolerance: if any single one of the three drives fails, the array remains online and readable. The missing data can be reconstructed from the parity blocks distributed across the other two drives.
- It uses storage efficiently compared to RAID 1: with three drives at 146 GB each, RAID 5 gives us approximately 292 GB of usable space (two drives worth), rather than RAID 1 which would give only 146 GB.
- The hardware controller offloads the parity calculation from the CPU, so performance is not impacted even under heavy write loads.

The volume was named areca_raid5 as instructed. Once the array was initialised and the build/synchronisation completed, it appeared as a single clean disk to the operating system installed later.

5.3 Screenshots & Photos





6. NAS Operating System — OpenMediaVault

6.1 Research: About OpenMediaVault

6.1.1 What is OpenMediaVault?

OpenMediaVault (OMV) is a Debian-based NAS distribution that comes with a browser-based web interface for managing storage, users, file shares, and services. It was designed to be an out-of-the-box solution, meaning even someone without deep Linux command-line knowledge can install and operate a fully functional NAS.

Under the hood, OMV is a standard Debian Linux system, but one where all the complexity of configuring services like Samba, NFS, SSH, or rsync has been wrapped into a clean graphical interface. Advanced users can still drop to the shell and customise things at the OS level when needed.

6.1.2 Who Created It?

The project was started and is still led by Volker Theile, who has been the primary developer since OMV's inception around 2009. The software is released under the GNU General Public License v3 (GPLv3), which means anyone can inspect, modify, and redistribute the source code freely. Copyright is held by Volker Theile for the period 2009–2025.

6.1.3 Latest Release

At the time of our project, the current stable major series is OMV 7, which became the official stable release in March 2024. It is based on Debian 12 "Bookworm". Updates within the OMV 7 series (point releases) are delivered through Debian's standard package management system.

6.1.4 Cost

OpenMediaVault is completely free. There are no license fees per disk, per user, or per feature. The GPLv3 license explicitly allows anyone to download, run, and modify it. This is one of its biggest advantages over commercial NAS solutions, which often require paid licenses for capacity expansion or advanced features.

6.1.5 Key Features

- Core platform: Built on Debian Linux, inheriting its stability, security updates, and wide hardware support.
- Web administration: A responsive browser-based GUI covers virtually all configuration tasks storage, services, networking, users, and notifications.
- Storage management: Supports volume management, filesystem creation (ext4, XFS, Btrfs), S.M.A.R.T. disk monitoring, Wake-on-LAN, and snapshot support.
- File sharing services: Native support for SMB/CIFS (Windows), NFS (Linux/macOS), SFTP, FTP, rsync, and SSH.
- RAID: Software RAID via both BTRFS profiles (RAID1, RAID5, etc.) and Linux MD RAID (mdadm), giving flexibility in how redundancy is implemented.

- Notifications: Email alerts for hardware events, disk failures, or service issues.
- Plugin system: A modular architecture allows community and official plugins to extend functionality for example, adding a BitTorrent client, Docker support, or extra RAID management tool.

6.1.6 When to Use OMV and When Not To

Reasons to use OpenMediaVault:

- You want a free, open-source NAS OS with a reasonably simple web UI and don't require enterprise vendor support.
- You prefer Debian's ecosystem and wide hardware compatibility, including low-power hardware like a Raspberry Pi.
- You want a modular system install only the plugins and services you actually need.

Reasons you might prefer an alternative:

- If you need deep ZFS integration and a polished management experience, TrueNAS Core or TrueNAS SCALE might be a better fit.
- If you need full-stack virtualisation baked in (containers, VMs) with tight storage integration, TrueNAS SCALE or Proxmox VE with shared storage might suit better.
- If you want a turnkey vendor-supported appliance where hardware and software come bundled with a support contract, a commercial NAS from Synology or QNAP would be more appropriate.

6.2 Creating the Installation Medium

We downloaded the latest OMV 7 ISO image from the official website (https://www.openmediavault.org/) and used a tool like Rufus or balenaEtcher to write it onto a USB stick. The installation medium was verified by checking its file hash against the checksum published on the download page.

6.3 Installation Procedure

The installation USB was plugged into one of the front USB ports of the server. We configured the BIOS to boot from USB first, then restarted. The OMV Debian-based installer loaded correctly. The full sequence of installer steps is documented below exactly as we went through them.

6.3.1 Language and Locale

The installer opened with a language selection screen. We selected English as the installation language. Since Luxembourg was not in the default shortlist of English-speaking locations, we chose Other and then navigated to Europe > Luxembourg. Because there is no dedicated Luxembourg locale for English, the installer prompted us to select a locale to base system settings on we chose United Kingdom (en_GB.UTF-8). For the keyboard layout we selected Swiss French, which corresponds to the physical keyboards in our classroom.

6.3.2 Network Configuration

The installer detected two network interfaces on the motherboard, both Intel Corporation 82575EB Gigabit Network Connection adapters. We selected enp1s0f0 as the primary interface.

The installer then attempted to auto-configure the network using DHCP but returned an error: Network autoconfiguration failed the network was not using the DHCP protocol or the DHCP server was unavailable. This was expected since the lab network uses static IP addressing. We dismissed the error and proceeded to configure the network manually with the following values:

IP address	10.0.13.2
Netmask	255.255.0.0
Gateway	10.0.0.1
Name server (DNS)	10.0.0.1
Hostname	MR3S01
Domain name	local

6.3.3 Root Password

The installer prompted us to set a root password, which was entered and confirmed. This password is used for direct console access to the underlying Debian system and is separate from the OMV web interface admin credentials configured later.

6.3.4 Disk Partitioning and Installation Target

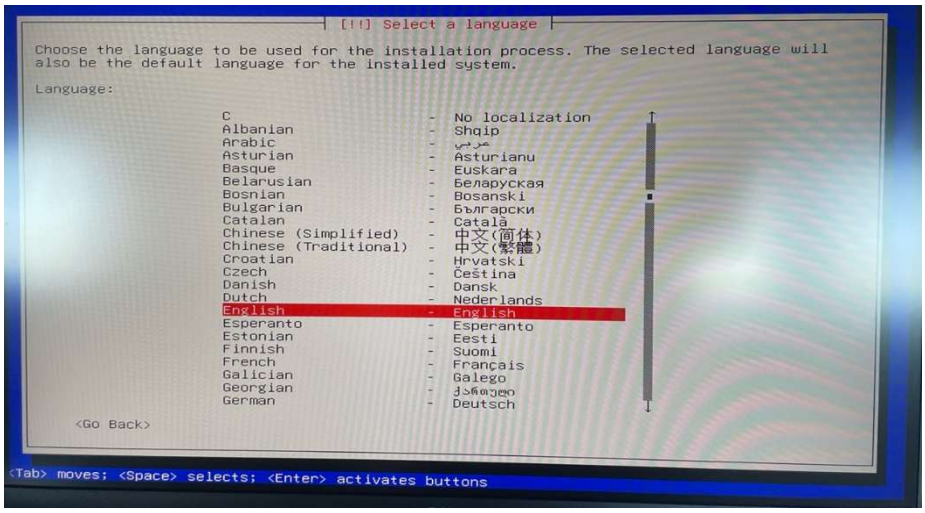
On the partitioning screen, the installer displayed a warning: More than one storage device has been detected. This is important the server had multiple drives visible (the two SATA disks, the RAID5 virtual disk from the Areca controller, and the two USB sticks). Selecting the wrong device here would have overwritten data drives.

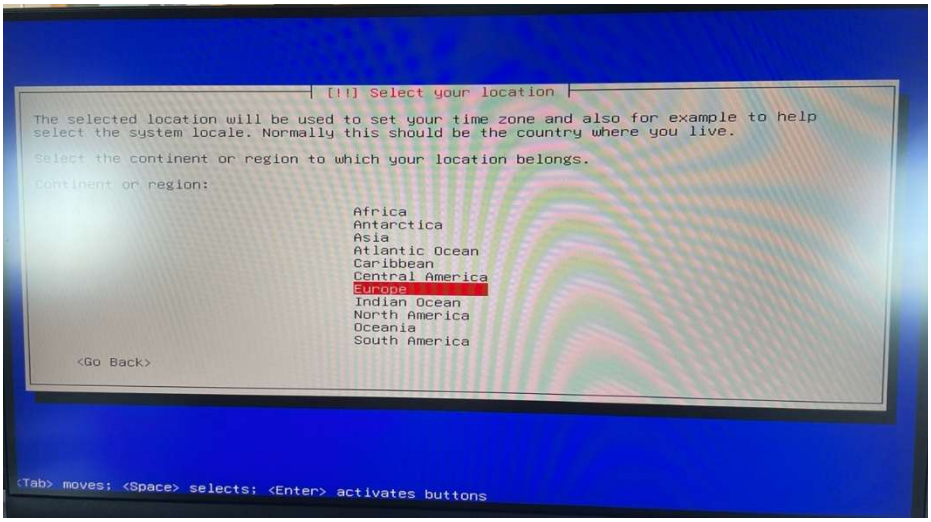
The installer also detected that the firmware had started in UEFI mode but found existing BIOS-mode installations on the system. We chose No to not force UEFI installation, keeping the system in BIOS compatibility mode. This was the correct choice given the age of the server hardware.

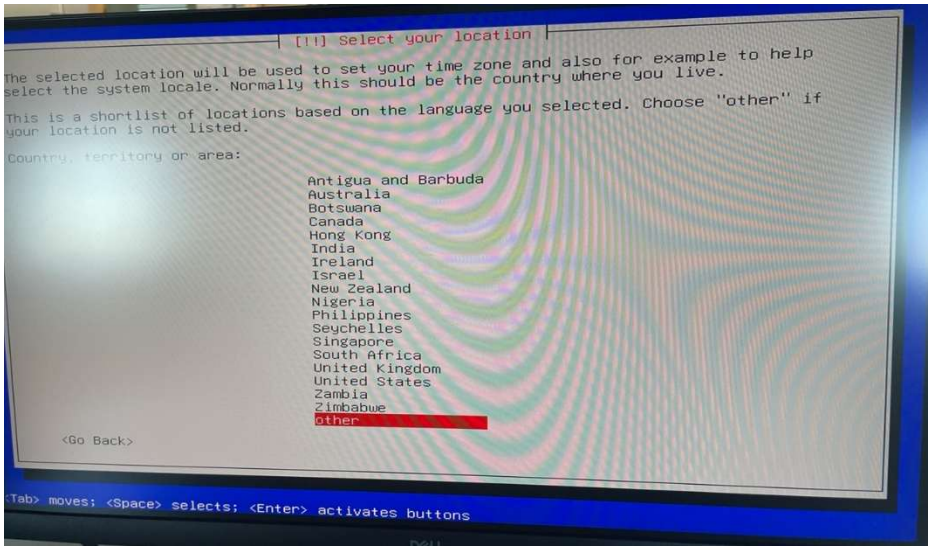
We carefully identified and selected the Samsung rear USB stick as the installation target (device /dev/sde). The installer formatted partition 2 of /dev/sde as ext4 for the root filesystem and installed the GRUB bootloader directly onto /dev/sde. Both operations completed successfully.

