

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект навчально-експериментальної
мережної лабораторії homelab»

Виконав студент 4-го курсу групи КІ-20-1
спеціальності 123 «Комп'ютерна інженерія»

 Магурін Олексій

Керівник доцент кафедри КІ та КБ, к.пед.н., доцент,

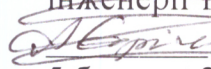
 Головня Олена

Рецензент завідувач кафедри КІ та КБ, к.т.н., доцент

 Андрій Єфіменко

Житомир – 2024

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки
 Андрій Єфіменко
5 березня 2024 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Магурін Олексій Олександрович

Тема роботи: «Проект навчально-експериментальної мережної лабораторії
homelab»

затверджена Наказом університету від 5 березня 2024 р. № 97

Термін здачі студентом закінченої роботи 9 червня 2024 р.

Вихідні дані роботи:

Об'єктом дослідження є домашні мережеві технології, тоді як предметом є конкретні методи інтеграції та управління сервісами в рамках homelab.

Обладнання та програмне забезпечення:

1. Маршрутизатор Mikrotik hAP ax²
2. Комутатори TP-LINK LS1005G
3. SBC Raspberry Pi 4b 8GB
4. Raspberry Pi OS, Ubuntu, Proxmox, Windows 11, RouterOS 7, MacOS, android
5. PiHole, Docker, Wireguard, Nginx Proxy Manager, VScode, Samba, Jellyfin, qBittorrent, Filebrowser

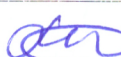
Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1			

2			
3			

Студент Магурін Олексій
Керівник Головня Олена

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	15.03.2024 – 05.04.2024	
2	Аналіз теоретичних засад побудови домашніх мереж	05.04.2024 – 13.04.2024	
3	Аналіз теоретичних засад homelab та можливих сервісів	13.04.2024 – 22.04.2024	
4	Проектування фізичної та логічної структури мережі	22.04.2024 – 26.05.2024	
5	Аналіз та придбання обладнання	26.05.2024 – 01.05.2024	
6	Збірка мережі homelab та її налаштування	01.05.2024 – 24.05.2024	
7	Проектування та конфігурація зовнішніх сервісів мережі	24.05.2024 – 28.05.2024	
8	Проектування та конфігурація внутрішніх сервісів мережі	28.05.2024 – 05.06.2024	
9	Оформлення кваліфікаційної роботи	06.06.2024 – 09.06.2024	
10	Підготовка презентації до захисту	09.06.2024 – 12.06.2024	

Студент Магурін Олексій
Керівник Головня Олена

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи на тему «Проект навчально-експериментальної мережної лабораторії homelab» складається з вступу, трьох розділів, висновків, списку використаної літератури. Текстова частина викладена на 120 сторінках друкованого тексту. Список використаних джерел містить 20 найменувань і займає 2 сторінки. Додатки займають 5 сторінок. В роботі наведено 104 рисунка. Загальний обсяг роботи: 127 сторінки.

У першому розділі детально розглядаються теоретичні основи створення домашньої мережі homelab. Описано феномен homelab, проаналізовані переваги та недоліки самостійного хостингу порівняно з хмарними та локальними розгортаннями.

У другому розділі викладено процес створення мережі для домашньої лабораторії homelab. Розглянуто роль домашньої мережі як основи для лабораторії, визначено вимоги до функцій та захисту мережі. Проаналізовано і сплановано фізичну та логічну структуру мережі, а також вибір обладнання. Описано процеси збірки та конфігурації пристроїв, перевірено працездатність мережі.

Третій розділ присвячений конфігурації зовнішніх і внутрішніх сервісів у homelab. Описано налаштування зовнішніх сервісів, Cloudflare, Nginx Proxy Manager, створення веб-серверу та впровадження серверу Proxmox для Minecraft. Описано розгортання допоміжних сервісів, таких як PiHole, SMB Share, VPN Wireguard, та медіа-стеку з Jellyfin, qBittorrent, і Filebrowser. Перевірено роботу homelab сервісів.

Ключові слова: HOMELAB, КОНФІГУРАЦІЯ МЕРЕЖІ, СИСТЕМНЕ АДМІНІСТРУВАННЯ, БЕЗПЕКА, САМОСТІЙНИЙ ХОСТИНГ, ВІРТУАЛІЗАЦІЯ, КОНТЕЙНЕРИЗАЦІЯ, VPN, DNS, МЕРЕЖЕВИЙ РЕСУРС, MICROTIK, ІНТЕГРАЦІЯ СЕРВІСІВ, PROXMOX, DOCKER

					KKIK.KP.B – 123 – 13 – ПЗ			
Змі.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Магурін О. О.		20.06.24	Проект навчально-експериментальної			
Керівник		Головня О. С.		20.06.24	мережної лабораторії homelab			
Реценз.		Єфіменко А.А.		20.06.24				
Н. контр.		Головня О. С.		20.06.24				
Затверд.		Єфіменко А.А.		20.06.24				
					Лит.	Арк.	Аркушів	
						4	127	
					ФІКТ, Гр. КІ-20-1			

ANNOTATION

The explanatory note to the qualification paper on the topic “Project of the educational and experimental network laboratory homelab” consists of an introduction, three chapters, conclusions, and a list of references. The text part is presented on 122 pages of printed text. The list of references includes 9 titles and occupies 1 page. The paper contains 104 figures. The total volume of the work: 123 pages.

The first chapter discusses in detail the theoretical foundations of the homelab home network. The phenomenon of homelab is described, and the advantages and disadvantages of self-hosting compared to cloud and local deployments are analyzed.

The second section describes the process of creating a network for a home lab. The role of the home network as the basis for the laboratory is considered, and the requirements for network functions and security are defined. The physical and logical structure of the network, as well as the choice of equipment, are analyzed and planned. The processes of assembling and configuring devices are described, and the network performance is tested.

The third chapter is devoted to the configuration of external and internal services in homelab. The chapter describes the configuration of external services, Cloudflare, Nginx Proxy Manager, creating a web server, and implementing the Proxmox server for Minecraft. The deployment of auxiliary services such as PiHole, SMB Share, VPN Wireguard, and a media stack with Jellyfin, qBittorrent, and Filebrowser is described. The work of homelab services is tested.

Keywords: HOMELAB, NETWORK CONFIGURATION, SYSTEM ADMINISTRATION, SECURITY, SELF-HOSTED, VIRTUALIZATION, CONTAINERIZATION, VPN, DNS, NETWORK SHARING, MICROTIK, SERVICE INTEGRATION, PROXMOX, DOCKER

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ ЩОДО HOMELAB ТА SELF-HOSTED СЕРВІСІВ	9
1.1 ФЕНОМЕН HOMELAB.	9
1.2 ПЛЮСИ ТА МІНУСИ САМОСТІЙНОГО ХОСТИНГУ ЧИ ГОТОВИХ РІШЕНЬ, ХМАРНИХ ЧИ ЛОКАЛЬНИХ РОЗГОРТАНЬ.	11
1.3 ВИБІР СЕРВІСІВ ДЛЯ HOMELAB, ВИМОГИ ТА НЕОБХІДНІ РЕСУРСИ, ЗНАННЯ ТА ТЕХНОЛОГІЇ	21
РОЗДІЛ 2 СТВОРЕННЯ МЕРЕЖІ ДЛЯ ДОМАШНЬОЇ ЛАБОРАТОРІЇ	25
2.1 ДОМАШНЯ МЕРЕЖА ЯК ОСНОВА ДЛЯ HOMELAB.....	25
2.2 ФОРМУВАННЯ ВИМОГ ЩОДО ФУНКЦІЙ ТА ЗАХИСТУ МЕРЕЖІ	28
2.3 ПРОЕКТУВАННЯ ДОМАШНЬОЇ МЕРЕЖІ	29
2.2.1 ПЛАНУВАННЯ ФІЗИЧНОЇ СТРУКТУРИ МЕРЕЖІ	29
2.2.2 ПЛАНУВАННЯ ЛОГІЧНОЇ СТРУКТУРИ МЕРЕЖІ	32
2.2.3 АНАЛІЗ ТА ВИБІР ОБЛАДНАННЯ.....	35
2.4 ПОБУДОВА ТА КОНФІГУРАЦІЯ ДОМАШНЬОЇ МЕРЕЖІ.....	41
2.4.1 ЗБІРКА ТА ПІДКЛЮЧЕННЯ ВСІХ ПРИСТРОЇВ МЕРЕЖІ	41
2.4.2 НАЛАШТУВАННЯ МАРШРУТИЗАТОРА ЧЕРЕЗ ROUTEROS TERMINAL	45
2.4.3 ОГЛЯД ТА ПЕРЕВІРКА НАЛАШТУВАНЬ МЕРЕЖІ	58
РОЗДІЛ 3 ПІДГОТОВКА ТА КОНФІГУРАЦІЯ СЕРВІСІВ.....	68
3.1 КОНФІГУРАЦІЯ ЗОВНІШНІХ СЕРВІСІВ	68
3.1.1 CLOUDFLARE - РЕЄСТРАЦІЯ ДОМЕНУ ТА СТВОРЕННЯ ЗАПИСІВ	68
3.1.2 NGINX PROXY MANAGER І ПЕРСОНАЛЬНА WEB-СТОРІНКА	73
3.1.3 НАЛАШТУВАННЯ PROXMOX ТА MINECRAFT СЕРВЕРУ	83
3.2 КОНФІГУРАЦІЯ ВНУТРІШНІХ СЕРВІСІВ	89
3.2.1 РОЗГОРТАННЯ ДОПОМІЖНИХ СЕРВІСІВ: PiHOLE, SMB SHARE, VPN WIREGUARD.	89
3.2.2 РОЗГОРТАННЯ МЕДІА-СТЕКУ: JELLYFIN, QBITTORRENT, FILEBROWSER, NGINX PROXY MANAGER	102
3.3 ОГЛЯД ТА ПЕРЕВІРКА HOMELAB.....	108
ВИСНОВКИ.....	120
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	121

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ОС – операційна система

TCP – протокол управління передачею

UDP – протокол користувацьких дейтаграм

DHCP – протокол динамічної конфігурації хостів

MAC – мережевий адрес фізичного пристрою

LAN – локальна обчислювальна мережа

VLAN – віртуальна локальна обчислювальна мережа

IP – інтернет-протокол

NAT – мережева адресна трансляція

VM – віртуальна машина

VPN – віртуальна приватна мережа

SSL – протокол шару сокетів безпеки

TLS – транспортний шар безпеки

IaaS – інфраструктура як послуга

PaaS – платформа як послуга

SaaS – програмне забезпечення як послуга

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми даної роботи полягає в зростаючій потребі освоєння та використання технологій домашніх лабораторій (HomeLab), що дозволяють інтегрувати зовнішні та внутрішні сервіси, підвищуючи рівень особистої ІТ-грамотності та незалежності. Завдяки homelab можна розв'язати проблеми контролю за даними, безпеки інформації та розробки персоналізованих ІТ-рішень.

Метою кваліфікаційної роботи є створення домашньої лабораторії, що інтегрує різноманітні сервіси для комплексного вивчення і застосування в сучасних технологічних умовах.

Завдання, що впливають із поставленої мети, включають:

- Дослідження теоретичних основ створення HomeLab.
- Проектування та конфігурацію мережевої інфраструктури.
- Інтеграцію та налаштування внутрішніх та зовнішніх сервісів.
- Перевірку та аналіз ефективності розгортання і функціонування системи.

Об'єктом дослідження є домашні мережеві технології. Предметом дослідження є конкретні методи інтеграції та управління сервісами в рамках homelab.

У процесі було розроблено проект домашньої лабораторії, що включає технічні та програмні рішення для створення функціонального і безпечного ІТ-середовища.

Апробація результатів:

1. Mahurin O.O., Shadura V.A. Technical Aspects and Comparison of Virtualization and Containerization. X All Ukrainian Scientific and Practical Conference “Current Trends in Young Scientists’ Research”, 20 квітня 2023 року. Житомир : «Житомирська політехніка», 2023. С. 195-197.
2. Магурін О.О., Головня О.С.. Використання NGINX у домашній мережі для хостингу веб-застосунків та оптимізації серверних функцій. Тези XIV Міжнародної науково-технічної конференції "Інформаційно-комп'ютерні технології - 2024", 28-29 березня 2024 року. Житомир : «Житомирська політехніка», 2024. С. 45-46.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ ЩОДО HOMELAB ТА SELF-HOSTED СЕРВІСІВ

1.1 Ф е н о м е n e m e l a b

Домашня лабораторія - це персоналізоване налаштування, яке складається з різних апаратних та програмних компонентів, призначених для відтворення професійного середовища у межах власної домівки. Вона може включати в себе сервери, мережеві пристрої, рішення для зберігання та віртуалізаційні платформи. Домашні лабораторії служать платформою для навчання чи хостингу зручних сервісів, надаючи практичний досвід з технологій корпоративного рівня у середовищі з низьким ризиком. Вони надають ентузіастам можливості для експериментів, розвитку навичок та реалізації особистих проектів.

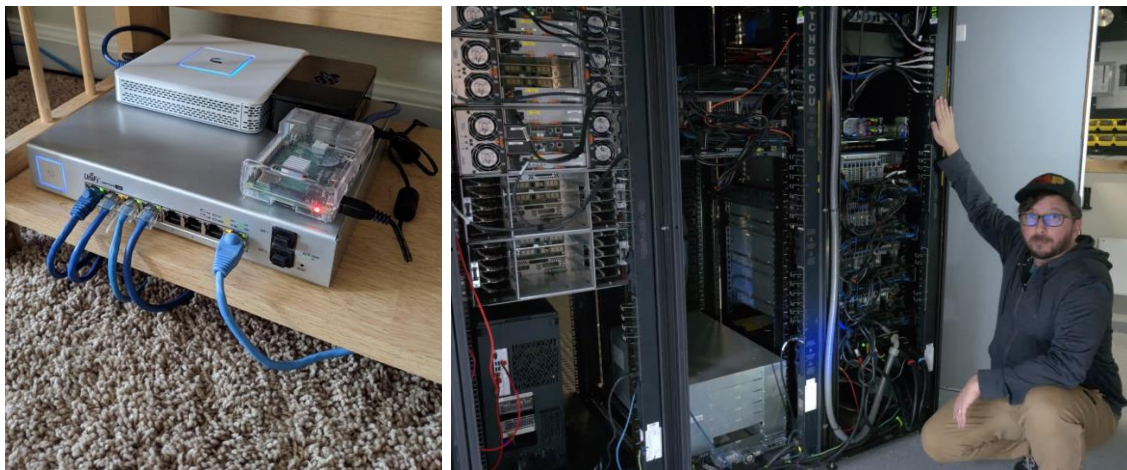


Рисунок 1.1 – Домашні лабораторії різних масштабів

Домашні лабораторії зазвичай охоплюють різноманітні апаратні та програмні компоненти, призначені задовольняти конкретні потреби та вподобання. Серед апаратних компонентів можна виділити сервери, мережеві пристрої (такі як маршрутизатори, комутатори та брандмауери), рішення для зберігання (наприклад, пристрої NAS або сервери зберігання) та системи керування енергією [1][2]. У сфері програмного забезпечення віртуалізаційні платформи (наприклад, VMware ESXi, Proxmox або Hyper-V), інструменти контейнеризації (такі як Docker або Kubernetes), а також різні операційні системи відіграють ключову роль у створенні універсального та динамічного середовища домашньої лабораторії.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

Створення домашньої лабораторії може вимагати ретельного планування та уважного врахування різних чинників так і розвиватись поступово та хаотичною, це про свободу. Для початківців та професіоналів бажано вибирати обладнання відповідно до навантаження та бюджетних обмежень, забезпечуючи баланс між продуктивністю, масштабованістю та енергоефективністю, але це не є обов'язковим. Якщо підходити до створення домашньої лабораторії професійно, то налаштування мережі відіграє важливу роль у забезпеченні безперервного зв'язку між пристроями, вимагаючи твердого розуміння мережевих принципів та протоколів. Встановлення та налаштування програмного забезпечення охоплює розгортання операційних систем, віртуалізаційних платформ та інших програмних компонентів з акцентом на оптимізацію та безпеку. Створення сприятливого середовища для експериментів передбачає належні моніторинг та резервне копіювання, а також впровадження найкращих практик у сфері адміністрування та обслуговування системи.

Жива спільнота домашніх лабораторій пропонує величезний обсяг ресурсів та підтримки для ентузіастів на всіх рівнях експертизи. Онлайн-форуми, такі як r/homelab на Reddit [2] та різноманітні сервери Discord (наприклад, [3], [4]), виступають центрами для обміну знаннями, вирішення проблем та співпраці над проектами. Крім того, безліч блогів, посібників та каналів на YouTube орієнтовані на ентузіастів домашніх лабораторій, надаючи цінні інсайти, посібники та реальні досвіди. Залучення до спільноти домашніх лабораторій не лише сприяє товариськості, але й полегшує неперервне навчання та інновації.

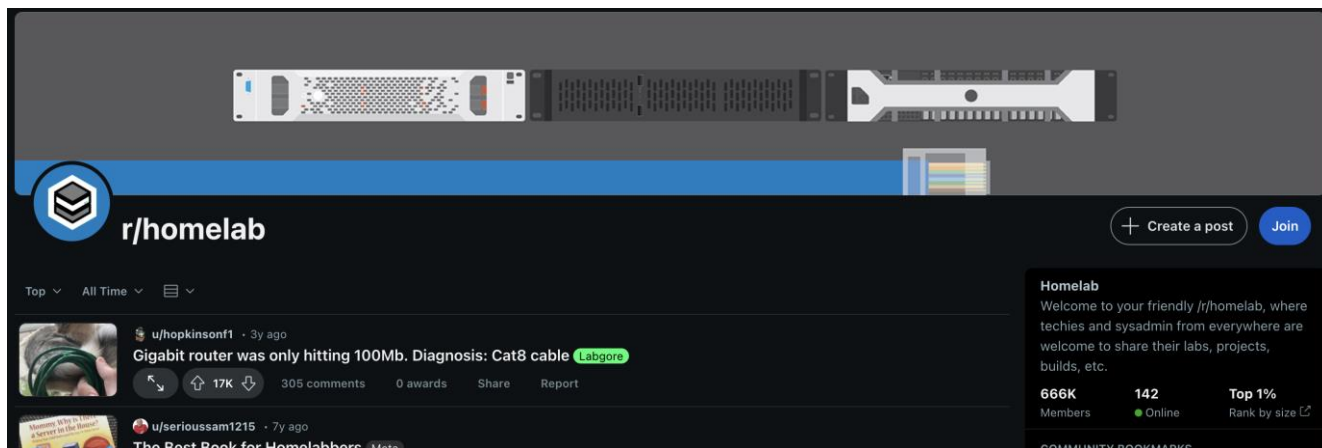


Рисунок 1.2 – Homelab спільнота на вебсайті Reddit

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

1.2 Переваги та недоліги хмарних чи локальних розгортань

При розробці та розгортанні ІТ-сервісів одне з найважливіших рішень, з яким стикаються організації та приватні особи, - це вибір між придбанням готових рішень та самостійним хостингом своїх послуг. Кожен варіант має свої переваги та виклики, що робить його важливим для кожного, хто займається управлінням ІТ-інфраструктурою та сервісами. При виборі самостійного хостингу виникає ще одне важливе рішення: де розгорнути інфраструктуру - у хмарі чи локально. Це рішення суттєво впливає на управління, масштабованість та безпеку сервісів.

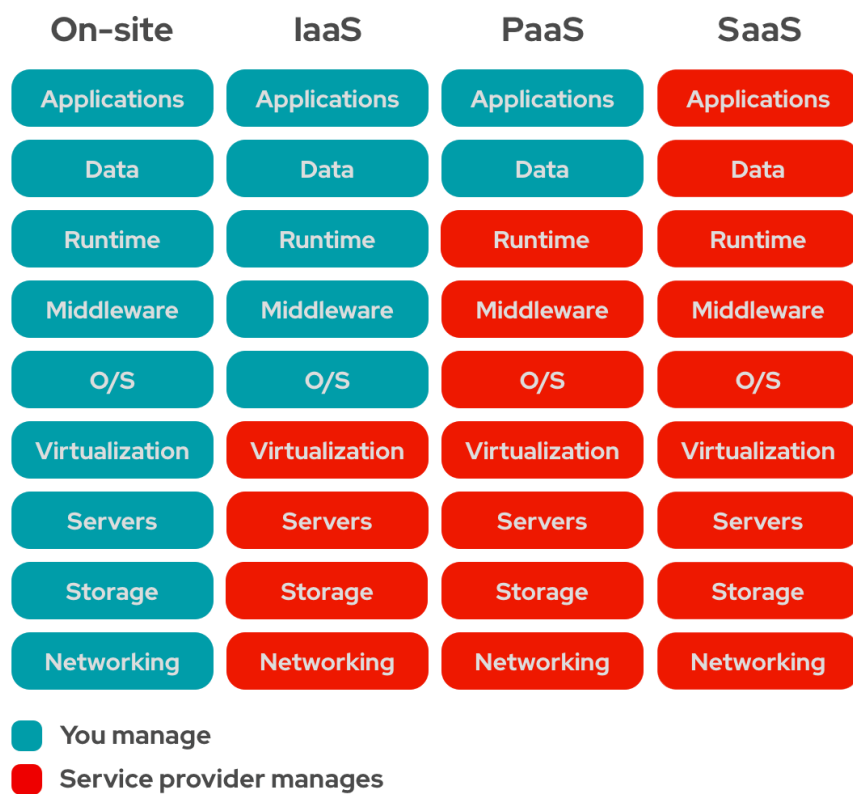


Рисунок 1.3 – Варіанти розгортання сервісів

Метою даної кваліфікаційної роботи є створення домашньої лабораторії - персонального середовища для навчання, експериментів та тестування саморозміщених сервісів з використанням локальної інфраструктури. Хоча домашня лабораторія за своєю суттю передбачає локальне розгортання, розуміння ширшого ландшафту готових і саморозміщених рішень, а також хмарних і локальних розгортань є важливим. Ці знання не тільки забезпечують всебічну

перспективу, але й допомагають приймати обґрунтовані рішення для майбутніх проектів, які можуть виходити за межі домашньої лабораторії.

Готові рішення, також відомі як готове програмне забезпечення, - це заздалегідь створені додатки або послуги, що надаються постачальниками. Ці рішення призначені для швидкого розгортання з мінімальною кастомізацією, пропонуючи негайну функціональність для задоволення загальних бізнес-потреб. Приклади включають платформи "програмне забезпечення як послуга" (SaaS)[3], такі як Salesforce, Microsoft Office 365 та різні галузеві додатки.

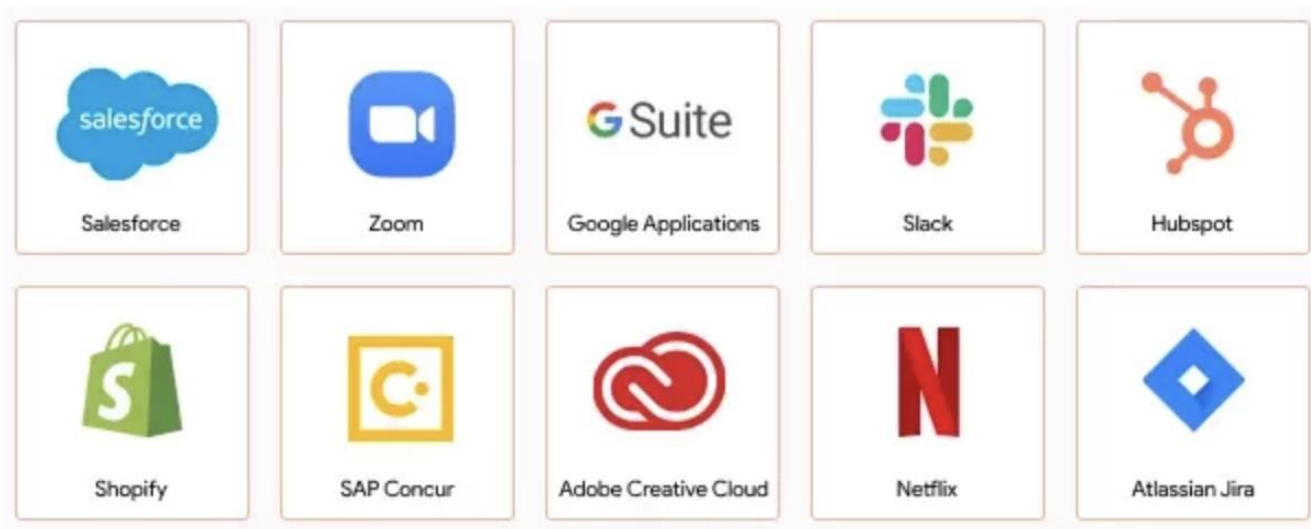


Рисунок 1.4 – Готові рішення (SaaS) на прикладі популярних програм
Використання готових рішень має наступні переваги.

- Швидке розгортання. Готові рішення можна швидко розгорнути, що дозволяє організаціям почати використовувати програмне забезпечення з мінімальною затримкою. Це особливо вигідно для компаній, яким потрібно швидко впроваджувати нові рішення.
- Передбачуваність витрат. Готові рішення часто поставляються з передбачуваними ціновими моделями, як правило, на основі підписки. Це дозволяє організаціям ефективніше планувати бюджет, не турбуючись про непередбачувані витрати, пов'язані з обладнанням та обслуговуванням.
- Підтримка та оновлення. Постачальники готових рішень надають регулярні оновлення, патчі безпеки та технічну підтримку. Це зменшує навантаження на ІТ-персонал організації та гарантує, що програмне забезпечення

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

залишається актуальним з найновішими функціями та покращеннями безпеки.

- Масштабованість. Готові рішення, особливо ті, що розміщені в хмарі, пропонують гнучку масштабованість. Організації можуть легко налаштувати їх використання відповідно до своїх потреб без значних інвестицій у додаткову інфраструктуру.
- Доступність. Хмарні готові рішення забезпечують високу доступність, дозволяючи користувачам отримувати доступ до програмного забезпечення з будь-якого місця, де є підключення до Інтернету. Це ідеальний варіант для організацій з віддаленою або розподіленою робочою силою.

Водночас, застосування готових рішень має наступні недоліки.

- Обмежена кастомізація. Одним з основних недоліків готових рішень є обмежена можливість кастомізації. Організації можуть виявити, що програмне забезпечення не ідеально відповідає їхнім конкретним потребам і процесам, що призводить до потенційної неефективності.
- Безпека та конфіденційність даних. Зберігання даних у сторонніх середовищах може викликати занепокоєння щодо безпеки та конфіденційності даних. Організації повинні довіряти постачальнику в тому, що він буде безпечно поводитися з їхніми даними і дотримуватися відповідних норм, що може бути значним ризиком.
- Залежність від постачальника. Організації, які використовують готові рішення, залежать від постачальника щодо оновлень, підтримки та безперервності обслуговування. Якщо у постачальника виникають проблеми або він припиняє випуск продукту, це може серйозно вплинути на діяльність організації.
- Поточні витрати. Хоча готові рішення часто мають нижчі початкові витрати, поточні витрати на підписку з часом можуть зростати. При довгостроковому використанні це може стати дорожчим, ніж самостійне рішення.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

- Проблеми інтеграції. Інтеграція готових рішень з існуючими системами та процесами може бути складною. Організації можуть зіткнутися з проблемами сумісності, що потребують додаткового часу та ресурсів для досягнення безперешкодної інтеграції.

Самостійно розміщені рішення передбачають встановлення та запуск програмного забезпечення на серверах, якими керує організація або окрема особа. Такий підхід дає змогу краще налаштовувати і контролювати програмне забезпечення та дані, які воно обробляє. Самостійно розміщені рішення вимагають, щоб користувач сам займався всіма аспектами обслуговування, безпеки та оновлень. Приклади включають розміщення веб-сайту з використанням WordPress/Apache/Nginx тощо на приватному сервері або запуск корпоративних додатків, таких як ERP/CRM-системи, на власному обладнанні.

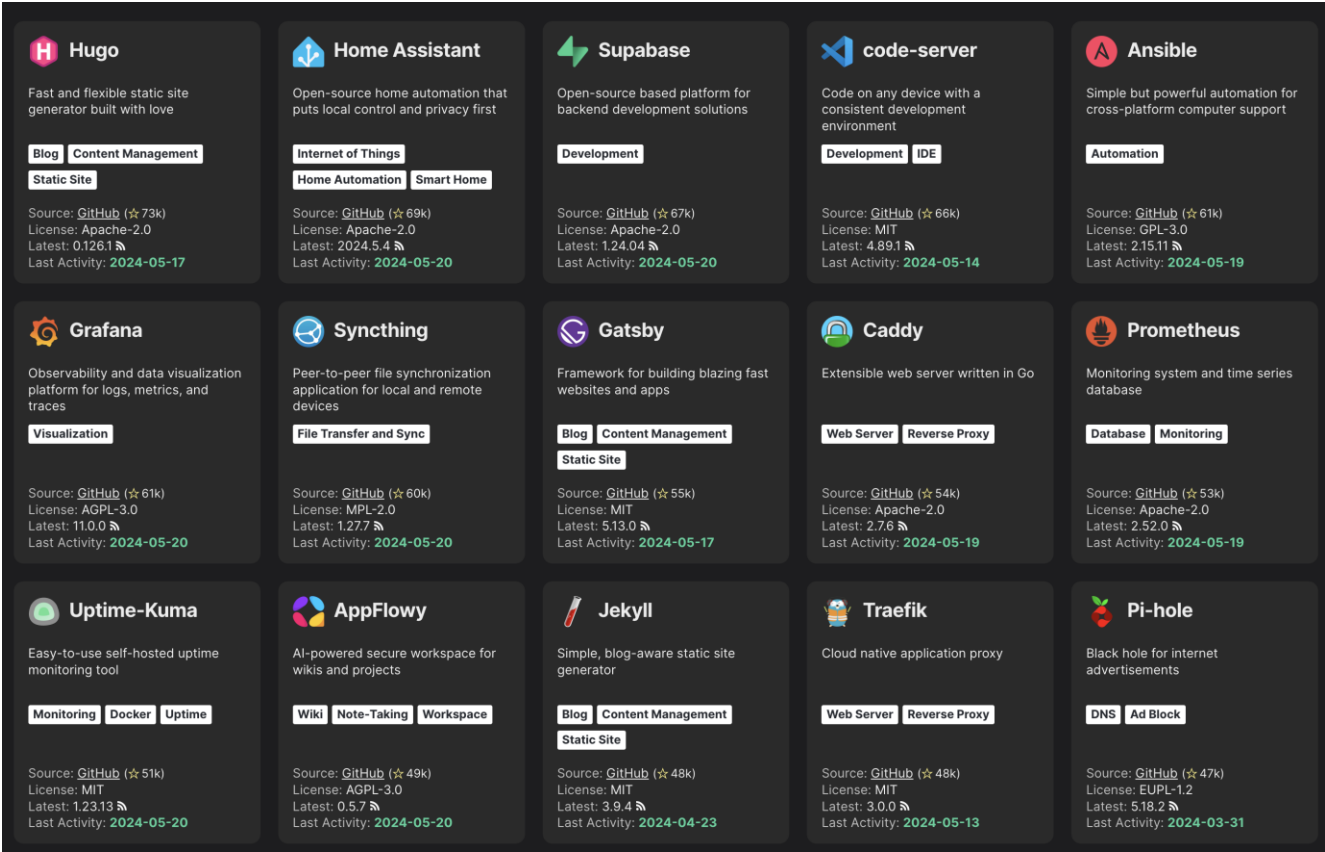


Рисунок 1.5 – Популярні сервіси самостійного хостингу

Самостійний хостинг має наступні переваги.

- **Контроль і кастомізація.** Однією з головних переваг саморозміщуваних рішень є високий рівень контролю, який вони пропонують. Організації можуть адаптувати програмне забезпечення до своїх потреб, впроваджуючи кастомні функції та конфігурації, які можуть бути неможливими у готових рішеннях. Це особливо вигідно для компаній зі специфічними операційними вимогами, які не можуть бути задоволені стандартними програмними пакетами.
- **Безпека.** Самостійно розміщені рішення можуть забезпечити підвищену безпеку, оскільки організація має повний контроль над даними та інфраструктурою. Це дозволяє впроваджувати суворі заходи безпеки, адаптовані до конкретних потреб організації. Конфіденційні дані залишаються на власних серверах компанії, що зменшує ризик витоку даних, пов'язаний зі сторонніми провайдерами.
- **Продуктивність.** Завдяки рішенням з власним хостингом організації можуть оптимізувати продуктивність відповідно до своїх конкретних потреб. Вони можуть вибирати апаратні та мережеві конфігурації, які найкраще підтримують їхні додатки, що призводить до потенційно вищої продуктивності порівняно із загальним середовищем у хмарних рішеннях.
- **Економічна ефективність в довгостроковій перспективі.** Хоча початкові витрати на налаштування самостійних рішень можуть бути високими, в довгостроковій перспективі вони можуть бути більш економічно ефективними. Організації уникають постійної абонентської плати, пов'язаної з хмарними сервісами, і можуть амортизувати вартість інфраструктури з часом.
- **Відповідність вимогам і контроль даних.** Для компаній, що працюють у регульованих галузях, рішення з власним хостингом забезпечують більшу відповідність галузевим нормам і законам про суверенітет даних. Організації можуть гарантувати, що дані зберігаються та обробляються відповідно до

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

місцевих законодавчих вимог, що має вирішальне значення в таких секторах, як охорона здоров'я, фінанси та уряд.

Використання самостійного хостингу також має недоліки, перелічень нижче.

- Високі початкові витрати. Початкові витрати на налаштування рішень, розміщених на власному хостингу, можуть бути непомірно високими. Організаціям потрібно інвестувати в обладнання, ліцензії на програмне забезпечення та інфраструктуру. Крім того, існують витрати, пов'язані зі створенням і підтримкою безпечного та ефективного ІТ-середовища.
- Обслуговування та оновлення. Самостійно розміщені рішення потребують постійного обслуговування, включаючи обслуговування обладнання, оновлення програмного забезпечення та патчі безпеки. Це може бути ресурсномістким і вимагати спеціальної ІТ-команди, яка може бути дорогою і складною в управлінні.
- Проблеми з масштабуванням. Масштабування самостійних рішень може бути складним і дорогим. У міру зростання організації виникає потреба в додатковому обладнанні та ресурсах, що може спричинити значні інвестиції та логістичні проблеми. Це контрастує з хмарними рішеннями, які пропонують гнучке масштабування.
- Простоти та надійності. Забезпечення високої доступності та надійності в середовищах з власним хостингом може бути складним завданням. Організаціям необхідно впроваджувати надійні рішення для резервного копіювання та аварійного відновлення, які можуть бути складними і дорогими. Простоти можуть безпосередньо вплинути на бізнес-операції та призвести до потенційної втрати доходів.
- Обмежена доступність. Самостійно розміщені рішення можуть мати обмежений доступ у порівнянні з хмарними рішеннями. Віддалений доступ може бути складніше реалізувати безпечно, що потенційно обмежує гнучкість віддаленої роботи та співпраці.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

Хмарне розгортання - це використання сторонніх сервісів для розміщення та управління інфраструктурою, платформами або програмним забезпеченням. Доступ до цих сервісів здійснюється через Інтернет і надається, зокрема, такими компаніями, як Amazon Web Services (AWS), Microsoft Azure та Google Cloud Platform (GCP). Хмарне розгортання пропонує масштабовані ресурси, що дозволяє користувачам платити лише за те, що вони використовують, з можливістю гнучко налаштовувати ресурси за потреби.



Рисунок 1.6 – Логотипи популярних хмарних провайдерів

Переваги хмарних розгортань є наступними.

- Масштабованість і гнучкість. Хмарні розгортання пропонують неперевершену масштабованість. Організації можуть легко збільшувати або зменшувати використання хмарних сервісів відповідно до своїх поточних потреб без значних інвестицій в обладнання. Така гнучкість дозволяє компаніям швидко реагувати на мінливі вимоги.
- Економічна ефективність. Хмарні розгортання зазвичай працюють за моделлю "оплата за фактом" (pay-per-use), що дозволяє організаціям уникати великих авансових витрат. Ця модель може бути більш економічно вигідною, особливо для малих і середніх підприємств, які не хочуть вкладати значні кошти в інфраструктуру.
- Доступність та співпраця. Хмарні рішення забезпечують легкий доступ з будь-якого місця, де є підключення до Інтернету. Це покращує співпрацю

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		17

між командами, особливо у віддаленому робочому середовищі. Співробітники можуть отримати доступ до потрібних їм інструментів і даних з будь-якого місця, що сприяє кращій командній роботі та підвищенню продуктивності.

- Автоматичні оновлення та обслуговування. Провайдери хмарних сервісів займаються оновленням та обслуговуванням програмного забезпечення, гарантуючи, що найновіші функції та виправлення безпеки завжди доступні. Це зменшує навантаження на ІТ-відділ організації та гарантує, що система залишається безпечною та сучасною.
- Аварійне відновлення та резервне копіювання. Хмарні розгортання часто супроводжуються надійними рішеннями для аварійного відновлення та резервного копіювання. Це гарантує безпеку даних і можливість їх швидкого відновлення у разі збою, зменшуючи час простою та ризики втрати даних.

Втім, хмарні розгортання мають і недоліки, зокрема наступні.