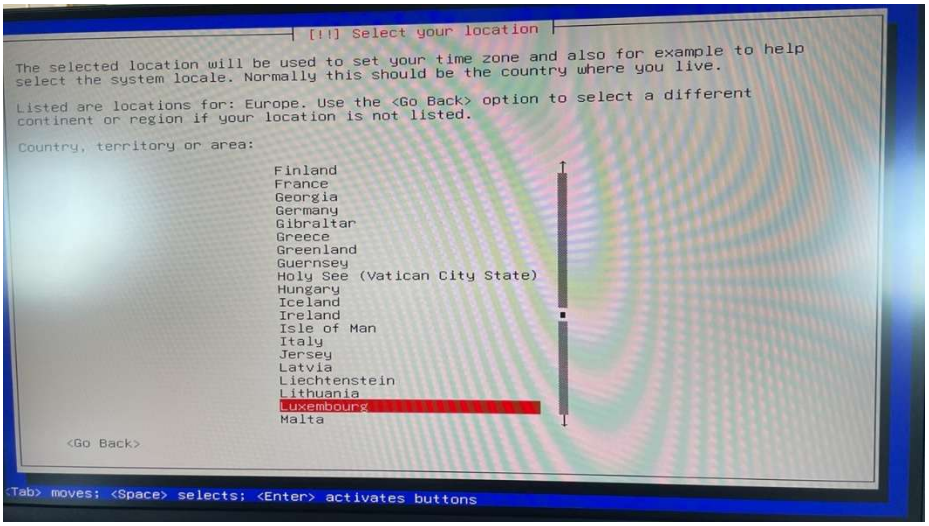
After installation completed, we removed the front installer USB and rebooted. The server loaded OMV from the rear USB stick and the web interface became accessible at http://10.0.13.2.

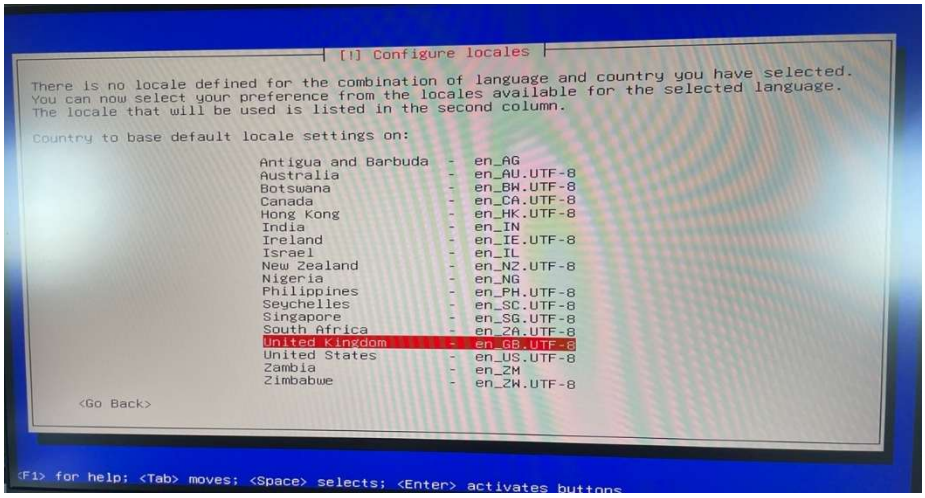
6.3.5 Photos

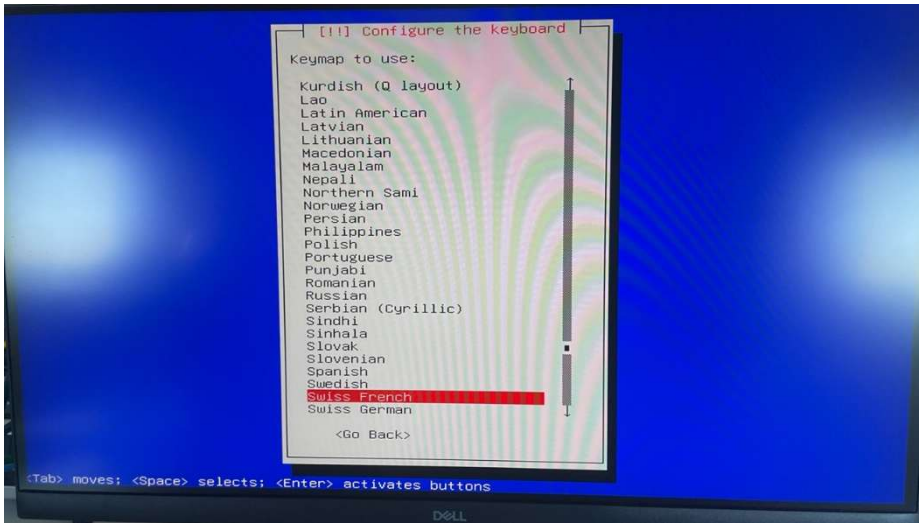

[!!] Select a language
Choose the language to be used for the installation process. The selected language will also be the default language for the installed system.
Language:
C - No localization
Albanian - Shqip
Arabic - عربي
Asturian - Asturianu
Basque - Euskara
Belarusian - Беларуская
Bosnian - Bosanski
Bulgarian - Български
Catalan - Català
Chinese (Simplified) - 中文(简体)
Chinese (Traditional) - 中文(繁體)
Croatian - Hrvatski
Czech - Čeština
Danish - Dansk
Dutch - Nederlands
English - English
Esperanto - Esperanto
Estonian - Eesti
Finnish - Suomi
French - Français
Galician - Galego
Georgian - ქართული
German - Deutsch
<Go Back>
<Tab> moves; <Space> selects; <Enter> activates buttons

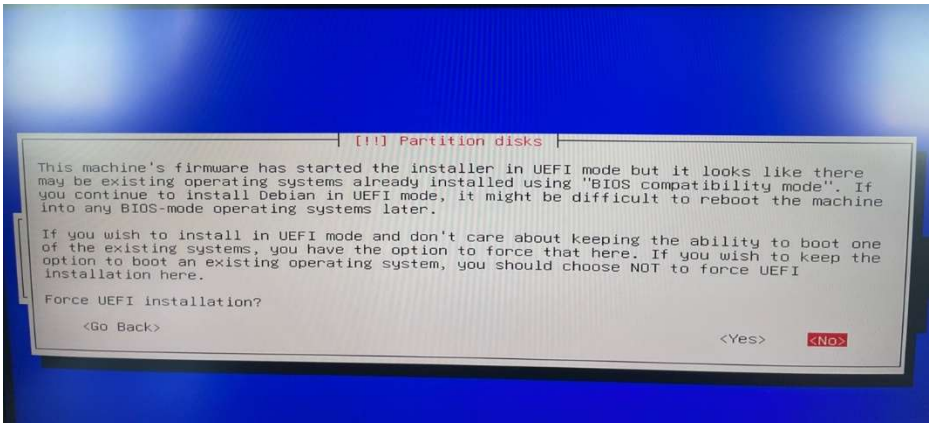

[!!] Select your location
The selected location will be used to set your time zone and also for example to help select the system locale. Normally this should be the country where you live.
Select the continent or region to which your location belongs.
Continent or region:
Africa
Antarctica
Asia
Atlantic Ocean
Caribbean
Central America
Europe
Indian Ocean
North America
Oceania
South America
<Go Back>
<Tab> moves; <Space> selects; <Enter> activates buttons

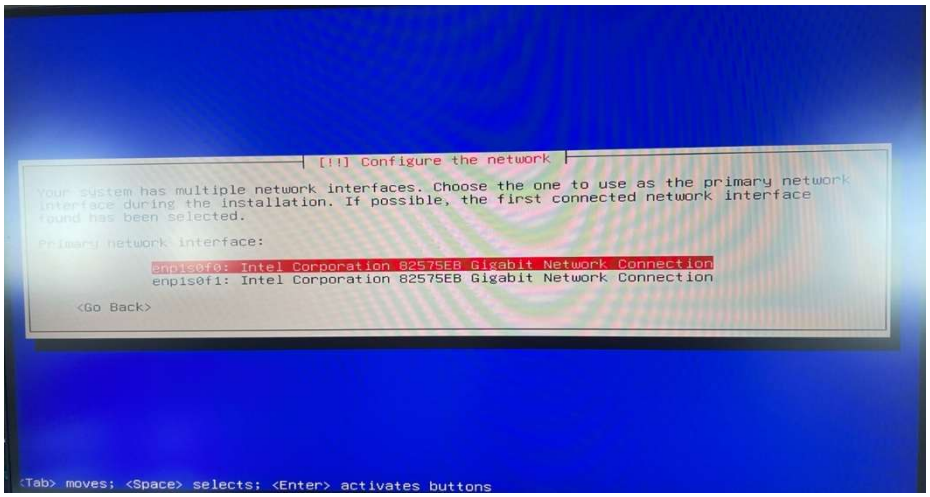

[!!] Select your location
The selected location will be used to set your time zone and also for example to help select the system locale. Normally this should be the country where you live.
This is a shortlist of locations based on the language you selected. Choose "other" if your location is not listed.
Country, territory or area:
Antigua and Barbuda
Australia
Botswana
Canada
Hong Kong
India
Ireland
Israel
New Zealand
Nigeria
Philippines
Seychelles
Singapore
South Africa
United Kingdom
United States
Zambia
Zimbabwe
other
<Go Back>
<Tab> moves; <Space> selects; <Enter> activates buttons

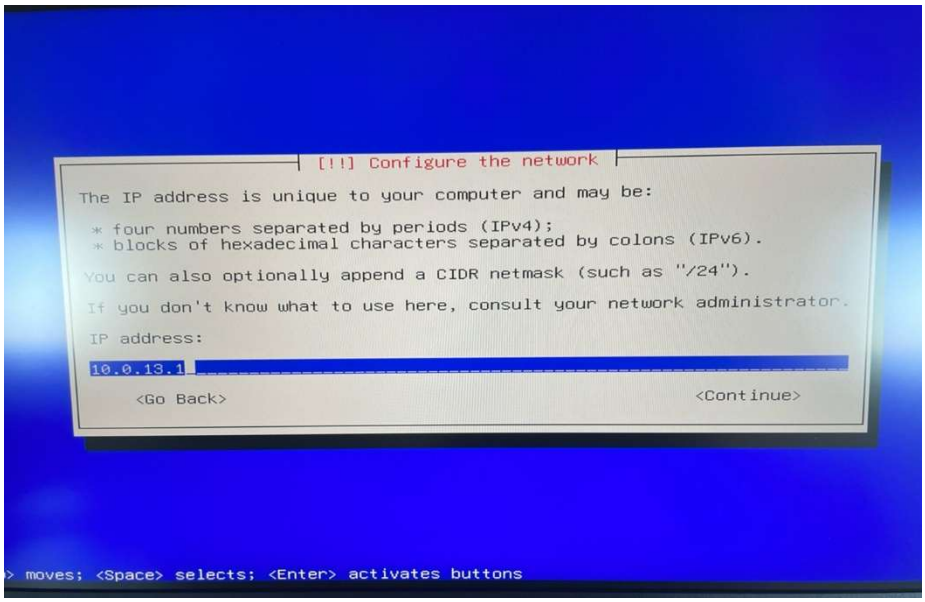

[!!] Select your location
The selected location will be used to set your time zone and also for example to help select the system locale. Normally this should be the country where you live.
Listed are locations for: Europe. Use the <Go Back> option to select a different continent or region if your location is not listed.
Country, territory or area:
Finland
France
Georgia
Germany
Gibraltar
Greece
Greenland
Guernsey
Holy See (Vatican City State)
Hungary
Iceland
Ireland
Isle of Man
Italy
Jersey
Latvia
Liechtenstein
Lithuania
Luxembourg
Malta
<Go Back>
<Tab> moves; <Space> selects; <Enter> activates buttons

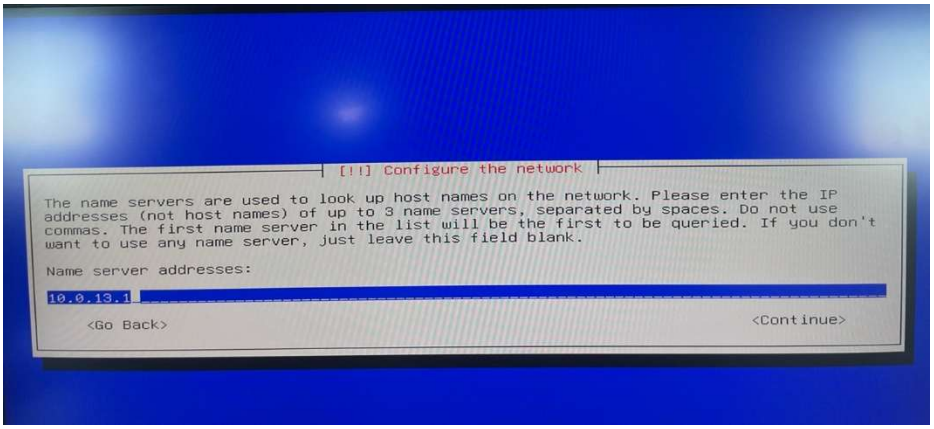

[!] Configure locales
There is no locale defined for the combination of language and country you have selected. You can now select your preference from the locales available for the selected language. The locale that will be used is listed in the second column.
Country to base default locale settings on:
Antigua and Barbuda - en_AG
Australia - en_AU.UTF-8
Botswana - en_BW.UTF-8
Canada - en_CA.UTF-8
Hong Kong - en_HK.UTF-8
India - en_IN
Ireland - en_IE.UTF-8
Israel - en_IL
New Zealand - en_NZ.UTF-8
Nigeria - en_NG
Philippines - en_PH.UTF-8
Seychelles - en_SC.UTF-8
Singapore - en_SG.UTF-8
South Africa - en_ZA.UTF-8
United Kingdom - en_GB.UTF-8
United States - en_US.UTF-8
Zambia - en_ZM
Zimbabwe - en_ZW.UTF-8
<Go Back>
<F1> for help; <Tab> moves; <Space> selects; <Enter> activates buttons

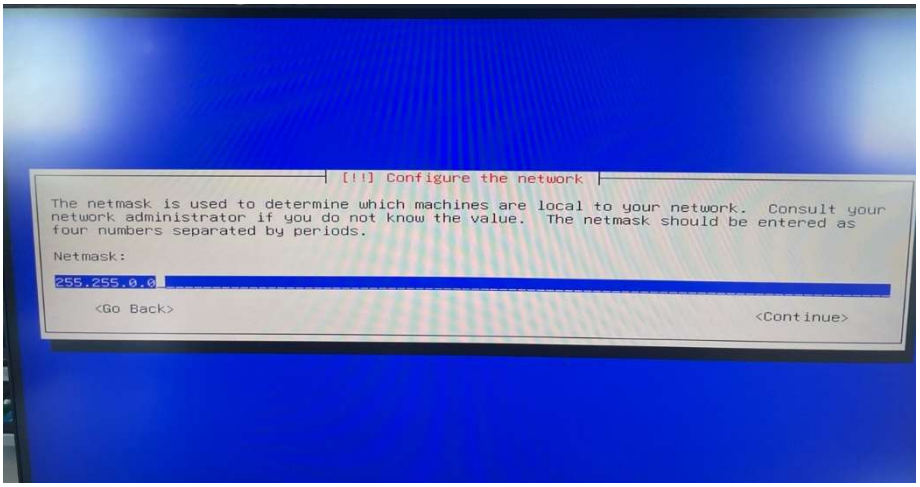

[!!] Configure the keyboard
Keymap to use:
Kurdish (Q layout)
Lao
Latin American
Latvian
Lithuanian
Macedonian
Malayalam
Nepali
Northern Sami
Norwegian
Persian
Philippines
Polish
Portuguese
Punjabi
Romanian
Russian
Serbian (Cyrillic)
Sindhi
Sinhala
Slovak
Slovenian
Spanish
Swedish
Swiss French
Swiss German
<Go Back>
<Tab> moves; <Space> selects; <Enter> activates buttons

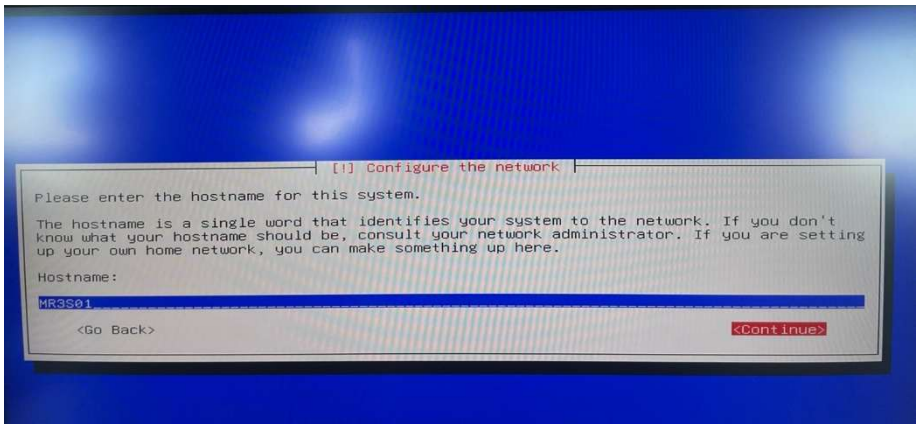

[!!] Partition disks
This machine's firmware has started the installer in UEFI mode but it looks like there may be existing operating systems already installed using "BIOS compatibility mode". If you continue to install Debian in UEFI mode, it might be difficult to reboot the machine into any BIOS-mode operating systems later.
If you wish to install in UEFI mode and don't care about keeping the ability to boot one of the existing systems, you have the option to force that here. If you wish to keep the option to boot an existing operating system, you should choose NOT to force UEFI installation here.
Force UEFI installation?
<Go Back>
<Yes>
<No>

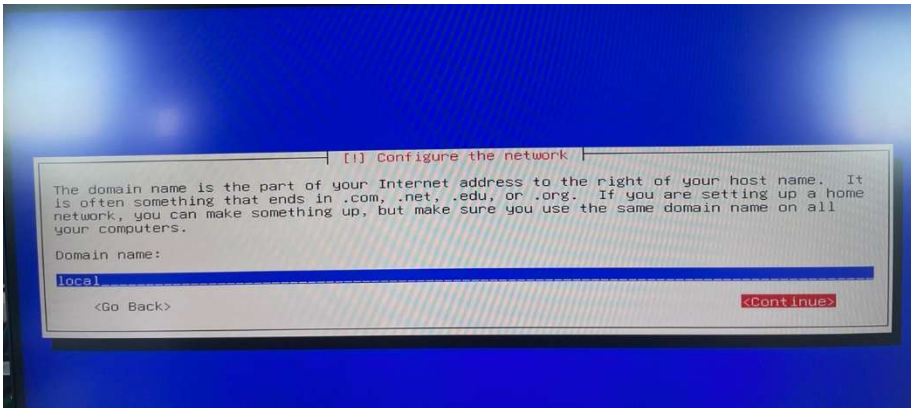

[!!] Configure the network
Your system has multiple network interfaces. Choose the one to use as the primary network interface during the installation. If possible, the first connected network interface found has been selected.
Primary network interface:
enp1s0f0: Intel Corporation 82575EB Gigabit Network Connection
enp1s0f1: Intel Corporation 82575EB Gigabit Network Connection
<Go Back>
<Tab> moves; <Space> selects; <Enter> activates buttons

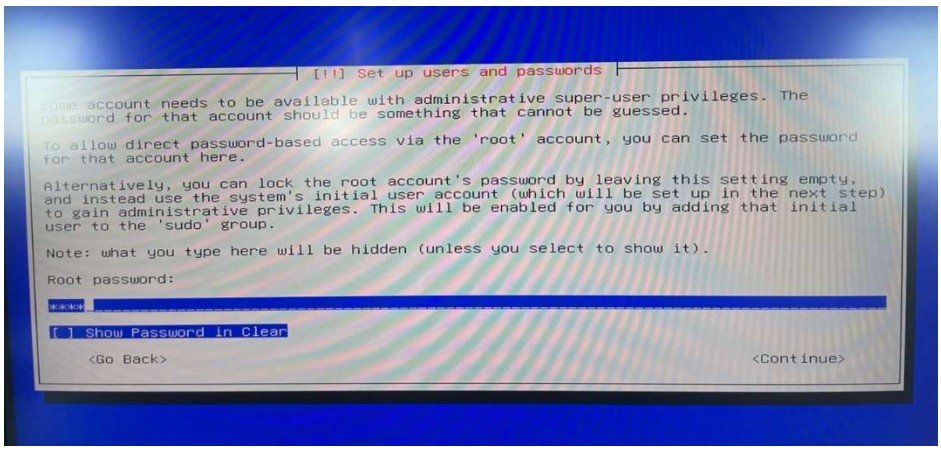

[!!] Configure the network
The IP address is unique to your computer and may be:
* four numbers separated by periods (IPv4);
* blocks of hexadecimal characters separated by colons (IPv6).
You can also optionally append a CIDR netmask (such as "/24").
If you don't know what to use here, consult your network administrator.
IP address:
10.0.13.1
<Go Back>
<Continue>
moves; <Space> selects; <Enter> activates buttons