- Безпека та конфіденційність даних. Хоча хмарні провайдери впроваджують надійні заходи безпеки, зберігання даних за межами майданчика може викликати занепокоєння щодо конфіденційності даних та відповідності нормативним стандартам. Організації повинні переконатися, що їхній хмарний провайдер відповідає всім необхідним вимогам безпеки та відповідності.
- Залежність від Інтернету. Хмарні рішення покладаються на стабільне підключення до Інтернету. Будь-які перебої з підключенням можуть вплинути на доступ до критично важливих додатків і даних, що потенційно може вплинути на бізнес-операції.
- Обмеження контролю. Розгортання хмарних рішень може обмежити рівень контролю організації над своїм ІТ-середовищем. Налаштування інфраструктури або конкретних конфігурацій може бути обмежена політикою та можливостями хмарного провайдера.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

- Поточні витрати. Хоча модель оплати за фактом використання пропонує гнучкість, сукупні витрати з часом можуть бути значними. Організаціям потрібно ретельно контролювати використання хмарних сервісів, щоб уникнути несподівано високих рахунків.
- Прив'язка до постачальника. Перехід від одного постачальника хмарних послуг до іншого може бути складним і дорогим. Організації можуть виявитися прив'язаними до екосистеми певного постачальника, що ускладнює перехід до іншого постачальника або інтеграцію з іншими сервісами.

Локальне розгортання передбачає розміщення та управління інфраструктурою у фізичному приміщенні організації. Сюди входить налаштування серверів, мережевого обладнання та сховищ у власному дата-центрі або офісі компанії. Локальне розгортання пропонує повний контроль над апаратним і програмним середовищем, але вимагає значних інвестицій у фізичну інфраструктуру та постійне обслуговування.

Переваги локального розгортання є наступними.

- Контроль та налаштування. Локальне розгортання забезпечує повний контроль над апаратним і програмним середовищем. Організації можуть налаштовувати свої системи відповідно до конкретних вимог і мати повний контроль над даними та заходами безпеки.
- Безпека. Завдяки локальному розгортанню організації можуть впроваджувати та керувати власними протоколами безпеки. Це може бути особливо корисно для підприємств, що працюють з конфіденційною інформацією, гарантуючи, що дані не покинуть межі приміщення.
- Продуктивність. Локальні розгортання можуть запропонувати чудову продуктивність, оскільки вони не залежать від підключення до Інтернету. Це може призвести до швидшого доступу до даних і додатків, особливо для завдань, що вимагають великих обчислень.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

- Відповідність нормативним вимогам. Для організацій, що працюють в галузях з високим рівнем регулювання, локальне розгортання може запропонувати кращу відповідність суверенітету даних і нормативним вимогам. Дані можуть зберігатися і оброблятися відповідно до конкретних правових і регуляторних стандартів.
- Незалежність від підключення до Інтернету. Локальні розгортання не залежать від підключення до Інтернету для доступу, що робить їх більш надійними в середовищах, де інтернет-сервіс нестабільний або недоступний.

Локальне розгортання також має недоліки, зокрема наступні.

- Високі початкові витрати. Початкові інвестиції в локальне розгортання можуть бути значними, включаючи витрати на обладнання, ліцензії на програмне забезпечення та інфраструктуру. Це може стати перешкодою для малих і середніх підприємств.
- Обслуговування та оновлення. Управління локальними розгортаннями вимагає постійного обслуговування, включаючи обслуговування обладнання, оновлення програмного забезпечення та патчі безпеки. Це може бути ресурсномістким і вимагати створення спеціальної ІТ-команди.
- Проблеми з масштабуванням. Масштабування локального розгортання може бути складним і дорогим. У міру зростання організації виникає потреба в додатковому обладнанні та ресурсах, що може спричинити значні інвестиції та логістичні проблеми.
- Аварійне відновлення та резервне копіювання. Впровадження надійних рішень для аварійного відновлення та резервного копіювання для локальних розгортань може бути складним і дорогим. Організації повинні переконатися, що вони мають ефективні заходи для захисту від втрати даних і простоїв.
- Доступність. Локальні розгортання можуть обмежувати доступність, особливо для віддалених працівників. Впровадження безпечних рішень для віддаленого доступу може бути складним і не забезпечувати такий же рівень зручності, як хмарні рішення.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		20

Отже, хоча створення домашньої лабораторії в першу чергу передбачає використання локальної інфраструктури, розуміння спектру доступних рішень і стратегій розгортання є важливим. Воно дає знання для прийняття обґрунтованих рішень як для особистих проєктів, так і для професійного середовища, гарантуючи, що обраний підхід відповідає конкретним потребам і цілям. У майбутньому, з розвитком сервісів у домашній лабораторії настане момент, коли набагато краще буде зробити специфічний сервіс у хмарі та інтегрувати з локальною інфраструктурою, ніж розгортати локально.

1.3 В и б і р с е рвісів для та не обхвднн технології

Подорож до створення домашньої лабораторії починається з вибору правильних сервісів для розміщення. Це передбачає всебічне розуміння як вимог, так і ресурсів, необхідних для ефективної підтримки цих сервісів. Діапазон можливостей дуже широкий - від самостійного розміщення медіа-серверів та ігрових серверів до розгортання віртуалізованих середовищ і передових мережевих рішень. Кожен вибір, зроблений на етапі планування, впливає на загальну функціональність, продуктивність і безпеку домашньої лабораторії.

Домашня лабораторія може підтримувати різноманітні сервіси, кожен з яких відповідає різним інтересам і функціональним вимогам. Ось деякі з найпомітніших прикладів [5].

- Ігрові сервер. Запуск ігрових серверів, наприклад, для Minecraft, дозволяє проводити приватні ігрові сесії з друзями або надає платформу для експериментів з конфігураціями серверів та ігровими модами.
- Медіа-сервери. Централізоване управління мультимедіа полегшують такі сервери, як Plex, Emby та Jellyfin, які обробляють бібліотеки музики та фільмів. Додаткові інструменти, такі як Tautulli, Sonarr, Radarr, Couch-Potato та Mylar, допомагають в організації та автоматизації медіа-колекцій.
- Рішення для зберігання. Такі рішення, як NextCloud, пропонують персональне хмарне сховище, Veeam надає можливості резервного

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

копіювання для віртуальних машин, а FreeNAS забезпечує надійне мережеве сховище.

- Веб-хостинг. Розміщення веб-сайтів на таких серверах, як IIS, Apache або Nginx, задовольняє особисті потреби або потреби малого бізнесу, пропонуючи гнучкість у веб-розробці та розгортанні.
- Віртуалізація. Використання гіпервізорів, таких як Docker, ESXi, unRAID, Hyper-V або Proxmox, дозволяє створювати кілька віртуальних середовищ на одній апаратній платформі, сприяючи ізольованому та безпечному налаштуванню різних додатків.
- Робота в мережі. Такі інструменти, як GNS3 для моделювання мережі, pfSense або OPNsense для брандмауерів, Pi-hole для блокування реклами та OpenVPN для безпечного віддаленого доступу, дозволяють комплексно керувати мережею та підвищувати її безпеку.
- Моніторинг та безпека. Такі рішення, як Splunk для аналізу даних, Snort для запобігання вторгненням та інструменти мережевого моніторингу, такі як Network Monitor, забезпечують безпеку та працездатність домашньої лабораторії.
- Управління інфраструктурою. Керування конфігураціями серверів за допомогою Chef, Puppet або Ansible, а також віддалений доступ до домашньої лабораторії за допомогою Guacamole додає ще один рівень ефективності та зручності.

Успіх домашньої лабораторії залежить від ретельного планування, яке передбачає підхід "зверху-вниз". Це означає, що потрібно почати з визначення бажаних послуг, вибору необхідного програмного забезпечення, визначення вимог до обладнання та планування мережевої інфраструктури. Кожен крок ґрунтується на попередньому, забезпечуючи узгоджене та функціональне налаштування.

Першим кроком у плануванні є чітке визначення сервісів, які необхідно розмістити. Сервіси можна умовно поділити на приватні (внутрішні) та публічні (зовнішні).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

Приватні сервіси призначені для особистого використання і зазвичай доступні лише у локальній мережі. Прикладами можуть слугувати мережеві сховища для централізованого зберігання даних, віртуальні приватні мережі (VPN) для безпечного віддаленого доступу та Pi-hole для блокування реклами у всій мережі. Приватні служби мають вирішальне значення для підвищення особистої продуктивності та безпеки. Розміщення цих сервісів всередині зменшує вразливість до зовнішніх загроз і конфіденційні дані залишаються під контролем.

Загальнодоступні сервіси, з іншого боку, доступні за межами домашньої мережі і обслуговують ширшу аудиторію. До них відносяться веб-сервери, такі як Nginx, на яких можуть розміщуватися веб-сайти і веб-додатки, а також ігрові сервери, які надають платформу для друзів або більшої спільноти для спілкування і гри.

Поділ на приватні та публічні сервіси важливий з кількох причин. По-перше, це допомагає керувати безпекою, відокремлюючи внутрішні ресурси від тих, що доступні в інтернеті. По-друге, це впливає на контроль доступу та конфігурацію мережі, забезпечуючи належний захист послуг відповідно до їхнього ризику витоку. Нарешті, така сегрегація дозволяє підвищити продуктивність мережі, надаючи пріоритет внутрішньому трафіку для приватних сервісів і оптимізуючи публічні сервіси для зовнішнього доступу.

Хоча існує безліч сервісів, які можна розгорнути, для цілей цього проекту домашньої лабораторії буде обрано декілька з них для створення мінімально життєздатного продукту (MVP). Для приватних сервісів буде реалізовано NAS, VPN та DNS. Для публічних сервісів буде створено розширене налаштування веб-сервера Nginx та ігровий сервер. Цей вибір забезпечує збалансоване поєднання функціональних можливостей, задовольняючи основні потреби у зберіганні даних, безпеці та розвагах.

Також було б корисно розгорнути медіа-сервер як внутрішній сервіс, який дозволить переглядати мультимедійний контент, завантажувати торренти, та керувати файлами через зручні веб-інтерфейси.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

В и с н о в к и д і о р о з д і л у

У цьому розділі було розглянуто концепцію домашньої лабораторії та різні підходи до розгортання ІТ-сервісів, включно з готовими рішеннями (SaaS), самохостингом, хмарними та локальними розгортаннями.

Готові рішення пропонують швидке розгортання, передбачувану модель витрат та регулярні оновлення, але мають обмежену кастомізацію та проблеми з безпекою даних. Самохостинг забезпечує високий контроль, кастомізацію та відповідність вимогам, проте потребує значних початкових інвестицій та постійного обслуговування.

Хмарні розгортання забезпечують неперевершену масштабованість, економічну ефективність за моделлю оплати за фактом та автоматичні оновлення, але можуть викликати занепокоєння щодо безпеки даних і залежність від інтернет-підключення. Локальні розгортання пропонують повний контроль та незалежність від інтернету, але також вимагають високих витрат та складного обслуговування.

Після аналізу був зроблений вибір на користь створення домашньої лабораторії з використанням локальної інфраструктури для практичного досвіду роботи з корпоративними технологіями у безпечному середовищі.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

РОЗДІЛ 2 СТВОРЕННЯ МЕРЕЖІ ДЛЯ ДОМАШНЬОЇ ЛАБОРАТОРІЇ

2 . 1 Д о м а ш н я м е р е ж а я к о с н о в а д л я

Налаштування homelab ґрунтується на надійній і добре спроектованій домашній мережі. Ця мережа слугує основою для всіх самостійних служб і систем, які складають homelab, включаючи різні пристрої, такі як сервери, рішення для зберігання даних і мережеве обладнання. Побудова домашньої мережі з нуля пропонує безпрецедентну можливість адаптувати кожен аспект до конкретних потреб, починаючи від вимог до продуктивності і закінчуючи протоколами безпеки. Починаючи з чіткої та контрольованої мережевої архітектури, користувачі гарантують, що їхня домашня лабораторія буде не тільки ефективною, але й масштабованою, адаптуючись до майбутніх розширень або технологічних оновлень з мінімальними перебоями в роботі.

Важливість побудови індивідуальної домашньої мережі неможливо переоцінити. Індивідуальна мережа, ретельно спроектована і впроваджена, забезпечує стабільну і безпечну основу, необхідну для будь-якого середовища homelab. Вона дозволяє глибше розуміти і контролювати потік даних і дозволи доступу в мережі, що має вирішальне значення для ефективного управління як внутрішніми, так і зовнішніми сервісами. Крім того, персоналізоване налаштування мережі гарантує дотримання конкретних вимог до продуктивності та безпеки, що є важливим для розміщення чутливих або критично важливих додатків. По суті, створення власної мережі з нуля дає можливість як ентузіастам, так і професіоналам оптимізувати налаштування homelab для максимальної продуктивності та надійності, гарантуючи, що кожен компонент працює в ідеальній синергії.

З початку, необхідно для себе зрозуміти, що таке домашня мережа. У широкому розумінні, домашня мережа - це інфраструктура, що об'єднує різні електронні пристрої в межах одного житлового приміщення чи будинку. Ця мережа дозволяє підключеним пристроям обмінюватися даними, отримувати доступ до Інтернету, спільно використовувати ресурси та послуги. Вона може включати в себе

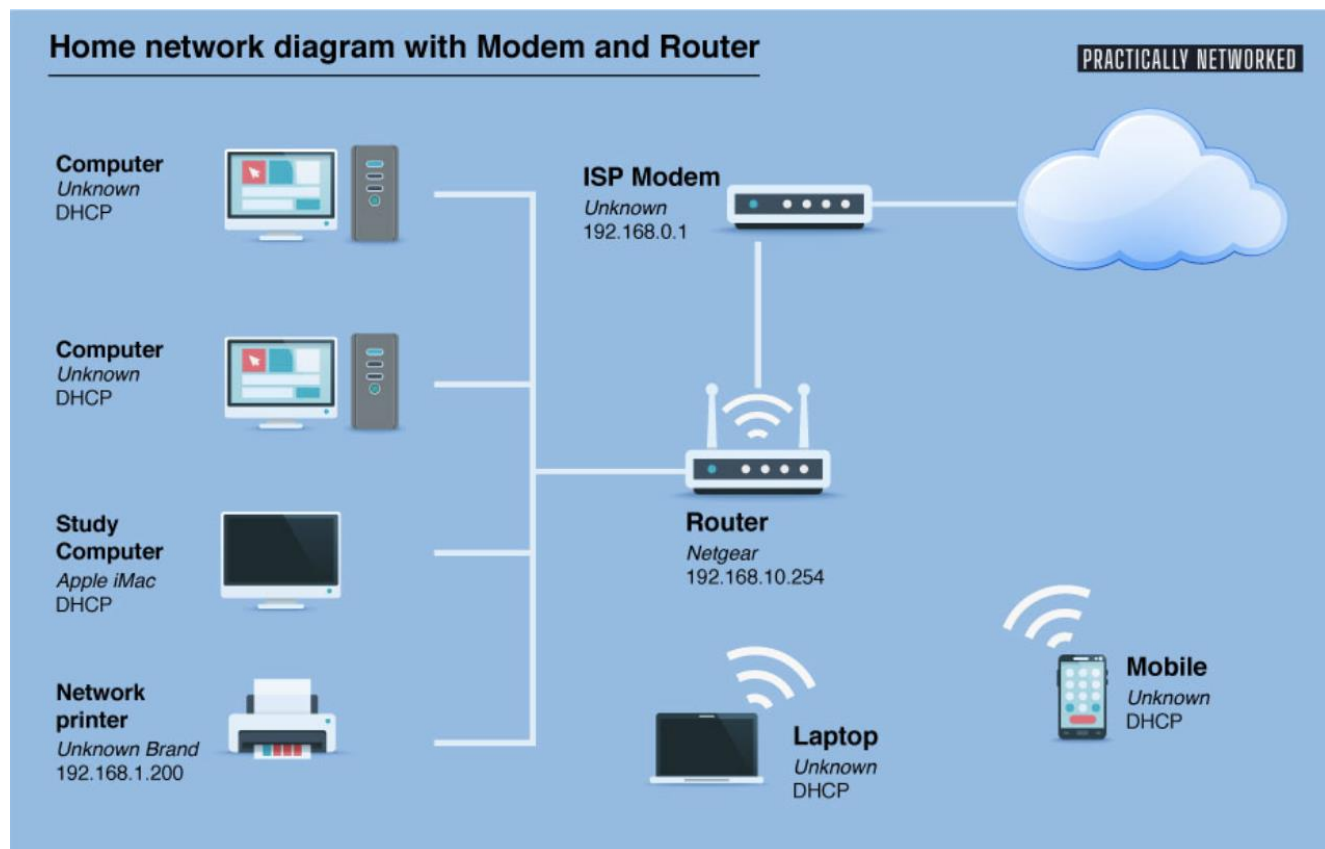
		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

комп'ютери, смартфони, планшети, роутери, мережеві сховища даних, IoT-пристрої, які контролюють освітлення, температуру, безпеку тощо.

Основна мета домашньої мережі - забезпечити зручність, ефективність та безпеку використання технологій вдома. Вона створює затишне інтерактивне середовище, де користувачі можуть не лише споживати інформацію, а й активно взаємодіяти з мережею, обмінюючись даними та керуючи пристроями.

З технічного погляду, домашня мережа складається з різних компонентів, які забезпечують її функціонування. Основні складові такої мережі включають в себе маршрутизатор, комутатори, точки доступу Wi-Fi, кабельну або бездротову інфраструктуру, мережеві кабелі та з'єднання. Всі ці компоненти спільно працюють, щоб забезпечити надійний, швидкий та безпечний обмін даними між пристроями в мережі та Інтернетом.

Розумна конфігурація домашньої мережі враховує потреби користувачів у зручності використання, швидкості передачі даних, захисті від несанкціонованого доступу та збереженні конфіденційності даних. Таке розуміння поняття домашньої мережі є важливим для ефективної побудови та налаштування мережевої інфраструктури в домашньому середовищі.



		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

Рис. 2.1. Типовий проект простої домашньої мережі мережі

Фізичні межі між Інтернетом і домашньою мережею пролягають через декілька ключових пристроїв та технологій. Першим етапом є з'єднання з Інтернетом через провайдера може бути модем, хоча в Україні зазвичай провайдери прокладають ethernet кабель, без необхідності встановлення модему. Провайдер (ISP - Internet Service Provider) є компанією, яка надає доступ до Інтернету через свою мережу. Він забезпечує підключення до Інтернету через кабель, оптичний кабель, DSL (Digital Subscriber Line) або інші технології.

Логічне розділення мережевого трафіку у домашній мережі грає ключову роль у забезпеченні безпеки, ефективності та контролю над з'єднанням з Інтернетом. Один із основних елементів цього розділення - Network Address Translation (NAT), який виконує перетворення локальних IP-адрес пристроїв у домашній мережі на глобальні IP-адреси для зовнішнього з'єднання з Інтернетом. Такий механізм дозволяє заощадити глобальні IP-адреси та підвищує безпеку, оскільки зовнішні пристрої не можуть напряду звертатися до локальних пристроїв, що знаходяться за NAT.

Ще одним важливим елементом логічного розділення є Firewall (брандмауер), який фільтрує мережевий трафік та контролює доступ до мережних ресурсів. Він визначає правила доступу для пакетів даних, що перетинають межу між Інтернетом і домашньою мережею, блокуючи небажані підключення та захищаючи мережу від потенційних загроз, таких як вторгнення чи зловмисний код. Firewall може бути реалізований як апаратний (наприклад, вбудований у роутер) або програмний (віддільно встановлений на пристрої).

Крім того, логічне розділення може включати в себе створення Virtual Private Network (VPN) - віртуальної приватної мережі, що забезпечує безпечний та зашифрований канал зв'язку між різними мережами або пристроями. Це особливо корисно для віддаленого доступу до домашніх ресурсів з будь-якої точки Інтернету, забезпечуючи високий рівень конфіденційності та захисту даних.

Усі ці технології та методи логічного розділення мережевого трафіку в домашній мережі допомагають забезпечити безпеку, ефективність та контроль над

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

з'єднанням з Інтернетом, що є важливими аспектами для сучасних користувачів з урахуванням зростаючих вимог до захисту даних та конфіденційності.

2.2 Ф о р м у в а н н я в и м о г щ о д о ф у н к ц і й т а

Перед початком будь-якого проекту зі створення мережі належить вкласти час у те, щоб ретельно проаналізувати потреби, визначити цілі та сформулювати вимоги, яким вона повинна відповідати. Цей етап підготовки дуже важливий, оскільки від правильно визначених вимог залежить ефективність, безпека та зручність використання мережі. Перший крок – визначити призначення мережі. Це дасть зрозуміти масштаби і орієнтовну структуру мережі.

Homelab – це інфраструктура, що основана на домашній мережі, створена для експериментів, розвитку технічних навичок та реалізації персональних проектів, з можливістю впроваджувати сервіси для зручності в щоденному використанні комп'ютера. Вона є своєрідним лабораторним майданчиком, де можна тестувати нові технології, розгортати власні веб-сервіси, гостити ігрові сервери та багато іншого. Для успішного функціонування homelab необхідно мати добре організовану та захищену домашню мережу.

Перед початком створення домашньої мережі важливо провести аналіз потреб та сформулювати вимоги до її функціонування.

Економічна ефективність є одним із ключових аспектів при плануванні мережі. Оскільки бюджет обмежений, важливо ретельно розпланувати витрати. З урахуванням обмеженого бюджету у розмірі 20 000 гривень для homelab, необхідно збалансувати якість та кількість придбаних пристроїв.

Надійність мережі - це ще один важливий аспект. Оскільки мережа призначена для використання невеликою сім'єю та хобі проектів, вимога до 24/7 доступності сервісів є не настільки критичною. У випадку виникнення проблем з доступністю, їх можна вирішити вручну, тому резервні джерела живлення та мережеві з'єднання не є обов'язковими.

Продуктивність мережі визначається її здатністю забезпечувати стабільне та швидке Інтернет-з'єднання. Для домашньої мережі важливо мати можливість без

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

перешкод переглядати відео високої якості та грати в онлайн ігри з низькою затримкою. Тому потрібно забезпечити максимальну швидкість з'єднання у 1000 Mb/s, що визначатиме моделі пристроїв та тип ethernet кабелів.

Масштабованість мережі - це важлива характеристика, яка визначає, наскільки ефективно мережа може розширюватися та пристосовуватися до зростання обсягів даних та кількості підключених пристроїв. Оскільки домашня мережа не буде розвиватися у щось дуже велике, на відміну від мереж компаній тощо, то масштабованість у цьому проекті – це більше про логічне налаштування мережі – планування адресного простору, VLAN, розбиття мережі на окремі логічні компоненти.

Безпека є одним з найважливіших аспектів будь-якої мережі. Вона включає в себе захист від зовнішніх атак, вірусів та зловживань доступом. Забезпечення високого рівня безпеки дозволяє уникнути втрати даних, порушення приватності та забезпечити стабільну роботу мережі. Безпека в плануванні мережі – це про її дизайн. Homelab мережа матиме веб-сервіси, відкриті для світу, отже це необхідно відокремити в окремий компонент, під захистом підмереж, VLAN, чи фаєрволів. Також, непогано мати гостьову мережу, щоб розмежувати домашній та гостьовий трафік і не надавати доступ до мережі, де не треба.

Керованість мережі передбачає наявність зручних та ефективних засобів для керування всіма аспектами мережі. Це включає в себе можливість налаштування, моніторингу та розвитку мережевої інфраструктури без зайвих труднощів та затрат часу. Моє бачення зручної керованої мережі – це централізована конфігурація, проста, але водночас ефективна і безпечна мережа.

2.3 Проект домашньої мережі

2.2.1 Планування фізичної структури мережі

Основна концепція домашньої мережі homelab базується на розділенні мережі на підмережі, кожна з яких виконуватиме визначену роль. Ця стратегія дозволяє створити як фізичні, так і логічні кордони, що розширює можливості налаштування підмереж та контролю їх доступу відповідно до конкретних завдань. Більше того, це значно підвищує рівень безпеки мережі, оскільки дозволяє ізолювати різні

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

частини мережі від небезпек та забезпечує ефективний механізм контролю доступу до ресурсів. Такий підхід не лише забезпечує структурованість та організованість мережі, але й дозволяє максимально використовувати її потенціал для різноманітних завдань та проектів.

Ще однією перевагою підходу до розділення мережі на підмережі є її масштабованість. У випадку потреби у створенні нового сегменту, наприклад, для Інтернету речей (IoT), не потрібно змінювати всі налаштування, що дозволяє уникнути порушень у роботі попередніх сегментів через невідомі фактори. Крім того, такий підхід забезпечує зручність обслуговування та користування мережею. У разі необхідності масштабування будь-якого сегмента також не виникне проблеми, якщо це запланувати на етапі логічного дизайну мережі.

Створення мережі відбувається не з нуля. Наразі, мій комп'ютер і ноутбук підключенні до домашньої мережі батьків, тому при створенні мережі існує два можливих сценарії розвитку переходу. Перший – це провести у квартиру ще один кабель поточного чи іншого провайдера і вважати це повністю незалежним підключенням. Це доволі простий і надійний варіант, але мінусом є те, що необхідно платити за окреме підключення. Наступний варіант – це запросити у поточного провайдера додаткову зовнішню адресу, підключити між маршрутизатором поточної домашньої мережі та кабелем провайдера комутатор, що надасть змогу поставити ще один маршрутизатор, вже homelab мережі, і завдяки DHCP з можливістю закріплення IP-адреси до MAC-адреси можна зробити домашню мережу homelab, аналогічно до першого варіанту. У цього підходу зручність в економії грошей та часу, можливість масштабування окремих публічних мереж надалі, але є і мінус – це спільна швидкість інтернету на дві мережі, хоча в цьому разі це не є проблемою, так як на даний момент всі пристрої і так в єдиній мережі і поточної швидкості інтернету достатньо, а в крайньому випадку можна запросити більшу швидкість від провайдера. Мені пощастило, оскільки мій поточний провайдер Corbina Telecom[6], що надає інтернет у батьківську домашню мережу, має такі послуги надання додаткових статичних IP-

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

адрес, в такому випадку рахунок за інтернет на місяць підвищено лише на 30 гривень. Тепер можна створювати проект домашньої мережі.

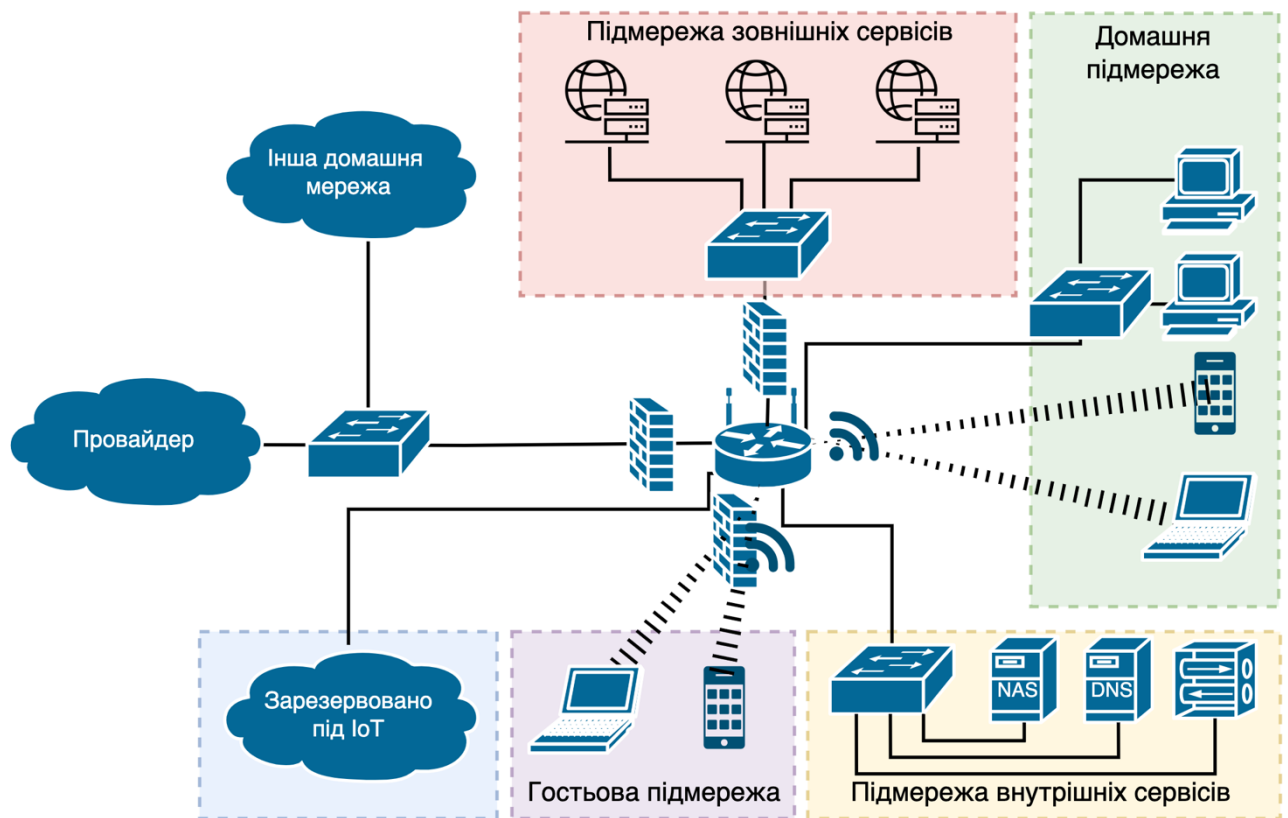


Рис. 2.1. Проект домашньої мережі homelab

Мережа homelab складається з ряду підмереж, кожна з яких виконує певні функції та має свої правила взаємодії:

- Підмережа А - це напіввідкрита мережа, призначена для сервісів, які потребують доступу зовнішніх користувачів. Тут розміщуються веб-сервери, хостинг ігрових серверів та інші відкриті сервіси. Однак доступ до інших частин мережі обмежений з міркувань безпеки.
- Підмережа Б - це мережа, призначена для домашніх пристроїв, таких як комп'ютери та телефони. Вона має підключення як по етапному кабелю, так і по бездротовому зв'язку Wi-Fi. Ця підмережа має широкий доступ до інших частин мережі для зручного керування та налаштування компонентів.
- Підмережа В - це мережа, яка використовується для внутрішніх сервісів, таких як VPN (віртуальна приватна мережа), NAS (мережеве сховище), pi-Hole DNS (система фільтрації реклами). Вона забезпечує безпеку та надійність внутрішніх комунікацій та сервісів.

- Підмережа Г - це зарезервована мережа для Інтернету речей (IoT). Ця підмережа не налаштовується детально, але вона ізольована від інших частин мережі.
- Підмережа Д - це гостьова мережа, яка призначена для бездротового підключення гостей та має обмежений доступ тільки до Інтернету. Це забезпечує безпеку основним частинам мережі в разі, якщо гості користуються бездротовим зв'язком.

2.2.2 Планування логічної структури мережі

Після створення плану фізичної структури мережі необхідно приділити уваги логічному плануванню мережі. Перше і головне у плануванні логічної структури – це вибір IP-адресної схеми. Це основа мережевого спілкування. Це включає в себе рішення про використання IPv4 або IPv6 і планування діапазону IP-адрес для призначення пристроям. Адреси IPv4 представлені в десятковій системі запису (наприклад, 192.168.1.1), тоді як адреси IPv6 представлені в шістнадцятковій системі (наприклад, 2001:0db8:85a3:0000:0000:8a2e:0370:7334). У створенні домашньої мережі під час планування схеми IP-адресації йдеться про непублічну IP-адресу, так як публічна IP-адреса взагалі часто може постійно змінюватись, але слід зазначити, що в моєму випадку адреса повністю статична і IPv4. Інше питання – вибір адресної схеми приватної мережі, що знаходитиметься за NAT маршрутизатора. Для зручності і доступності в домашній мережі буде використано IPv4, бо нема проблеми з обмеженим адресним простором за NAT. Хоча можна обрати будь-яку адресу, краще притримуватися стандартів.

Приватні адресні простори – це зарезервовані діапазони IP-адрес, призначені для використання в приватних мережах, ізольованих від загальнодоступного Інтернету. Ці адреси не маршрутизуються в Інтернеті та зазвичай використовуються в домашніх, офісних або корпоративних мережах. RFC 1918 визначає три основні простори приватних адрес:

- 10.0.0.0/8: цей діапазон складається з усіх IP-адрес від 10.0.0.0 до 10.255.255.255, загалом 16 777 216 адрес.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

- 172.16.0.0/12: цей діапазон охоплює IP-адреси від 172.16.0.0 до 172.31.255.255, що пропонує 1 048 576 адрес.
- 192.168.0.0/16: цей діапазон включає IP-адреси від 192.168.0.0 до 192.168.255.255, що забезпечує 65 536 адрес. Часто встановлений як простір за замовченням в більшості маршрутизаторів.

Я обираю адресний простір 10.0.0.0/8, бо він простий, дає великий простір для масштабування.

Наступний крок – створення підмереж. Це передбачає поділ великої мережі на менші підмережі. Кожна підмережа має власний діапазон IP-адрес і маску підмережі, яка визначає мережеву частину та частину хоста IP-адреси. Підмережі допомагають в організації та оптимізації мережевого трафіку та підвищують безпеку шляхом ізоляції різних частин мережі. Але перед тим, як віділяти маску підмережі, я хочу розглянути ще один компонент, що допоможе у майбутньому з масштабуванням – це VLAN (віртуальні локальні мережі).

VLAN використовуються для логічного сегментування однієї фізичної мережі на кілька віртуальних мереж, кожна з яких має власний широкомовний домен. VLAN покращують мережеву безпеку, продуктивність і керованість шляхом ізоляції трафіку та контролю зв'язку між різними сегментами мережі. На даний момент моя мережа homelab не потребуватиме VLAN, бо вона достатньо мала. Але у майбутньому це, можливо, знадобиться, особливо у створенні IoT, де, наприклад, знадобилося б відокремлення сенсорів, актуаторів та контролерів у окремі віртуальні мережі.

Отже, враховуючи підмережі та VLAN, я хочу зробити маску підмережі 10.0.0.0/24, де з 9 по 16 біт йде номер підмережі, а з 17 по 24 йде номер VLAN. Це було б неекономічно з точки зору публічних адрес, але в невеликих домашніх мережах це норма і дає зручно візуально розуміти, що, наприклад 10.20.10.1/24 – це домашня підмережа, з номером мережі 20, номером підмережі 10 і хостом 1. Ця схема надає можливість мати 256 підмереж, в кожній з якій може бути 256 віртуальних локальних мереж, в кожній з якій може бути 254 хоста.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Наступним кроком буде визначення системи найменувань та планування DNS. Система найменувань забезпечує структурований підхід до найменування пристроїв мережі, серверів, доменів та ресурсів у мережі. Ця систематична схема найменувань гарантує однорідність та зручність управління. Доменна система імен (DNS) доповнює систему найменувань, служачи ієрархічною та децентралізованою системою для перекладу доменних імен (наприклад, `www.example.com`) на IP-адреси (наприклад, `192.0.2.1`). DNS сприяє зручному доступу до ресурсів, дозволяючи користувачам використовувати зрозумілі для людини назви замість числових IP-адрес. Разом, системи найменувань та DNS спрощують управління мережею та покращують доступ користувачів до ресурсів мережі. Для проекту `homelab` я вибрав схему найменувань за хостом, кожен пристрій мережі має унікальне ім'я, аналогічно хімічному елементу в періодичній таблиці. По-перше, це просто цікавий спосіб називати хости з легко запам'ятовуваними назвами, по-друге, це зручно для створення коротких хостнейсів, наприклад (`hydrogen – h`, `helium – he`). Мій зареєстрований домен на `cloudflare` – `mahurin.xyz`. Для зовнішніх сервісів буде використовуватися саме цей домен, наприклад `blog.mahurin.xyz`, `cv.mahurin.xyz`, тощо. Для домашніх сервісів, можна буде використати внутрішній DNS-сервер, всі домашні сервіси знаходимуться в домені `home.mahurin.xyz` (`nas.mahurin.xyz`, `hydrogen.mahurin.xyz`). Окрім створення А записів, ще можна використати схему найменувань з назвами хімічних елементів для створення скорочених CNAME записів, наприклад `he.home -> helium.home.mahurin.xyz`.

Наступний крок – розподілення IP-адрес. Є два варіанти отримання IP-адреси хостом – або вводити статично, або отримувати завдяки DHCP. Від цього залежить, для яких хостів треба прописувати IP-адреси. У статичної IP-адресації є свої плюси – це те, що можна постійно знати, що дана IP-адреса відповідає конкретному хосту, це критично для серверів та мережевих пристроїв, але мінус в тому, що це треба це налаштовувати вручну і налаштування відрізняється залежно від типу пристрою та операційної системи, що не є зручним. Друга опція – отримання адрес за DHCP, це ідеально підходить для пристроїв користувачів через свою зручність, але не для серверів. Є рішення – це налаштувати DHCP сервер, який буде прив'язувати IP-

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

адреси до MAC-адрес інтерфейсів хостів, що дуже полегшить керування та розподіл IP-адрес, бо всі налаштування будуть проведені на маршрутизаторі, і, у випадку зміни мережевих налаштувань (зміна бажаної IP-адреси, шлюз по замовчуванню, DNS сервери), це робиться в пару кліків. Як було зазначено вище, для клієнтів немає необхідності закріплювати IP-адреси до MAC-адрес, але через те, що моє середовище тісно пов'язано з створенням та розробкою проектів, було б непогано виділити статичну адресу через DHCP, а для телефонів та інших невідомих пристроїв буде видаватися адреса випадково. Для цього необхідно створити пул IP-адрес, нехай адреси з 10.x.x.1/24 по 10.x.x.99/24 будуть зарезервовані для мережевих пристроїв, серверів, та комп'ютерів, щоб вручну закріплювати їм IP-адреси до MAC-адрес і вони видавалися завдяки DHCP. Адреси з 10.x.x.100/24 по 10.x.x.254/24 будуть пулом адрес для динамічної видачі DHCP.

На даному етапі було би добре створити таблицю IP-адресації мережі, але для цього спочатку необхідно закупити обладнання і визначити назви інтерфейсів та їх MAC-адреси, тому цю таблицю буде створено у третьому розділі.

2.2.3 Аналіз та вибір обладнання

Для реалізації проекту наступним кроком буде аналіз та вибір обладнання та його придбання. До цього кроку треба поставитись відповідально та вибрати пристрої, що задовільняють вимоги з першого розділу та зможуть бути використані для побудови відповідно фізичному та логічному плану мережі, розроблених у другому розділі. Перше і найпростіше, з чого я би хотів почати – це вибір кінцевих пристроїв. Кінцеві пристрої користувачів вже наявні, тому необхідно вибрати сервери. Тип та кількість серверів залежить від поставлених задач і сервісів, що у мають бути у homelab мережі.

Базові сервіси, які можуть бути розгорнуті в мережі homelab:

- Веб-сервер(и)
- Платформа для віртуалізації
- NAS сховище
- VPN концентратор
- piHole DNS сервер

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

Raspberry Pi може бути ідеальним вибором для хостингу різноманітних сервісів у мережі homelab. Raspberry Pi визначається як міні-комп'ютер розміром з кредитну картку, який сумісний з будь-яким апаратним пристроєм введення та виведення, таким як монітор, телевізор, миша чи клавіатура, – ефективно перетворюючи налаштування на повноцінний ПК низька вартість.

Raspberry Pi можна використовувати для різних цілей, включаючи вивчення мов програмування та управління мережею. Він багатофункціональний і за останні кілька років набув навіть більшої популярності, ніж передбачалося спочатку.

Raspberry Pi - це програмований пристрій. Він оснащений усіма критичними функціями материнської плати середнього комп'ютера, але без периферійних пристроїв і внутрішньої пам'яті. Щоб налаштувати комп'ютер Raspberry, вам знадобиться SD-карта, вставлена у відведене місце. На карті SD повинна бути встановлена операційна система, і вона потрібна для завантаження комп'ютера. Комп'ютери Raspberry сумісні з ОС Linux. Це зменшує необхідний обсяг пам'яті та створює середовище для різноманітності.



Рис. 2.2. Raspberry Pi 4B

Raspberry 4B, випущений у 2019 році, є суттєвим удосконаленням своїх попередників із різним об'ємом пам'яті від 2 до 8 ГБ оперативної пам'яті. Він також

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

має швидший процесор 1,5 ГГц і хороше поєднання USB-портів 2.0 і 3.0. Pi 4B — ідеальна модель Raspberry, оскільки вона підходить практично для будь-якого випадку використання з більшим об'ємом оперативної пам'яті, щоб задовольнити навіть найвідданіших програмістів. Залежно від пам'яті ціна коливається від 35 до 75 доларів США, але кожен має всі варіанти підключення.

На жаль, в Україні нема офіційного представника Raspberry pi, що робить їх ціну вищою через посередників. Є варіант замовити Raspberry Pi 4B 8GB з Китаю, це буде набагато дешевше, хоч і з ризиками підробки. В Україні середня ціна за Raspberry Pi 4B 8GB – 4000 гривень, на Aliexpress можливо замовити набір з Raspberry Pi 4B 8GB з блоком живлення, корпусом, охолодженням, hdmi кабелем, sd картою за 3900 гривень.



гpn.3,901.05
 gpn.3,880.74 each, ≥ 3 pieces
 Extra 2% off with coins

Choice
 · Free shipping over gpn.380.00
 · Fast delivery by May 21

Raspberry Pi 4 Model B 4B RAM 1GB 2GB 4GB 8GB Kit+ Case+32 64 128 GB TF Card+ Power Adapter+ Fan+ Heat Sink+Video Cable for Pi 4

★★★★★ 4.8 168 Reviews | 900+ sold

Bundle: 8GB RAM
 1GB RAM 2GB RAM 4GB RAM 8GB RAM

Color: 64GB Card
 32GB Card 64GB Card 128GB Card No SD Card

BUY NOW >

Рис. 2.3. Замовлення Raspberry Pi 4B з Китаю

У контексті віртуалізації, Raspberry Pi не є оптимальним вибором через обмежені ресурси та використання архітектури ARM. Якщо метою є віртуалізація Linux та Windows на гіпервізорі типу 1, необхідно мати пристрій з архітектурою x86. Для цього підходить старий комп'ютер. Він має 16 ГБ оперативної пам'яті та процесор Intel i5-2500 (4) @ 3.700GHz, що вважається доволі слабким у сучасному світі комп'ютерів, але достатнім для використання як сервер віртуалізації для тестування або хостингу невеликих проектів.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

Сервери вибрано, тепер необхідно вибрати мережеві пристрої. На черзі – бездротовий маршрутизатор. Це серце домашньої мережі, він виконує функцію маршрутизатора, комутатора, точки доступу, мережевого екрану, DHCP серверу, DNS серверу, тощо. Через те, що це головний компонент мережі, його потрібно обирати дуже ретельно. Перед вибором моделі спочатку треба ознайомитися з брендами. Було розглянуто таких виробників, як TP-Link, ASUS, Netgear, Ubiquiti, MikroTik та Linksys. Більше інформації подано в Додатку А.

Серед розглянутих брендів я обрав MikroTik – хоча їх налаштування може бути доволі складним для початківців, оволодіння та налаштування цієї техніки надасть корисні знання, які можуть стати в нагоді при працевлаштуванні, оскільки багато компаній все частіше використовують обладнання MikroTik для корпоративної мережі. Крім того, це найкращий варіант для детального налаштування мережі. За співвідношенням ціна-якість це один із найкращих варіантів.

Розглянемо різні моделі маршрутизаторів MikroTik. Для побудови мережі homelab підійде бездротовий маршрутизатор з категорії для дому і офісу. У MikroTik наявні такі моделі:

- hAP lite: hAP lite — це невеликий, доступний роутер, який підходить для невеликих домашніх мереж. Він має чотири порти Gigabit Ethernet і одноядерний процесор.
- RB951Ui-2HnD: RB951Ui-2HnD — це більш потужний роутер, що підходить для середніх домашніх лабораторій. Він має п'ять портів Gigabit Ethernet, двоядерний процесор та вбудований USB-порт.
- hAP ac²: hAP ac² — це дводіапазонний роутер, який підтримує бездротове підключення 802.11ac. Він підходить для домашніх лабораторій, що потребують високошвидкісного бездротового з'єднання.
- cAP ac: cAP ac — це стельовий точковий доступ, який підтримує бездротове підключення 802.11ac. Він підходить для домашніх лабораторій, що потребують широкого покриття бездротової мережі.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

- wAP ac: wAP ac — це настінний точковий доступ, який підтримує бездротове підключення 802.11ac. Він підходить для домашніх лабораторій, що потребують широкого покриття бездротової мережі.
- hAP ax²: hAP ax² — це дводіапазонний роутер, який підтримує бездротове підключення 802.11ax (Wi-Fi 6). Він підходить для домашніх лабораторій, що потребують найновішої та найшвидшої бездротової технології.
- L009UiGS-2HaxD-IN: L009UiGS-2HaxD-IN — це дводіапазонний роутер, який підтримує бездротове підключення 802.11ax (Wi-Fi 6). Він має п'ять портів Gigabit Ethernet та чотириядерний процесор.
- hAP ax³: hAP ax³ — це тридіапазонний роутер, який підтримує бездротове підключення 802.11ax (Wi-Fi 6). Він підходить для домашніх лабораторій, що потребують високошвидкісного бездротового з'єднання та широкого покриття.
- Chateau LTE6: Chateau LTE6 — це стільниковий роутер, який підтримує підключення LTE. Він підходить для домашніх лабораторій, що потребують доступу до інтернету в віддаленому місці.
- Audience: Audience — це система сітчастої мережі, яку можна використовувати для створення цілісної домашньої мережі Wi-Fi. Вона підходить для домашніх лабораторій, що потребують широкого покриття бездротової мережі.
- RB4011iGS+5HacQ2HnD-IN: RB4011iGS+5HacQ2HnD-IN — це високопродуктивний роутер, який підходить для великих домашніх лабораторій. Він має десять портів Gigabit Ethernet, чотириядерний процесор та вбудований порт SFP+.

Після аналізу було обрано hAP ax² - це дуже хороший варіант, який підходить під вимоги, має баланс між ціною та можливостями. Проблем з покупкою в Україні цього пристрою немає, було замовлено на Rozetka за 3700 грн, що відповідає рекомендованій ціні на сайті MikroTik.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		39



Рис. 2.4. MikroTik hAP ax²

Наступна задача – знайти комутатори. На них не плануються жодні додаткові налаштування, лише для розширення кількості підключених пристроїв до мережі. Для цього підійдуть неконтрольовані комутатори, що значно дешевші за контрольовані. Їх роль в мережі не критична – у випадку несправності буде просто їх замінити, оскільки налаштувань ніяких не треба, потреби у проведенні ґрунтового аналізу не було. Було переглянуто сайт Rozetka і вибрано дешевий з найкращими відгуками що задовільняють наші вимоги - комутатор TP-LINK LS1005G. За проектом мережі їх необхідно 4 штуки, але наразі немає необхідності підключати один до підмережі доманшніх сервісів через те, що в тій підмережі на даний момент лише один пристрій, а цей пристрій в широкій наявності на різних сайтах і легке отримання у цей же чи наступний день після замовлення. Тому було куплено лише 3 таких по 529 грн кожен.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

Тепер наявні усі пристрої мережі, їх необхідно з'єднати, для цього треба ethernet кабель, мінімум це cat5e зі швидкістю до 1000 Мбіт на секунду. Було придбано різні кабелі з різних сайтів з запасом. Для критично важливих вузлів мережі було придбано Cat8. Також придбано для живлення мережі необхідний мережевий фільтр. В майбутньому для NAS серверу, необхідно придбати SSD накопичувач. Вибір впав на Kingston SSDNow A400 480GB SATA III, бренд відомий своєю надійністю.

Таблиця 2.1. Вартість придбаного обладнання

Назва	Кількість	Ціна, грн	Сума, грн
Маршрутизатор MikroTik hAP ax2	1	3699	3699
Комутатор TP-LINK LS1005G	3	529	1587
Raspberry Pi 4b 8gb kit	2	3126	6252
Патч корд Cat8 3 м	2	106	212
Патч корд Cat5E 3 м	2	66	132
Патч корд Cat5E 1.5 м	6	50	300
Мережевий фільтр-подовжувач RZTK	1	599	599
Kingston SSDNow A400 480GB 2.5" SATAIII	1	1349	1349
Зовнішня кишеня RZTK 2.5" SATA USB Type-C 3.1	1	299	299
Разом			14429

2.4 Побудова та конфігурація домашньої

2.4.1 Збірка та підключення всіх пристроїв мережі

Побудова та конфігурація домашньої мережі починається з підготовки до монтажу всіх компонентів мережі. Оскільки все необхідне обладнання для проекту homelab вже придбано, постає наступне завдання — правильне розміщення та монтаж мережевого обладнання для забезпечення максимальної ефективності та зручності користування мережею.

Перед початком монтажу необхідно оцінити доступний простір у квартирі та визначити оптимальні місця для розміщення мережевого обладнання. Серед можливих варіантів розміщення: просте розміщення на поверхні столу або в тумбочці, розміщення в серверній стійці або шафі, настінне монтування обладнання на стіні або дошці, а також розміщення під стільницею.

Настінне монтування обладнання є найкращим варіантом для більшості домашніх мереж. Воно дозволяє розмістити обладнання на стіні або дошці, заощаджуючи

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

корисний простір та забезпечуючи хороший повітряний потік для охолодження. Крім того, настінне монтування надає зручний доступ до обладнання для обслуговування та налаштування. Це також надає мобільність мережі – у випадку переїзду не треба від'єднувати всі мережеві пристрої, лише кінцеві та WAN кабель.

Розміщення під стільницею також може бути практичним варіантом для розміщення обладнання, однак, щоб зробити нормально і не зіпсувати стіл треба мати деякі навички та витратити значний обсяг часу, що не є оптимальним.

З урахуванням усіх перерахованих варіантів, найкращим вибором буде настінне монтування обладнання на дошці. Це оптимальне рішення, яке поєднує в собі економію місця, зручність обслуговування та легкий доступ до всіх компонентів мережі. Обладнання буде надійно прикріплене до дошки, що запобігає випадковому пошкодженню або переміщенню. Настінне монтування також надає більш організований вигляд домашній мережі.

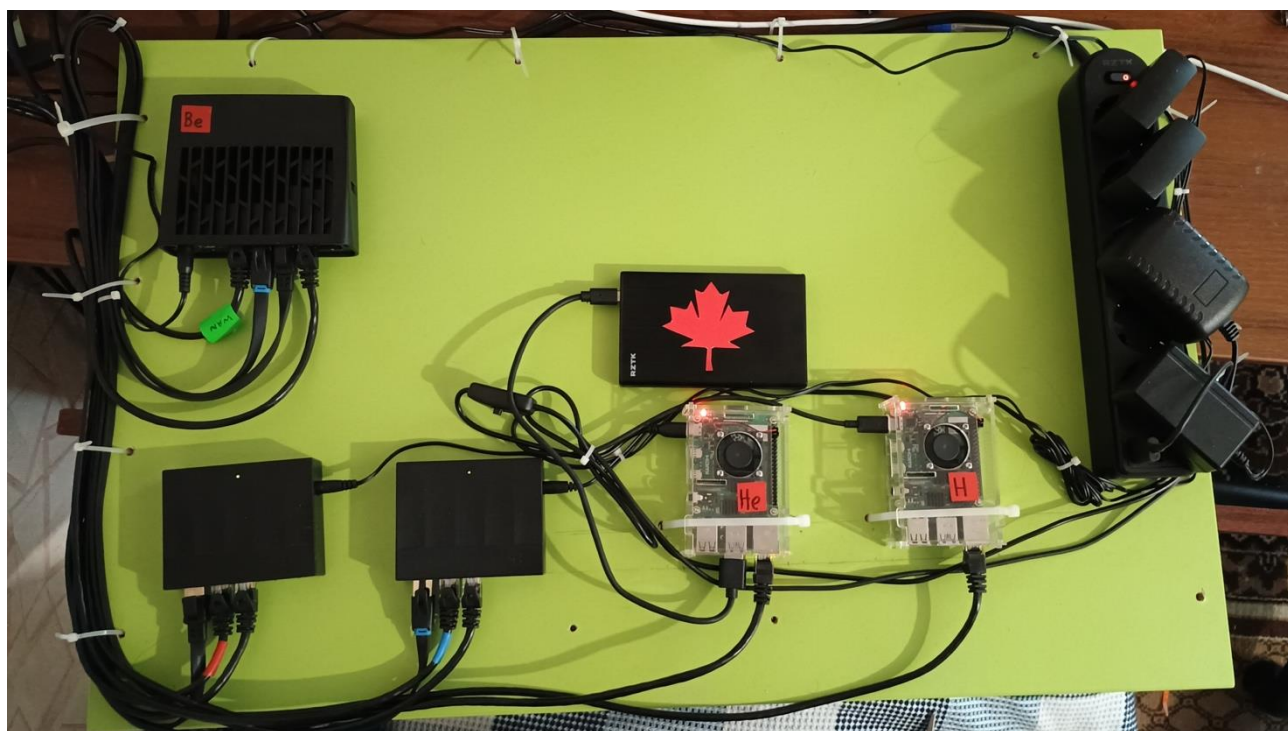


Рис. 2.5. Зібрана домашня мережа

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

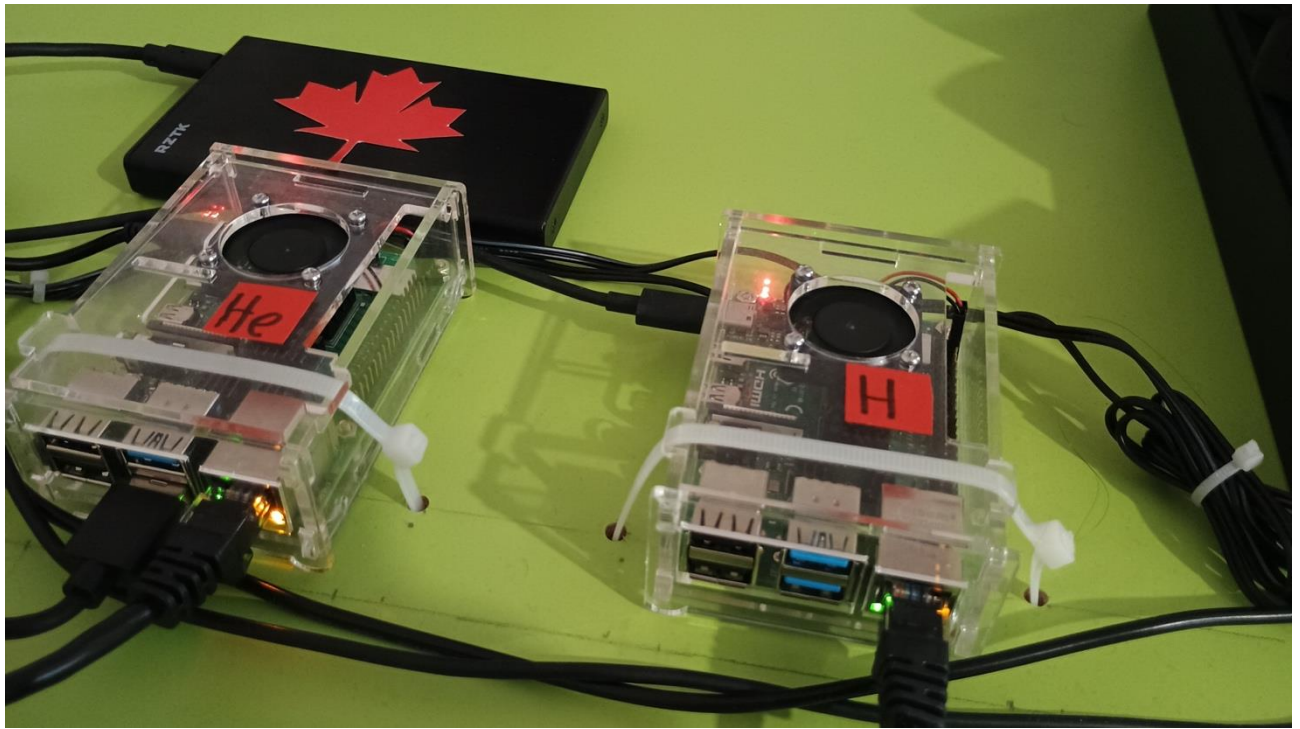


Рис. 2.6. Raspberry Pi сервери



Рис. 2.6. Сегмент мережі перед маршрутизатором

Дошка служить основою для кріплення різних компонентів мережі, які розташовані таким чином, щоб займати оптимально місце і бути достатньо близько до інших взаємопов'язаних компонентів.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

Маршрутизатор, два комутатори та мережевий фільтр закріплені на саморізах завдяки спеціальним кріпленням, передбаченим виробником. Це дозволяє надійно утримувати обладнання на місці, запобігаючи його руху чи падінню. Кріплення також забезпечує зручний доступ до кожного пристрою для обслуговування, налаштування або заміни.

Дві Raspberry Pi, закріплені завдяки просвердленим діркам та стяжкам, займають своє місце на дошці. Це дозволяє утримувати їх надійно на місці, водночас забезпечуючи гнучкість у разі необхідності зміни конфігурації або додавання нових компонентів. Такий спосіб кріплення також забезпечує достатню циркуляцію повітря навколо пристроїв, запобігаючи їх перегріванню. SSD диск у зовнішній кишені закріплений на двосторонній скотч, він доволі легкий і не потребує циркуляції повітря.

У правій верхній частині дошки залишено достатньо місця для розміщення нового обладнання у майбутньому. Це дозволяє розширювати мережу у разі необхідності, наприклад, додавати нові пристрої або компоненти для підвищення продуктивності або масштабованості мережі.

По трьох краях дошки йдуть ethernet кабелі та кабелі живлення, які закріплені стяжками, що надає всьому монтажу охайний та акуратний вигляд. Таке кріплення також дозволяє забезпечити зручний доступ до кабелів у разі необхідності їх заміни або перестановки.

Живлення мережі забезпечується мережевим фільтром, до якого підключені всі блоки живлення від різних пристроїв.

На рис. 2.6 показано початок мережі, пристрій по центру – комутатор, що керує з'єднанням між домашньою мережею батьків, моєю новою homelab мережею та провайдером.

Загалом, на мою думку, для свого бюджету і першої спроби вийшло дуже непогано.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

2.4.2 Налаштування маршрутизатора через RouterOS terminal

Налаштування маршрутизатора за допомогою інтерфейсу командного рядка RouterOS terminal є одним з найважливіших у побудові ефективної та надійної домашньої мережі. Конфігурування маршрутизатора через інтерфейс командного рядка RouterOS terminal є критичним, оскільки стандартні налаштування можуть бути недостатньо захищеними та не відповідати конкретним потребам користувача. Змінюючи параметри маршрутизатора вручну, можна підвищити рівень безпеки мережі, налаштувати функціональність відповідно до власних вимог та оптимізувати роботу всіх підключених пристроїв.

Маршрутизатор MikroTik з операційною системою RouterOS [7] - це потужний і гнучкий інструмент для налаштування та управління домашньою мережею. RouterOS - це операційна система, розроблена компанією MikroTik для їхніх маршрутизаторів і мережевих пристроїв. Вона базується на ядрі Linux і має великий набір функцій, що дозволяють налаштовувати різні аспекти мережі, включаючи маршрутизацію, безпеку, керування трафіком, VPN, файервол, DHCP і багато іншого. RouterOS надає користувачам доступ до інтерфейсу командного рядка (CLI) і веб-інтерфейсу, що дозволяє зручно і ефективно керувати всіма аспектами мережі. Його широкі можливості та гнучкість роблять його дуже хорошим вибором для створення та управління домашньою мережею homelab.

Існують два основних способи налаштування маршрутизатора MikroTik: через графічний інтерфейс (WebFig або WinBox) або консоль. Графічний інтерфейс забезпечує зручне керування усіма налаштуваннями через веб-браузер або окрему програму WinBox. Однак, консольний інтерфейс, який працює через командний рядок, є більш потужним і гнучким засобом налаштування. Він надає доступ до всіх функцій і можливостей RouterOS, дозволяючи більш точно керувати мережею та налаштовувати її за потреби. Хоча графічний інтерфейс може бути простішим у використанні, консольний інтерфейс дозволяє більше можливостей для досвідчених користувачів також, він дозволить зрозуміти детальніше внутрішні процеси та поглибить знання роботи мережевих пристроїв MikroTik. Тому в даному випадку вибір консольного налаштування є більш відповідним, оскільки це

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		45

дозволить отримати більше гнучкості та контролю над мережею, а також розширити свої технічні навички у цій області.

Перший крок - це скинути пристрій до заводських налаштувань, що забезпечить повне видалення попередніх конфігурацій та повернення пристрою до початкового стану. Це важливо, оскільки дозволяє позбутися будь-яких непередбачених змін, які могли б вплинути на подальшу роботу мережі. У моєму випадку пристрій повністю новий, тому цей крок можна пропустити.

Заводські налаштування – це все ж таки якісь налаштування. Необхідно їх зрозуміти, так що вони стануть основою, у яку будуть вноситися зміни. Цю інформацію можна отримати з офіційного сайту MikroTik або при першому підключенні до маршрутизатора. Цей тип конфігурації застосовується до домашніх точок доступу маршрутизаторів, які можуть використовуватися без додаткової настройки (крім паролів маршрутизатора та бездротових ключів). Перший Ethernet завжди налаштований як WAN-порт (захищений брандмауером, включено DHCP-клієнт та вимкнено підключення/виявлення MAC-адреси). Інші порти Ethernet та бездротові інтерфейси додаються до локального моста LAN з встановленим адресом 192.168.88.1/24 та налаштованим DHCP-сервером. У випадку двосмугових маршрутизаторів, один бездротовий налаштовується як точка доступу 5 ГГц, а інший - як точка доступу 2,4 ГГц. Отже, налаштування доволі прості.

Наступним кроком буде підключення до маршрутизатора. Було перевірено і доступ через веб-браузер, так і підключення по ssh, стандартний пароль від акаунта адміністратора знаходиться на коробці. Важливий момент – продумати, як підключатися до комутатора для налаштувань. На жаль, ця модель не має окремого інтерфейсу для конфігурації, тому треба бути обережним при налаштуванні, аби не обірвати собі зв'язок з маршрутизатором. Отже, було підключено з ноутбуку через wi-fi2 інтерфейс, з якого буде налаштовано всі інші інтерфейси, а в кінці, після перепідключення на інших інтерфейс, наприклад eth2, буде доналаштовано wi-fi2 як гостьовий wi-fi.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

```

MMM      MMM      KKK      TTTTTTTTTT      KKK
MMMM     MMMM     KKK      TTTTTTTTTT      KKK
MMM MMMM MMM III KKK KKK RRRRRR      000000      TTT      III KKK KKK
MMM MM  MMM III KKKKKK      RRR RRR 000 000      TTT      III KKKKK
MMM      MMM III KKK KKK RRRRRR      000 000      TTT      III KKK KKK
MMM      MMM III KKK KKK RRR RRR      000000      TTT      III KKK KKK

MikroTik RouterOS 7.8 (c) 1999-2023      https://www.mikrotik.com/

Press F1 for help

-----
---
The following default configuration has been installed on your router:

Welcome to RouterOS!
1) Set a strong router password in the System > Users menu
2) Upgrade the software in the System > Packages menu
3) Enable firewall on untrusted networks
4) Set your country name to observe wireless regulations
-----
-
RouterMode:
* WAN port is protected by firewall and enabled DHCP client
* Wireless and Ethernet interfaces (except WAN port/s)
  are part of LAN bridge
LAN Configuration:
  IP address 192.168.88.1/24 is set on bridge (LAN port)
  DHCP Server: enabled;
  DNS: enabled;
wifi Configuration:
  mode:                ap;
  band:                5ghz-ax;
  tx-chains:           0;1;
  rx-chains:           0;1;
  installation:        indoor;
  ht-extension:        20/40/80mhz;
  wpa2:                yes;

```

Рис. 2.7. MOTD при першому запуску маршрутизатора

Тепер, у межах безпекових налаштувань потрібно створити нового користувача з правами адміністратора та унікальним паролем (рис. 2.8-2.9).

```

[admin@MikroTik] > /user add name=dedly password=[REDACTED] group=full
[admin@MikroTik] > /system identity set name=beryllium
[admin@beryllium] > /user print
Columns: NAME, GROUP
# NAME  GROUP
;;; system default user
0 admin  full
1 dedly  full
Connection to 192.168.88.1 closed.
PS C:\Users\dedly> ssh dedly@192.168.88.1

```

Рис. 2.8. Створення нового користувача-адміністратора

```
[dedly@beryllium] > /user remove admin
[dedly@beryllium] > /user print
Columns: NAME, GROUP
# NAME    GROUP
0 dedly   full
```

Рис. 2.9. Видалення стандартного користувача-адміністратора

Щоб створити окремі підмережі та ізолювати певні порти для окремих сегментів мережі, необхідно спочатку виконати такі дії:

- Визначити порти, які потрібно розділити на власні підмережі, ефективно ізолюючи їх від стандартної конфігурації мосту.
- Видалити ці ідентифіковані порти з інтерфейсу мосту, щоб створити окремі сегменти мережі.
- Перевірити оновлену конфігурацію, щоб переконатися, що потрібні порти успішно видалено з інтерфейсу мосту, підтверджуючи створення окремих підмереж.

Все це можна виконати командами, наведеними на рис. 2.10.

```
[dedly@beryllium] > /interface bridge port print
Flags: I - INACTIVE
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COS
T, HORIZON
# INTERFACE BRIDGE HW PVID PRIORITY PATH-COST IN HORIZON
;;; defconf
0 ether2     bridge yes 1 0x80      10 10 none
;;; defconf
1 ether3     bridge yes 1 0x80      10 10 none
;;; defconf
2 ether4     bridge yes 1 0x80      10 10 none
;;; defconf
3 I ether5     bridge yes 1 0x80      10 10 none
;;; defconf
4 I wifi1     bridge      1 0x80      10 10 none
;;; defconf
5 wifi2      bridge      1 0x80      10 10 none
[dedly@beryllium] > /interface bridge port remove 0,1,2,3,4
[dedly@beryllium] > /interface bridge port print
Columns: INTERFACE, BRIDGE, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COST, H
ORIZON
# INTERFACE BRIDGE PVID PRIORITY PATH-COST IN HORIZON
;;; defconf
0 wifi2      bridge 1 0x80      10 10 none
[dedly@beryllium] >
```

Рис. 2.10. Видалення інтерфейсів з віртуального інтерфейсу моста

Wifi2 не видалено поки що, щоб не обірвалося з'єднання.

Для створення моста для домашньої мережі та забезпечення того, щоб пристрої, підключені як до бездротового інтерфейсу wifi1, так і до ethernet-інтерфейсу ether3, були частиною однієї підмережі, необхідно виконати наступні кроки (рис. 2.11-2.12):

- Створити новий мостовий інтерфейс з назвою "lan2-bridge", який буде слугувати основою для домашньої мережі.
- Додати ethernet-інтерфейс ether3 до "lan2-bridge", щоб включити пристрої з підключенням по кабелю до домашньої мережі.
- Додати бездротовий інтерфейс wifi1 до "lan2-bridge", щоб включити бездротові пристрої до домашньої мережі.
- Перевірити конфігурацію виведенням списку портів на інтерфейсі "lan2-bridge".

```
[dedly@beryllium] > /interface bridge add name=lan2-bridge comment="LAN-2 -
Home Network"
[dedly@beryllium] > /interface bridge port add bridge=lan2-bridge interface=
ether3
[dedly@beryllium] > /interface bridge port add bridge=lan2-bridge interface=
wifi1
[dedly@beryllium] > /interface bridge port print
Flags: I - INACTIVE
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COS
T, HORIZON
# INTERFACE BRIDGE HW PVID PRIORITY PATH-COST IN HORIZON
;;; defconf
0 wifi2 bridge 1 0x80 10 10 none
1 ether3 lan2-bridge yes 1 0x80 10 10 none
2 I wifi1 lan2-bridge 1 0x80 10 10 none
[dedly@beryllium] > /interface bridge print
Flags: X - disabled, R - running
0 R ;;; defconf
name="bridge" mtu=auto actual-mtu=1500 l2mtu=65535 arp=enabled
arp-timeout=auto mac-address=48:A9:8A:DF:D2:71 protocol-mode=rstp
fast-forward=yes igmp-snooping=no auto-mac=no
admin-mac=48:A9:8A:DF:D2:71 ageing-time=5m priority=0x8000
max-message-age=20s forward-delay=15s transmit-hold-count=6
vlan-filtering=no dhcp-snooping=no

1 R ;;; LAN-2 - Home Network
name="lan2-bridge" mtu=auto actual-mtu=1500 l2mtu=1568 arp=enabled
arp-timeout=auto mac-address=48:A9:8A:DF:D2:72 protocol-mode=rstp
fast-forward=yes igmp-snooping=no auto-mac=yes ageing-time=5m
priority=0x8000 max-message-age=20s forward-delay=15s
transmit-hold-count=6 vlan-filtering=no dhcp-snooping=no
[dedly@beryllium] >
```

Рис. 2.11. Створення моста для домашньої підмережі та підключення інтерфейсів

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

```

[dedly@beryllium] > /interface set comment="WAN-1 - Internet" ether1
[dedly@beryllium] > /interface set comment="LAN-1 - DMZ" ether2
[dedly@beryllium] > /interface set comment="LAN-2 - Home Network slave" ether3
[dedly@beryllium] > /interface set comment="LAN-3 - Home Service Network" ether4
[dedly@beryllium] > /interface set comment="LAN-4 - IOT" ether5
[dedly@beryllium] > /interface set comment="LAN-2 - Home Network slave" wifi1
[dedly@beryllium] > /interface set comment="LAN-5 - Guest Network" wifi2
[dedly@beryllium] > /interface set comment="Needs to be removed" bridge
[dedly@beryllium] > /interface set comment="LAN-2 - Home Network" lan2-bri
dge
[dedly@beryllium] > /interface print
Flags: R - RUNNING; S - SLAVE
Columns: NAME, TYPE, ACTUAL-MTU, L2MTU, MAX-L2MTU, MAC-ADDRESS
# NAME TYPE ACTUAL-MTU L2MTU MAX-L2MTU MAC-ADDRESS
;;; WAN-1 - Internet
0 R ether1 ether 1500 1568 9214 48:A9:8A:DF:D2:70
;;; LAN-1 - DMZ
1 R ether2 ether 1500 1568 9214 48:A9:8A:DF:D2:71
;;; LAN-2 - Home Network slave
2 RS ether3 ether 1500 1568 9214 48:A9:8A:DF:D2:72
;;; LAN-3 - Home Service Network
3 R ether4 ether 1500 1568 9214 48:A9:8A:DF:D2:73
;;; LAN-4 - IOT
4 ether5 ether 1500 1568 9214 48:A9:8A:DF:D2:74
;;; Needs to be removed
5 R bridge bridge 1500 65535 48:A9:8A:DF:D2:71
;;; LAN-2 - Home Network
6 R lan2-bridge bridge 1500 1568 48:A9:8A:DF:D2:72
;;; LAN-2 - Home Network slave
7 S wifi1 wifi 1500 48:A9:8A:DF:D2:75
;;; LAN-5 - Guest Network
8 RS wifi2 wifi 1500 48:A9:8A:DF:D2:76
[dedly@beryllium] >

```

Рис. 2.12. Додавання опису для інтерфейсів відповідно до їхньої ролі

Для забезпечення належного зв'язку та сегментації мережі IP-адреси призначаються певним інтерфейсам. IP-адреси призначаються різним підмережам, щоб логічно відокремити різні сегменти мережі з метою безпеки, управління та організації (рис. 2.13). Призначення IP-адрес інтерфейсам забезпечує правильну маршрутизацію та з'єднання в кожному сегменті мережі, зберігаючи розділення між ними.

```
[dedly@beryllium] > /ip address add address=10.10.10.1/24 interface=ether2 comment="LAN-1 - DMZ"
[dedly@beryllium] > /ip address add address=10.20.10.1/24 interface=lan2-bridge comment="LAN-2 - Home Network"
[dedly@beryllium] > /ip address add address=10.30.10.1/24 interface=ether4 comment="LAN-3 - Home Service Network"
[dedly@beryllium] > /ip address add address=10.40.10.1/24 interface=ether5 comment="LAN-4 - IOT"
[dedly@beryllium] > /ip address print
Flags: D - DYNAMIC
Columns: ADDRESS, NETWORK, INTERFACE
# ADDRESS NETWORK INTERFACE
;;; defconf
0 192.168.88.1/24 192.168.88.0 bridge
1 D 31.43.154.6/24 31.43.154.0 ether1
;;; LAN-1 - DMZ
2 10.10.10.1/24 10.10.10.0 ether2
;;; LAN-2 - Home Network
3 10.20.10.1/24 10.20.10.0 lan2-bridge
;;; LAN-3 - Home Service Network
4 10.30.10.1/24 10.30.10.0 ether4
;;; LAN-4 - IOT
5 10.40.10.1/24 10.40.10.0 ether5
[dedly@beryllium] >
```

Рис. 2.13. Видача IP-адрес інтерфейсам з розподіленням на підмережі

Сервер DHCP (Dynamic Host Configuration Protocol) — це мережева служба, яка автоматично призначає IP-адреси та інші налаштування мережевої конфігурації пристроям у мережі. Це спрощує мережеве адміністрування шляхом динамічного розподілу IP-адрес із попередньо визначених пулів, усуваючи потребу в ручній конфігурації IP. Сервери DHCP також надають додаткову інформацію про мережу, таку як маска підмережі, шлюз за замовчуванням, адреси DNS-серверів та інші параметри конфігурації, необхідні для зв'язку пристроїв у мережі.