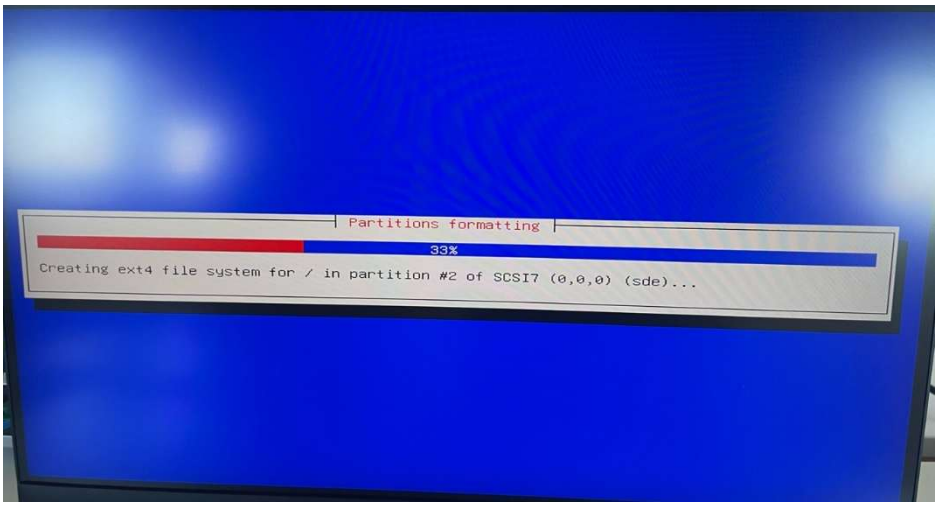

[!!] Configure the network
The name servers are used to look up host names on the network. Please enter the IP addresses (not host names) of up to 3 name servers, separated by spaces. Do not use commas. The first name server in the list will be the first to be queried. If you don't want to use any name server, just leave this field blank.
Name server addresses:
10.0.13.1
<Go Back>
<Continue>


[!!] Configure the network
The netmask is used to determine which machines are local to your network. Consult your network administrator if you do not know the value. The netmask should be entered as four numbers separated by periods.
Netmask:
255.255.0.0
<Go Back>
<Continue>


[!] Configure the network
Please enter the hostname for this system.
The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.
Hostname:
MR3S01
<Go Back>
<Continue>


[!] Configure the network
The domain name is the part of your Internet address to the right of your host name. It is often something that ends in .com, .net, .edu, or .org. If you are setting up a home network, you can make something up, but make sure you use the same domain name on all your computers.
Domain name:
local
<Go Back>
<Continue>


[!!] Set up users and passwords
Some account needs to be available with administrative super-user privileges. The password for that account should be something that cannot be guessed.
To allow direct password-based access via the 'root' account, you can set the password for that account here.
Alternatively, you can lock the root account's password by leaving this setting empty, and instead use the system's initial user account (which will be set up in the next step) to gain administrative privileges. This will be enabled for you by adding that initial user to the 'sudo' group.
Note: what you type here will be hidden (unless you select to show it).
Root password:
[ ] Show Password in Clear
<Go Back>
<Continue>


Partitions formatting
33%
Creating ext4 file system for / in partition #2 of SCSI7 (0,0,0) (sde)...

7. Rack Installation

With the server configured and tested at the bench, the next step was installing it into the 19-inch server rack. This stage required coordination with the other teams to plan the layout of the rack correctly.

7.1 Pre-Installation Planning

We first mapped out which U positions in the rack were allocated to which team's server. Proper planning prevents mechanical conflicts (servers blocking each other's rails or cable runs) and ensures enough airflow. In a real data centre, airflow direction cold air intake at the front, hot air exhaust at the rear dictates how servers should be orientated and spaced.

7.2 Rail Installation

The sliding rails were installed horizontally inside the rack. These rails need to be clicked into both the front and rear vertical rails of the rack at the same U position on each side, otherwise the server will not slide in straight. We double-checked that both rails were level before proceeding.

7.3 Mounting the Server

The server was carefully slid onto the installed rails from the front of the rack. Rails use a spring-latch mechanism the server needs to be pushed in fully until the front ears snap into the rack posts and the latch engages. We confirmed the server was fully seated and could not be pulled forward without pressing the release latches.

7.4 Cable Management

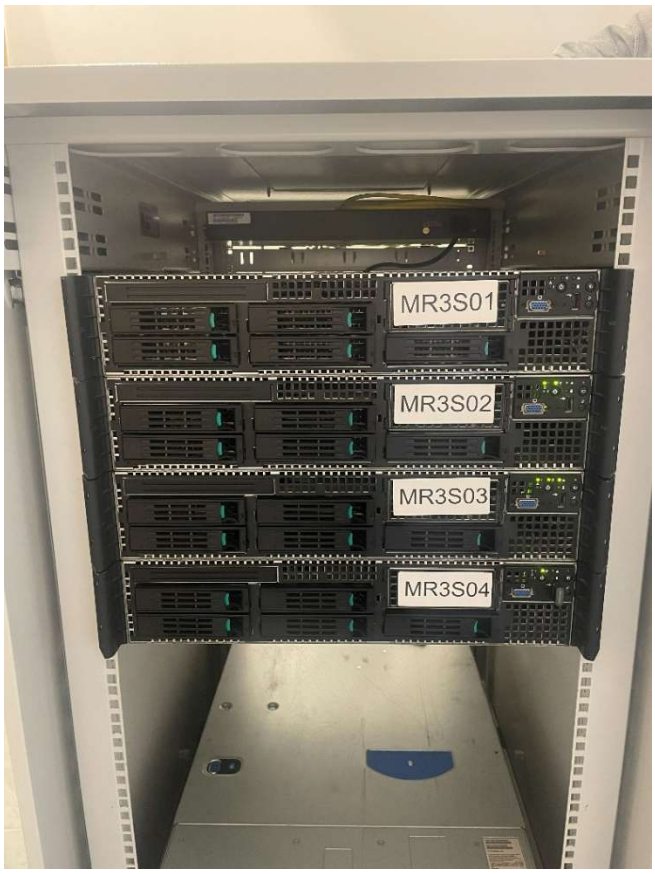
Once the server was physically mounted, we connected the required cables at the rear:

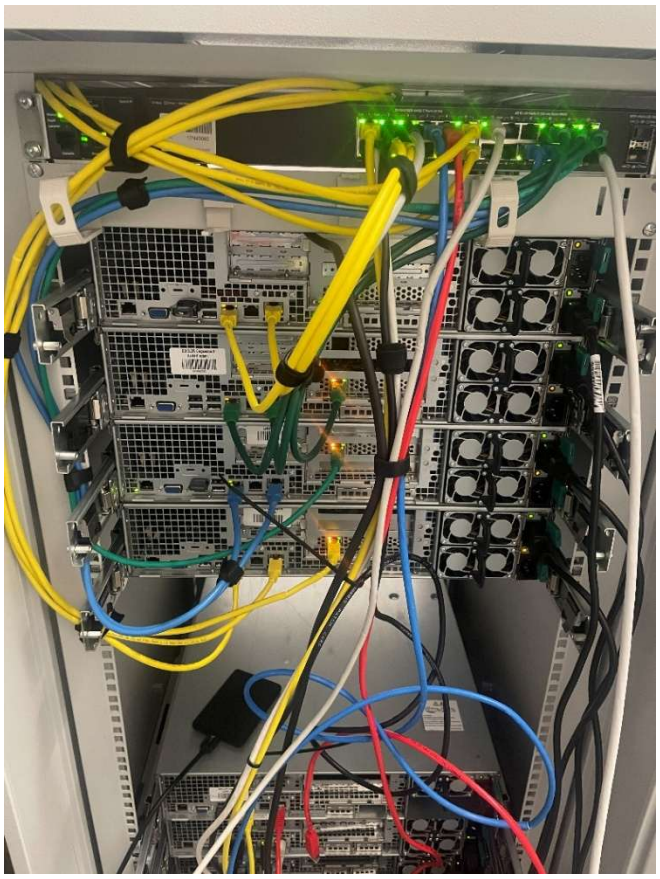
- 1 Ethernet cable from the mainboard LAN1 port to the lab network switch providing normal network access at 10.0.13.2.
- 1 Ethernet cable from the RMM port to the management network allowing remote access to the server console at 192.168.0.40 even when the OS is down.
- 1 Ethernet cable from the Areca RAID controller port enabling access to the RAID management web interface at 192.168.0.140.
- 2 power cables from both PSUs to the rack PDU (Power Distribution Unit).

7.5 Remote Management Module (RMM) Connection

We tested the RMM connection by opening a browser and navigating to 192.168.0.40. The RMM (essentially Intel's equivalent of IPMI/iDRAC) presented a web interface where we could view the server's console remotely, see hardware sensor readings (temperatures, fan speeds, voltages), and power cycle the machine without physical access. This is a critical feature in a real server environment where physical access to the rack may be restricted or inconvenient

7.6 Photos





8. NAS Configuration and Testing

8.1 File Systems — Mounting Both Volumes

After logging into the OMV web interface at http://10.0.13.2, the first step was to get both RAID volumes mounted and visible to the OS. Under Storage > File Systems, OMV showed two devices available:

Device	Filesystem Type	Volume Tag	Available Space	RAID Source
/dev/sda	BTRFS	software_raid1	296.08 GiB	Software BTRFS RAID1 (2x WD 149 GiB)
/dev/sdb1	EXT4	areca_raid5	255.59 GiB	Hardware RAID5 (3x Seagate SAS 146 GiB)

Both filesystems were mounted successfully (checkmark in the Mounted column, Status: Available). The usage warning threshold was set to 85% so that OMV sends an alert before the volumes run completely full.

8.2 Creating the Software BTRFS RAID 1 Volume (software_raid1)

For the software RAID volume we used OMV's built-in BTRFS filesystem support rather than Linux mdadm. BTRFS is a modern copy-on-write filesystem that has native RAID capabilities built directly into the filesystem layer meaning RAID and filesystem management are handled together, which simplifies the setup compared to layering mdadm underneath a separate filesystem.

The creation was done through Storage > File Systems > Create, selecting type BTRFS and profile RAID1. We selected both SATA drives as members:

-
- WDC WD1600AAJS-07M0A0 [/dev/sda, 149.05 GiB]
-
- WDC WD1600AAJS-07M0A0 [/dev/sdc, 149.05 GiB]

With BTRFS RAID1, every data block is written to two different devices simultaneously, which is the BTRFS equivalent of mirroring. If one of the two SATA drives fails, the other continues serving data without interruption. After creation, the volume was tagged software_raid1 and mounted from Storage > File Systems > Mount, with the usage warning threshold set to 85%.

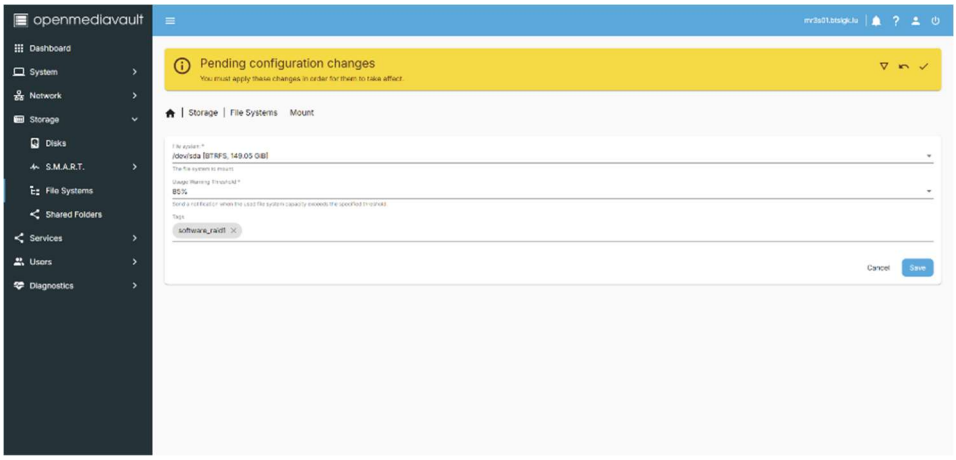


Figure 1 Creation of the file system

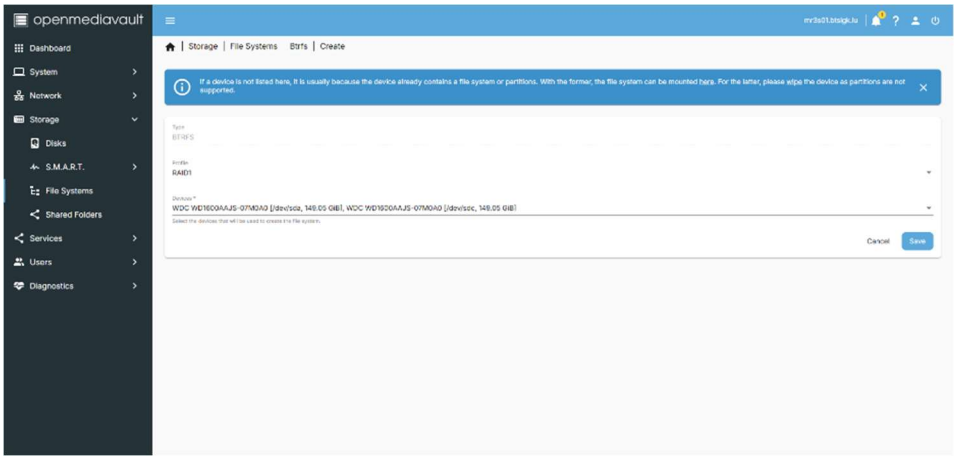


Figure 2 Mountig the file system

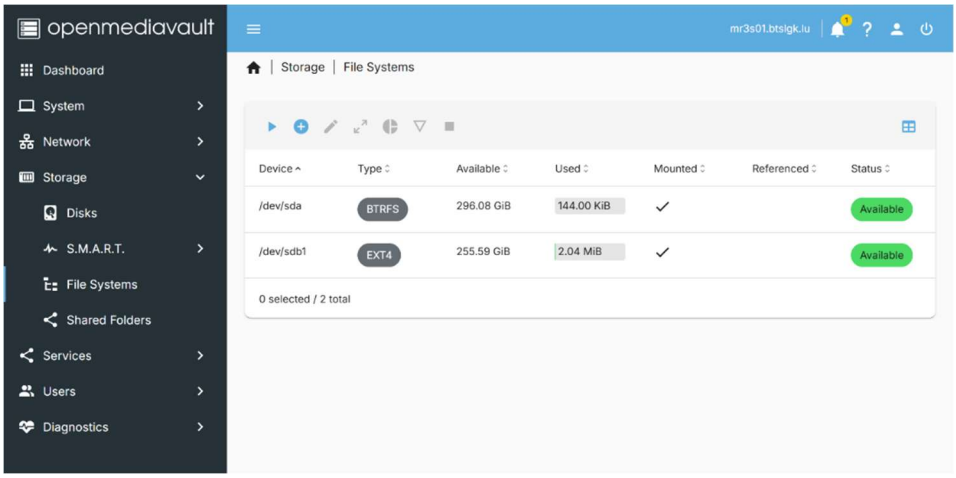


Figure 3 overview of the mounted file systems

8.3 Family Home Storage — Shared Folder Design

We were asked to design a complete file sharing solution for a family wanting to use the server as central home storage, with mixed access rights and support for Windows, Linux, and macOS clients.

8.3.1 Actual Shared Folder Structure

All shared folders were created under Storage > Shared Folders. Personal folders for each family member were placed on /dev/sdb1 (areca_raid5, EXT4), benefiting from hardware RAID5 redundancy. The backup folder was placed on /dev/sda (software_raid1, BTRFS) to give it a second, separate layer of protection on different physical disks.

Shared Folder Name	Device	Purpose	Access
andrea	/dev/sdb1 [areca_raid5]	Private folder for Andrea	Andrea: Read/Write — All others: No access
christophe	/dev/sdb1 [areca_raid5]	Private folder for Christophe	Christophe: Read/Write — All others: No access
donny	/dev/sdb1 [areca_raid5]	Private folder for Donny	Donny: Read/Write — All others: No access
marios	/dev/sdb1 [areca_raid5]	Private folder for Marios	Marios: Read/Write — All others: No access
shared-media-folder	/dev/sdb1 [areca_raid5]	Common shared media (photos, videos, etc.)	All users: Read/Write; Others: Read-only
backup	/dev/sda [software_raid1]	Centralised backup storage	Admin: Read/Write; clients: write

The folder creation screen in OMV showed the default permission template set to: Administrator: read/write, Users: read/write, Others: read-only. For personal folders these were then overridden through the per-folder Permissions panel to enforce private access.