Серверам DHCP потрібні заздалегідь визначені пули адрес, з яких вони можуть динамічно призначати IP-адреси пристроям. Ці пули мають бути налаштовані з діапазоном IP-адрес, які ще не використовуються статично призначеними пристроями, щоб уникнути конфліктів (рис. 2.14).

```
[dedly@beryllium] > /ip pool add name=LAN1-Pool ranges=10.10.10.100-10.10.10.254
[dedly@beryllium] > /ip pool add name=LAN2-Pool ranges=10.20.10.100-10.20.10.254
[dedly@beryllium] > /ip pool add name=LAN3-Pool ranges=10.30.10.100-10.30.10.254
[dedly@beryllium] > /ip pool add name=LAN4-Pool ranges=10.40.10.100-10.40.10.254
[dedly@beryllium] > /ip pool print
Columns: NAME, RANGES
# NAME RANGES
0 default-dhcp 192.168.88.10-192.168.88.254
1 LAN1-Pool 10.10.10.100-10.10.10.254
2 LAN2-Pool 10.20.10.100-10.20.10.254
3 LAN3-Pool 10.30.10.100-10.30.10.254
4 LAN4-Pool 10.40.10.100-10.40.10.254
[dedly@beryllium] >
```

Рис. 2.14. Створення пулів IP-адрес для підмереж

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

Кожен сервер DHCP має бути пов'язаний із певною мережею або інтерфейсом на маршрутизаторі. Цей зв'язок гарантує, що запити DHCP, отримані на цьому інтерфейсі, обробляються відповідним сервером DHCP (рис. 2.15).

```
[dedly@beryllium] > /ip dhcp-server network add address=10.10.10.0/24 gateway=10.10.10.1 dns-server=10.10.10.1 comment="LAN-1 - DMZ"
[dedly@beryllium] > /ip dhcp-server network add address=10.20.10.0/24 gateway=10.20.10.1 dns-server=10.20.10.1 comment="LAN-2 - Home Network"
[dedly@beryllium] > /ip dhcp-server network add address=10.30.10.0/24 gateway=10.30.10.1 dns-server=10.30.10.1 comment="LAN-3 - Home Service Network"
[dedly@beryllium] > /ip dhcp-server network add address=10.40.10.0/24 gateway=10.40.10.1 dns-server=10.40.10.1 comment="LAN-4 - IOT"
[dedly@beryllium] > /ip dhcp-server network print
Columns: ADDRESS, GATEWAY, DNS-SERVER
# ADDRESS GATEWAY DNS-SERVER
;;; LAN-1 - DMZ
0 10.10.10.0/24 10.10.10.1 10.10.10.1
;;; LAN-2 - Home Network
1 10.20.10.0/24 10.20.10.1 10.20.10.1
;;; LAN-3 - Home Service Network
2 10.30.10.0/24 10.30.10.1 10.30.10.1
;;; LAN-4 - IOT
3 10.40.10.0/24 10.40.10.1 10.40.10.1
;;; defconf
4 192.168.88.0/24 192.168.88.1 192.168.88.1
[dedly@beryllium] >
```

Рис. 2.15. Створення мереж dhcp-серверу

Сервери DHCP повинні знати певні параметри мережі, такі як адреса підмережі, адреса шлюзу та адреси DNS-серверів, щоб надати клієнтам під час процесу призначення IP-адреси (рис. 2.16).

```
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN1 lease-time=1d authoritative=yes interface=ether2 address-pool=LAN1-Pool
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN2 lease-time=1d authoritative=yes interface=ether3 address-pool=LAN2-Pool
failure: can not run on slave interface
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN3 lease-time=1d authoritative=yes interface=ether4 address-pool=LAN3-Pool
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN4 lease-time=1d authoritative=yes interface=ether5 address-pool=LAN4-Pool
[dedly@beryllium] > /ip dhcp-server print
Flags: I - INVALID
Columns: NAME, INTERFACE, ADDRESS-POOL, LEASE-TIME
# NAME INTERFACE ADDRESS-POOL LEASE-TIME
0 defconf bridge default-dhcp 10m
1 dhcp-LAN1 ether2 LAN1-Pool 1d
2 dhcp-LAN3 ether4 LAN3-Pool 1d
3 I dhcp-LAN4 ether5 LAN4-Pool 1d
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN2 lease-time=1d authoritative=yes interface=lan2-bridge address-pool=LAN2-Pool
[dedly@beryllium] > /ip dhcp-server print
Flags: I - INVALID
Columns: NAME, INTERFACE, ADDRESS-POOL, LEASE-TIME
# NAME INTERFACE ADDRESS-POOL LEASE-TIME
0 defconf bridge default-dhcp 10m
1 dhcp-LAN1 ether2 LAN1-Pool 1d
2 dhcp-LAN3 ether4 LAN3-Pool 1d
3 I dhcp-LAN4 ether5 LAN4-Pool 1d
4 dhcp-LAN2 lan2-bridge LAN2-Pool 1d
[dedly@beryllium] >
```

Рис. 2.16. Створення мереж dhcp-серверу

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Сервери DHCP повинні знати певні параметри мережі, такі як адреса підмережі, адреса шлюзу та адреси DNS-серверів, щоб надати клієнтам під час процесу призначення IP-адреси (рис. 2.17).

```
[dedly@beryllium] > /ip dhcp-server print
Columns: NAME, INTERFACE, ADDRESS-POOL, LEASE-TIME
# NAME      INTERFACE ADDRESS-POOL LEASE-TIME
0 defconf   bridge    default-dhcp 10m
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN1 lease-time=1d authoritative=
yes interface=ether2 address-pool=LAN1-Pool
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN2 lease-time=1d authoritative=
yes interface=ether3 address-pool=LAN2-Pool
failure: can not run on slave interface
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN3 lease-time=1d authoritative=
yes interface=ether4 address-pool=LAN3-Pool
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN4 lease-time=1d authoritative=
yes interface=ether5 address-pool=LAN4-Pool
[dedly@beryllium] > /ip dhcp-server print
Flags: I - INVALID
Columns: NAME, INTERFACE, ADDRESS-POOL, LEASE-TIME
# NAME      INTERFACE ADDRESS-POOL LEASE-TIME
0 defconf   bridge    default-dhcp 10m
1 dhcp-LAN1 ether2     LAN1-Pool    1d
2 dhcp-LAN3 ether4     LAN3-Pool    1d
3 I dhcp-LAN4 ether5     LAN4-Pool    1d
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN2 lease-time=1d authoritative=
yes interface=lan2-bridge address-pool=LAN2-Pool
[dedly@beryllium] > /ip dhcp-server print
Flags: I - INVALID
Columns: NAME, INTERFACE, ADDRESS-POOL, LEASE-TIME
# NAME      INTERFACE ADDRESS-POOL LEASE-TIME
0 defconf   bridge    default-dhcp 10m
1 dhcp-LAN1 ether2     LAN1-Pool    1d
2 dhcp-LAN3 ether4     LAN3-Pool    1d
3 I dhcp-LAN4 ether5     LAN4-Pool    1d
4 dhcp-LAN2 lan2-bridge LAN2-Pool    1d
[dedly@beryllium] >
```

Рис. 2.17. Створення dhcp серверів та алокація пулів IP-адресів

Призначення IP-адрес MAC-адресам на сервері DHCP дуже зручне для керування мережею, гарантуючи, що певні пристрої постійно отримують однакову IP-адресу під час підключення. Цей процес, відомий як резервування DHCP, спрощує ідентифікацію пристрою та полегшує адміністрування мережі.

Прив'язуючи IP-адреси до MAC-адрес, встановлюється прямий зв'язок між унікальною апаратною адресою (MAC) пристрою та його відповідною IP-адресою (рис. 2.18). Цей зв'язок гарантує, що кожен пристрій зберігає свою призначену IP-адресу незалежно від перезавантаження мережі чи закінчення терміну оренди. Це особливо зручно у випадку, коли у мережі наявно багато різних пристроїв з різними ОС чи системи, які можуть часто перевстановлюватися, наприклад, raspberry pi.

Таким чином конфігурація тримається в одному зручному місці і зменшує ризик неправильної конфігурації та полегшує роботу.

```
[dedly@beryllium] > /ip dhcp-server print
Columns: NAME, INTERFACE, ADDRESS-POOL, LEASE-TIME
# NAME      INTERFACE ADDRESS-POOL LEASE-TIME
0 defconf   bridge    default-dhcp 10m
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN1 lease-time=1d authoritative=
yes interface=ether2 address-pool=LAN1-Pool
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN2 lease-time=1d authoritative=
yes interface=ether3 address-pool=LAN2-Pool
failure: can not run on slave interface
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN3 lease-time=1d authoritative=
yes interface=ether4 address-pool=LAN3-Pool
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN4 lease-time=1d authoritative=
yes interface=ether5 address-pool=LAN4-Pool
[dedly@beryllium] > /ip dhcp-server print
Flags: I - INVALID
Columns: NAME, INTERFACE, ADDRESS-POOL, LEASE-TIME
# NAME      INTERFACE ADDRESS-POOL LEASE-TIME
0 defconf   bridge    default-dhcp 10m
1 dhcp-LAN1 ether2     LAN1-Pool    1d
2 dhcp-LAN3 ether4     LAN3-Pool    1d
3 I dhcp-LAN4 ether5     LAN4-Pool    1d
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN2 lease-time=1d authoritative=
yes interface=lan2-bridge address-pool=LAN2-Pool
[dedly@beryllium] > /ip dhcp-server print
Flags: I - INVALID
Columns: NAME, INTERFACE, ADDRESS-POOL, LEASE-TIME
# NAME      INTERFACE ADDRESS-POOL LEASE-TIME
0 defconf   bridge    default-dhcp 10m
1 dhcp-LAN1 ether2     LAN1-Pool    1d
2 dhcp-LAN3 ether4     LAN3-Pool    1d
3 I dhcp-LAN4 ether5     LAN4-Pool    1d
4 dhcp-LAN2 lan2-bridge LAN2-Pool    1d
[dedly@beryllium] >
```

Рис. 2.18. Закріплення IP-адреси до MAC-адреси пристроїв

У конфігурації мережі списки інтерфейсів забезпечують зручний спосіб організації та керування мережевими інтерфейсами на основі конкретних критеріїв або цілей. Необхідно додати згрупувати інтерфейси ether2, lan2-bridge, ether4, ether5, wifi1 у список LAN, вони до цього фактично в ньому знаходились, але під контролем стандартного мосту.

Групуючи інтерфейси в список, ми можемо застосовувати конфігурації або правила брандмауера до всіх інтерфейсів у списку одночасно, оптимізуючи завдання керування мережею (рис. 2.19).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

```
[dedly@beryllium] > /interface list member add list=LAN interface=ether2
[dedly@beryllium] > /interface list member add list=LAN interface=lan2-bridge
[dedly@beryllium] > /interface list member add list=LAN interface=ether4
[dedly@beryllium] > /interface list member add list=LAN interface=ether5
[dedly@beryllium] > /interface list member print
Columns: LIST, INTERFACE
# LIST INTERFACE
;;; defconf
0 LAN bridge
;;; defconf
1 WAN ether1
2 LAN ether2
3 LAN lan2-bridge
4 LAN ether4
5 LAN ether5
[dedly@beryllium] > /interface list member add list=LAN interface=wifi1
[dedly@beryllium] > /interface list member print
Columns: LIST, INTERFACE
# LIST INTERFACE
;;; defconf
0 LAN bridge
;;; defconf
1 WAN ether1
2 LAN ether2
3 LAN lan2-bridge
4 LAN ether4
5 LAN ether5
6 LAN wifi1
[dedly@beryllium] > |
```

Рис. 2.19. Групування інтерфейсів у список для фаєрфолу

Основну частину налаштувань підмереж завершено, тепер необхідно доналаштувати інше, переходимо з інтерфейсу wi-fi2 до підмережі з lan2-bridge інтерфейсом, до якої можна підключитись ether3 (wifi1 ще буде налаштовуватися).

```
[dedly@beryllium] > /interface bridge port print
Flags: I - INACTIVE
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COST, HORIZON
# INTERFACE BRIDGE HW PVID PRIORITY PATH-COST INTERNAL-PATH-COST HORIZON
;;; defconf
0 wifi2 bridge 1 0x80 10 10 none
1 ether3 lan2-bridge yes 1 0x80 10 10 none
2 I wifi1 lan2-bridge 1 0x80 10 10 none
[dedly@beryllium] > /interface bridge port remove 0
[dedly@beryllium] > /interface bridge remove [find name=bridge]
[dedly@beryllium] > /interface bridge port print
Flags: I - INACTIVE
Columns: INTERFACE, BRIDGE, HW, PVID, PRIORITY, PATH-COST, INTERNAL-PATH-COST, HORIZON
# INTERFACE BRIDGE HW PVID PRIORITY PATH-COST INTERNAL-PATH-COST HORIZON
0 ether3 lan2-bridge yes 1 0x80 10 10 none
1 I wifi1 lan2-bridge 1 0x80 10 10 none
[dedly@beryllium] > /interface bridge print
Flags: X - disabled, R - running
0 R ;; LAN-2 - Home Network
name="lan2-bridge" mtu=auto actual-mtu=1500 l2mtu=1568 arp=enabled arp-timeout=auto
mac-address=48:A9:8A:DF:D2:72 protocol-mode=rstp fast-forward=yes igmp-snooping=no auto-mac=yes
ageing-time=5m priority=0x8000 max-message-age=20s forward-delay=15s transmit-hold-count=6
vlan-filtering=no dhcp-snooping=no
```

Рис. 2.20. Видалення стандартного інтерфейса мосту

Розширюючи параметри мережі, ми зараз налаштовуємо окрему гостьову мережу, відмінну від основної локальної мережі, щоб забезпечити доступ до

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

Інтернету для гостей, зберігаючи безпеку та ізоляцію від основної мережі. Налаштування інтерфейсу поки що аналогічне до попередніх кроків, а саме алокація IP-адреси, створення пулу IP-адрес, dhcp мережі та серверу, додання інтерфейсу до списку LAN для фаєрволу (рис. 2.21).

```
[dedly@beryllium] > /ip address add address=10.50.10.1/24 interface=wifi2 comment="LAN-5 - Guest Network"
[dedly@beryllium] > /ip pool add name=LAN5-Pool ranges=10.50.10.100-10.50.10.254
[dedly@beryllium] > /ip dhcp-server network add address=10.50.10.0/24 gateway=10.50.10.1 dns-server=10.50.10.1
comment="LAN-5 - Guest Network"
[dedly@beryllium] > /ip dhcp-server add name=dhcp-LAN5 lease-time=1d authoritative=yes interface=wifi2 address-
pool=LAN5-Pool
[dedly@beryllium] > /interface list member add list=LAN interface=wifi2
```

Рис. 2.21. Налаштування wifi2 інтерфейсу

Оновлення налаштувань бездротового зв'язку як для основної домашньої мережі (wifi1), так і для гостьової мережі (wifi2), щоб забезпечити належне підключення до мережі та заходи безпеки. Ці команди змінюють SSID (ідентифікатор набору послуг), код країни та пароль для обох бездротових мереж. Крім того, вказівка правильного коду країни допомагає оптимізувати роботу бездротового зв'язку в рамках правових норм (рис. 2.22).

```
[dedly@beryllium] > /interface wifiwave2 set [find name=wifi1] configuration.ssid=dedly configuration.country=U
kraine security.passphrase=[REDACTED]
[dedly@beryllium] > /interface wifiwave2 set [find name=wifi2] configuration.ssid=dedly-guest configuration.cou
ntry=Ukraine security.passphrase=[REDACTED]
[dedly@beryllium] >
```

Рис. 2.22. Налаштування wifi інтерфейсів

Наступні налаштування будуть стосуватися фаєрволу і безпеки. До цього моменту ніяких налаштувань не було. Налаштування фаєрволу є дуже важливим етапом, бо він і є основною складовою безпеки мережі. Варто оглянути стандартні налаштування фаєрволу та вирішити, як його змінити для підвищення безпеки. Фактично, фаєрвол це список правил, кожен пакет проходить з першого правила до наступного і так далі, поки не потрапить на правило, яке є прийме, перенаправить чи відкине, або не дійде до кінця правил, що також відкине пакет (рис. 2.23)..

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

```
[dedly@beryllium] > /ip firewall filter print
Flags: X - disabled, I - invalid; D - dynamic
0 D ;;; special dummy rule to show fasttrack counters
  chain=forward action=passthrough

1   ;;; defconf: accept established,related,untracked
  chain=input action=accept connection-state=established,related,untracked

2   ;;; defconf: drop invalid
  chain=input action=drop connection-state=invalid

3   ;;; DMZ devices cannot initiate connections to other LANs
  chain=forward action=drop connection-state=new src-address=10.10.10.0/24 dst-address=10.0.0.0/8

4   ;;; Guest devices cannot initiate connections to other LANs
  chain=forward action=drop connection-state=new src-address=10.50.10.0/24 dst-address=10.0.0.0/8

5   ;;; Allow ICMP only from LANs
  chain=input action=accept protocol=icmp in-interface-list=LAN

6   ;;; defconf: accept to local loopback (for CAPsMAN)
  chain=input action=accept dst-address=127.0.0.1

7   ;;; defconf: drop all not coming from LAN
  chain=input action=drop in-interface-list=!LAN

8   ;;; defconf: accept in ipsec policy
  chain=forward action=accept ipsec-policy=in,ipsec

9   ;;; defconf: accept out ipsec policy
  chain=forward action=accept ipsec-policy=out,ipsec

10  ;;; defconf: fasttrack
  chain=forward action=fasttrack-connection hw-offload=yes connection-state=established,related

11  ;;; defconf: accept established,related, untracked
  chain=forward action=accept connection-state=established,related,untracked

12  ;;; defconf: drop invalid
  chain=forward action=drop connection-state=invalid

13  ;;; defconf: drop all from WAN not DSTNATed
  chain=forward action=drop connection-state=new connection-nat-state=!dstnat in-interface-list=WAN
[dedly@beryllium] >
```

Рис. 2.23. Перегляд правил фаєрвола

Детальний огляд правил фаєрвола наведено у Додатку Б.

Наведені правила брандмауєра служать для регулювання та забезпечення безпеки мережевого трафіку, гарантуючи допуск лише авторизованих з'єднань та запобігаючи потенційно шкідливому або несанкціонованому доступу. Правила, такі як приймання встановлених, пов'язаних та невідстежених з'єднань, сприяють плавному потоку законного трафіку, тоді як відкидання неправильних з'єднань зменшує ризик зловмисної діяльності. Додатково, дозвіл трафіку ICMP забезпечує необхідні функції мережевої діагностики. Однак, для подальшого підвищення безпеки та контролю, вводяться конкретні обмеження. Додавання правил зміцнює сегментацію мережі та ізолює потенційно вразливі пристрої від критичних внутрішніх ресурсів (рис. 2.24).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

```
[dedly@beryllium] > /ip firewall filter set 3 in-interface-list=LAN comment="Allow ICMP only from LANs"
[dedly@beryllium] > /ip firewall filter add chain=forward src-address=10.10.10.0/24 dst-address=10.0.0.0/8 conn
action-state=new action=drop place-before=3 comment="DMZ devices cannot initiate connections to other LANs"
[dedly@beryllium] > /ip firewall filter add chain=forward src-address=10.50.10.0/24 dst-address=10.0.0.0/8 conn
action-state=new action=drop place-before=3 comment="Guest devices cannot initiate connections to other LANs"
```

Рис. 2.24. Додання правил фаєрвола

- Правило 1: Обмежує трафік ICMP лише до інтерфейсів LAN, підвищуючи контроль над мережевою діагностикою та обмежуючи зовнішні атаки на основі ICMP.
- Правило 2: Блокує спроби нових з'єднань від пристроїв DMZ до інших ЛВС, ізолюючи потенційні вразливості та зменшуючи ризик несанкціонованого доступу.
- Правило 3: Забороняє спроби нових з'єднань від пристроїв Гостей до інших ЛВС, підвищуючи безпеку шляхом обмеження доступу та потенційних загроз, що походять з мережі Гостей.

2.4.3 Огляд та перевірка налаштувань мережі

На цьому етапі всі налаштування мережі зроблені, і вона повинна повністю працювати, але перед завершенням необхідно перевірити працездатність мережі, так як система доволі складна і може бути непередбачувана поведінка у зв'язку з проблемами конфігурації чи підключення на фізичному рівні, також в доволі рідкісних випадках можуть бути проблеми у ПЗ та фізичні вади пристроїв.

Перше, що можна зробити – перевірити мережеву конфігурацію на кінцевих точках, а саме чи видано IP адресу завдяки DHCP відповідно до закріплення по MAC-адресі, чи правильно налаштований шлюз за замовчуванням, маска підмережі, сама IP мережі. Було перевірено на двох пристроях – робоча станція Lithium у домашній підмережі на операційній системі Windows (рис. 2.25), та на сервері Helium у домашній сервісній підмережі на операційній системі Linux (рис. 2.26). Ці налаштування на інших пристроях в інших підмережах буде перевірено опосередковано у наступних кроках при перевірці зв'язку та інших компонентів мережі.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

```

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . . : 
Description . . . . . : Realtek Gaming 2.5GbE Family Controller
Physical Address. . . . . : 74-56-3C-73-67-D9
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::e8b4:2d67:b960:20e8%5(Preferred)
IPv4 Address. . . . . : 10.20.10.2(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 29 квітня 2024 р. 10:42:21
Lease Expires . . . . . : 30 квітня 2024 р. 10:42:22
Default Gateway . . . . . : 10.20.10.1
DHCP Server . . . . . : 10.20.10.1
DHCPv6 IAID . . . . . : 108287548
DHCPv6 Client DUID. . . . . : 00-01-00-01-2C-95-58-4F-74-56-3C-73-67-D9
DNS Servers . . . . . : 10.20.10.1
NetBIOS over Tcpip. . . . . : Enabled

```

Рис. 2.25. Перевірка мережевих налаштувань на Lithium

```

dedly@helium:~$ sudo dhclient -v eth0
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/d8:3a:dd:1a:bf:67
Sending on   LPF/eth0/d8:3a:dd:1a:bf:67
Sending on   Socket/fallback
DHCPREQUEST for 10.30.10.2 on eth0 to 255.255.255.255 port 67
DHCPACK of 10.30.10.2 from 10.30.10.1
Error: ipv4: Address already assigned.
bound to 10.30.10.2 -- renewal in 42187 seconds.
dedly@helium:~$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether d8:3a:dd:1a:bf:67 brd ff:ff:ff:ff:ff:ff
    inet 10.30.10.2/24 brd 10.30.10.255 scope global dynamic noprefixroute eth0
        valid_lft 74002sec preferred_lft 74002sec
    inet6 fe80::b8d4:4ace:clc2:17e1/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
dedly@helium:~$ ip route show
default via 10.30.10.1 dev eth0
default via 10.30.10.1 dev eth0 proto dhcp src 10.30.10.2 metric 100
10.30.10.0/24 dev eth0 proto kernel scope link src 10.30.10.2 metric 100
10.30.250.0/24 dev wg0 proto kernel scope link src 10.30.250.1
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
172.18.0.0/16 dev br-79542640401b proto kernel scope link src 172.18.0.1
dedly@helium:~$ dig google.com

; <<>> DiG 9.18.24-1-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 38977
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                 274     IN      A      172.217.16.46

;; Query time: 39 msec
;; SERVER: 10.30.10.1#53(10.30.10.1) (UDP)
;; WHEN: Tue May 07 11:25:49 EEST 2024

```

Рис. 2.26. Перевірка мережевих налаштувань на Helium

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

RouterOS v7.8 (stable)									
DHCP Leases									
Add New									
7 items									
	Address	MAC Address	Client ID	Server	Active Address	Active MAC Address	Active Host Name	Bridge Port	Expires After
boron									
- D	10.10.10.3	54:04:A6:C5:78:7		dhcp-LAN1			boron		
hydrogen									
- D	10.10.10.2	E4:5F:01:14:62:BD		dhcp-LAN1	10.10.10.2	E4:5F:01:14:62:BD	hydrogen		19:51:03
- D	10.20.10.101	C8:9F:0C:63:40:11	1:c8:9f:c:63:40:11	dhcp-LAN2	10.20.10.101	C8:9F:0C:63:40:16			19:42:02
lithium									
- D	10.20.10.2	74:56:3C:73:67:D9		dhcp-LAN2	10.20.10.2	74:56:3C:73:67:D9	lithium		07:43:16
- D	10.20.10.100	8C:85:90:7C:CD:A3	1:8c:85:90:7c:cd::	dhcp-LAN2	10.20.10.100	8C:85:90:7C:CD:A3	nitrogen		20:51:51
- D	10.20.10.103	74:A6:CD:E1:76:20	1:74:a6:cd:e1:76:20	dhcp-LAN2	10.20.10.103	74:A6:CD:E1:76:20	dedlys-Laptop	wifi1	18:58:55
helium									
- D	10.30.10.2	D8:3A:DD:1A:BF:67		dhcp-LAN3	10.30.10.2	D8:3A:DD:1A:BF:67	helium		23:17:37

Рис. 2.27. Перевірка виданих IP-адрес на маршрутизаторі beryllium

Для початку необхідно зрозуміти, чи є доступ до всіх пристроїв мережі, для цього виконано просту команду ping, в результаті отримано те, що всі кінцеві пристрої та маршрутизатор відповідають на перевірку протоколом ICMP (рис. 2.28). Бажано зробити аналогічну перевірку з інших хостів, але, поки що, нічого не свідчить про проблеми, тому будемо вважати, що все нормально, все одно зв’язок опосередковано буде перевірено в процесі.

```

dedly@lithium: ~
Windows PowerShell
PS C:\Users\dedly> ping -n 2 10.10.10.2 # hydrogen

Pinging 10.10.10.2 with 32 bytes of data:
Reply from 10.10.10.2: bytes=32 time<1ms TTL=63
Reply from 10.10.10.2: bytes=32 time<1ms TTL=63

Ping statistics for 10.10.10.2:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\dedly> ping -n 2 10.10.10.3 # boron

Pinging 10.10.10.3 with 32 bytes of data:
Reply from 10.10.10.3: bytes=32 time<1ms TTL=63
Reply from 10.10.10.3: bytes=32 time<1ms TTL=63

Ping statistics for 10.10.10.3:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\dedly> ping -n 2 10.20.10.2 # lithium (itself)

Pinging 10.20.10.2 with 32 bytes of data:
Reply from 10.20.10.2: bytes=32 time<1ms TTL=128
Reply from 10.20.10.2: bytes=32 time<1ms TTL=128

Ping statistics for 10.20.10.2:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\dedly> ping -n 2 10.20.10.1 # beryllium

Pinging 10.20.10.1 with 32 bytes of data:
Reply from 10.20.10.1: bytes=32 time<1ms TTL=64
Reply from 10.20.10.1: bytes=32 time<1ms TTL=64

Ping statistics for 10.20.10.1:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\dedly> ping -n 2 10.30.10.2 # helium

Pinging 10.30.10.2 with 32 bytes of data:
Reply from 10.30.10.2: bytes=32 time<1ms TTL=63
Reply from 10.30.10.2: bytes=32 time<1ms TTL=63

Ping statistics for 10.30.10.2:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Users\dedly>

```

Рис. 2.28 Перевірка зв'язку за протоколом ICMP

Наступним кроком буде перевірка зв'язку з Інтернетом та якості з'єднання. Першим чином, перевірено підключення до інтернету, через браузер на Lithium. У відповідь на Google-запит speedtest.net успішно завантажився сайт, що значить що DNS та доступ в інтернет працює. Робимо перевірку швидкості дротового з'єднання та отримуємо 93.79 Mbps на завантаження 94.61 Mbps та вивантаження. Нагадую, що за поточним тарифом провайдера наявна максимальна швидкість 100 Mbps (рис. 2.29), що значить, що, враховуючи можливі втрати швидкості у провайдера, швидкість чудова і мережа справляється при 100 Mbps.

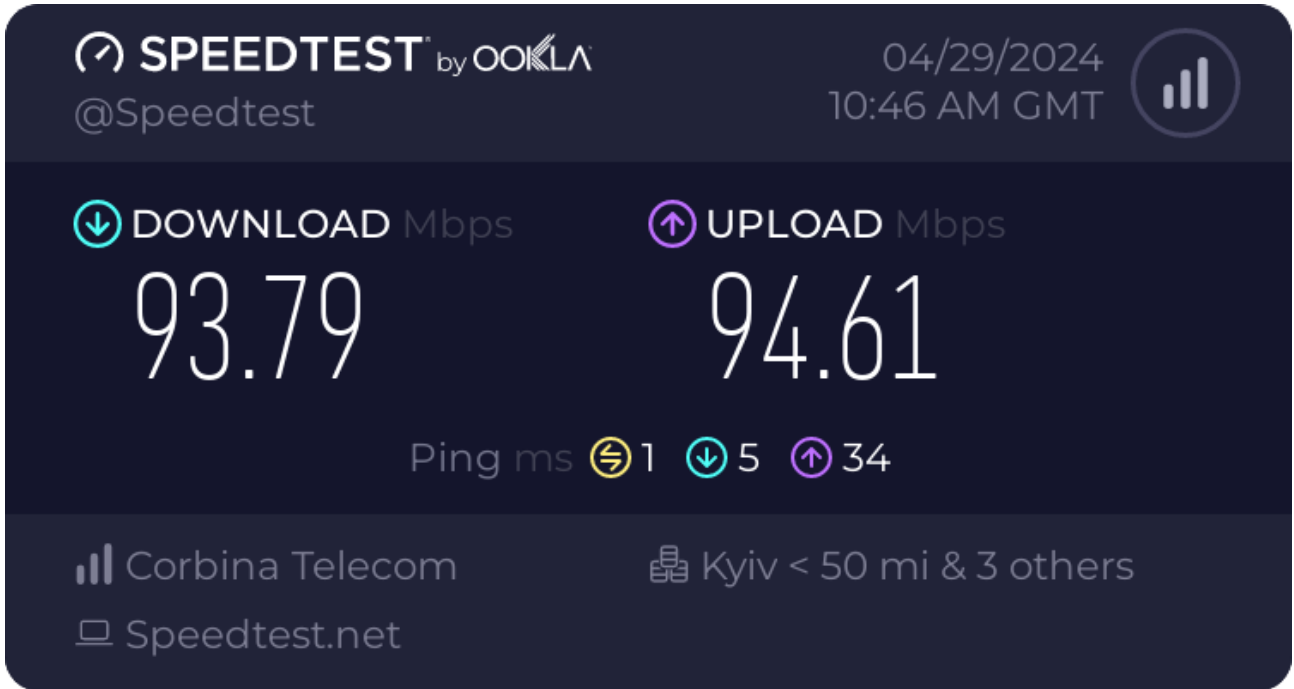


Рис. 2.29. Перевірка швидкості 100 Mbps з'єднання Інтернет

Для того, щоб дізнатись швидкість мережі і не бути обмеженим тарифом провайдера, можна зробити швидкісних тест між кінцевими точками homelab. Для цього було обрано hydrogen (10.10.10.2) та helium (10.30.10.2). Для цього використано утіліту iperf3, на hydrogen, що виступить у ролі сервера прописано команду iperf3 -s, це переводить його у режим сервера і він очікує на клієнта. З боку helium введено iperf3 -c 10.10.10.2. Після 10 секунд, ми отримуємо результат орієнтовано 942 Mbps для відправника та отримувача, що показує, що мережа дійсно має пропускну здатність 1 Gbps і справляється чудово (рис. 2.30).

```
dedly@helium:~$ iperf3 -c 10.10.10.2
Connecting to host 10.10.10.2, port 5201
[ 5] local 10.30.10.2 port 49660 connected to 10.10.10.2 port 5201
[ ID] Interval      Transfer    Bitrate      Retr  Cwnd
[ 5]  0.00-1.00    sec    113 MBytes  948 Mbits/sec    0   400 KBytes
[ 5]  1.00-2.00    sec    112 MBytes  941 Mbits/sec    0   400 KBytes
[ 5]  2.00-3.00    sec    113 MBytes  946 Mbits/sec    0   420 KBytes
[ 5]  3.00-4.00    sec    112 MBytes  942 Mbits/sec    0   420 KBytes
[ 5]  4.00-5.00    sec    112 MBytes  940 Mbits/sec    0   420 KBytes
[ 5]  5.00-6.00    sec    112 MBytes  943 Mbits/sec    0   420 KBytes
[ 5]  6.00-7.00    sec    112 MBytes  936 Mbits/sec    0   420 KBytes
[ 5]  7.00-8.00    sec    112 MBytes  941 Mbits/sec    0   420 KBytes
[ 5]  8.00-9.00    sec    112 MBytes  942 Mbits/sec    0   420 KBytes
[ 5]  9.00-10.00   sec    113 MBytes  947 Mbits/sec    0   420 KBytes
-----
[ ID] Interval      Transfer    Bitrate      Retr
[ 5]  0.00-10.00   sec    1.10 GBytes  943 Mbits/sec    0
[ 5]  0.00-10.00   sec    1.10 GBytes  941 Mbits/sec
sender
receiver
iperf Done.
```

Рис. 2.30. Перевірка швидкості локального з'єднання

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

Було протестовано Wi-Fi. Я не змінював стандартні налаштування, окрім назв SSID та passphrase, тому бажано проаналізувати, як інтерфейси поведуться в цих умовах.



Рис. 2.31. Перевірка сигналу Wi-Fi інтерфейсів

Wi-Fi сканер дозволяє зрозуміти і вивчити сильні та слабкі сигнали, їхню частоту, потужність та інші параметри. Це дає можливість отримати інформацію про доступні мережі, їхні зони покриття. Більше того, аналіз сигналів вай-фай дозволяє виявити можливі джерела перешкод або інтерференції, що може бути корисно при оптимізації роботи мережі та підвищенні її ефективності.

З рисунку 2.31 видно, що сканер не бачить багато мереж, що свідчить про сильну ізоляцію стінами, що може бути й добре, бо не дає чужим мережам перешкоджати сигналу, і погано, бо може погіршуватися якість в інших кімнатах. Для обох інтерфейсів сигнал чудовий, про це свідчить сила сигналу -40db для 5 GHz і -50db для 2 GHz. Також видно, що сигнали з іншими мережами майже не

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		63

перетинаються і це не заважає зв'язку. Маршрутизатор MikroTik автоматично сканує канали та обирає найменш забиті. Варто зазначити, що обидва Wi-Fi використовують WPA3-personal – що надає найкращу безпеку на сьогодні для домашніх мереж.

Далі, перевірено саме підключення до Wi-Fi мереж. Першим буде інтерфейс Wi-Fi 5 GHz, що має SSID dedly та по налаштуванню повинен бути підключений до мосту домашньої підмережі, це і буде перевірено.

The screenshot shows a terminal window with network configuration details on the left and ping test results on the right.

Network details:

- Type: Wi-Fi 5
- Device MAC address: c8:9f:0c:63:40:16
- IP address: 10.20.10.101
- Gateway: 10.20.10.1
- Subnet mask: 255.255.255.0
- DNS: 10.20.10.1
- Transmit link speed: 390 Mbps
- Receive link speed: 351 Mbps

Ping tests:

```
$ ping -c 2 10.20.10.2
PING 10.20.10.2 (10.20.10.2) 56(84) bytes of data.
64 bytes from 10.20.10.2: icmp_seq=1 ttl=128 time=4.36 ms
64 bytes from 10.20.10.2: icmp_seq=2 ttl=128 time=6.29 ms

--- 10.20.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 4.362/5.327/6.293/0.968 ms

$ ping -c 2 10.30.10.2
PING 10.30.10.2 (10.30.10.2) 56(84) bytes of data.
64 bytes from 10.30.10.2: icmp_seq=1 ttl=63 time=6.38 ms
64 bytes from 10.30.10.2: icmp_seq=2 ttl=63 time=5.61 ms

--- 10.30.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 5.616/6.001/6.386/0.385 ms

$ ping -c 2 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data.
64 bytes from 10.10.10.2: icmp_seq=1 ttl=63 time=4.99 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=63 time=6.98 ms

--- 10.10.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 4.990/5.986/6.982/0.996 ms
```

Рис. 2.32. Перевірка Wi-Fi 5 GHz

З рис. 2.32 видно, що налаштування IP-адресації було успішно отримано, всі параметри правильні. Також видно, що швидкість з'єднання 350-390 Mbps, що є доволі високою для домашніх мереж, що зможе витримати великі навантаження і, в цілому, не буде ніяких проблем зі швидкістю при роботі з будь-яким трафіком. Також перевірено підключення до інших підмереж – доступ є, як і заплановано. Але щоб переконатися остаточно, що інтерфейс належить до мосту домашньої мережі було виконано пінг за цієї IP-адреси смартфона з підключеного до eth3 через комутатор хост Lithium та перевірено ARP таблицю на наявність запису. У результаті успішно перевірено і можна стверджувати, що інтерфейси eth3 та

wifi1 дійсно належать віртуальному інтерфейсу lan2-bridge та поводяться коректно згідно налаштувань (рис. 2.33).

```
PS C:\Users\dedly> ping -n 1 10.20.10.101

Pinging 10.20.10.101 with 32 bytes of data:
Reply from 10.20.10.101: bytes=32 time=50ms TTL=64

Ping statistics for 10.20.10.101:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 50ms, Maximum = 50ms, Average = 50ms
PS C:\Users\dedly> arp -a

Interface: 10.20.10.2 --- 0x5
Internet Address      Physical Address      Type
10.20.10.1             48-a9-8a-df-d2-72    dynamic
10.20.10.101           c8-9f-0c-63-40-16    dynamic
10.20.10.255           ff-ff-ff-ff-ff-ff    static
224.0.0.22             01-00-5e-00-00-16    static
224.0.0.251            01-00-5e-00-00-fb    static
224.0.0.252            01-00-5e-00-00-fc    static
239.255.102.18         01-00-5e-7f-66-12    static
239.255.255.250        01-00-5e-7f-ff-fa    static
255.255.255.255        ff-ff-ff-ff-ff-ff    static
```

Рис. 2.33. Перевірка зв'язку та ARP таблиці

Наступне – це перевірка гостьового Wi-Fi. Окрім тих самих перевірок, що було зроблено для домашнього Wi-Fi, тут також можна перевірити налаштування фаєрволу – пристрої не повинні мати можливість ініціалізувати зв'язок в межах локальних мереж homelab. На рис. 3.34 видно налаштування мережі та перевірку зв'язку, смартфон може без проблем мати доступ до мережі Інтернет, але не може ініціалізувати зв'язок з пристроями локальних підмереж.