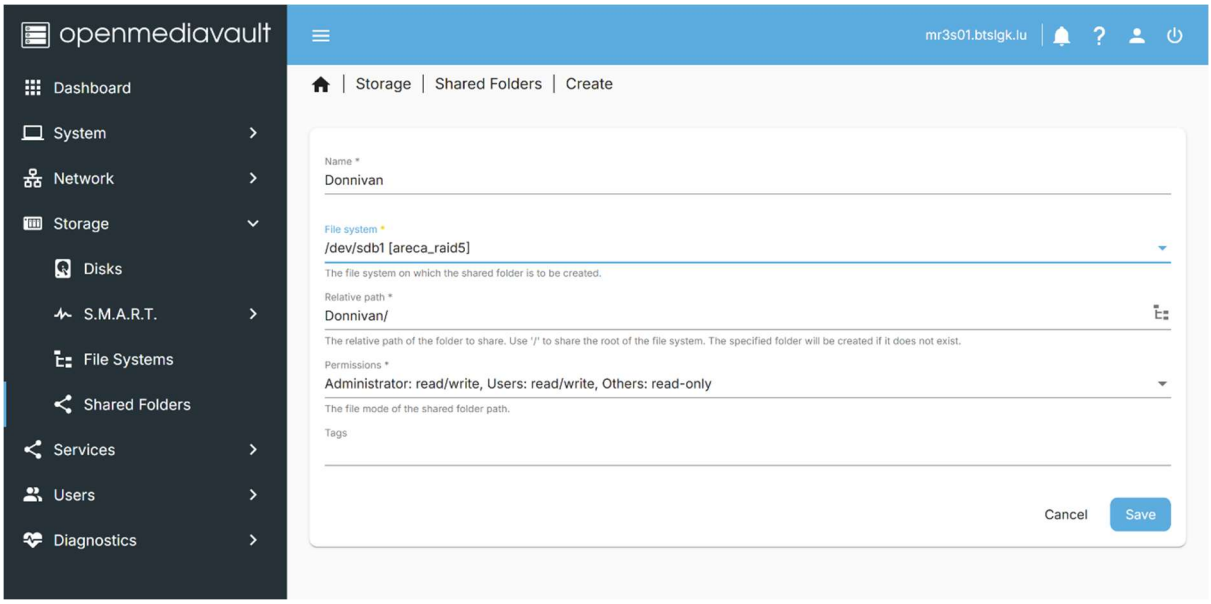


Figure 4 Creation of an folder

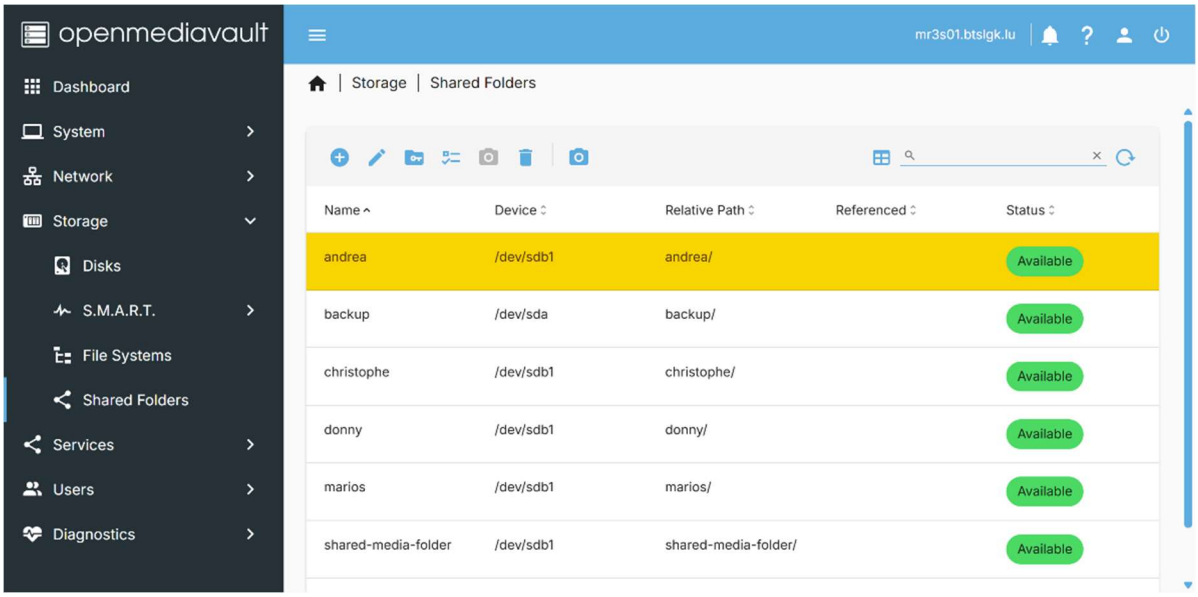


Figure 5 Overview of all folders

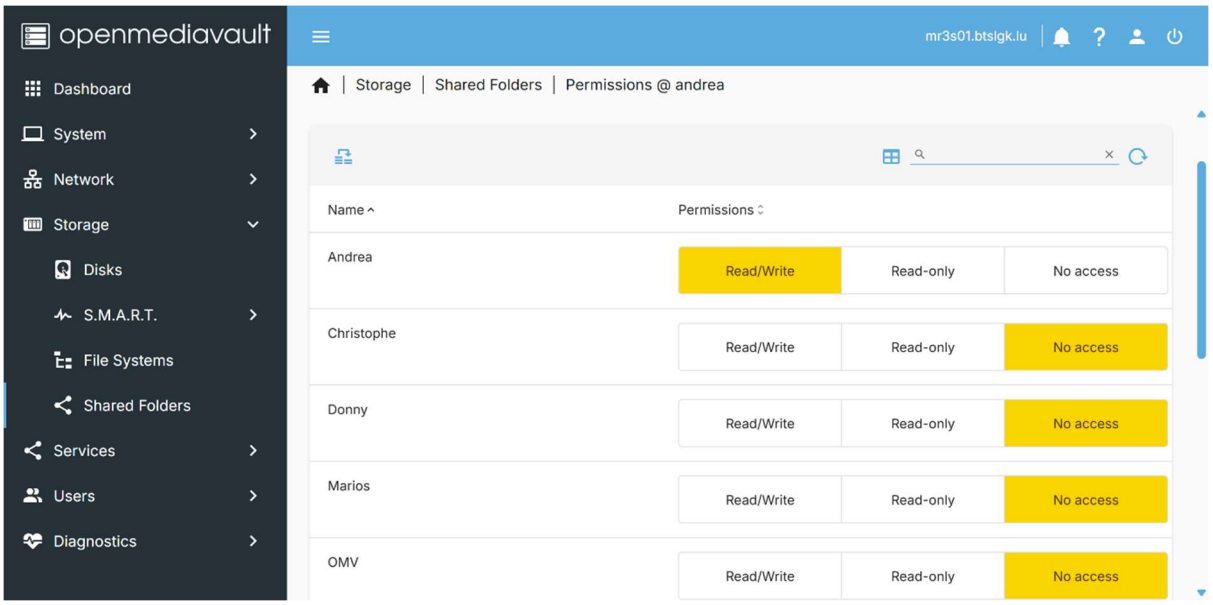


Figure 6 Permissions for a personal folder

8.3.2 User Accounts and Groups

We created four user accounts in OMV under User Management > Users one per family member: Andrea, Christophe, Donny, and Marios. A system user called OMV also exists for internal service management.

The group structure was kept clean and logical:

- family group: contains all four family members (Andrea, Christophe, Donny, Marios). This group was used to grant shared access to the shared-media-folder.
- Individual groups (andrea, christophe, donny, marios): each user has their own dedicated group this is standard Linux practice and is what OMV uses to enforce per-folder private permissions.
- OMV group: contains only the OMV system user for internal service tasks.

Each user was created with a password and the shell /usr/bin/sh. They were assigned to the family group at creation time, with their personal group added automatically.

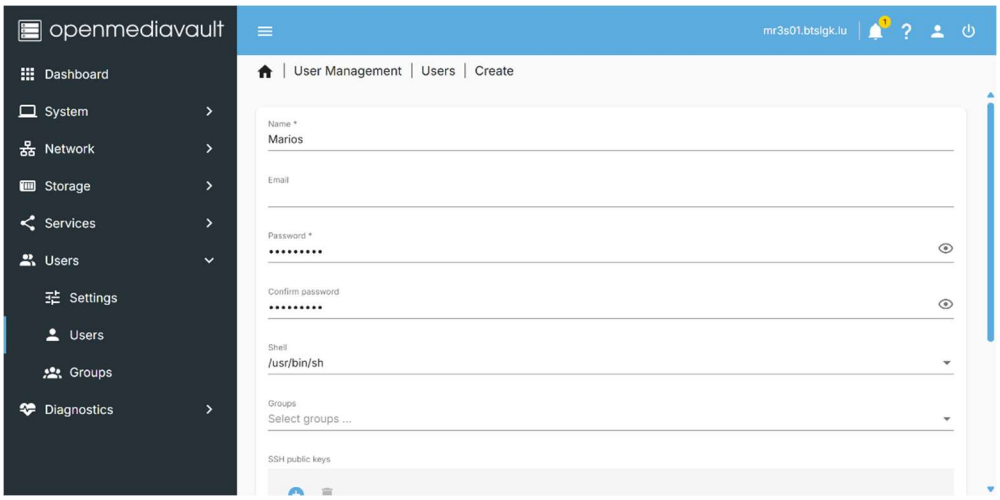


Figure 7 Creation of a user

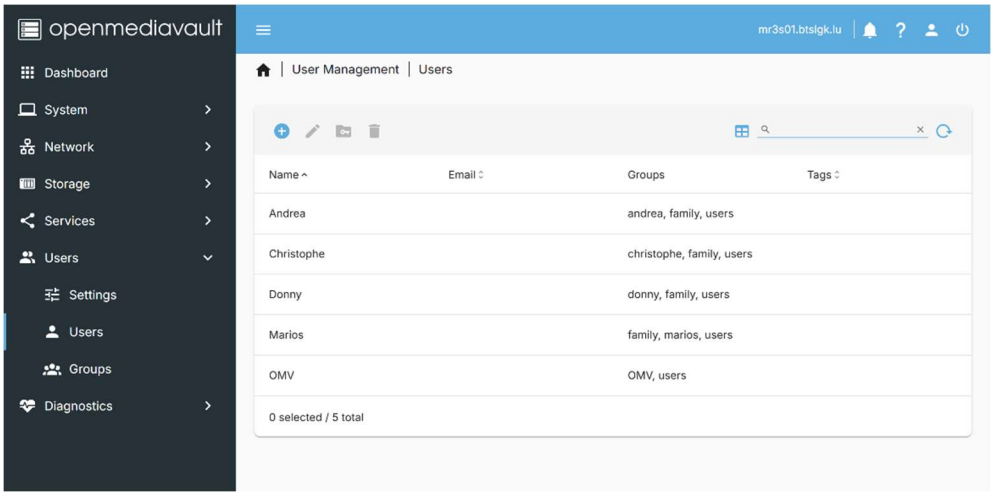


Figure 8 Overview of all users

8.3.3 Per-Folder Permission Configuration

The most important part of the configuration was setting the correct permissions on each personal folder. Taking the andrea folder as an example visible in the Permissions @ andrea screen the settings were:

- Andrea: Read/Write (highlighted as active)
- Christophe: No access
- Donny: No access
- Marios: No access
- OMV: No access

The same pattern was applied to all personal folders: only the owning user receives read/write access, and every other user is explicitly set to no access. This ensures that even if a user is connected to the server via SMB, they cannot see or open another family member's private folder. The shared-media-folder had a more open permission set, allowing all family members to read and write, making it the common collaborative space.

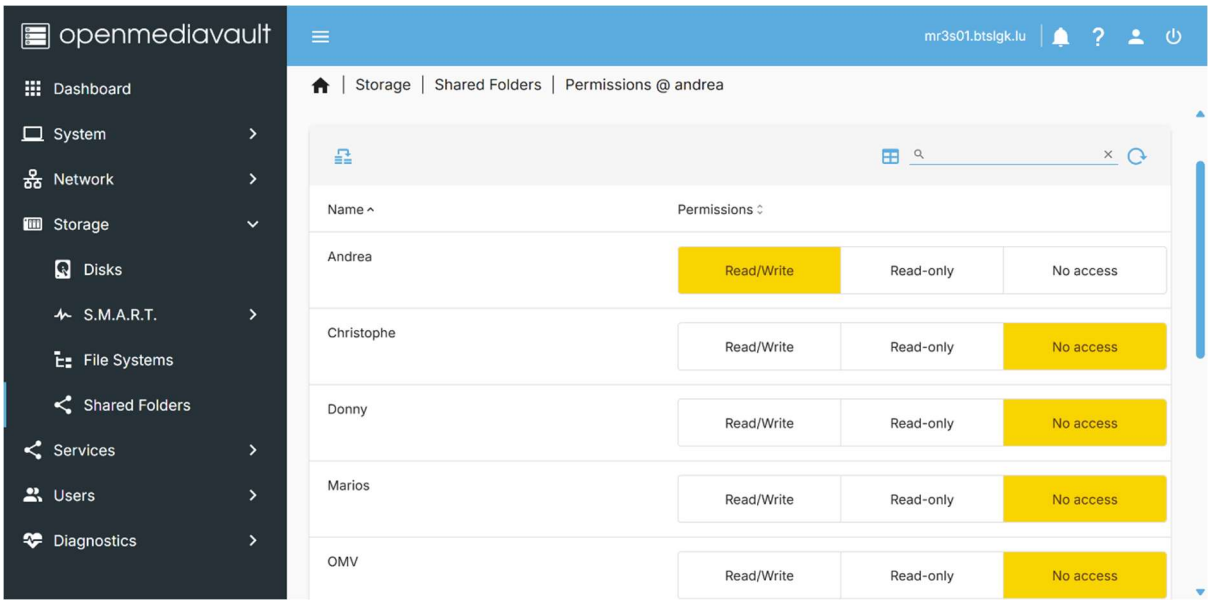


Figure 9 User permissions for his folder

8.3.4 Network Protocols

To support the variety of client devices (Windows, Linux, macOS) in the family scenario, we enabled the following protocols in OMV:

- SMB/CIFS: The primary protocol for Windows clients. OMV uses Samba under the hood. All shares were published as SMB shares, accessible from Windows Explorer with standard drive mapping or UNC path (\\10.0.13.2\andrea for example).
- NFS: Configured for Linux clients. NFS exports were created pointing to the same shared folders.
- SFTP/SSH: Enabled for scripted backup solutions and macOS/Linux terminal access.

macOS clients connect natively via SMB modern macOS has solid SMB support and does not require the older AFP protocol.

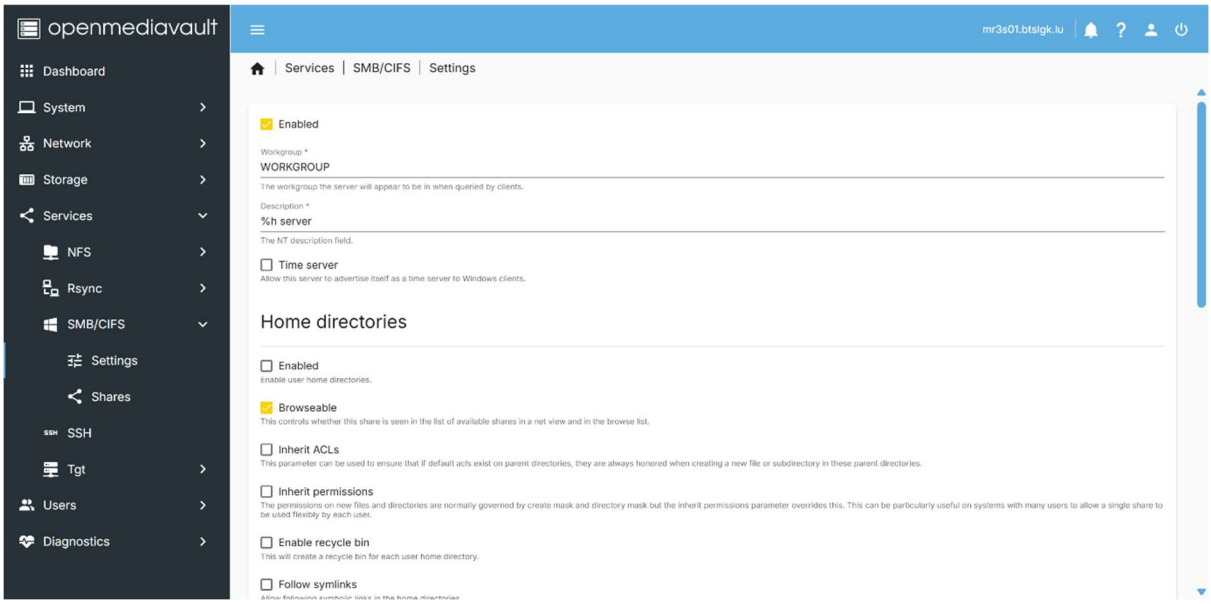


Figure 10 Enabling SMB

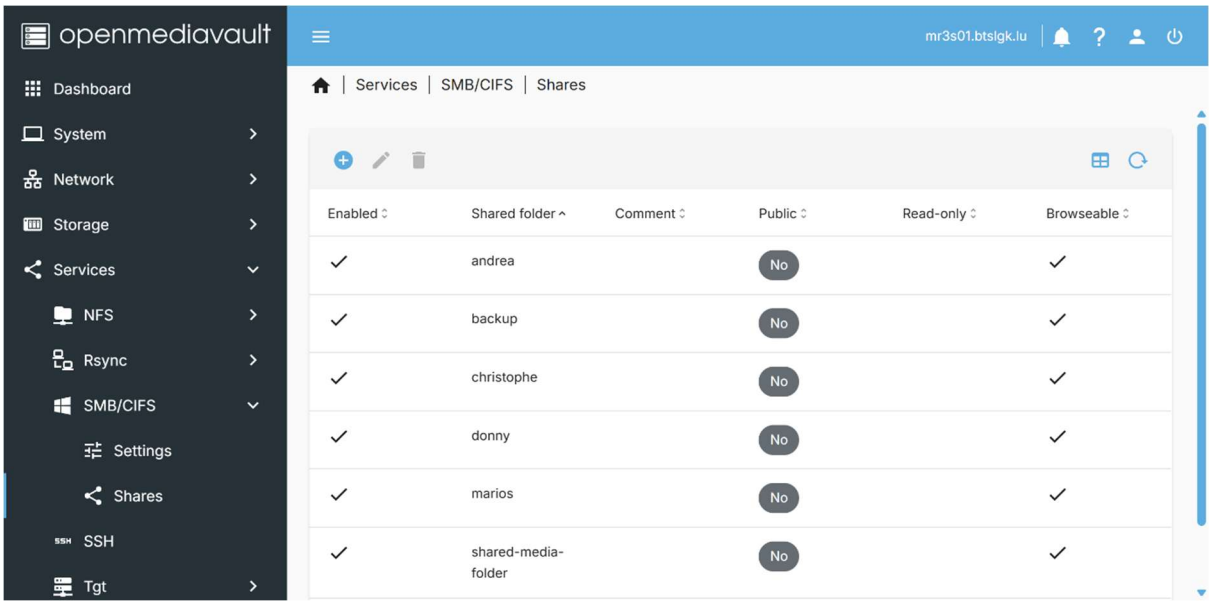


Figure 11 Overview of all shard folder

8.4 Testing File Access

We tested the full configuration from client machines on the lab network:

1. Opened File Explorer on a Windows client and navigated to \\10.0.13.2 all shared folders were visible in the share list.
2. Logged in as user Marios and confirmed read/write access to the marios personal folder and to shared-media-folder.
3. Attempted to open the andrea folder while logged in as Marios access was correctly denied, confirming the per-folder permission settings were working.
4. Uploaded a test image file to shared-media-folder and confirmed it was immediately accessible when logged in as a different user on a second machine.
5. Verified that the backup folder was accessible to the admin account and confirmed files could be written to it.

All permission rules behaved exactly as configured. The two RAID volumes remained mounted and stable throughout all tests.

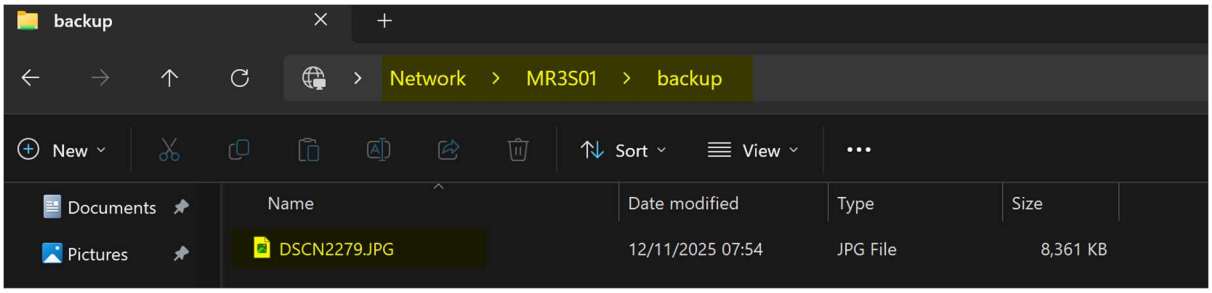


Figure 12 Testing of the backup folder

8.5 iSCSI Research and Testing

8.5.1 What is iSCSI?

iSCSI (Internet Small Computer Systems Interface) is a storage protocol that transmits standard SCSI commands over a regular IP network. Where NFS and SMB share files and

directories at the filesystem level, iSCSI presents a raw block device (essentially a virtual hard drive) to the client over the network. The client (called the initiator) then treats this block device exactly as if it were a locally attached disk it can partition it, format it with any filesystem, and use it for any purpose including running databases or virtual machine disk images.

Feature	NFS	SMB/CIFS	iSCSI
Access type	File-level	File-level	Block-level
Filesystem managed by	NAS server	NAS server	Client (initiator)
Multi-client write	Yes (with locking)	Yes (with locking)	Normally single-client
Use case	Linux shared files	Windows shared files	VMs, databases, raw storage
Protocol layer	IP (TCP/UDP)	IP (TCP)	IP (TCP)

8.5.2 iSCSI in OpenMediaVault

OMV supports iSCSI through the openmediavault-iscsitarget plugin, which is available in the OMV plugin repository. After installing the plugin from the web interface, we could define iSCSI targets and present logical unit numbers (LUNs) essentially virtual disk images backed by our RAID volumes.

We created a test LUN of 5 GB backed by the areca_raid5 volume, defined an iSCSI target, and connected to it from a Windows client using the built-in Windows iSCSI Initiator tool (Start > iSCSI Initiator). Once connected, the LUN appeared in Windows Disk Management as a new unformatted disk. We formatted it as NTFS and wrote a test file to confirm the iSCSI connection was working correctly.

The main practical difference we noticed compared to SMB: the iSCSI disk behaved exactly like a local drive Windows assigned it a drive letter and we could set NTFS permissions directly on it. However, it could only safely be used by one client at a time, unlike SMB shares which multiple users can access simultaneously.

8.5.3 Screenshots

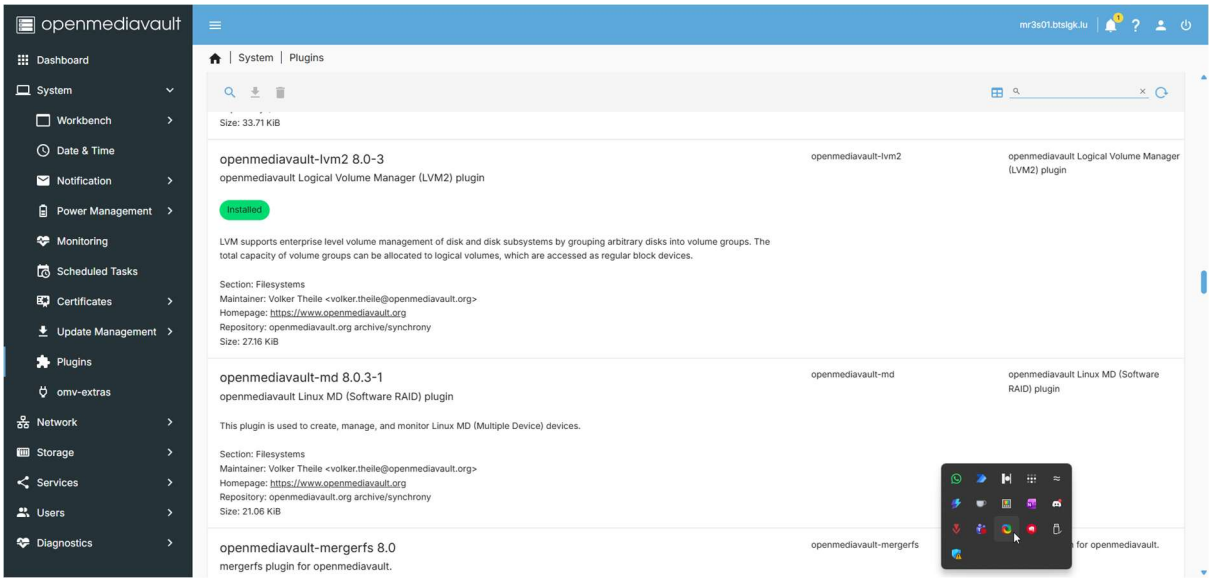


Figure 13 Download of the LVM pluggin

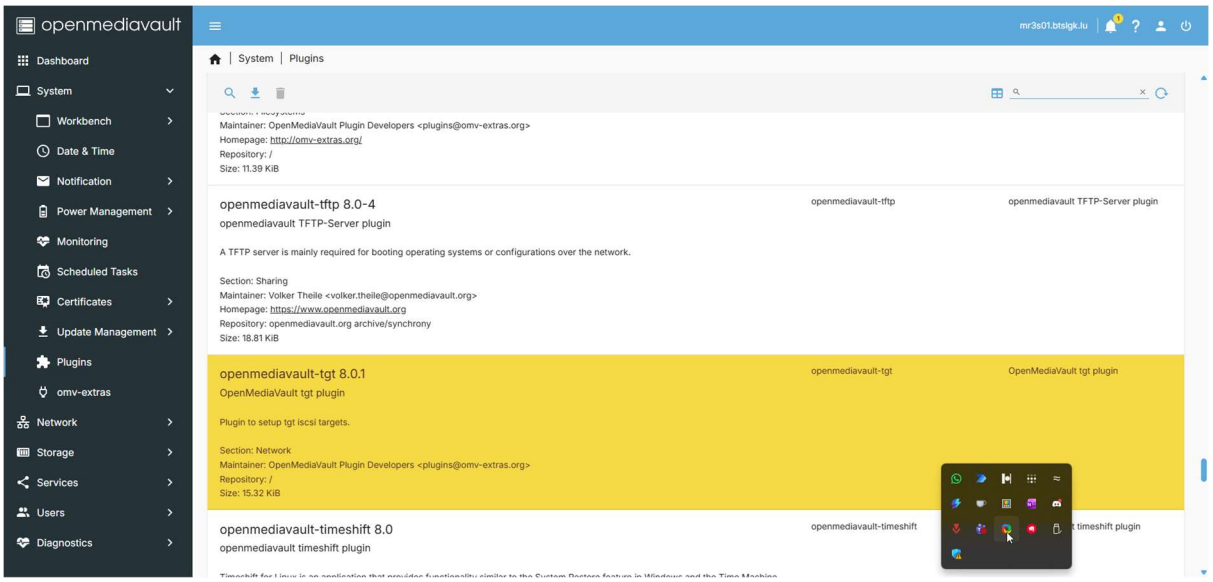


Figure 14 Download of the tgt pluggin

9. Project Work Log

The following section documents the day-by-day progress of the project, including the problems we ran into along the way and how each one was resolved. Troubleshooting real hardware and software issues made up a significant part of the practical experience, and it is documented here honestly alongside the progress made.