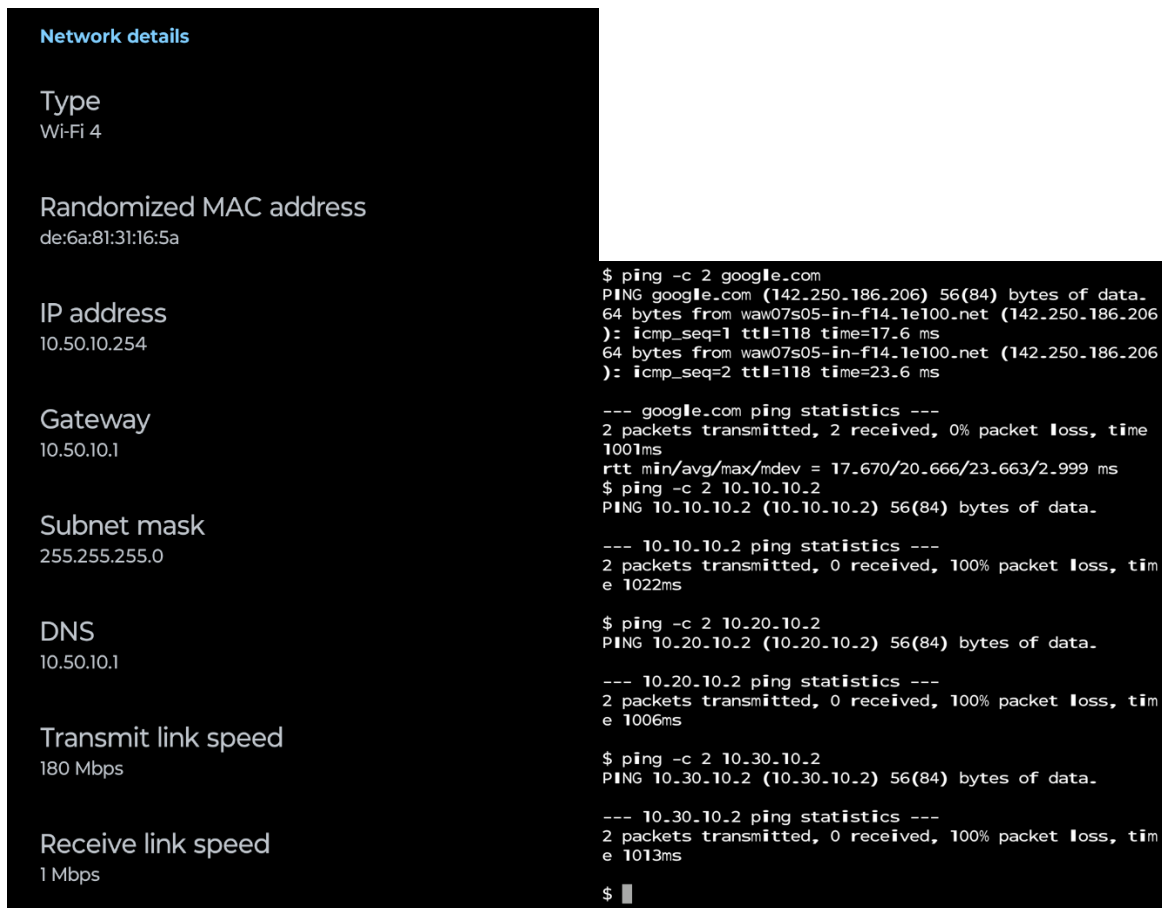


Рис. 2.34. Перевірка Wi-Fi 2.4 GHz

Наостанок було перевірено фаєрвол для першої підмережі, що використовуватиметься для веб-сервісів. Helium, що знаходиться у домашній сервісній мережі повинен успішно зв'язуватися з hydrogen, але не навпаки, на рис. 2.35 та 2.36 було успішно перевірено це.

```
dedly@helium:~$ ip address show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether d8:3a:dd:1a:bf:67 brd ff:ff:ff:ff:ff:ff
    inet 10.30.10.2/24 brd 10.30.10.255 scope global dynamic noprefixroute eth0
        valid_lft 53349sec preferred_lft 53349sec
    inet6 fe80::b8d4:4ace:c1c2:17e1/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
dedly@helium:~$ ping -c 2 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data:
64 bytes from 10.10.10.2: icmp_seq=1 ttl=63 time=0.416 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=63 time=0.278 ms

--- 10.10.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 0.278/0.347/0.416/0.069 ms
dedly@helium:~$
```

Рис. 2.35. Успішна перевірка зв'язку з helium до hydrogen

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

```

dedly@hydrogen:~ $ ip address show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether e4:5f:01:14:62:bd brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.2/24 brd 10.10.10.255 scope global dynamic noprefixroute eth0
        valid_lft 60740sec preferred_lft 60740sec
    inet6 fe80::dd5d:ae74:a476:55bd/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
dedly@hydrogen:~ $ ping -c 2 10.30.10.2
PING 10.30.10.2 (10.30.10.2) 56(84) bytes of data.

--- 10.30.10.2 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1013ms

dedly@hydrogen:~ $ █

```

Рис. 2.36. Відсутність зв'язку з hydrogen до helium

В и с н о в к и д 20 р о з д і л у

У цьому розділі всебічно розглянуто основні аспекти проектування та побудови надійної мережевої основи, необхідної для ефективної роботи домашньої лабораторії. Глибоко кастомізована мережа - це не просто допоміжна структура, а ядро, яке буде визначати ефективність і безпеку всього середовища homelab.

Завдяки ретельному плануванню як фізичної, так і логічної структури мережі, ми застосували системний підхід, щоб забезпечити масштабованість, безпеку та оптимальну конфігурацію мережі відповідно до вимог різних розміщених на ній сервісів. Вибір відповідного мережевого обладнання, а також покрокові інструкції зі встановлення та налаштування цих компонентів були розроблені з метою сприяння глибокому розумінню та практичним навичкам.

Крім того, розглянуті процедури конфігурації, особливо з акцентом на терміналі RouterOS, були розроблені для того, щоб покращити здатність слухачів усувати несправності та обслуговувати свою мережу, забезпечуючи довготривалу стабільність та продуктивність. Заключні огляди і перевірка мережевих налаштувань підкреслюють важливість старанності і постійного моніторингу для підтримки мережі, яка не тільки функціональна, але і стійка до потенційних загроз безпеці.

По суті, цей розділ не лише надав базові знання та практичні кроки для побудови мережі homelab, але й підкреслив критичну важливість особистої участі та розуміння кожного кроку конфігурації мережі для досягнення оптимального налаштування, яке ефективно підтримує як внутрішні, так і зовнішні сервіси.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

РОЗДІЛ 3 ПІДГОТОВКА ТА КОНФІГУРАЦІЯ СЕРВІСІВ

3.1 Конфігурація ервїїні

3.1.1 Cloudflare - реєстрація домену та створення записів

При створенні homelab, що містить низку внутрішніх і зовнішніх сервісів, важливим першим кроком є встановлення доменного імені. Цей процес є фундаментальним, оскільки він встановлює основні засоби доступу та управління послугами в межах homelab. Доменне ім'я слугує ключовою точкою входу, що дозволяє конфігурувати мережу та розширювати її в майбутньому. Тому його придбання - це не просто купівля веб-адреси, а стратегічний вибір домену, який підтримуватиме заплановані функціональні можливості та заходи безпеки homelab. Придбання відповідного домену передбачає планування та врахування як безпосередніх, так і довгострокових вимог, щоб забезпечити його відповідність операційним цілям homelab. Домен стає сполучною ланкою мережі, впливаючи на те, як здійснюється доступ до послуг, керування ними та їх захист, тому його ретельний вибір є критично важливим для фундаментального налаштування домашньої лабораторії.

Доменне ім'я, за своєю структурою та технічними особливостями, функціонує як ієрархічне представлення адресного простору Інтернету[8]. На найвищому рівні цієї ієрархії знаходяться домени верхнього рівня (TLD)[9], такі як .com, .org, .net, а також національні закінчення, такі як .uk або .de. Безпосередньо під доменом верхнього рівня в ієрархії знаходиться домен другого рівня, який часто є ім'ям, що найбільше асоціюється з веб-сайтом у громадськості - наприклад, «google» у «google.com». За бажанням, домени третього рівня, або субдомени, такі як «maps.google.com», можуть використовуватися для переходу на певні служби або відділи організації.

Структурування доменних імен має значення, оскільки воно безпосередньо впливає на маршрутизацію DNS, по суті, спрямовуючи інтернет-трафік до потрібних місць. Кожна частина доменного імені відокремлюється крапками (.) і читається справа наліво з точки зору ієрархії; система аналізує цю ієрархію, щоб відповідно спрямовувати запити. Ефективне управління цією системою передбачає

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

налаштування записів DNS, які необхідні для правильного перетворення доменного імені у відповідну IP-адресу. DNS слугує телефонною книгою Інтернету, переводячи доменні імена, що читаються людиною, у машинозчитувані IP-адреси, що дозволяє браузерам завантажувати інтернет-ресурси. Таким чином, доменне ім'я - це не просто мітка, а важливий компонент мережевої навігації та управління, що забезпечує як технічний, так і зручний для користувача спосіб доступу до послуг homelab.

Доменні імена відіграють подвійну роль - як утилітарну, так і брендингову, з особливо сильним акцентом на їх функціональних перевагах у середовищі homelab. По суті, доменне ім'я забезпечує запам'ятовуваний і доступний спосіб доступу до різноманітних мережевих ресурсів, що важливо для ефективного управління як внутрішніми, так і зовнішніми сервісами. Доменне ім'я діє як стабільний ідентифікатор, який залишається незмінним, навіть якщо основні IP-адреси змінюються через реконфігурацію або оновлення мережі, тим самим зберігаючи цілісність зв'язків і шляхів доступу з плином часу.

Утилітарно, у випадку homelab, доменні імена полегшують організацію та сегментацію послуг. Наприклад, різні субдомени можуть бути призначені різним серверам або службам, що полегшує навігацію та управління. Це особливо корисно в середовищах зі складними мережами, в яких розміщуються кілька додатків або сервісів. Крім того, доменні імена підвищують рівень безпеки мережі, дозволяючи використовувати сертифікати для HTTPS, що гарантує зашифровані з'єднання та захист даних під час передачі.

Крім того, наявність виділеного домену для домашньої лабораторії підвищує професіоналізм будь-яких послуг, що надаються на хостингу, роблячи їх більш достовірними та надійними для кінцевих користувачів. Це дозволяє краще контролювати брендинг, узгоджуючи домен з ідентичністю та репутацією особи чи організації. Таким чином, доменне ім'я - це не лише технічна необхідність, а й стратегічний актив, який підвищує функціональність, безпеку та професійний імідж.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		69

Процес придбання та налаштування домену складається з кількох структурованих етапів, кожен з яких має вирішальне значення для забезпечення ефективного функціонування домену у середовищі homelab. Спочатку необхідно вибрати реєстратора домену, тобто організацію, уповноважену реєструвати доменні імена. Реєстратори можуть сильно відрізнятися за послугами, які вони пропонують, ціною та додатковими функціями, такими як опції конфіденційності або пакетні послуги хостингу. Після того, як обрано відповідного реєстратора і визначено, що доменне ім'я доступне, наступним кроком є процес реєстрації, під час якого ви офіційно заявляєте домен на своє ім'я або організацію на певний період часу.

Вибір правильного сервісу для реєстрації та управління доменом має вирішальне значення для ефективної роботи домашньої лабораторії. На ринку є кілька авторитетних реєстраторів, кожен з яких пропонує різні функції та переваги. Наприклад, GoDaddy і Namecheap відомі своєю широкою підтримкою клієнтів і зручними інтерфейсами, що робить їх популярним вибором як для початківців, так і для професіоналів. Ці платформи часто надають додаткові послуги, такі як веб-хостинг, SSL-сертифікати та хостинг електронної пошти, що може бути корисним для користувачів, які прагнуть консолідувати свою присутність в Інтернеті в рамках одного провайдера.

Однак для тих, хто особливо зосереджений на продуктивності та безпеці мережі, Cloudflare є кращим варіантом. На відміну від традиційних реєстраторів, Cloudflare спеціалізується на послугах з підвищення продуктивності та безпеки Інтернету, які легко інтегруються з реєстрацією доменів. Обираючи Cloudflare, користувачі отримують доступ до розширених інструментів управління DNS, надійних функцій безпеки, таких як захист від DDoS-атак і брандмауерів для веб-додатків, а також оптимізацію продуктивності за допомогою послуг CDN. Це робить Cloudflare чудовим вибором для технічних ентузіастів та професіоналів, які надають пріоритет стабільності та безпеці мережевої інфраструктури своєї homelab (рис. 3.2).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

Manage mahurin.xyz

[← Back to Manage Domains](#)

[Registration](#) [Contacts](#) [Configuration](#)

Registration

Automatic renewal is on

Your domain will automatically renew every year.



[Renew/Extend Domain](#) ▶

Expires on

December 1, 2024

Renews on

November 1, 2024

Price ⓘ

\$10.18 / year

You will be charged to the card ending in 5724. [Change](#)

Domain Details

Status

Active

Auto-renew

✓ Auto

Expiry date

December 1, 2024

Quick actions

[Update DNS configuration](#)

[Add Pages site](#)

Рис. 3.1. Зареєстрований домен на cloudflare

Крім того, служба DNS Cloudflare відома своєю швидкістю та надійністю, які є критично важливими для забезпечення швидкого та стабільного доступу до розміщених сервісів. Платформа також пропонує прості у впровадженні заходи безпеки, які гарантують, що і домен, і сервіси, на які він вказує, захищені від поширених загроз. Таке поєднання продуктивності, безпеки та простоти використання робить Cloudflare відмінним вибором для управління доменними аспектами налаштування homelab, незважаючи на наявність інших життєздатних варіантів на ринку.

Після реєстрації домену необхідно налаштувати так, щоб він вказував на IP-адреси вашої мережі homelab. Це робиться шляхом налаштування записів DNS, які спрямовують трафік на ваші сервери на основі доменного імені. Найпоширенішими записами є записи типу A, які зіставляють домен з адресою IPv4, і записи типу AAAA, які зіставляють домен з адресою IPv6. Інші важливі типи включають MX-записи для поштових серверів, CNAME-записи для псевдонімізації одного домену іншим і TXT-записи, які часто використовуються для перевірки права власності на домен і реалізації заходів безпеки електронної пошти.

Для homelab, що надає як внутрішні, так і зовнішні послуги, стратегічні конфігурації DNS має особливі налаштування. Для зовнішнього доступу такі домени, як `mahurin.xyz` і `\*.mahurin.xyz` повинні вказувати на публічну IP-адресу

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		71

мережі homelab за допомогою записів типу A. Така конфігурація використовує маршрутизатор, налаштованого на перенаправлення трафіку на зворотний проксі-сервер, який потім спрямовує запити на відповідні внутрішні служби.

І навпаки, для служб, призначених виключно для внутрішнього використання, такі домени, як `home.mahurin.xyz` і `\*.home.mahurin.xyz` повинні вказувати на приватні IP-адреси за допомогою записів типу A. Ці конфігурації забезпечують маршрутизацію трафіку у внутрішній мережі, оминаючи загальнодоступні інтернет-маршрути, що підвищує безпеку та продуктивність. Крім того, хоча проксі-сервіси Cloudflare використовуються для підвищення безпеки та продуктивності, можуть виникнути проблеми при розміщенні сервісів, які використовують не-HTTP-вміст, наприклад, сервер Minecraft, який вимагає як TCP, так і UDP-потоків на порту 25565. Щоб вирішити цю проблему, можна створити альтернативний DNS-запис `minecraft.mahurin.xyz` без використання проксі-сервера Cloudflare. Цей обхідний шлях відкриває публічну IP-адресу, пов'язану з `\*.mahurin.xyz`, що потенційно може розкрити зовнішній шлюз вашої мережі. Однак цей ризик наразі вважається керованим. Такий структурований підхід до керування записами DNS не лише оптимізує мережевий трафік, але й підвищує загальний рівень безпеки homelab.

DNS management for mahurin.xyz

Review, add, and edit DNS records. Edits will go into effect once saved.

DNS Setup: Full ⓘ

Import and Export ▼

⚙

Search DNS Records

▼ Add filter

Q

Type ▲	Name	Content	Proxy status
A	*.home	10.30.10.2	DNS only - reserved IP
A	home	10.30.10.2	DNS only - reserved IP
A	*	31.43.154.6	 Proxied
A	mahurin.xyz	31.43.154.6	 Proxied
 A	minecraft	31.43.154.6	 DNS only

Рис. 3.2. Створення DNS записів

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				72
Змн.	Арк.	№ докум.	Підпис	Дата		

Останній крок передбачає налаштування серверів імен, тобто серверів, які зберігатимуть і обслуговуватимуть DNS-записи. Можна використовувати сервери імен, надані вашим реєстратором, або вибрати сторонні сервери імен, якщо вам потрібен більший контроль або додаткові функції. У випадку реєстрації домену на Cloudflare – не потрібно нічого робити, вони самі встановлять іменні сервери (рис. 3.3).

Cloudflare Nameservers Every DNS zone on Cloudflare is assigned a set of Cloudflare-branded nameservers.	
Type	Value
NS	magnolia.ns.cloudflare.com
NS	razvan.ns.cloudflare.com

Рис. 3.3. Іменні сервери cloudflare, встановлені автоматично

3.1.2 Nginx Proxy Manager і персональна web-сторінка

Наступним кроком у налаштуванні домашньої лабораторії буде конфігурація та розгортання зовнішніх сервісів.

Конфігурація зовнішніх служб розширює можливості мережі за її внутрішні межі до ширшого Інтернету. На цьому етапі відбувається налаштування та керування службами, доступними через Інтернет, що забезпечує віддалений доступ і ширші можливості підключення. Це передбачає ретельне планування та надійні заходи безпеки, щоб гарантувати, що надані послуги будуть не тільки функціональними, але й захищеними від зовнішніх загроз.

Зовнішні сервіси слугуватимуть платформою для тестування і розгортання веб-додатків у реальних умовах. Це дозволяє отримати практичний досвід роботи з серверами, що виходять в Інтернет, навчитися керувати та зменшувати унікальні виклики та ризики безпеки, пов'язані з доступом до Інтернету.

Крім того, хостинг зовнішніх сервісів може забезпечити шлях до автоматизації та інтеграції з іншими хмарними сервісами. Це може підвищити продуктивність і

ефективність завдяки використанню API та підключенню до глобальних систем, тим самим розширюючи функціональний ландшафт homelab.

Приклади зовнішніх сервісів, які можна розмістити на homelab, різноманітні, вони задовольняють різні потреби та інтереси, що розширюють загальнодоступність і функціональність. Одним із поширених прикладів є хостинг персональних або бізнес-сайтів, які слугують як цифрові портфоліо, блоги або платформи електронної комерції. Ці сайти можна розробляти і керувати ними з homelab, забезпечуючи повний контроль над хостинговим середовищем і потенційно зменшуючи залежність від зовнішніх послуг веб-хостингу.

Ще однією популярною послугою є хостинг ігрових серверів, наприклад, для Minecraft або інших багатокористувацьких ігор, де можна мати повний контроль над власними приватними спільнотами.

Вирішено як proof of concept (перевірка концепції) зробити зовнішніми веб-сервісами персональний веб-сайт, а також сервер Minecraft. Веб-сервер наразі буде лише один, цього більш ніж достатньо, але конфігурація мережі має бути запланована для того, щоб підтримувати більше веб-серверів, для того, щоб мати велику кількість веб-серверів в домашній мережі на одному порті 443, необхідно мати зворотній проксі, щоб мати змогу маршрутизувати трафік до відповідних серверів керуючись доменими іменами (mahurin.xyz, blog.mahurin.xyz. тощо).

Зворотний проксі-сервер[10] - це проміжний сервер, який знаходиться між клієнтом і внутрішніми серверами. Він отримує запити від клієнтів, перенаправляє їх на відповідний сервер, а потім повертає відповідь сервера клієнту. Таке налаштування забезпечує такі переваги, як покращена безпека, балансування навантаження та підвищена продуктивність завдяки кешуванню вмісту.

Для homelabs популярними варіантами проксі-серверів є Nginx, Apache і Caddy. Nginx відомий своєю продуктивністю і гнучкістю, Apache - надійністю і великою системою модулів, а Caddy - автоматичним налаштуванням HTTPS. Кожен з них має свої сильні сторони: Nginx відмінно справляється з високим навантаженням трафіку, Apache пропонує широкі можливості налаштування, а Caddy спрощує безпечний веб-сервіс.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

Nginx Proxy Manager - це інтерфейсний інструмент для веб-сервера Nginx, який спрощує процес хостингу та управління веб-сайтами за допомогою графічного інтерфейсу користувача [11]. Він дозволяє користувачам легко керувати правилами проксі, SSL-сертифікатами та контролем доступу без необхідності ручного редагування конфігураційних файлів. Nginx Proxy Manager особливо підходить для homelab, оскільки поєднує в собі ефективність Nginx зі зручними функціями управління, що робить його ідеальним для тих, хто віддає перевагу графічному підходу до управління сервером.

Веб-сервер[12] - це програмно-апаратний комплекс, який використовує протокол HTTP (Hypertext Transfer Protocol) для обслуговування запитів користувачів на файли, що формують веб-сторінки. Він доставляє такий вміст, як документи HTML, таблиці стилів CSS та медіафайли до браузерів користувачів. Основне призначення Інтернету та сервера, що лежить в його основі, - розміщувати веб-сайти, забезпечуючи доступ до даних і додатків через Інтернет, полегшуючи таким чином комунікацію між користувачами та сервісами.

Вибір веб-сервера є ключовим кроком у створенні домашньої лабораторії, оскільки він передбачає навігацію в широкому спектрі технологій. Вибір веб-сервера залежить від поєднання мов програмування, внутрішніх серверів і фреймворків. Цей вибір може суттєво вплинути на розробку, розгортання та обслуговування розміщених додатків, впливаючи на продуктивність, масштабованість і безпеку.

Розглядаючи варіанти налаштування веб-сервера, можна виділити різні комбінації мов програмування та фреймворків. Для наших цілей обрано Flask через його простоту та ефективність. Flask[13] - це легкий фреймворк веб-додатків WSGI на Python, з яким легко почати роботу, і який є достатньо потужним для складних додатків. Він використовує шаблони Jinja для модуляризації фронтенду, а його можливості маршрутизації ефективно керують бекенд-логікою, яка рендерить ці шаблони, що робить його ідеальним вибором як для початківців, так і для досвідчених розробників.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

Хоча веб-розробка не є центральною темою даної кваліфікаційної роботи, персональна веб-сторінка стратегічно розроблена для досягнення кількох цілей. Перш за все, вона функціонує як динамічне портфоліо, орієнтоване на потенційних роботодавців, демонструючи навички розробника та забезпечуючи присутність особистого бренду. Крім того, веб-сайт може сприяти спілкуванню та обміну ідеями між однодумцями. Важливо, що основна мета цього проекту - продемонструвати підтвердження концепції, показати практичне застосування і розгортання веб-проектів, тим самим закріпивши знання в цих сферах.

Оскільки ми переходимо до етапу розгортання, використання Raspberry Pi для розміщення Nginx Proxy Manager і веб-сервера є ефективним вибором, враховуючи їх низьке навантаження і вимогу до постійної працездатності. Хоча ці програми та їх залежності можна встановити вручну, я надаю перевагу використанню Docker та Docker Compose. Цей підхід не тільки спрощує встановлення та оновлення, але й покращує організацію системи, ізолюючи програми в контейнерах.

Docker[14] - це платформа з відкритим вихідним кодом, яка дозволяє розробникам пакувати додатки в контейнери. Ці контейнери інкапсулюють всі необхідні компоненти, такі як код, час виконання та залежності, забезпечуючи узгоджену роботу програми в різних обчислювальних середовищах. Docker забезпечує стандартизовану одиницю програмного забезпечення, спрощуючи процеси розробки, тестування та розгортання.

Docker Compose - це інструмент для визначення та запуску багатоконтейнерних Docker-додатків. За допомогою Compose ви використовуєте YAML-файл для налаштування служб, мереж і томів вашого додатка, а потім створюєте і запускаєте всі служби з вашої конфігурації за допомогою однієї команди. Це спрощує робочий процес і гарантує, що всі контейнери працюють злагоджено, підтримуючи визначені взаємозв'язки та конфігурації налаштувань.

Розгортання таких програм, як Nginx Proxy Manager і веб-сервер, у вигляді контейнерів забезпечує чисте і ефективне рішення для управління. Завдяки контейнеризації цих програм, хост-система залишається недоторканою, на неї не

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		76

впливають залежності та потенційні конфлікти, які ці програми можуть спричинити в іншому випадку. Docker Compose полегшує керування цими контейнерами, дозволяючи виконувати прості операції з їх запуску та відключення як єдиної системи.

Перейдемо до розгортання. Оскільки веб-розробка виходить за межі теми даної кваліфікаційної роботи, то написання коду та його деталі пропущено. У нас є веб-сервер flask, що запускається завдяки команді `python app.py`, але такий підхід не підходить до production середовища. Для умовного продакшену, нам необхідно `gunicorn` – це Python WSGI HTTP-сервер.

За допомогою Docker було створено образ контейнера, пристосований для розгортання веб-додатку на Python. Цей образ починається з легкої бази Python Alpine, що забезпечує мінімальний розмір файлу. Середовище програми налаштовується в каталозі `/app`, куди копіюються всі необхідні файли. Це включає встановлення залежностей з файлу `requirements.txt` для підтримки чистого та ефективного управління необхідними пакунками. Крім того, для обслуговування програми встановлено Gunicorn, налаштований на прослуховування порту 5000. Додаток запускається з використанням Gunicorn як WSGI-сервера, оптимізованого для одночасної роботи з двома працівниками, прив'язаними до всіх інтерфейсів, що забезпечує надійну продуктивність додатка (рис. 3.4).

```

Dockerfile > ...
1 FROM python:3.9.19-alpine3.20
2 WORKDIR /app
3 COPY . /app
4 RUN pip install --no-cache-dir -r requirements.txt
5 RUN pip install gunicorn
6 EXPOSE 5000
7 ENV FLASK_APP=app.py
8 CMD ["gunicorn", "-w", "2", "-b", "0.0.0.0:5000", "app:app"]

```

Рис. 3.4. Dockerfile веб-серверу

Було успішно перевірено роботу на локальному хості. Репозиторій завантажений на github: <https://github.com/catloaf9000/homelab-webpage>.

Далі необхідно створити docker-compose, в ньому буде описано всі майбутні веб-сервіси, що пов'язані з зворотнім проксі також. Для початку треба підготувати

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

хост - hydrogen, наразі на ньому нічого не встановлено, окрім Raspberry Pi OS (рис. 3.5).

```
Shell
# Remove conflicting packages:
for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do
sudo apt-get remove $pkg; done

# Set up docker apt repository:
## Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install -y ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

## Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-
by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

## Install packages
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-
plugin docker-compose-plugin docker-compose

sudo systemctl enable docker.service
sudo systemctl enable containerd.service
```

Рис. 3.5. Встановлення необхідних сервісів та утиліт на hydrogen

Налаштування передбачають підготовку системи до встановлення Docker шляхом видалення будь-яких конфліктуючих пакунків, таких як старіші версії Docker або подібні інструменти для роботи з контейнерами. Далі налаштовується репозиторій Docker шляхом встановлення необхідних пакунків, таких як са-сертифікати та curl, створення каталогу для GPG-ключа Docker, а також завантаження та налаштування дозволів для ключа. Після цього сховище Docker буде додано до списку джерел системи. Після оновлення списку пакунків буде встановлено Docker та пов'язані з ним пакунки. У кінці, служби Docker увімкнено для автоматичного запуску разом із системою (рис. 3.6).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		78

```
sudo apt-get install -y git vim
mkdir -p ~/projects
cd ~/projects
git clone https://github.com/catloaf9000/homelab-webpage.git
```

```
vim ~/projects/docker-compose.yaml
```

```
---
services:
  homelab-webpage:
    build: ./homelab-webpage
    ports:
      - 8080:5000
    restart: unless-stopped

  nginxproxymanager:
    image: 'docker.io/jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    environment:
      - PUID=$PUID
      - PGID=$PGID
    ports:
      - '80:80'
      - '81:81'
      - '443:443'
    volumes:
      - nginxpm-data:/data
      - nginxpm-certs:/etc/letsencrypt

volumes:
  nginxpm-data:
  nginxpm-certs:
```

Рис. 3.6 Створення docker-compose.yaml на hydrogen

Процедура починається з встановлення необхідних інструментів, Git і Vim. Створюється каталог для проектів, який встановлюється як поточний робочий каталог. У цей каталог клонується репозиторій GitHub моєї веб-сторінки. Після цього створюється і редагується файл 'docker-compose.yaml' для визначення і налаштування сервісів, включаючи 'homelab-webpage' і 'nginxproxymanager'.

Конфігурація визначає контексти збірки, зіставлення портів, політики перезапуску, змінні середовища та зіставлення томів для постійного сховища. Це налаштування призначене для організації розгортання цих служб за допомогою Docker Compose.

```

vim ~/projects/.env

PUID=1000
PGID=1000

cd ~/projects/homelab-webpage
git pull
cd ~/projects
sudo docker-compose up --build --force-recreate -d
  
```

Рис. 3.7 Налаштування .env та підняття збірки

Процес починається з редагування файлу `.env` для встановлення змінних оточення (`PUID` і `PGID`), щоб забезпечити належні дозволи для сервісів. Після оновлення локального репозиторію найновішим кодом з репозиторію GitHub `'homelab-webpage'`, команди переходять до розгортання сервісів за допомогою Docker Compose. Команда `'docker-compose up'` запускається з опціями побудови контейнерів з нуля, примусового їх відтворення та запуску в автономному режимі. Такий підхід не тільки ініціалізує сервіси, але й гарантує, що будь-які оновлення або зміни в конфігурації Docker або коді веб-сторінки будуть негайно застосовані, а контейнери будуть відповідно оновлені.

Тепер, після успішного розгортання сервісів можна налаштовувати nginx проху manager. Для цього необхідно перейти на веб-сторінку <http://10.10.10.2:81>, вхід за стандартними логіном `admin@example.com` і паролем `changeme`.

Після форми з заміною стандартних даних для входу, необхідно отримати сертифікат для того, щоб працював SSL термінатор на цьому сервісі. Завдяки ньому клієнтам можна подавати веб-сайти захищено через https без потреби

налаштовувати https на самих серверах з контентом, що зазвичай має різну і незручну імплементацію.

Certbot використовує два основні методи для отримання сертифікатів: HTTP-01 challenge і DNS-01 challenge. HTTP-01 challenge вимагає запуску веб-сервера для відповіді на HTTP-запити від Let's Encrypt на порту 80, підтверджуючи володіння доменом шляхом надання певного файлу. На відміну від цього, DNS-01 challenge передбачає створення TXT-запису DNS для підтвердження контролю над доменом. Цей метод підходить, коли прямий доступ до веб-сервера неможливий, і дозволяє випускати сертифікати. Обидва методи призначені для автоматизації процесу перевірки для безпечної видачі SSL/TLS сертифікатів.

Автоматизація виклику DNS-01 передбачає використання API провайдера DNS для програмного створення TXT-записів DNS. Certbot або подібні клієнти можуть автоматизувати випуск сертифікатів шляхом інтеграції з цими API. Nginx Proху Manager може створювати та оновлювати сертифікати завдяки модулям до certbot, що надає доступ до всіх популярних провайдерів. У випадку Cloudflare, необхідно створити токен, що має доступи до зони DNS mahurin.xyz (рис. 3.8).

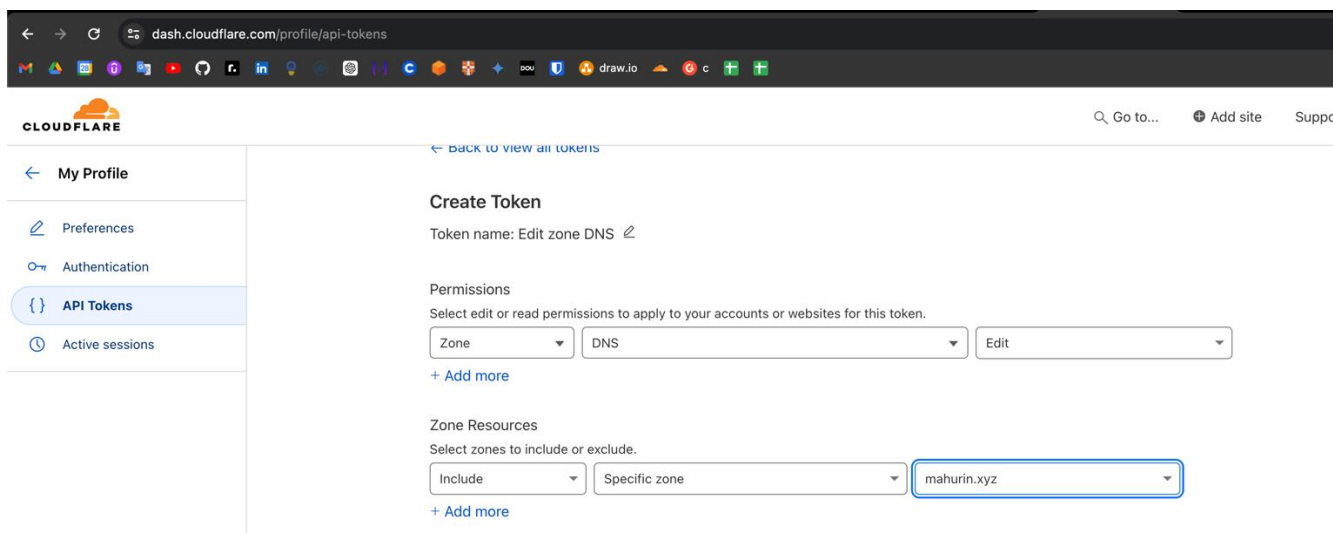


Рис. 3.8 API токен для генерації сертифікатів

Тепер, маючи токен, можна замовити сертифікати через веб-інтерфейс зворотнього проксі (рис. 3.9).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

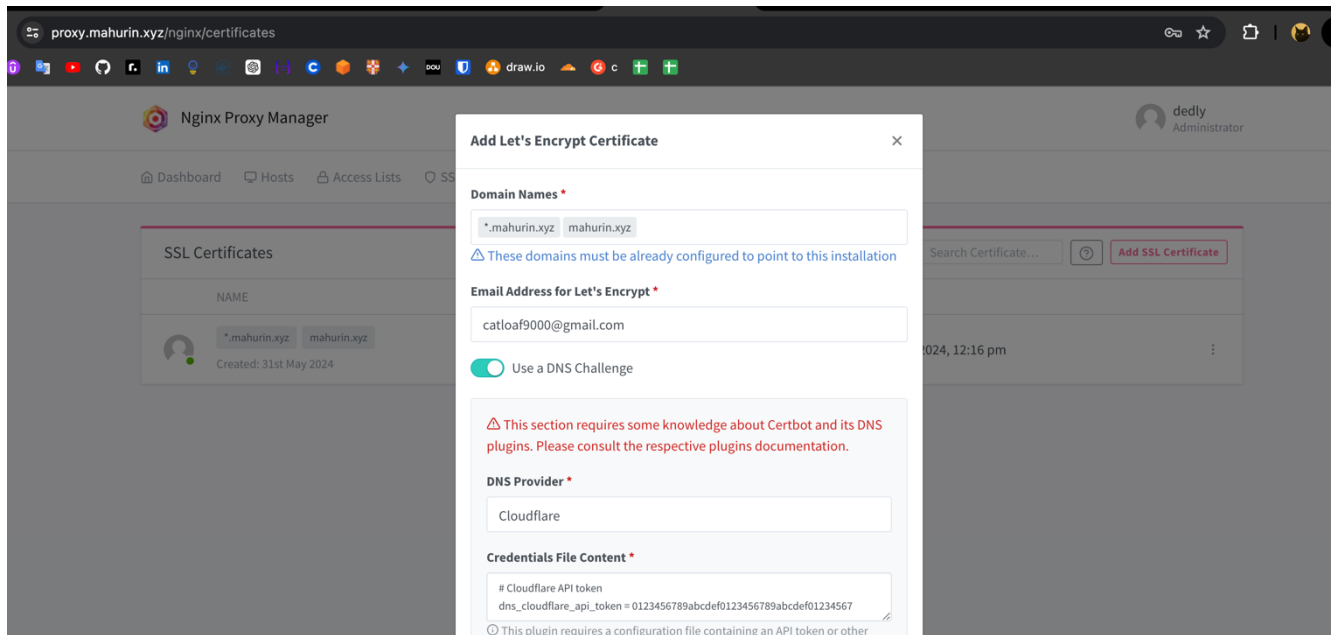


Рис. 3.9 Додання letsencrypt сертифікату з certbot dns-01 challenge

Додавання проксі-хоста в Nginx Proxy Manager є простим процесом. З дашборду Nginx Proxy Manager перехід до розділу «Проксі-хости». Тут можна додати новий проксі-вузол, вказавши доменне ім'я, яким треба керувати. Потім Налаштовуємо URL-адресу переадресації та порт, на якому працює веб-сервіс. Крім того, ви можете встановити розширені налаштування, такі як користувацьке розташування, контроль доступу, SSL-сертифікати та кешування. Після збереження цих конфігурацій Nginx Proxy Manager виконує маршрутизацію запитів до відповідного місця призначення на основі налаштувань.

Наразі нам необхідно встановити всього два проксі хоста – для mahurin.xyz та проху.mahurin.xyz. З налаштуванням mahurin.xyz все просто, а ось з проху.mahurin.xyz є нюанс – необхідно не дозволяти підключатися з зовнішнього світу з будь-якої локації, бо доступ до конфігурації зворотнього проксі – це ризик безпеки. Отож, необхідно створити список доступу, щоб лише локальні IP-адреси та адрес офісу, де я працюю, могли підключатися та налаштовувати зворотній проксі без використання VPN (рис. 3.10).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				82
Змн.	Арк.	№ докум.	Підпис	Дата		

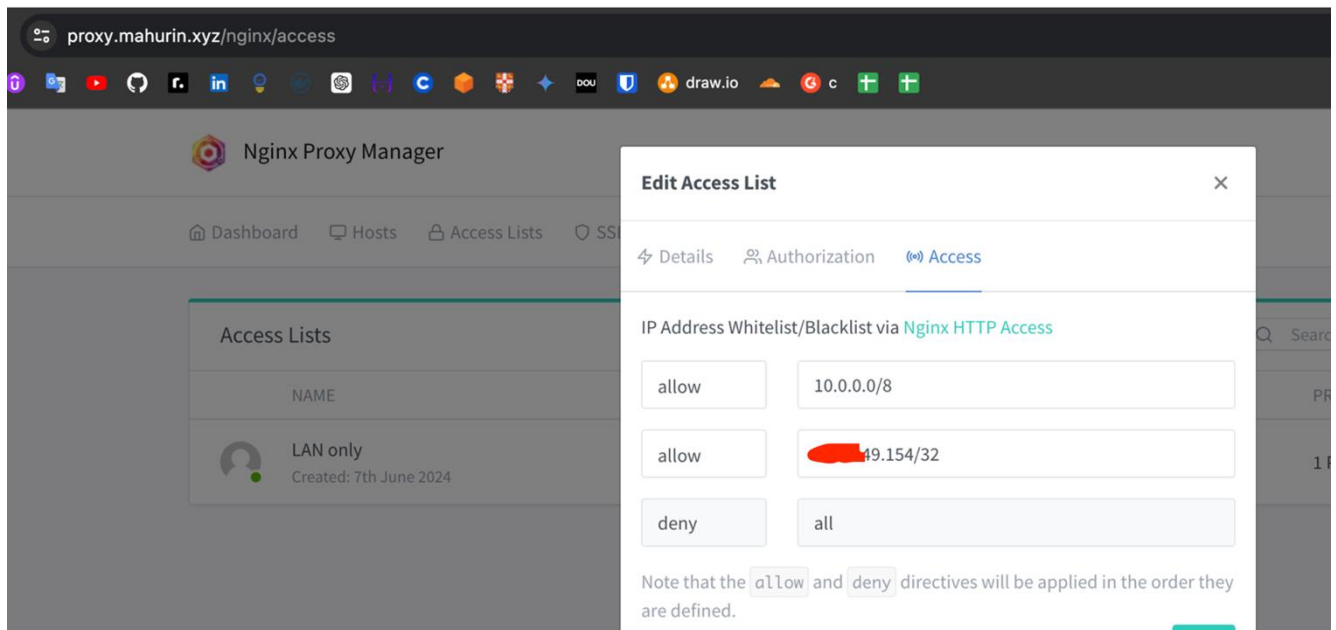


Рис. 3.10 Створений список доступу

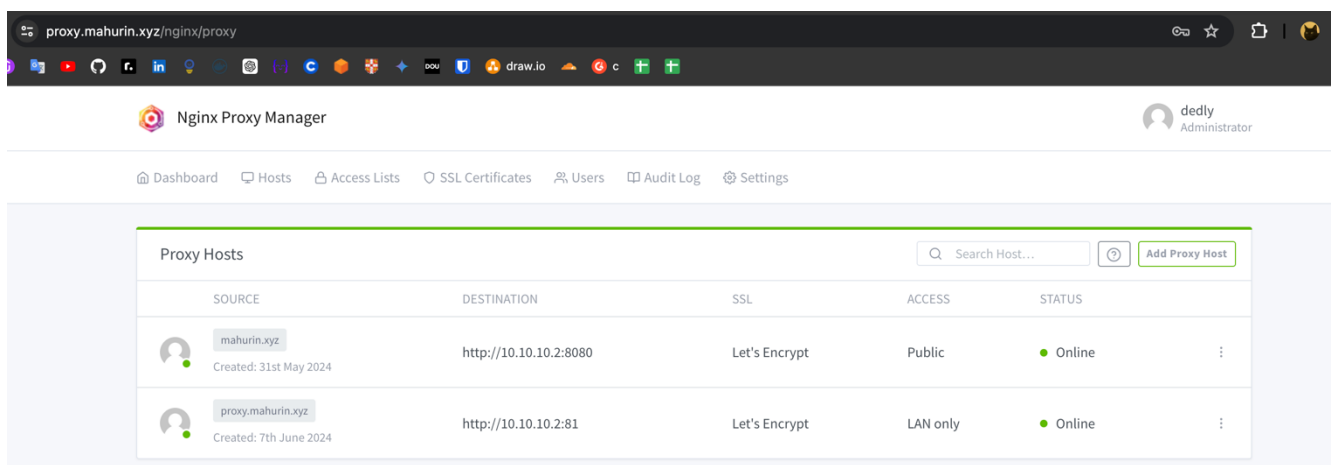


Рис. 3.11 Створені проксі хости

3.1.3 Налаштування Proxmox та Minecraft серверу

Proxmox VE - це платформа управління серверами з відкритим вихідним кодом, розроблена для полегшення віртуалізації підприємств та любителів [15]. Вона інтегрує технології контейнерної та повної віртуалізації за допомогою єдиного інтерфейсу управління, що дозволяє ефективно налаштовувати та керувати віртуальними машинами та контейнерами.

Proxmox поєднує в собі ядро Debian Linux з віртуальними машинами KVM (Kernel-based Virtual Machine) і LXC (Linux Containers), керованими через веб-

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				83
Змн.	Арк.	№ докум.	Підпис	Дата		

інтерфейс. Він підтримує кластеризацію, що дозволяє спільно використовувати ресурси, реплікацію даних та обхід відмов між серверами.

Proxmox необхідний для початкового розміщення віртуальної машини сервера Minecraft, забезпечуючи стабільне і масштабоване середовище віртуалізації. Його здатність керувати різноманітними технологіями віртуалізації робить його придатним для майбутніх проектів, що передбачають більш складну віртуалізацію та контейнеризацію, таким чином підтримуючи широкий спектр сервісів та експериментів.

Проблема в тому, що на початку було складно вибрати, до якої підмережі віднести Proxmox – до внутрішніх сервісів (тоді не буде доступу до зовнішнього світу), або до зовнішньої (в такому випадку неможлива зворотня комунікація з внутрішніми сервісами мережі). Варіант взяти ще один фізичний сервер наразі не актуальний, бо потреба невелика та це дороге. Було вирішено це питання компромісом – докупити до фізичного хоста також ще мережеву карту, в цьому випадку він матиме два мережевих інтерфейсів та зможе мати як внутрішні, так і зовнішні віртуальні машини. Один мінус в цьому – це питання безпеки, якщо хост буде скомпроментовано, буде прямий доступ до захищеної підмережі через DMZ мережу, але з використанням надійних вбудованих фаєрволів, на мою думку, це безпечно. Також, було вирішено докупити оперативної пам'яті 4 x 8 DDR3 1600 MT/s, щоб система мала 32 ГБ оперативної пам'яті, що збільшило її вдвічі.

Перед встановленням PVE, треба перевірити, чи дійсно система розпізнала мережеву картку та додаткову пам'ять і виставила правильну частоту (рис. 3.12).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		84

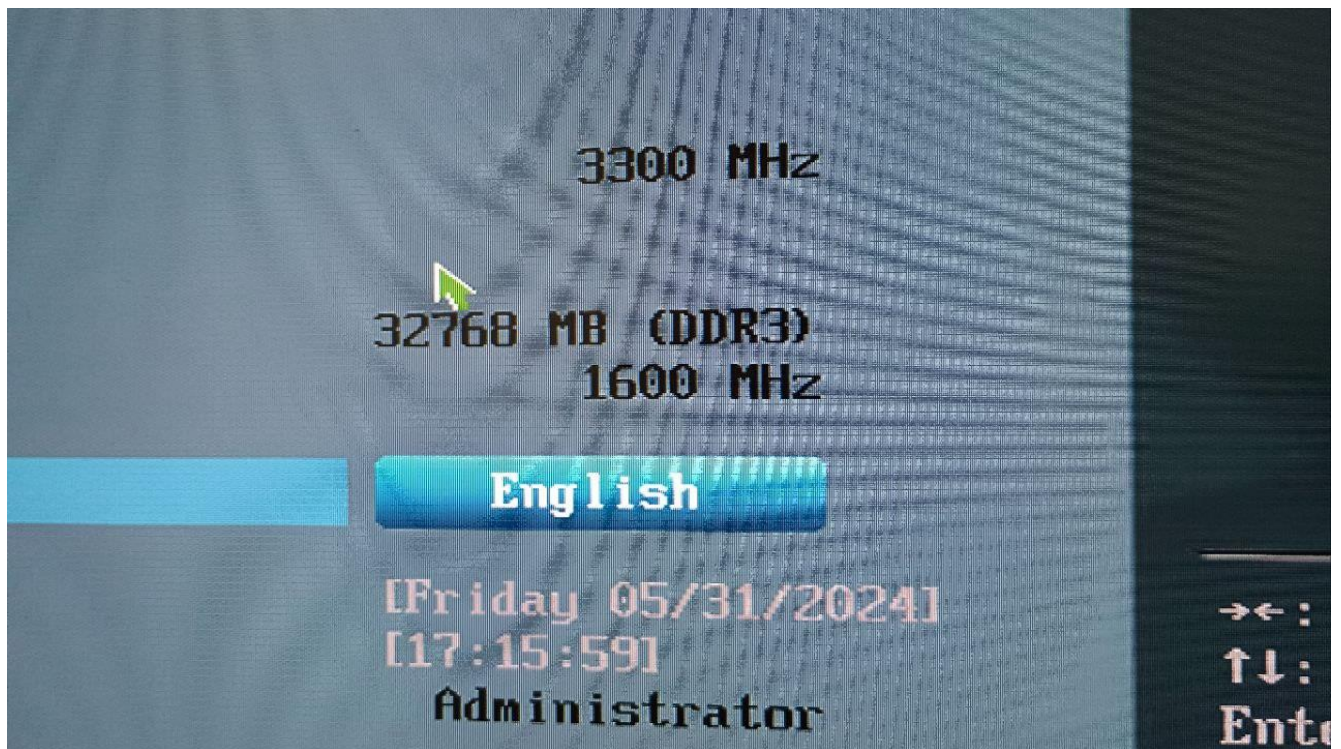


Рис. 3.12 Перевірка пам'яті у bios

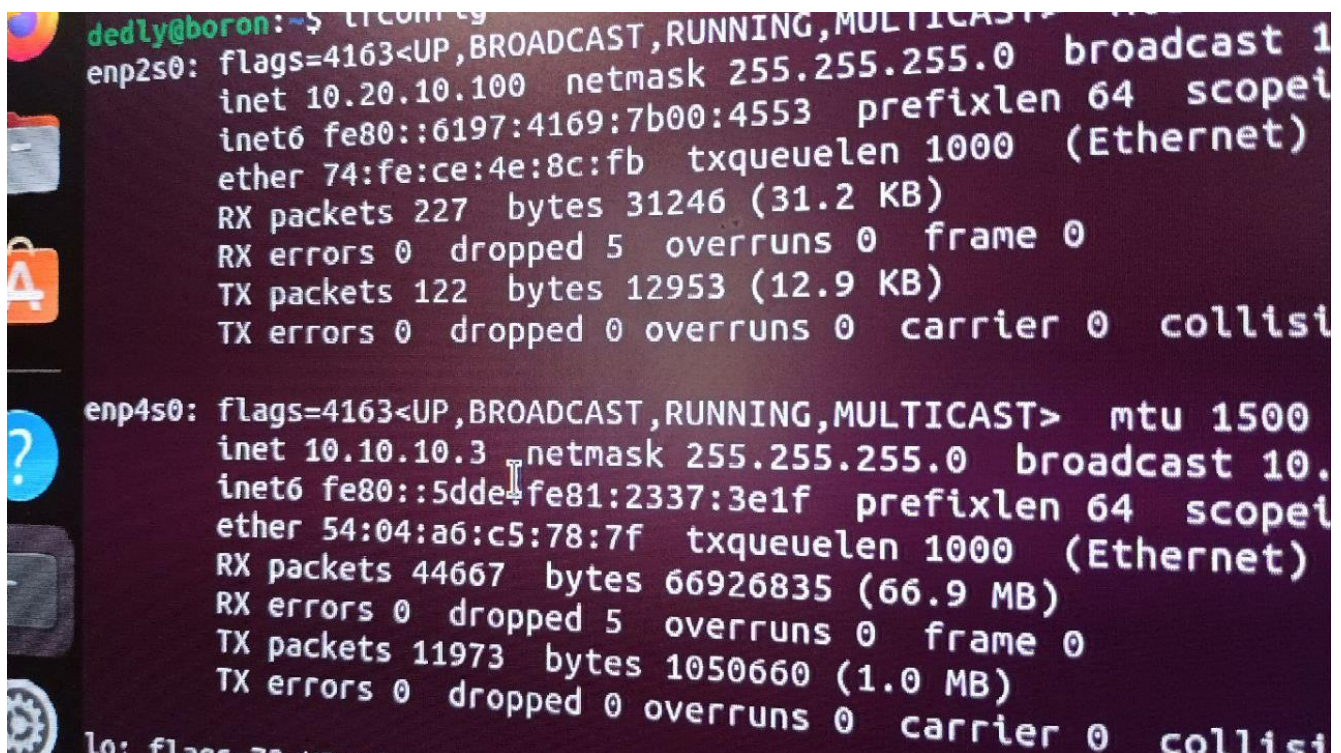


Рис. 3.13 Перевірка мережевих інтерфейсів

Система працює, а значить наступний крок – встановлення ОС Proxmox Virtual Environment.

Щоб встановити Proxmox VE без графічного інтерфейсу, можна завантажити ISO-образ Proxmox VE і записати його на USB-накопичувач або компакт-диск. Під

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				85
Змн.	Арк.	№ докум.	Підпис	Дата		

час інсталяції налаштовується ім'я хоста системи, мережеві налаштування та параметри зберігання. Після встановлення Proxmox VE, ним можна керувати через інтерфейс командного рядка або через веб-панель управління на іншому комп'ютері, що забезпечує повну функціональність без необхідності прямого графічного інтерфейсу користувача на самому сервері.

Вогон має дві статичні IP адреси – 10.10.10.3 для зовнішніх сервісів, 10.20.10.3 для внутрішніх сервісів та для керування веб-консоллю.

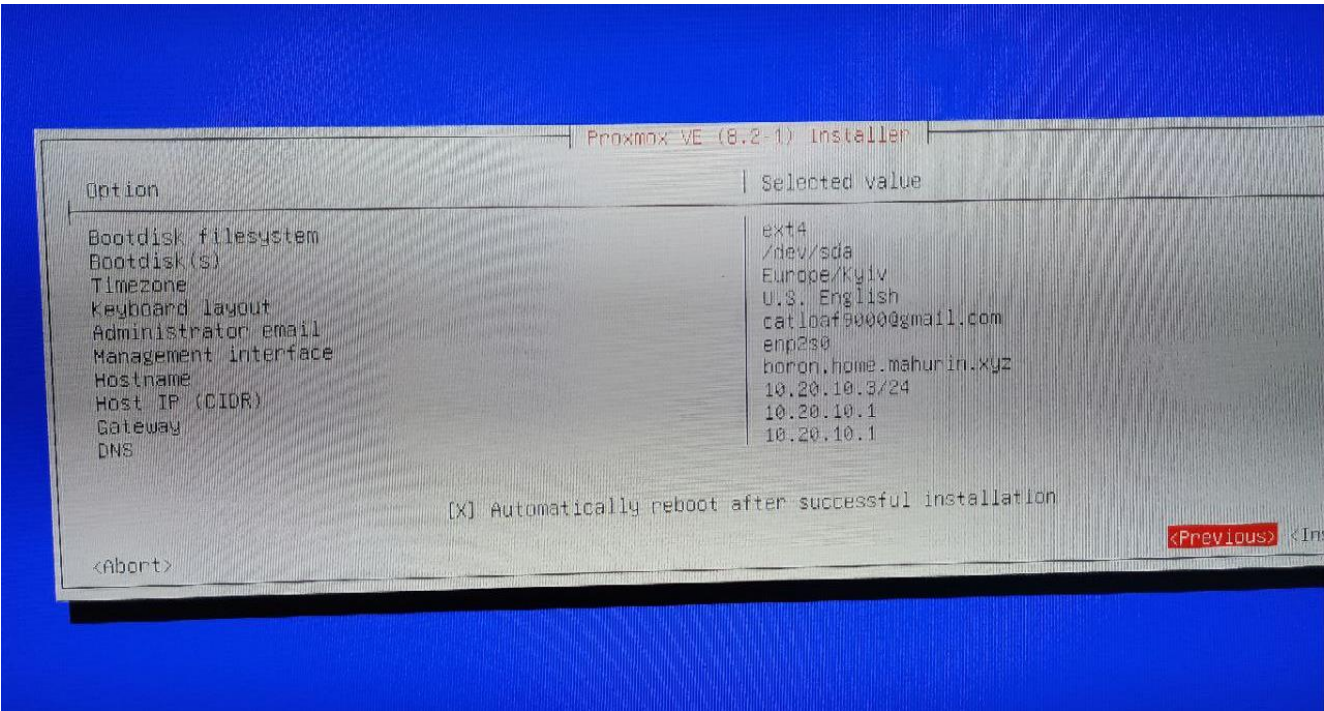


Рис. 3.14 Встановлення Proxmox VE

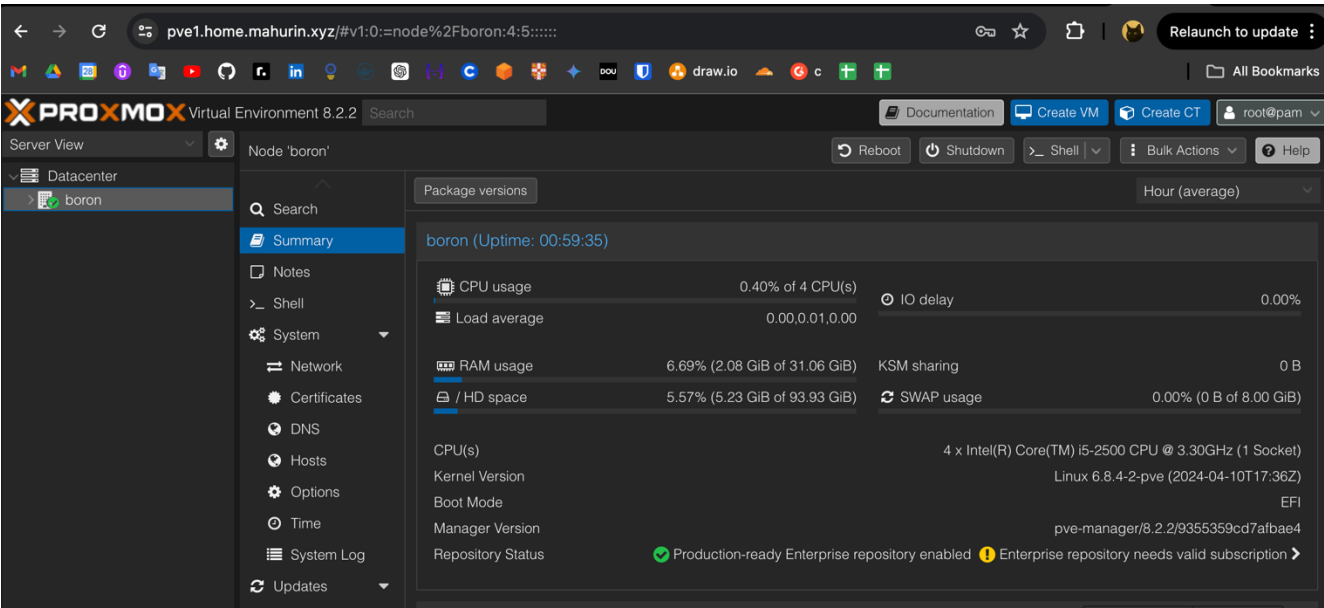


Рис. 3.15 Встановлений Proxmox VE

Процес створення віртуальної машини для сервера Minecraft на Proxmox VE складається з декількох кроків. Для нової VM вибрано в якості операційної системи Linux Ubuntu Server 24.04. Для VM виділяється два ядра процесора і 8 ГБ оперативної пам'яті для забезпечення достатньої продуктивності. Для зберігання даних налаштовано SSD-накопичувач об'ємом 32 ГБ з увімкненою опцією discard для оптимізації продуктивності та довговічності SSD. Ця конфігурація призначена для ефективного запуску сервера Minecraft, використовуючи можливості віртуалізації Proxmox VE для забезпечення стабільного та масштабованого ігрового середовища.

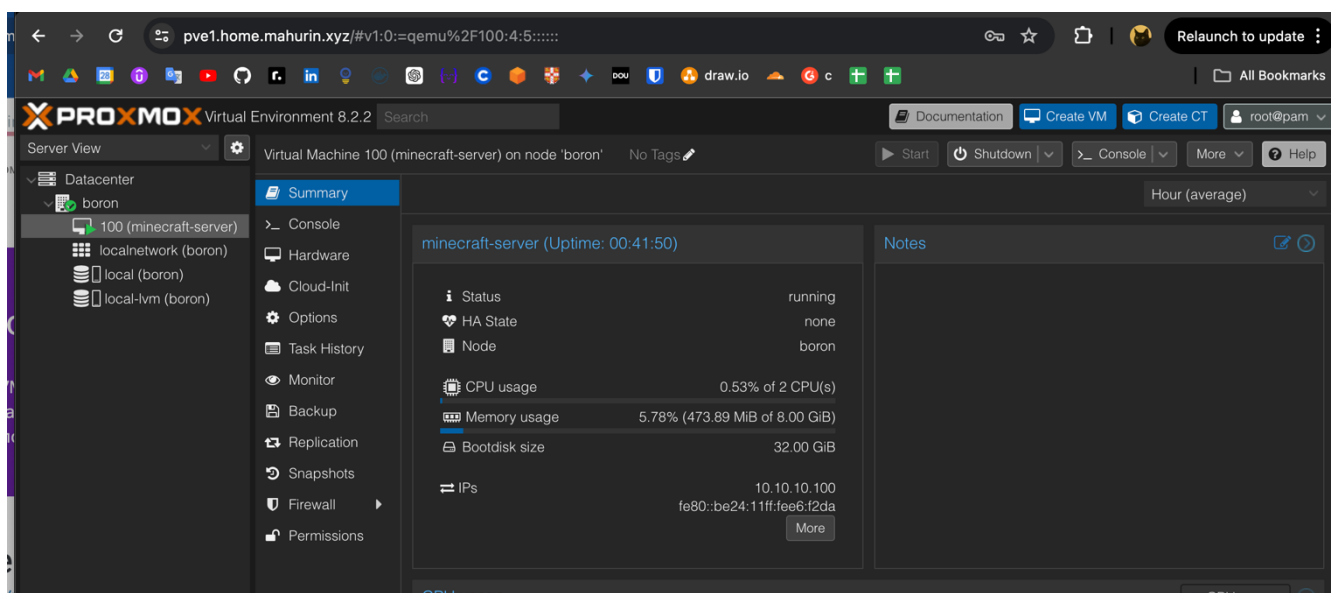


Рис. 3.16 Створена і запущена VM minecraft-server

Процес починається з оновлення та модернізації операційної системи, щоб переконатися, що всі пакунки актуальні. Після цього встановлюється гостьовий агент QEMU для підвищення продуктивності та управління віртуальною машиною, що працює на Proxmox. Далі встановлюється необхідне програмне забезпечення для запуску сервера Minecraft, таке як Java Runtime Environment, unzip та screen. Потім створюється спеціальний каталог для файлів сервера Minecraft, які завантажуються і розпаковуються в цей каталог. Право власності на каталог призначається відповідному користувачеві для цілей управління. Нарешті, скрипт запуску сервера редагується, щоб скоригувати налаштування пам'яті Java для оптимізації продуктивності сервера (рис. 3.17).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				87
Змн.	Арк.	№ докум.	Підпис	Дата		

```

sudo apt update && sudo apt upgrade -y
sudo apt install qemu-guest-agent
Shell

sudo apt install openjdk-17-jre unzip screen -y
sudo mkdir -p /opt/minecraft/GT_New_Horizons_2.6.1_Server_Java_17-21
sudo chown -R dedly:dedly /opt/minecraft
cd /opt/minecraft/GT_New_Horizons_2.6.1_Server_Java_17-21
wget
https://downloads.gtnewhorizons.com/ServerPacks/GT_New_Horizons_2.6.1_Server_Java_17-21.zip
unzip GT_New_Horizons_2.6.1_Server_Java_17-21.zip
rm GT_New_Horizons_2.6.1_Server_Java_17-21.zip
Shell

vim startserver-java9.sh
Shell

```

Рис. 3.17 Конфігурація серверу та завантаження серверу

Скрипту запуску сервера надаються дозволи на виконання (рис. 3.18). Створюється файл «eula.txt» з угодою з ліцензійною угодою Minecraft. Сервер Minecraft запускається в новому сеансі екрану, що дозволяє йому продовжувати працювати у фоновому режимі, і запускається скрипт запуску для початку роботи сервера.

```

chmod +x startserver-java9.sh
echo "eula=true" > eula.txt
Shell

screen -R MC
Shell

./startserver.sh
Shell

```

Рис. 3.18 Конфігурація серверу та запуск

3.2 Конфігурація внутрішніх сервісів

3.2.1 Розгортання допоміжних сервісів: PiHole, SMB share, VPN Wireguard

Після того, як зовнішні служби запущено, наступним кроком є налаштування внутрішніх служб homelab. Ці служби можна умовно поділити на дві групи: мережеві служби підтримки та група медіа-стеку для користувачів. Спочатку треба зробити перше, бо воно є основою для роботи медіа-стеку.

Служби підтримки мережі охоплюють кілька важливих компонентів: PiHole для блокування реклами в мережі, налаштування NAS через SMB для спільного зберігання даних і Wireguard VPN для безпечного тунелювання інтернет-трафіку.

Samba - це програмне забезпечення з відкритим вихідним кодом, яке забезпечує безперебійну роботу з файлами та послугами друку для клієнтів SMB/CIFS. Вона забезпечує сумісність між серверами Linux/Unix і клієнтами на базі Windows, Linux та MacOS, дозволяючи цим системам взаємодіяти один з одним і спільно використовувати файли і принтери. У homelab необхідно зробити лише обмін файлів, створити дві samba share: перша для персональних файлів, з обмеженим доступом користувачів, друга – публічна, з більш відкритим доступом та загальними файлами та мультимедіа.

Перша частина створення samba share – монтування та форматування диску. Під час підключення диску видно, що він підключений як /dev/sda, відформатований під формат NTFS (рис. 3.19), що не підходить до нашого проекту, необхідно відформатувати до Ext4. Хоча ця файлова система не розпізнається ОС Windows і MacOS, це не має значення, оскільки при створенні samba share на це не звертається уваги. Форматування відбувається утилітою fdisk (рис. 3.20).

```
dedly@helium:~$ lsblk -f
NAME        FSTYPE FSVER LABEL        UUID                                FSAVAIL FSUSE% MOUNTPOINTS
sda
├─sda1
└─sda2      ntfs      NAS drive EE50955A509529FB
mmcblk0
├─mmcblk0p1 vfat     FAT32 bootfs      44FC-6CF2              447.9M   12% /boot/firmware
└─mmcblk0p2 ext4      1.0    rootfs      93c89e92-8f2e-4522-ad32-68faed883d2f  53.1G    3% /
dedly@helium:~$ sudo umount /dev/sda1
sudo umount /dev/sda2
umount: /dev/sda1: not mounted.
umount: /dev/sda2: not mounted.
dedly@helium:~$ sudo fdisk /dev/sda
```

Рис. 3.19 Перегляд диску /dev/sda

```

Command (m for help): d
Partition number (1,2, default 2): 1

Partition 1 has been deleted.

Command (m for help): d
Selected partition 2
Partition 2 has been deleted.

Command (m for help): p
Disk /dev/sda: 447.13 GiB, 480103981056 bytes, 937703088 sectors
Disk model: Generic
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 4EF3B813-E789-4A78-938D-2314CB867D1F

Command (m for help): n
Partition number (1-128, default 1):
First sector (34-937703054, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-937703054, default 937701375):

Created a new partition 1 of type 'Linux filesystem' and of size 447.1 GiB.

Command (m for help): p
Disk /dev/sda: 447.13 GiB, 480103981056 bytes, 937703088 sectors
Disk model: Generic
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 4EF3B813-E789-4A78-938D-2314CB867D1F

Device      Start      End      Sectors  Size Type
/dev/sda1   2048 937701375 937699328 447.1G Linux filesystem

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

dedly@helium:~$ lsblk -f
NAME        FSTYPE FSVER LABEL  UUID                                  FSAVAIL FSUSE% MOUNTPOINTS
sda
└─sda1
mmcblk0
├─mmcblk0p1 vfat   FAT32 bootfs 44FC-6CF2                447.9M   12% /boot/firmware
└─mmcblk0p2 ext4    1.0   rootfs 93c89e92-8f2e-4522-ad32-68faed883d2f 53.1G   3% /
dedly@helium:~$ sudo mkfs.ext4 /dev/sda1

```

Рис. 3.20 Форматування диску /dev/sda, створення розділів

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

dedly@helium:~$ sudo mkfs.ext4 /dev/sda1
mkfs.ext4 1.47.0 (5-Feb-2023)
Creating filesystem with 117212416 4k blocks and 29310976 inodes
Filesystem UUID: a6bd3764-eacc-4609-9abd-e1af838f4419
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done

dedly@helium:~$ lsblk -f
NAME        FSTYPE FSVER LABEL        UUID                                FSAVAIL FSUSE% MOUNTPOINTS
sda
└─sda1      ext4    1.0          a6bd3764-eacc-4609-9abd-e1af838f4419
mmcblk0
├─mmcblk0p1 vfat    FAT32 bootfs 44FC-6CF2          447.9M   12% /boot/firmware
└─mmcblk0p2 ext4    1.0   rootfs 93c89e92-8f2e-4522-ad32-68faed883d2f 53.1G    3% /
dedly@helium:~$

```

Рис. 3.21 Форматування диску /dev/sda, створення файлової системи

```

dedly@helium:~$ sudo mkdir /mnt/nas1
dedly@helium:~$ sudo blkid /dev/sda1
/dev/sda1: UUID="a6bd3764-eacc-4609-9abd-e1af838f4419" BLOCK_SIZE="4096"
TYPE="ext4" PARTUUID="e24d21f3-4346-5043-a199-336939c38843"

dedly@helium:~$ sudo vim /etc/fstab
...
dedly@helium:~$ sudo cat /etc/fstab
proc          /proc          proc          defaults      0              0
PARTUUID=49bf66d5-01 /boot/firmware vfat          defaults      0              2
PARTUUID=49bf66d5-02 /              ext4          defaults,noatime 0              1
# a swapfile is not a swap partition, no line here
# use dphys-swapfile swap[on|off] for that
UUID=a6bd3764-eacc-4609-9abd-e1af838f4419 /mnt/nas1 ext4 defaults 0 0

dedly@helium:~$ sudo mount /mnt/nas1
mount: (hint) your fstab has been modified, but systemd still uses
the old version; use 'systemctl daemon-reload' to reload.
dedly@helium:~$ sudo systemctl daemon-reload
dedly@helium:~$ df -h /mnt/nas1
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       440G   28K  417G   1% /mnt/nas1

dedly@helium:~$ sudo reboot
...
dedly@helium:~$ df -h /mnt/nas1
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       440G   28K  417G   1% /mnt/nas1
dedly@helium:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0     0 447.1G  0 disk
└─sda1       8:1     0 447.1G  0 part /mnt/nas1
mmcblk0     179:0    0   59.5G  0 disk
├─mmcblk0p1 179:1    0    512M  0 part /boot/firmware
└─mmcblk0p2 179:2    0     59G   0 part /

```

Рис. 3.22 Перманентне монтування файлової системи і перевірка

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				91
Змн.	Арк.	№ докум.	Підпис	Дата		

```
Shell
dedly@helium:~ $ sudo apt-get update
dedly@helium:~ $ sudo apt-get install samba

dedly@helium:~ $ sudo adduser nas-user
...
dedly@helium:~ $ sudo adduser nas-guest
...
dedly@helium:~ $ sudo mkdir -p /mnt/nas1/dedly
dedly@helium:~ $ sudo mkdir -p /mnt/nas1/public
dedly@helium:~ $ sudo chown dedly:dedly /mnt/nas1/dedly
dedly@helium:~ $ sudo chown dedly:nas-user /mnt/nas1/public
dedly@helium:~ $ sudo chmod 770 /mnt/nas1/dedly
dedly@helium:~ $ sudo chmod 775 /mnt/nas1/public
dedly@helium:~ $ ls -la /mnt/nas1
total 32
drwxr-xr-x 5 root root 4096 Mar 27 13:22 .
drwxr-xr-x 3 root root 4096 Mar 27 12:29 ..
drwxrwxr-x 2 dedly dedly 4096 Mar 27 13:22 dedly
drwx----- 2 root root 16384 Mar 27 12:06 lost+found
drwxrwxr-x 2 dedly nas-user 4096 Mar 27 13:22 public
```

Рис. 3.23 Встановлення пакунків та підготовка до створення samba share

Послідовність конфігурації включає оновлення списку пакунків та встановлення Samba у системі. Далі до системи було додано двох користувачів, nas-user і nas-guest, і призначено їх до відповідних груп, для кожного з яких було створено домашні каталоги. Потім були створені каталоги /mnt/nas1/dedly і /mnt/nas1/public для зберігання даних у мережевому сховищі. Право власності на ці каталоги було встановлено для конкретних користувачів і груп, щоб керувати правами доступу. Нарешті, дозволи були скориговані, щоб дозволити відповідні рівні доступу до цих каталогів, що забезпечило безпечне та структуроване зберігання файлів і можливість спільного доступу до них.

Файл конфігурації Samba (smb.conf) було відредаговано для визначення двох спільних каталогів: DedlyShare і PublicShare (рис. 3.24). DedlyShare налаштовано на ексклюзивний доступ для користувача 'dedly', тоді як PublicShare доступний для користувачів 'nas-user', 'nas-guest' і 'dedly'. Обидва ресурси налаштовано так, щоб дозволити запис і перегляд файлів. Потім для користувачів 'dedly', 'nas-user' і 'nas-guest' були встановлені паролі Samba, щоб захистити доступ. Нарешті, служба Samba була перезапущена, щоб застосувати ці нові конфігурації, гарантуючи, що

вказані каталоги будуть правильно надані в загальний доступ відповідно до налаштувань.

```
dedly@helium:~ $ sudo vim /etc/samba/smb.conf
...
dedly@helium:~ $ sudo tail -n 15 /etc/samba/smb.conf

[DedlyShare]
comment = Dedlys Share
path = /mnt/nas1/dedly
valid users = dedly
read only = no
browseable = yes

[PublicShare]
comment = Public Share
path = /mnt/nas1/public
valid users = nas-user, nas-guest, dedly
read only = no
browseable = yes

dedly@helium:~ $ sudo smbpasswd -a dedly
New SMB password:
Retype new SMB password:
Added user dedly.
dedly@helium:~ $ sudo smbpasswd -a nas-user
New SMB password:
Retype new SMB password:
Added user nas-user.
dedly@helium:~ $ sudo smbpasswd -a nas-guest
New SMB password:
Retype new SMB password:
Added user nas-guest.

dedly@helium:~ $ sudo systemctl restart smbd
```

Рис. 3.24 Налаштування samba share

Це всі необхідні налаштування для samba share, перевірено їх буде у останньому пункті третього розділу, як і всі інші сервіси.

Wireguard - це сучасний протокол VPN, відомий своєю простотою і високою продуктивністю [16]. Він використовується для захисту мережевого трафіку шляхом створення зашифрованих тунелів для передачі даних. Це підвищує конфіденційність і безпеку, особливо при віддаленому доступі до homelab,

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		93

гарантуючи, що конфіденційна інформація залишається захищеною від зовнішніх загроз.

Процес конфігурації WireGuard включає оновлення системи та встановлення необхідних пакетів, WireGuard і утиліт (рис. 3.25 – рис. 3.26). Після інсталяції генеруються приватний і публічний ключі, які надійно зберігаються в каталозі WireGuard, гарантуючи, що приватний ключ доступний тільки користувачеві root з метою безпеки. Публічний ключ є похідним від цього приватного ключа.

```

Shell
sudo apt update
sudo apt install wireguard ufw vim

# generate private key
wg genkey | sudo tee /etc/wireguard/private.key
# ensure that only root can access to private key
sudo chmod go= /etc/wireguard/private.key
# generate public key for private key
sudo cat /etc/wireguard/private.key | wg pubkey | sudo tee
/etc/wireguard/public.key

```

Рис. 3.25 Встановлення та конфігурація WireGuard

```

Shell
sudo vim /etc/wireguard/wg0.conf

/etc/wireguard/wg0.conf:

[Interface]
PrivateKey = base64_encoded_private_key_goes_here
Address = 10.30.250.1/24
ListenPort = 51820
SaveConfig = true

PostUp = ufw route allow in on wg0 out on eth0
PostUp = iptables -t nat -I POSTROUTING -o eth0 -j MASQUERADE
PostUp = ip6tables -t nat -I POSTROUTING -o eth0 -j MASQUERADE
PreDown = ufw route delete allow in on wg0 out on eth0
PreDown = iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE
PreDown = ip6tables -t nat -D POSTROUTING -o eth0 -j MASQUERADE

```

Рис. 3.26 Конфігурація WireGuard

Файл конфігурації WireGuard `wg0.conf` налаштовує інтерфейс VPN з певними мережевими налаштуваннями і правилами безпеки (рис. 3.27). У розділі `[Interface]` вказано приватний ключ, IP-адресу VPN, порт прослуховування та директиву для збереження конфігурації після вимкнення. Команди `PostUp` і `PreDown` налаштовують і очищають IP-маршрутизацію і правила NAT, використовуючи `ufw`, `iptables` і `ip6tables` для управління потоком трафіку через VPN, забезпечуючи безпечну і ефективну обробку пакетів, що входять і виходять з інтерфейсу WireGuard.

```

sudo vim /etc/sysctl.conf
Shell

Append to the end of the file /etc/sysctl.conf:

net.ipv4.ip_forward=1
Copy

Apply configuration:

sudo sysctl -p
Shell

sudo ufw allow 51820/udp
sudo ufw allow OpenSSH
sudo ufw disable
Shell

sudo systemctl enable wg-quick@wg0.service
sudo systemctl start wg-quick@wg0.service
Shell

```

Рис. 3.27 Конфігурація WireGuard

Щоб налаштувати мережу сервера WireGuard на належне пересилання пакетів, файл конфігурації sysctl редагується, щоб увімкнути пересилання IPv4. Ця зміна негайно застосовується за допомогою команди sysctl. Крім того, налаштування брандмауера коригуються, щоб дозволити UDP-трафік на порту, який

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				95
Змн.	Арк.	№ докум.	Підпис	Дата		

використовується WireGuard (51820), і забезпечити доступ по SSH. Потім брандмауер вимикається для цілей конфігурації. Навіть якщо брандмауером не користуються, скрипт WireGuard не виконується, коли не дозволені порти. Нарешті, служба WireGuard увімкнена для автоматичного запуску при завантаженні, і вона негайно запускається для активації VPN-сервера.