Session	Summary of Work
Day 1	Research, topic questions, and team planning.
Day 2	Hardware installation, BIOS verification, RAID 5 configuration. Issue: RAID initialisation frozen resolved by starting it from the on-board RAID BIOS.
Day 3	OS installation preparation. Issue 1: Rufus could not detect the USB ISO was saved directly onto it. Moving the ISO to the local drive fixed it. Issue 2: Server would not boot from the installation USB despite multiple sticks and both Rufus / balenaEtcher.Carried over to Day 4.
Day 4	Resolved boot issue with teacher: DD image mode + MBR format in Rufus allowed successful boot. OMV installed onto rear USB. Issue: Network settings were read-only in OMV cause was browsing as a non-admin. Logging in as admin unlocked all settings; network configured as required.
Day 5	RAID sharing and folder configuration. Issue: RMM interface refused connection an existing session from another browser was blocking access. Closing it resolved the problem. Configuration of RAID volumes and shared folder structure began.
Day 6	Completed all shared folders, user accounts, groups, and permissions. Issue: Users could not open their own personal folders permissions had not been fully applied. Resolved after saving ACL settings correctly. Documentation started.
Day 7	Start of testing/configure of iSCSI. Issue: iSCSI needs a clean hard drive, but the two hard drives are used for the RAID1.

Day 1 — Research and Planning

The first session was spent entirely on preparation. We worked through all of the research questions in the assignment document covering what a NAS is and what it is used for, how it compares to DAS and SAN, and why professional environments keep compute and storage on separate systems. Having a clear answer to each of those questions before touching the hardware made the later practical steps easier to understand in context.

We also used this time to plan how we would approach the remaining sessions agreeing on which tasks to tackle in what order (hardware first, then RAID, then OS, then NAS config) and how to split the workload between ourselves.

Day 2 — Hardware Installation and BIOS Verification

We started the physical work by installing all hardware components into the server: the eight RAM modules, the two native SATA drives, and the Areca RAID controller with its three SAS drives. Once everything was assembled and the server was connected to a monitor and keyboard, we powered it on for the first time.

The BIOS check confirmed that all components had been detected correctly — both CPUs, all eight RAM sticks, and the attached storage devices were visible with no POST errors. With the hardware verified, we moved on to RAID configuration.

The Areca controller is managed through a browser-based web interface. To access it we:

- Connected a laptop directly to the dedicated RAID controller network port on the rear panel.
- Made sure the laptop was configured with an IP on the same subnet (192.168.0.x).
- Opened a browser and navigated to 192.168.0.139:80.

We created the RAID 5 array through the web UI and named the volume areca_raid5. However, after saving the configuration, the initialisation progress bar did not move — the process appeared completely frozen. After some investigation, we found the cause: the initialisation had to be explicitly triggered from the RAID controller's own BIOS menu, which is accessible during server boot before the OS loads. Triggering it from the web interface alone was not sufficient. Once we restarted the server, entered the RAID BIOS, and confirmed the initialisation there, the process started and completed without any further issues.

Day 3 — OS Installation Preparation (With Complications)

The goal for this session was to create a bootable USB installer for OpenMediaVault and use it to install OMV onto the rear Samsung USB stick. The process ran into several obstacles.

The first issue appeared immediately when we opened Rufus: it could not see the USB drive we wanted to use as the installer. After checking the drive, we found the cause — the OMV ISO file had been downloaded directly onto that same USB stick, rather than to the computer's local drive. Because the drive was occupied and actively in use, Rufus could not detect it properly. Moving the ISO to the laptop's hard drive and re-plugging the USB stick resolved the detection issue, and Rufus was able to write the bootable image without any errors.

Despite that fix, the server still would not boot from the USB stick. We tried multiple different USB sticks in case one was faulty, tested different USB ports on the server, and re-created the boot drive using balenaEtcher as an alternative to Rufus — but none of these attempts resulted in a successful boot. The installer simply was not loading. We ran out of time in this session and carried the issue forward to the next day.

Day 4 — Resolving the Boot Issue and Configuring the Network

We picked up the OS installation problem with direct support from the teacher. After methodically testing more USB sticks and imaging options, the solution was eventually found: the ISO had to be written using DD image mode in Rufus, combined with an MBR (Master Boot Record) partition scheme, rather than the default ISO image mode that Rufus normally suggests. This distinction matters because the server's firmware expected a specific boot structure that only the DD+MBR combination produced correctly.

With the correctly formatted boot USB inserted into the front port, the server booted into the OMV installer successfully. We went through the full installation wizard:

- Language: English
- Location: Other > Europe > Luxembourg
- Locale: en_GB.UTF-8 (United Kingdom, as no Luxembourg English locale exists)
- Keyboard: Swiss French
- Network interface: enp1s0f0 (Intel 82575EB Gigabit)
- DHCP autoconfiguration: failed — manually entered static IP 10.0.13.2, netmask 255.255.0.0, gateway 10.0.0.1, DNS 10.0.0.1
- Hostname: MR3S01, Domain: local
- Root password: set and confirmed
- Partitioning: selected rear Samsung USB stick (/dev/sde) as target, declined forced UEFI installation to keep BIOS compatibility mode
- Formatter created an ext4 partition on /dev/sde and installed the GRUB bootloader on /dev/sde

After installation completed, we removed the front installer USB and rebooted. OMV loaded correctly from the rear USB stick and the web interface was accessible at http://10.0.13.2.

One further issue came up during the first login: the OMV network settings panel appeared read-only, and no configuration fields could be edited. The cause was straightforward — we had been browsing the interface without logging in as administrator. The OMV web GUI shows configuration panels to all visitors but only grants edit access once you authenticate with admin credentials. Once we logged in properly, every setting was editable and we confirmed the network configuration matched the values entered during installation.

Day 5 — RMM Access Issue and Volume Configuration

This session was focused on connecting to the Remote Management Module and progressing with the shared folder and RAID volume setup inside OMV. When we tried to open the RMM web interface at 192.168.0.40, the login page would not accept our credentials or behaved unexpectedly.

After checking IP settings and ruling out network configuration as the cause, we discovered that another browser session from a different machine was already authenticated into the RMM. The RMM only permits one active web session at a time — a design choice common in IPMI-style management interfaces to prevent simultaneous conflicting commands. The

existing session was silently blocking ours. Once the other session was closed, we connected to the RMM immediately and confirmed that hardware sensor readings, console access, and power control were all functioning correctly.

We then continued with the OMV configuration, mounting both RAID volumes in the File Systems panel and beginning the shared folder creation process.

Day 6 — Completing NAS Configuration and Beginning Documentation

The second to last practical session was used to complete everything that remained in the OMV configuration. We finished creating all six shared folders across the two RAID volumes the four personal folders (andrea, christophe, donny, marios) and the shared-media-folder on areca_raid5, and the backup folder on software_raid1. We created the four user accounts and the family group, then went through each personal folder to set the correct per-user ACL permissions.

One final issue appeared during permission testing: users were not able to open their own personal folders when connecting via SMB from a client machine. After reviewing the configuration, we found that the ACL settings in OMV had not been fully committed on the first save attempt — the permission changes need to be explicitly applied and the underlying Samba configuration updated before they take effect on active connections. After re-saving the permissions correctly, each user could access their own folder and was properly blocked from opening any other.

We also began writing this documentation during Day 6, working through the research notes, configuration steps, and screenshots captured across all sessions and compiling them into this report.

Day 7 — Completing NAS Configuration and finishing the Documentation

The last practical session was used to complete iSCSI and the documentation.

The final issue with the iSCSI was solved be not integrating iSCSI in our architecture, because we had to wipe a hard disk to create a LVM, which is used for iSCSI. The issue with that is that the hard drive is in use as a backup folder.

10. Personal Conclusion

This project was a genuinely hands-on introduction to the kind of work that happens in server rooms and data centres every day. Setting up the server from scratch physically handling enterprise hard drives and a hardware RAID controller, going through the BIOS, and troubleshooting why something doesn't appear in the OS is a very different experience from just reading about it.

A few things stood out during the project. The first was how much thought goes into something as basic as where to put the operating system. The decision to install OMV on a rear USB stick and leave all internal drives for data is the kind of architectural choice that would be easy to overlook without prior knowledge, but it makes a real difference in maintainability. If the OS USB stick fails, you replace it and reinstall OMV; the data on the RAID volumes is completely untouched.

The comparison between hardware RAID and software RAID was also valuable. Hardware RAID (via the Areca controller) was transparent to the OS OMV simply saw a healthy volume. Software RAID (mdadm inside OMV) required more configuration steps but gave us more direct visibility into the RAID state through the web interface. In a home or small business environment, software RAID through OMV would be perfectly adequate. In a higher-load environment, hardware RAID with a BBU is the safer choice.

OpenMediaVault proved to be a capable platform for the given scenario. The web interface covered everything we needed without requiring command-line work for the standard tasks. The family home storage scenario made us think through real-world permissions design how to give each family member a private space while still allowing shared access to common folders which is a challenge that scales directly into enterprise Active Directory and NTFS permission planning.

If we were to do this project again, we would explore OMV's notification system more thoroughly setting up email alerts for disk S.M.A.R.T. failures would be important in a real deployment, and we would look at the resync plugin for automated cross-site backup replication.

11. Sources

The following sources were consulted during the research and configuration phases of this project:

- TechTarget – Definition of Network Attached Storage: https://www.techtarget.com/searchstorage/definition/network-attached-storage
- Stonefly – SAN vs NAS vs DAS: A Closer Look: https://stonefly.com/blog/san-vs-nas-vs-das-a-closer-look/
- OpenMediaVault official website: https://www.openmediavault.org/
- Intel SR2600UR / SR2625UR Server Guide (PDF): https://downloads.bl4ckb0x.de/download.intel.com/support/motherboards/server/s5520ur/sb/E51243-006_SR2600UR_SR2625UR_SG.pdf
- OMV Installation Tutorial on YouTube: https://youtu.be/E0Mvsl0uTi0
- Areca Technology RAID Controller documentation — manufacturer manual.
- GNU General Public License v3: https://www.gnu.org/licenses/gpl-3.0.en.html