```
dedly@helium:~$ sudo systemctl status wg-quick@wg0
● wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0
   Loaded: loaded (/lib/systemd/system/wg-quick@.service; enabled; preset: enabled)
   Active: active (exited) since Wed 2024-03-27 15:14:41 EET; 16s ago
     Docs: man:wg-quick(8)
           man:wg(8)
           https://www.wireguard.com/
           https://www.wireguard.com/quickstart/
           https://git.zx2c4.com/wireguard-tools/about/src/man/wg-quick.8
           https://git.zx2c4.com/wireguard-tools/about/src/man/wg.8
   Process: 3716 ExecStart=/usr/bin/wg-quick up wg0 (code=exited, status=0/SUCCESS)
   Main PID: 3716 (code=exited, status=0/SUCCESS)
      CPU: 320ms

Mar 27 15:14:40 helium wg-quick[3716]: [#] ip link add wg0 type wireguard
Mar 27 15:14:40 helium wg-quick[3716]: [#] wg setconf wg0 /dev/fd/63
Mar 27 15:14:40 helium wg-quick[3716]: [#] ip -4 address add 10.30.250.1/24 dev wg0
Mar 27 15:14:40 helium wg-quick[3716]: [#] ip link set mtu 1420 up dev wg0
Mar 27 15:14:40 helium wg-quick[3716]: [#] ufw route allow in on wg0 out on eth0
Mar 27 15:14:41 helium wg-quick[3754]: Rules updated
Mar 27 15:14:41 helium wg-quick[3754]: Rules updated (v6)
Mar 27 15:14:41 helium wg-quick[3716]: [#] iptables -t nat -I POSTROUTING -o eth0 -j MASQUERADE
Mar 27 15:14:41 helium wg-quick[3716]: [#] ip6tables -t nat -I POSTROUTING -o eth0 -j MASQUERADE
Mar 27 15:14:41 helium systemd[1]: Finished wg-quick@wg0.service - WireGuard via wg-quick(8) for wg0.
dedly@helium:~$
```

Рис. 3.28 Перевірка служби WireGuard

Наступний етап – налаштування клієнтів (рис. 3.29). Щоб налаштувати WireGuard VPN на різних пристроях, треба завантажити та встановити програму WireGuard з AppStore для MacOS або Google Play для Android. На MacOS відкрийте WireGuard, перейдіть до «Керування тунелями Wireguard» і додайте порожній тунель. Налаштуйте тунель за допомогою автоматично згенерованого приватного ключа, встановіть IP-адресу VPN пристрою, вкажіть DNS і деталі сервера, включно з відкритим ключем і кінцевою точкою. Повторіть аналогічний процес на Android, використовуючи значок «+» і опцію «Створити з нуля», ввівши відповідні дані конфігурації для кожного пристрою. Таке налаштування забезпечує безпечне, зашифроване з'єднання між пристроями за допомогою узгоджених конфігурацій DNS і кінцевих точок сервера.

Home laptop:

```
[Interface]
PrivateKey = automatically_generated_private_key
Address = 10.30.250.2/24
DNS = 10.30.10.1

[Peer]
PublicKey = public_key_of_the_server
AllowedIPs = 10.0.0.0/8
Endpoint = 31.43.154.6:51820
```

Work laptop:

```
[Interface]
PrivateKey = automatically_generated_private_key
Address = 10.30.250.3/24
DNS = 10.30.10.1

[Peer]
PublicKey = public_key_of_the_server
AllowedIPs = 10.0.0.0/8
Endpoint = 31.43.154.6:51820
```

Android phone:

```
PrivateKey = automatically_generated_private_key
Address = 10.30.250.4/24
DNS = 10.30.10.1

[Peer]
PublicKey = public_key_of_the_server
AllowedIPs = 10.0.0.0/8
Endpoint = 31.43.154.6:51820
```

Рис. 3.29 Налаштування клієнтів WireGuard

Наведені команди налаштовують конфігурацію WireGuard для інтерфейсу `wg0`, призначаючи певні однорангові пристрої, ідентифіковані за допомогою відкритих ключів, відповідальними за певні IP-адреси в VPN-мережі. Кожна команда пов'язує різний одноранговий пристрій з унікальною внутрішньою IP-

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		97

адресою, ефективно керуючи доступом до мережі та маршрутизацією для кожного підключеного пристрою. Цей метод гарантує, що кожен пристрій або користувач, підключений через VPN, має визначену внутрішню IP-адресу, що забезпечує контрольоване та безпечне керування мережею в межах налаштування WireGuard (рис. 3.30).

```

sudo wg set wg0 peer x+RkoduqcQ0PZW0wDzage6HsgWmMx29Lc0+37WrYUVw= allowed-ips
10.30.250.2
sudo wg set wg0 peer Hhk2MA1bmj+qTZRpMivsP07Q1FsI/mbSNP+0BIE9n0s= allowed-ips
10.30.250.3
sudo wg set wg0 peer myZDdyIwDEfBdg8mu8mRxVNd9/PtguHJwatdaKQr0ww= allowed-ips
10.30.250.4
sudo wg set wg0 peer cLYGJIsG0mZHYAI3l4Y10JUIWIr5AGQI5w3yYfb8NRk= allowed-ips
10.30.250.5

```

Рис. 3.30 Налаштування клієнтів WireGuard

```

dedly@helium:~$ sudo wg
interface: wg0
  public key: Fw/ICTJQm/mN/xRKLp1idjICHa+4Dw7uZjL+0IRKIQ0=
  private key: (hidden)
  listening port: 51820

peer: x+RkoduqcQ0PZW0wDzage6HsgWmMx29Lc0+37WrYUVw=
  allowed ips: 10.30.250.2/32

peer: Hhk2MA1bmj+qTZRpMivsP07Q1FsI/mbSNP+0BIE9n0s=
  allowed ips: 10.30.250.3/32

peer: myZDdyIwDEfBdg8mu8mRxVNd9/PtguHJwatdaKQr0ww=
  allowed ips: 10.30.250.4/32
dedly@helium:~$

```

Рис. 3.31 Перевірка дозволених клієнтів WireGuard

PiHole працює як мережевий блокувальник реклами, який перехоплює запити до відомих рекламних доменів на рівні DNS, запобігаючи завантаженню небажаного контенту. Це підвищує швидкість і ефективність перегляду веб-сторінок, а також покращує мережеву безпеку, блокуючи потенційно шкідливу рекламу і трекери. Також PiHole може мати локальні записи DNS, які будуть використані для зручного доступу до внутрішніх сервісів.

Сервіс PiHole вирішено розгорнути у вигляді контейнеру, тому першим етапом буде встановлення Docker та всіх його додаткових пакунків. Через те, що цей процес всюди однаковий та це вже було продемонстровано у пункті про зовнішні сервіси, повторно команди наведено не буде.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

sudo mkdir -p /mnt/nas1/server
sudo chown dedly:dedly /mnt/nas1/server
mkdir -p /mnt/nas1/server/piHole
cd /mnt/nas1/server/piHole
vim docker-compose.yml

```

Shell

```

# More info at https://github.com/pi-hole/docker-pi-hole/ and https://docs.pi-hole.net/
services:
  pihole:
    container_name: pihole
    image: pihole/pihole:latest
    # For DHCP it is recommended to remove these ports and instead add:
    network_mode: "host"
    ports:
      - "53:53/tcp"
      - "53:53/udp"
      # - "67:67/udp" # Only required if you are using Pi-hole as your DHCP
server
      - "80:80/tcp"
    environment:
      TZ: 'Europe/Kyiv'
      WEBPASSWORD: '*****'
    # Volumes store your data between container upgrades
    volumes:
      - './etc-pihole:/etc/pihole'
      - './etc-dnsmasq.d:/etc/dnsmasq.d'
    # https://github.com/pi-hole/docker-pi-hole#note-on-capabilities
    # cap_add:
    # - NET_ADMIN # Required if you are using Pi-hole as your DHCP server, else
not needed
    restart: unless-stopped

```

YAML

Рис. 3.32 Налаштування PiHole через docker-compose

Процес включає в себе налаштування Pi-Hole на диску NAS за допомогою Docker, що починається зі створення спеціального каталогу та призначення відповідних дозволів. Потім створюється файл docker-compose.yml для визначення служби Pi-Hole. Конфігурація визначає використання найновішого образу Pi-Hole Docker, зіставлення необхідних портів для DNS і функцій веб-інтерфейсу, а також встановлення змінних середовища, таких як часовий пояс і пароль веб-

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		99

адміністратора. Томи налаштовано на збереження даних при оновленні контейнерів. Налаштування гарантує, що Pi-Hole працює безперервно і перезапускається автоматично, якщо його не зупинити вручну (рис. 3.32).

```

Shell
sudo ufw allow 53/tcp
sudo ufw allow 53/udp
sudo ufw allow 80/tcp
sudo ufw status

Shell
sudo docker compose up -d
  
```

Рис. 3.33 Налаштування PiHole та запуск

Команди на рис. 3.33 встановлюють правила брандмауера, щоб дозволити трафік на певних портах, необхідних для роботи Pi-Hole: порт 53 для трафіку DNS (як TCP, так і UDP) і порт 80 для доступу до веб-інтерфейсу (тільки TCP). Після цього Pi-Hole розгортається за допомогою Docker Compose, щоб працювати у фоновому режимі, забезпечуючи безперервну роботу сервісу без прив'язки до терміналу.

Також необхідно зробити налаштування PiHole як DNS-сервера через графічний інтерфейс для RouterOS v7.8 на хості beryllium (рис. 3.34 – рис. 3.35). На панелі конфігурації DNS IP-адреса PiHole вводиться як основний DNS-сервер, з додатковими налаштуваннями для управління динамічними серверами, DNS через HTTPS і властивостями обробки запитів, такими як максимальний розмір пакетів UDP і налаштування таймауту сервера. Це налаштування спрямоване на маршрутизацію всіх DNS-запитів через PiHole, що дозволяє фільтрувати небажаний вміст, відстежувати DNS-запити і підвищувати безпеку та ефективність мережі.

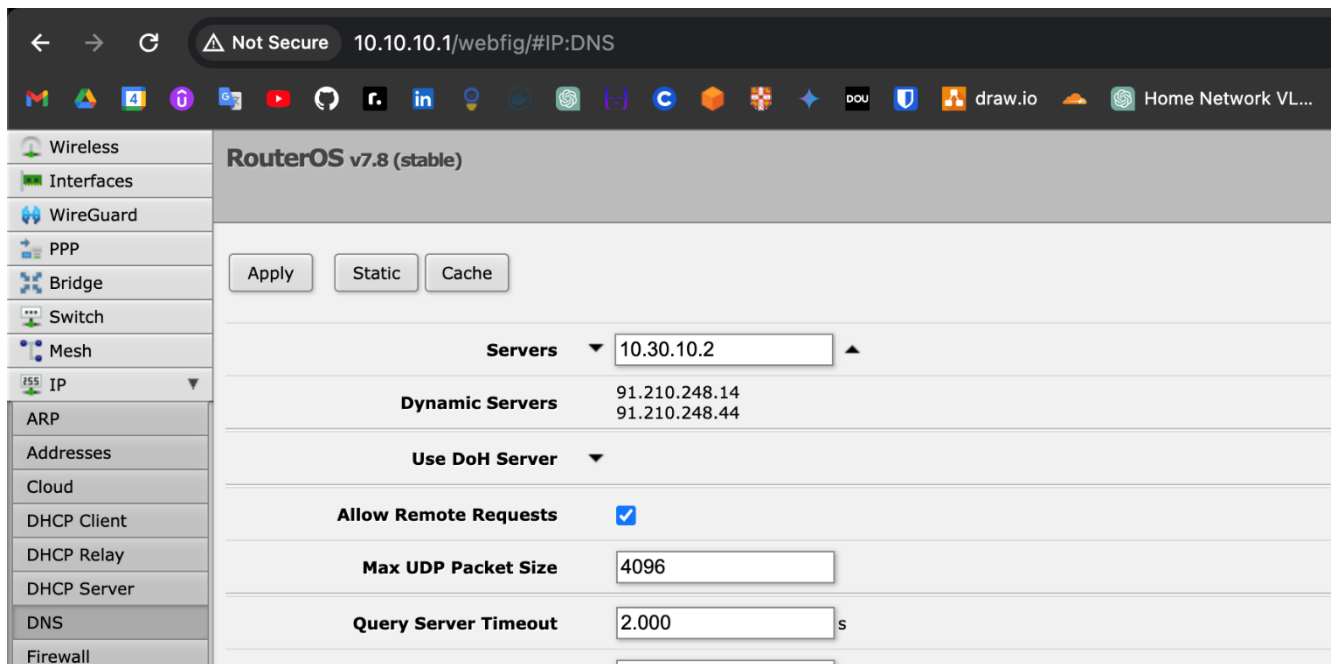


Рис. 3.34 Налаштування beryllium для використання PiHole

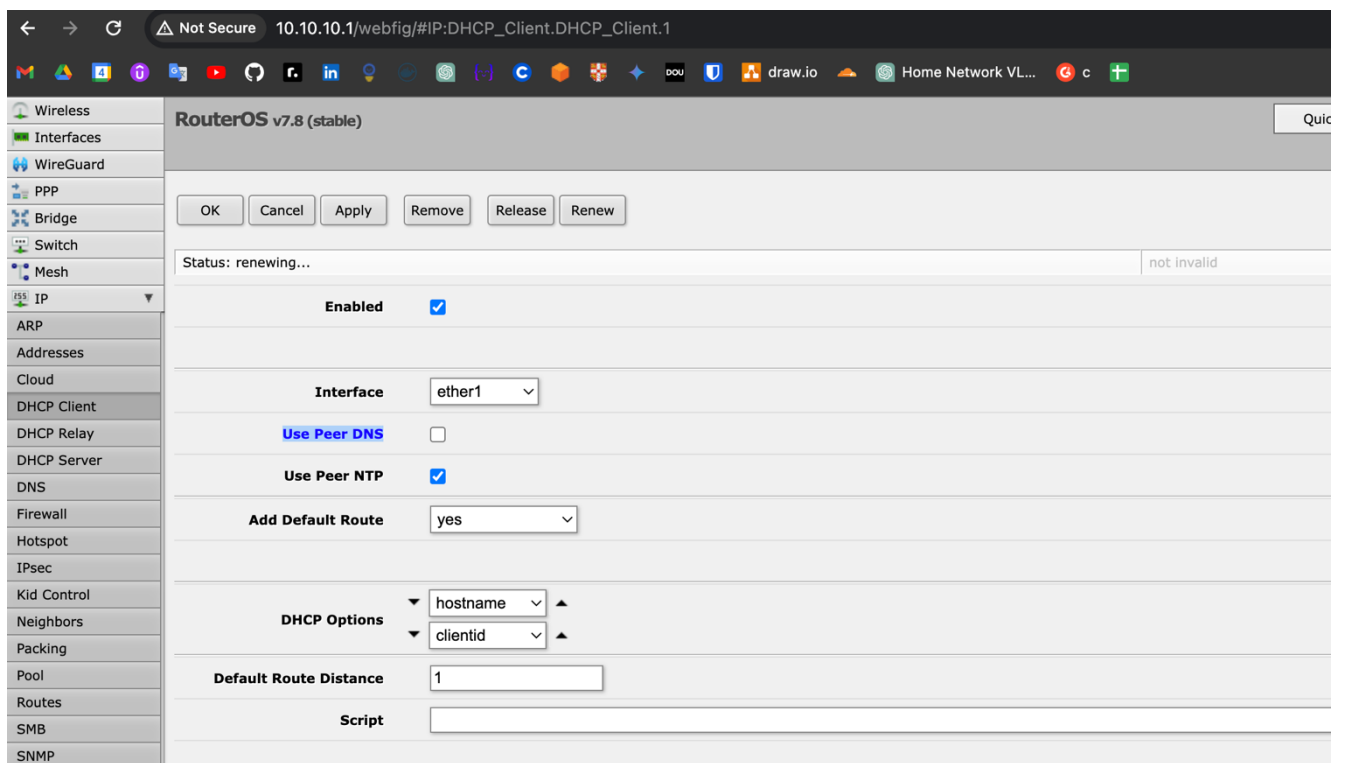


Рис. 3.35 Налаштування beryllium для використання PiHole

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				101
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2.2 Розгортання медіа-стеку: jellyfin, qbittorrent, filebrowser, nginx proxy manager

Розгортання медіа-стеку в середовищі homelab передбачає встановлення різних програм, таких як Jellyfin для керування медіа та потокового передавання, qBittorrent для обміну файлами, Filebrowser для керування файлами та Nginx Proxy Manager для маршрутизації та захисту доступу. Ця конфігурація розширює можливості домашньої лабораторії, надаючи комплексні медіа-сервіси та рішення для обробки даних, а також забезпечує безпечне та ефективне управління доступом за допомогою Nginx Proxy Manager. На відміну від попередніх внутрішніх сервісів, ці всі будуть розгортатись одночасно завдяки docker-compose.

Jellyfin - це безкоштовне програмне забезпечення для медіа-серверів з відкритим кодом, яке дозволяє користувачам керувати цифровими медіа та транслювати їх на різні пристрої [17]. Як самостійне рішення, Jellyfin пропонує інструменти для збору та управління медіафайлами, такими як фільми, телешоу, музика та фотографії, роблячи їх доступними через веб-інтерфейс або додатки для різних платформ. Користувачі можуть транслювати свої медіа без жодних комісій чи відстеження, зберігаючи повний контроль над своїм цифровим контентом та конфіденційністю.

qBittorrent - це крос-платформний BitTorrent-клієнт з відкритим вихідним кодом, який є альтернативою комерційним торрент-клієнтам [18]. Він має добре інтегровану та розширювану пошукову систему для одночасного пошуку торрентів на багатьох сайтах. Крім того, він пропонує детальне керування торрентами, наприклад, визначення пріоритетів файлів, пірінгове керування, планування пропускну здатності та послідовне завантаження, що є корисним для потокового відеофайлу.

Filebrowser - це легкий веб-інструмент для керування файлами, призначений для віддаленого керування файлами на сервері [19]. Він має зручний інтерфейс, який дозволяє користувачам завантажувати, видаляти, редагувати та впорядковувати файли і каталоги, а також надає такі можливості, як спільний доступ до файлів і управління дозволами. Filebrowser може легко інтегруватися в

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		102

персональне або професійне серверне середовище, полегшуючи ефективне керування файлами на сервері через браузер з будь-якого місця.

```

Shell
sudo chown dedly:dedly /srv
mkdir -p /srv/media
mkdir -p /srv/jellyfin/config
mkdir -p /srv/jellyfin/cache
mkdir -p /srv/qbittorrent/config
mkdir -p /srv/filebrowser
mkdir -p /srv/nginxproxymanager
touch /srv/filebrowser/filebrowser.db

vim /srv/filebrowser/.filebrowser.json

JSON
{
  "port": 80,
  "baseUrl": "",
  "address": "",
  "log": "stdout",
  "database": "/database.db",
  "root": "/srv"
}

```

Рис. 3.36 Підготовка до розгортання медіа-стеку

Послідовність налаштування починається зі створення структурованого середовища у каталозі /srv та встановлення дозволів для користувача «dedly». Каталоги для різних служб створюються для полегшення монтування томів Docker, впорядкування специфічних для служб даних, таких як медіафайли, конфігурації для Jellyfin, qBittorrent, а також папки налаштувань для Filebrowser і Nginx Proxy Manager. Файл бази даних для Filebrowser також створюється вручну, оскільки Filebrowser не генерує цей файл автоматично. Супровідна JSON-конфігурація для Filebrowser визначає основні робочі параметри, такі як порт, ведення журналу, розташування бази даних і кореневого каталогу, що мають вирішальне значення для його початкового налаштування та інтеграції з системою.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		103

vim /srv/docker-compose.yaml

Shell

YAML

```

---
services:
  jellyfin:
    image: jellyfin/jellyfin
    container_name: jellyfin
    user: $PUID:$PGID
    ports:
      - 8096:8096
    volumes:
      - /srv/jellyfin/config:/config
      - /srv/jellyfin/cache:/cache
      - /mnt/nas1:/media
    restart: unless-stopped
    environment:
      - JELLYFIN_PublishedServerUrl=https://jellyfin.home.mahurin.xyz

  qbittorrent:
    image: lscr.io/linuxserver/qbittorrent:latest
    container_name: qbittorrent
    environment:
      - PUID=$PUID
      - PGID=$PGID
      - TZ=Europe/Kyiv
      - WEBUI_PORT=8080
      - TORRENTING_PORT=6881
    volumes:
      - /srv/qbittorrent/config:/config
      - /mnt/nas1:/downloads
    ports:
      - 6881:6881
      - 6881:6881/udp
    restart: unless-stopped

  filebrowser:
    image: filebrowser/filebrowser
    container_name: filebrowser
    volumes:
      - /mnt/nas1:/srv
      - /srv/filebrowser/filebrowser.db:/database.db
      - /srv/filebrowser/.filebrowser.json:/filebrowser.json
    user: $PUID:$PGID
    restart: unless-stopped

```

Рис. 3.37 docker-compose файл для медіа-стеку

```

filebrowser:
  image: filebrowser/filebrowser
  container_name: filebrowser
  volumes:
    - /mnt/nas1:/srv
    - /srv/filebrowser/filebrowser.db:/database.db
    - /srv/filebrowser/.filebrowser.json:/.filebrowser.json
  user: $PUID:$PGID
  restart: unless-stopped

nginxproxymanager:
  image: 'docker.io/jc21/nginx-proxy-manager:latest'
  restart: unless-stopped
  environment:
    - PUID=$PUID
    - PGID=$PGID
  ports:
    - '82:80'
    - '81:81'
    - '443:443'
  volumes:
    - /srv/nginxproxymanager/data:/data
    - /srv/nginxproxymanager/letsencrypt:/etc/letsencrypt

```

```
vim /srv/.env
```

Shell

```

PUID=1000
PGID=1000

```



Рис. 3.38 docker-compose файл для медіа-стеку і файл змінних

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		105

Shell

```
mkdir -p /srv/qbittorrent/config/qBittorrent/  
vim /srv/qbittorrent/config/qBittorrent/qBittorrent.conf
```

Copy

```
[AutoRun]  
enabled=false  
program=  
  
[BitTorrent]  
Session\DefaultSavePath=/downloads/  
Session\Port=6881  
Session\QueueingSystemEnabled=true  
Session\TempPath=/downloads/incomplete/  
  
[LegalNotice]  
Accepted=true  
  
[Meta]  
MigrationVersion=6  
  
[Network]  
Cookies=@Invalid()  
PortForwardingEnabled=false  
Proxy\HostnameLookupEnabled=false  
Proxy\Profiles\BitTorrent=true  
Proxy\Profiles\Misc=true  
Proxy\Profiles\RSS=true  
  
[Preferences]  
Connection\PortRangeMin=6881  
Connection\UPnP=false  
Downloads\SavePath=/downloads/  
Downloads\TempPath=/downloads/incomplete/  
WebUI\Address=*  
WebUI\HostHeaderValidation=false  
WebUI\ServerDomains=*  
WebUI\Password_PBKDF2="@ByteArray(ARQ77eY1NUZaQsuDHbIMCA==:0WMRkYTUWVT9wVvdDtHAjU  
9b3b7uB8NR1Gur2hmQCvCDpm39Q+PsJRJPaCU51dEiz+dTzh8qbPsL8WkFljQYFQ==)"
```

Рис. 3.39 Файл конфігурації qbittorrent

Аналогічно до конфігураційного файлу filebrowser (рис. 3.38), qbittorrent також потребує свого (рис. 3.39), інакше він буде працювати лише в режимі командного

рядка, а з веб-браузера повертатиме код 401 – unauthorized, бо очікує авторизацію через заголовки. Найголовніші рядки:

- WebUI\HostHeaderValidation=false – дозволяє авторизуватись через WebUI.
- WebUI\Password_PBKDF2="@ByteArray(ARQ77eY1NUZaQsuDHbIMCA==:0WMRkYTUWVT9wVvdDtHAjU9b3b7uB8NR1Gur2hmQCvCDpm39Q+PsJRJPaCU51dEiz+dTzh8qbPsL8WkFljQYFQ==)" – встановлює стандартний пароль adminadmin

```
cd /srv
sudo docker-compose up -d && sudo docker-compose logs -f
```

Рис. 3.40 Запуск медіа-стеку через docker-compose

Конфігурація в себе також включає процес налаштування Nginx Proxy Manager після розгортання медіа-стеку, процес аналогічний до того, що було зроблено при налаштуванні зовнішніх сервісів. Спочатку отримається сертифікат для доменів home.mahurin.xyz та *.home.mahurin.xyz, це надає змогу перейти до наступного кроку і налаштувати проксі хости з SSL termination.

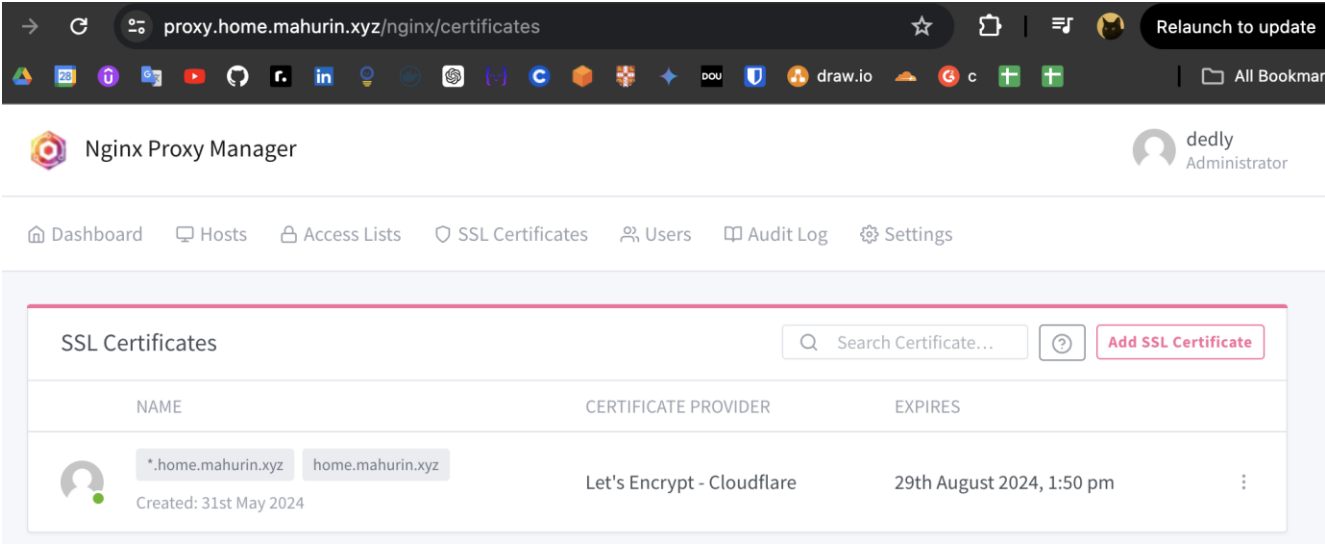


Рис. 3.41 Отриманий сертифікат для внутрішніх сервісів

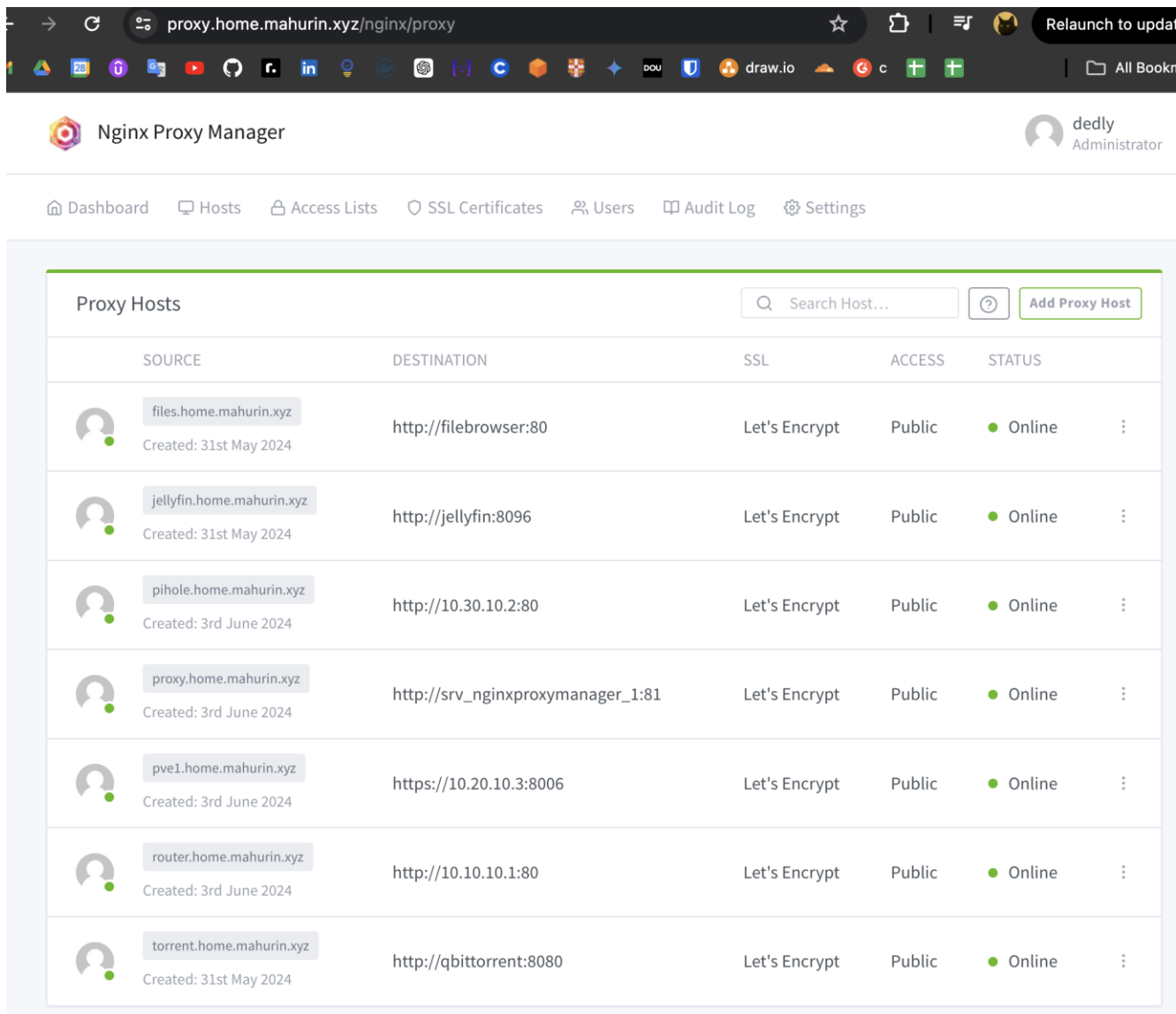


Рис. 3.42 Налаштування проксі хостів внутрішніх сервісів

Загалом, інші сервіси працюють одразу і не потребують додаткового налаштування. Було змінено на всіх сервісах стандартного користувача та пароль заради безпеки. Встановлення Jellyfin потребувало додатково ще вказання бібліотек, де знаходиться медіа, та як йому поводитись з метаданими.

3.3 Огляд та перевірка home lab

Усе готово, тепер треба перевірити працездатність сервісів, як зовнішніх, так і зовнішніх. Почнімо з зовнішніх сервісів, через те, що сервіси взаємопов'язані, можна буде перевірити їх працездатність через роботу інших сервісів. Наприклад, якщо з мережі Інтернет зробити запит у браузері <https://mahurin.xyz> та отримати сторінку, що працює без проблем, то це, по-перше, значить, що WSGI сервер, побудований і запущений як Docker контейнер і написаний завдяки Flask і Python

працює успішно. По-друге, це значить, що nginx проху manager для зовнішніх сервісів також працює справно, дозволяючи зовнішнім клієнтам підключатися через нього та розривати SSL. По-третє, це свідчить що маршрутизатор begyllium налаштовано також правильно, що робить трансляцію NAT і пересилає TCP пакети на порті 443 до nginx проху manager. Нарешті, це також свідчить про те, що clouflare DNS працює справно.

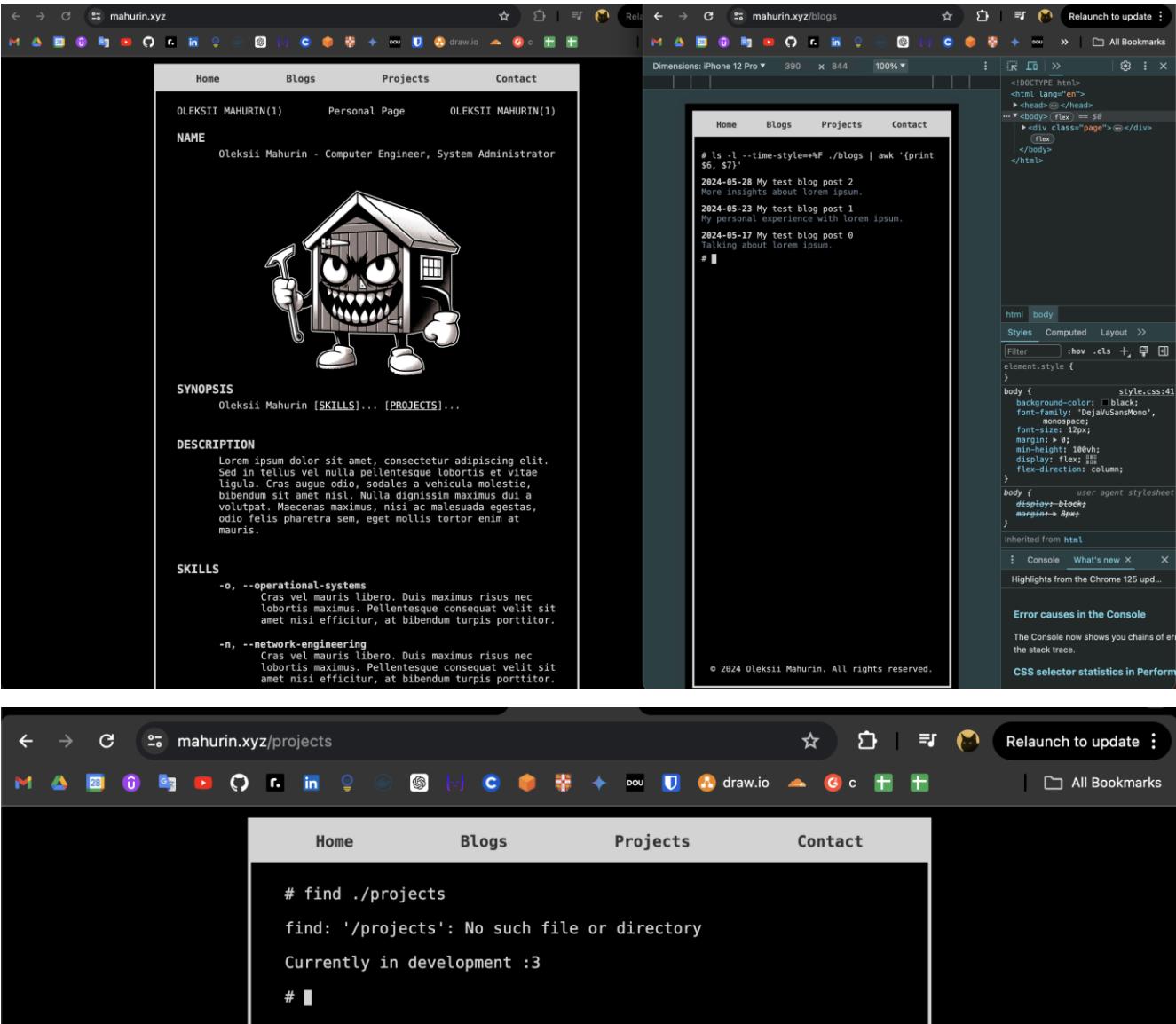


Рис. 3.43 Успішна перевірка веб-серверу

Наступне що треба перевірити у зовнішніх сервісах – це працездатність серверу майнкрафт. Локальні підключення можуть відбуватися через адресу 10.10.10.100 або minecraft.home.mahurin.xyz, ззовні підключається через адресу minecraft.mahurin.xyz.

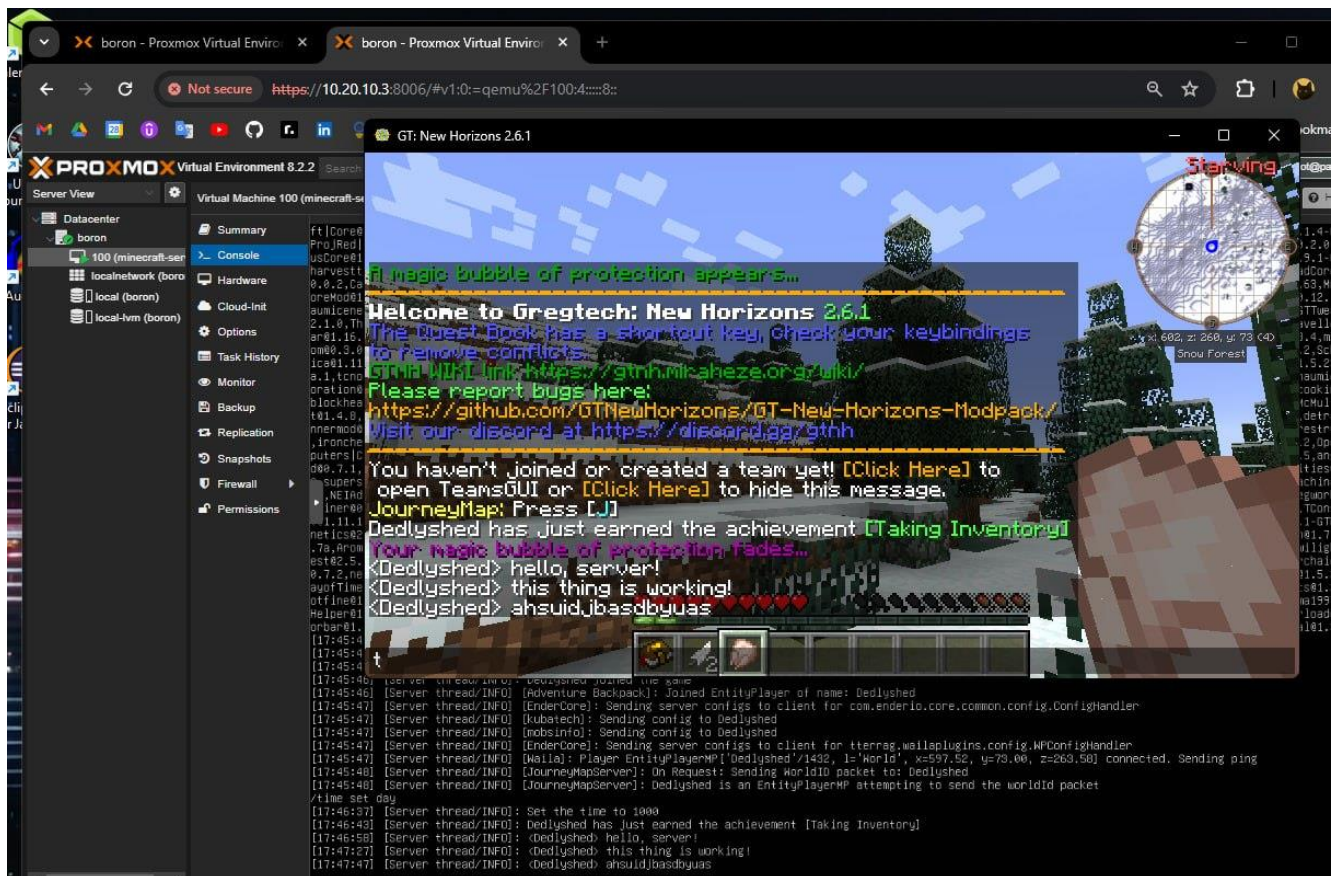


Рис. 3.44 Успішна перевірка майнкрафт серверу

При роботі було знайдено помилку – не працювала веб-консоль у прохтох, якщо був доступ через зворотній проксі pve1.home.mahurin.xyz, через це на рис. 3.44 до прохтох під'єднано через IP-адрес. Рішення – дозволити веб-сокети у налаштуванні nginx проху manager.

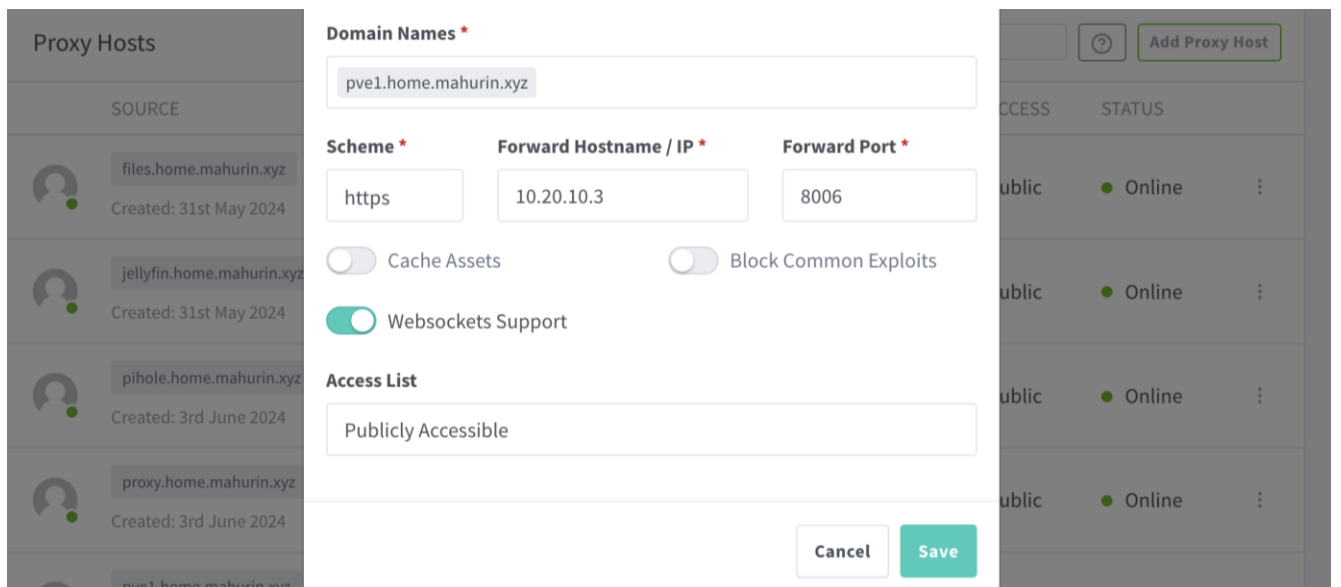


Рис. 3.45 Увімкнення підтримки веб-сокетів

Зовнішні сервіси перевірено, на черзі – внутрішні. Почнімо з PiHole, перевірка доволі проста, просто перевірити, чи блокуються домени за правилами та чи видаються домени з локального DNS-серверу.

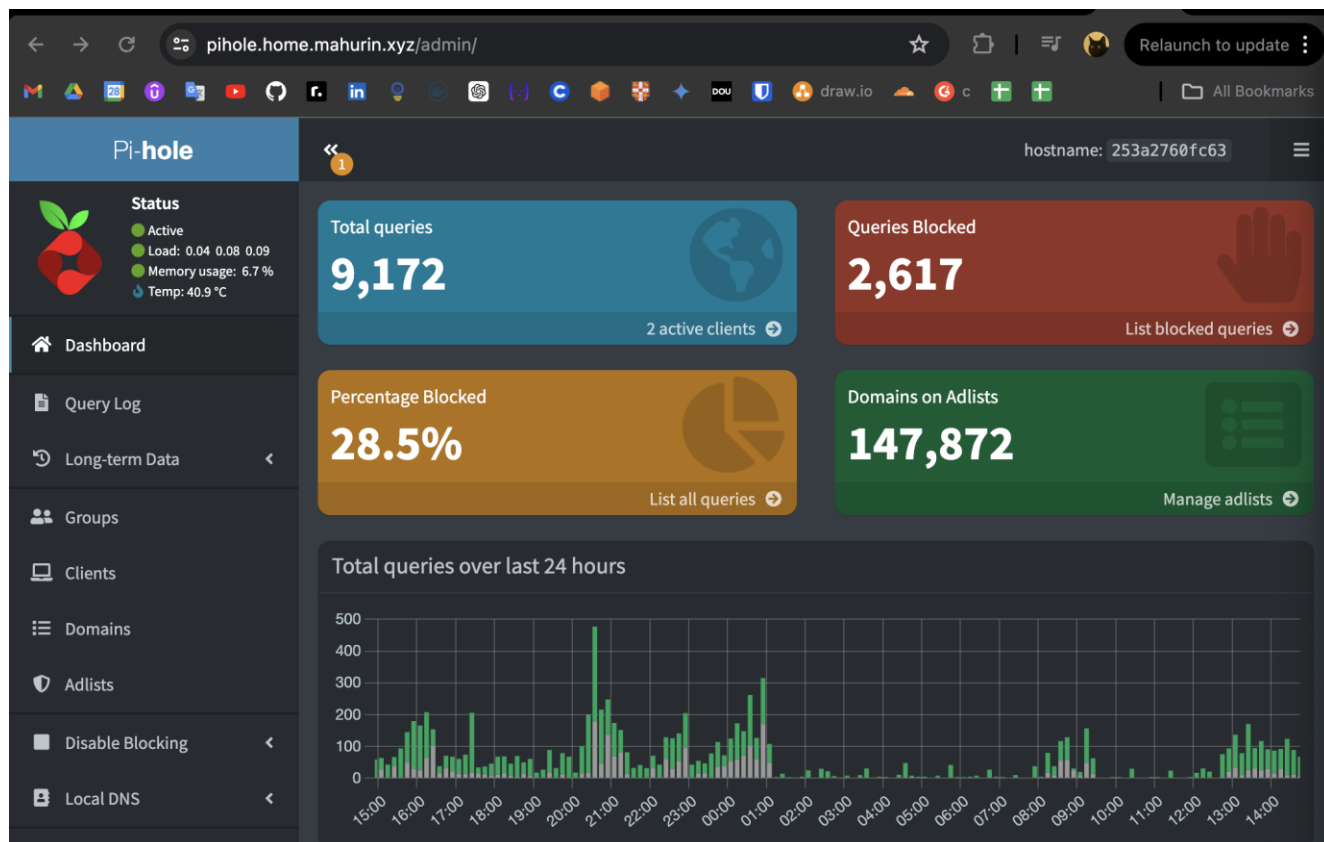


Рис. 3.46 Дашбоард pihole.home.mahurin.xyz

List of domains

✓

Exact whitelist

✓

Regex

Show

10

▼

entries

	Domain/RegEx	Type	Status	Comment
	\.ru	Regex blacklist	Enabled	ruski korabl idi nah

Рис. 3.47 Список правил для блокування небажаного трафіку

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		111

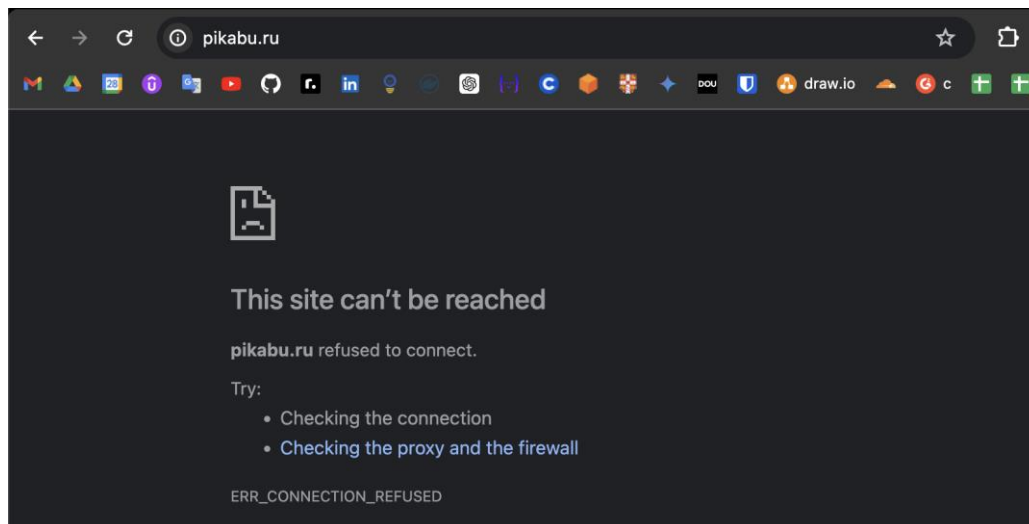


Рис. 3.48 Успішна перевірка правила чорного списку

List of local DNS domains

Show10▼entries

Search:

Domain	IP	Action
beryllium.home.mahurin.xyz	10.10.10.1	
boron.home.mahurin.xyz	10.10.10.3	
helium.home.mahurin.xyz	10.30.10.2	
hydrogen.home.mahurin.xyz	10.10.10.2	
lithium.home.mahurin.xyz	10.20.10.2	

Showing 1 to 5 of 5 entries

List of local CNAME records

Show10▼entries

Search:

Domain	Target	Action
b.home	boron.home.mahurin.xyz	
be.home	beryllium.home.mahurin.xyz	
h.home	hydrogen.home.mahurin.xyz	
he.home	helium.home.mahurin.xyz	
li.home	lithium.home.mahurin.xyz	
nas.home	helium.home.mahurin.xyz	

Рис. 3.49 Локальні DNS записи

```

>> dig he.home

; <<>> DiG 9.10.6 <<>> he.home
;; global options: +cmd
;; Got answer:
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 7890
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;he.home.                IN      A

;; ANSWER SECTION:
he.home.                  0       IN      CNAME   helium.home.mahurin.xyz.
helium.home.mahurin.xyz. 0       IN      A       10.30.10.2

;; Query time: 23 msec
;; SERVER: 10.30.10.1#53(10.30.10.1)
;; WHEN: Mon Jun 10 14:59:29 EEST 2024
;; MSG SIZE rcvd: 78

>> dig helium.home.mahurin.xyz

; <<>> DiG 9.10.6 <<>> helium.home.mahurin.xyz
;; global options: +cmd
;; Got answer:
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 6311
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;helium.home.mahurin.xyz. IN      A

;; ANSWER SECTION:
helium.home.mahurin.xyz. 0       IN      A       10.30.10.2

;; Query time: 6 msec
;; SERVER: 10.30.10.1#53(10.30.10.1)
;; WHEN: Mon Jun 10 14:59:52 EEST 2024
;; MSG SIZE rcvd: 57

```

Рис. 3.50 Перевірка роботи локального DNS серверу

Наступний сервіс для перевірки – VPN. Перевірка зроблена з усіх пристроїв, але продемонстрована у пояснювальній записці з робочого ноутбука. З мережі офіса підключаємо тунель, який було сконфігуровано до цього.

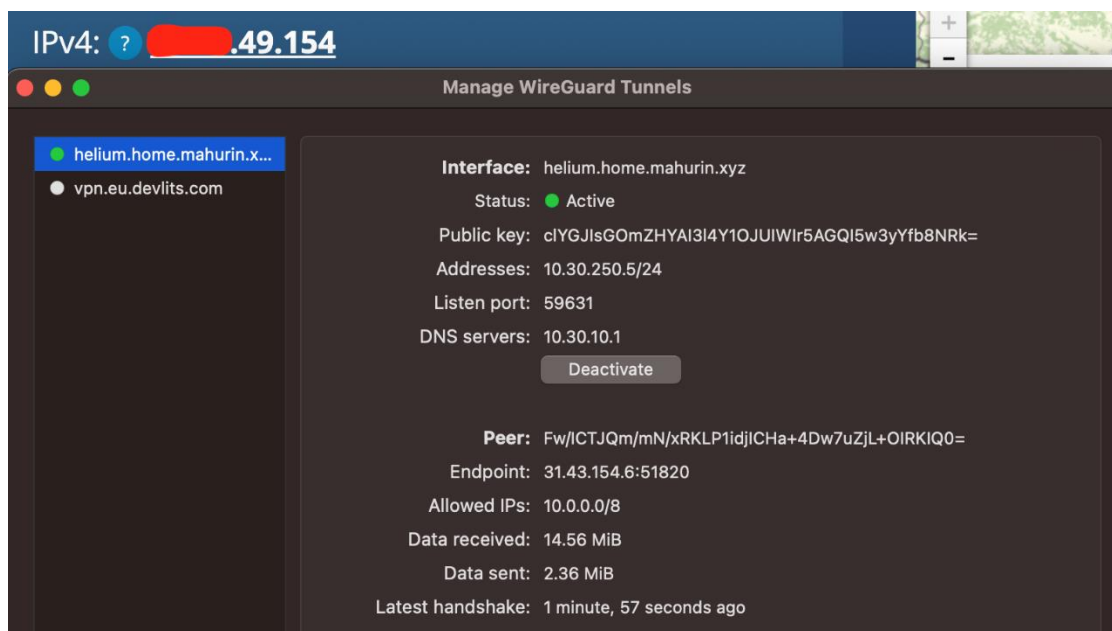


Рис. 3.51 WireGuard VPN тунель

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		113

```

dedly@helium:~$ sudo wg
interface: wg0
  public key: Fw/ICTJQm/mN/xRKLP1idjICHa+4Dw7uZjL+0IRKIQ0=
  private key: (hidden)
  listening port: 51820

peer: cLYGJIsG0mZHYAI3L4Y10JUIWIr5AGQI5w3yYfb8NRk=
  endpoint: [REDACTED].49.154:59631
  allowed ips: 10.30.250.5/32
  latest handshake: 1 minute, 16 seconds ago
  transfer: 3.01 MiB received, 17.19 MiB sent

```

Рис. 3.52 Перевірка WireGuard тунелю з серверу

```

>> ping -c 2 google.com
PING google.com (142.250.186.206): 56 data bytes
64 bytes from 142.250.186.206: icmp_seq=0 ttl=119 time=46.108 ms
64 bytes from 142.250.186.206: icmp_seq=1 ttl=119 time=17.201 ms

--- google.com ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.201/31.654/46.108/14.454 ms
>> ping -c 2 he.home
PING helium.home.mahurin.xyz (10.30.10.2): 56 data bytes
64 bytes from 10.30.10.2: icmp_seq=0 ttl=64 time=34.072 ms
64 bytes from 10.30.10.2: icmp_seq=1 ttl=64 time=47.367 ms

--- helium.home.mahurin.xyz ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 34.072/40.719/47.367/6.648 ms
>> ping -c 2 10.30.10.2
PING 10.30.10.2 (10.30.10.2): 56 data bytes
64 bytes from 10.30.10.2: icmp_seq=0 ttl=64 time=13.735 ms
64 bytes from 10.30.10.2: icmp_seq=1 ttl=64 time=26.205 ms

--- 10.30.10.2 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 13.735/19.970/26.205/6.235 ms

```

Рис. 3.53 Перевірка зв'язку через VPN тунель

Далі перевірено роботу мережевого сховища. Це необхідно перевірити на своїх платформах. Також, треба впевнитись, що користувачі, що були створені для цього мережевого сховища мають відповідні права. Спочатку, було перевірено працездатність на WSL2, що фактично є віртуальною машиною Ubuntu всередині ОС Windows. Для перевірки вибрано користувача nas-guest. Він має доступ лише на читання до PublicShare. Спочатку було невдала спроба підключення до DedlyShare, у цього користувача навіть немає правил. Потім йде успішна спроба підключення користувачем nas-guest, до Public-Share, що є успішним. Була спроба зробити файл, але система не дозволила, отже, всі правила працюють правильно.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		114

```

dedly@lithium:~$ sudo mkdir -p /mnt/samba
dedly@lithium:~$ sudo mount -t cifs //10.30.10.2/DedlyShare /mnt/samba -o username=nas-guest,password=
"g&AF27b5&2EgF6",vers=3.0
mount: /mnt/samba: cannot mount //10.30.10.2/DedlyShare read-only.
dedly@lithium:~$ sudo mount -t cifs //10.30.10.2/PublicShare /mnt/samba -o username=nas-guest,password=
"g&AF27b5&2EgF6",vers=3.0
dedly@lithium:~$ cd /mnt/samba/
dedly@lithium:/mnt/samba$ ls
'Dune (2021) [1080p] [BluRay] [5.1] [YTS.MX]'      tmp
'Oppenheimer (2023) [1080p] [BluRay] [5.1] [YTS.MX]'
dedly@lithium:/mnt/samba$ cat tmp
hello, world!
dedly@lithium:/mnt/samba$ touch myfile
touch: cannot touch 'myfile': Permission denied
dedly@lithium:/mnt/samba$

```

Рис. 3.54 Перевірка підключення samba share на ОС Linux

Далі, була здійснена перевірка з ОС Windows для користувача nas-user Також, було перевірено швидкість з'єднання, в результаті отримано 100 Мб в секунду, що є дуже хорошим результатом для такого простого варіанту на Raspberry Pi.

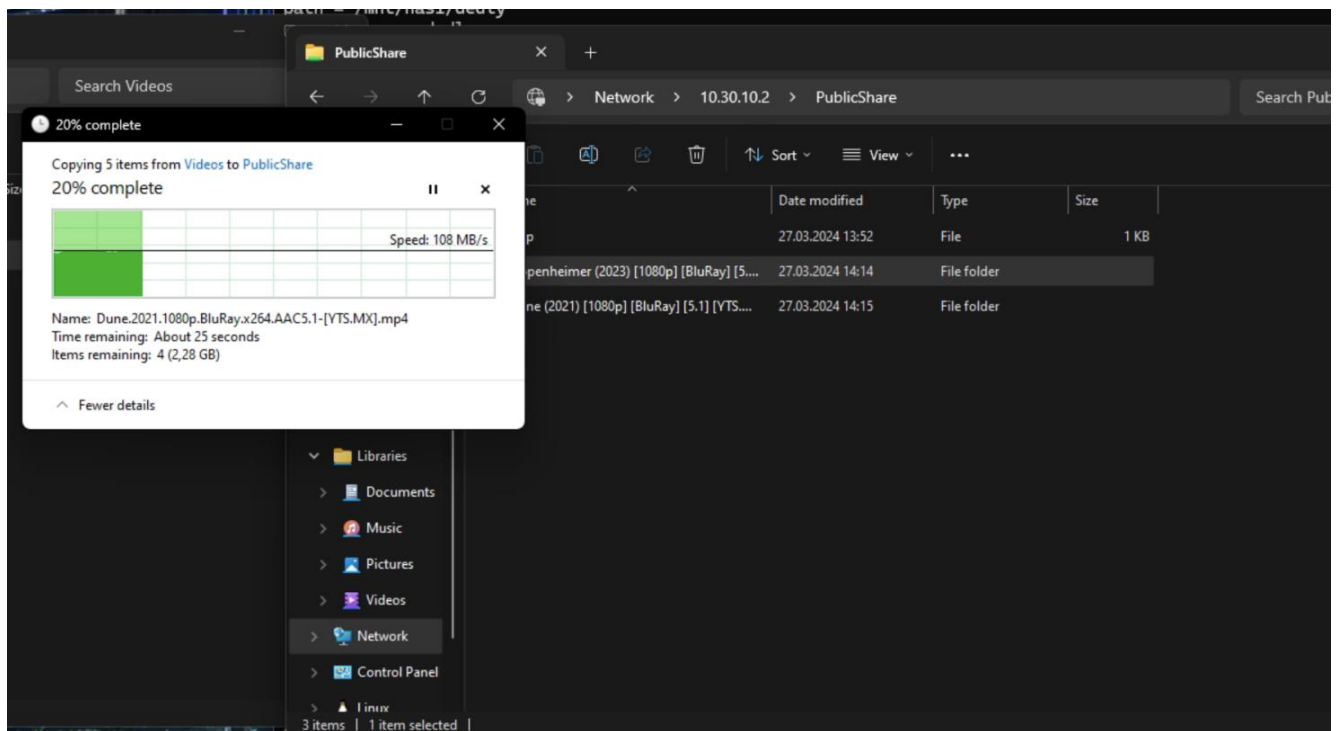


Рис. 3.55 Перевірка підключення samba share на ОС Windows

Перевірку з MacOS було зроблено з робочого ноутбуку через VPN через користувача dedly. Швидкість стала на рівні 4-5 Мб в секунду, що значно менше, але тут фактором зниження швидкості може бути інша ОС, протокол SMB розрахований більше для Windows, з'єднання відбулось через VPN та ноутбук підключений до Wi-Fi точки. Хоча це набагато менше, це все одно дуже непоганий результат.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				115
Змн.	Арк.	№ докум.	Підпис	Дата		

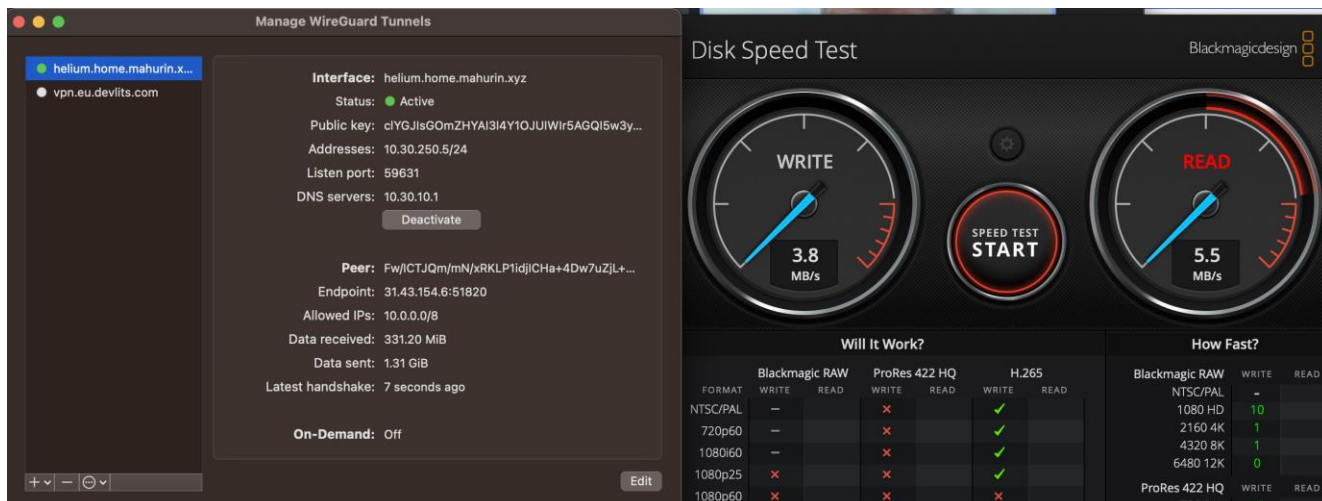


Рис. 3.56 Перевірка підключення samba share на MacOS через VPN

Наступне – це перевірка медіа-стеку. Для того, щоб перевірити, відтворимо процес додання фільму на реальному прикладі. Спочатку завантажимо торент-файл фільму. Буде завантажено фільм, що знаходиться у публічному домені - Steamboat Willie (1928).

Спочатку необхідно завантажити торент на робочу станцію, це можна зробити тут: <https://archive.org/details/steamboat-willie-mickey>. Далі, у веб-інтерфейсі <https://torrent.home.mahurin.xyz> створюємо торент, завантаживши цей файл відповідно через веб-інтерфейс.

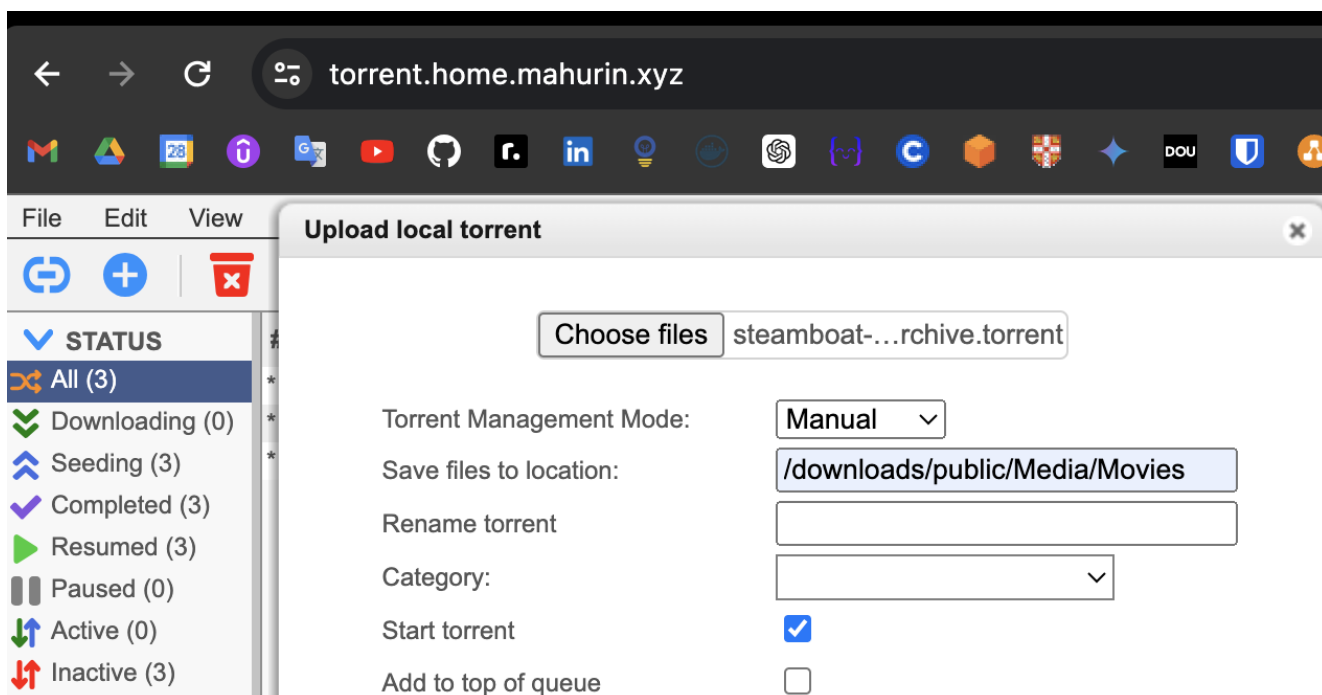


Рис. 3.57 Створення торенту

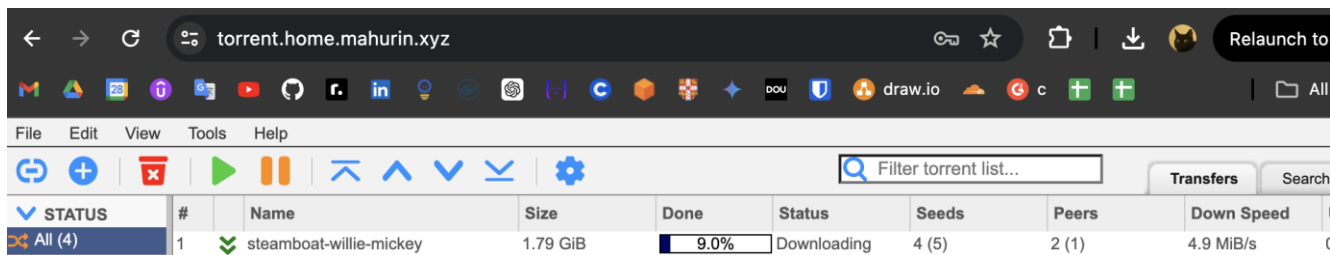


Рис. 3.58 Завантаження торенту

Після завантаження фільму, можна перевірити роботу filebrowser. У <https://files.home.mahurin.xyz> можна побачити завантажений файл та редагувати текстові файли, завантажувати та вивантажувати файли, переміщати та копіювати.

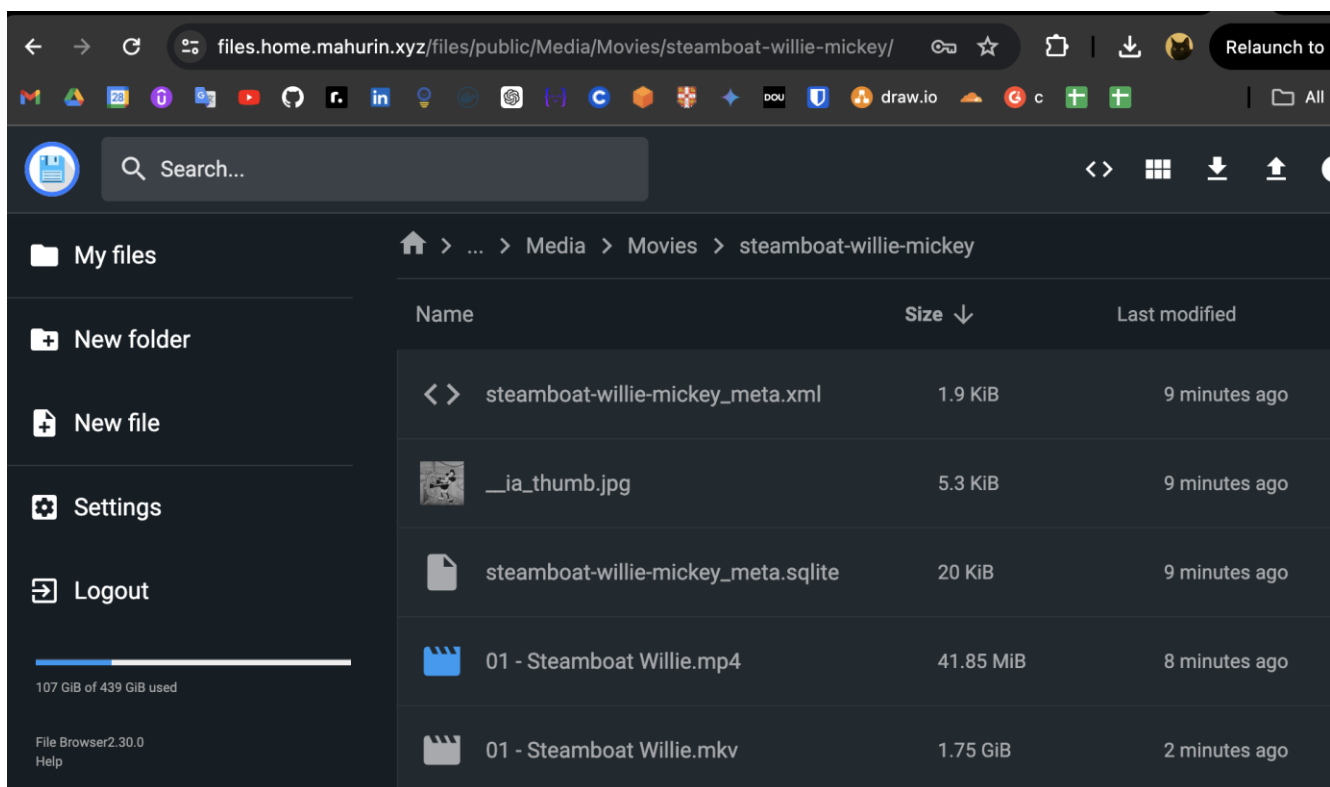


Рис. 3.59 Перегляд завантаженого торенту через filebrowser

Далі, можна переходити до медіа-центру jellyfin. На головній сторінці відображатиметься дашборд з доступними бібліотеками, нещодавно доданим медіа, тощо. У категорії фільми вже додано фільм і можна перевіряти роботу. Фільми відтворюються без проблем, підтримка full hd контенту без затримок, але необхідно було вимкнути апаратну обробку відео, оскільки Raspberry Pi не має відеокarti для обробки, а цей неоптимізований процес навантажуватиме процесор на всі 100 відсотків.

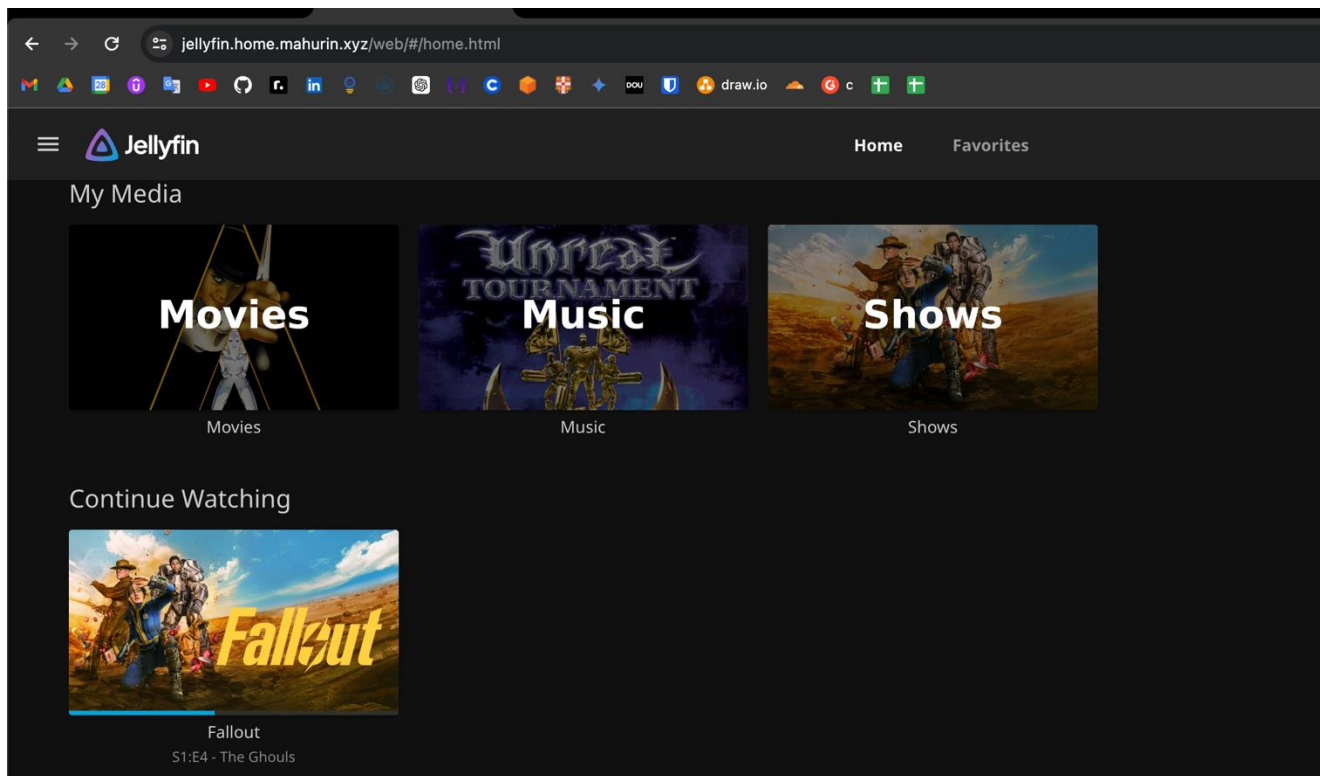


Рис. 3.60 Головна сторінка jellyfin

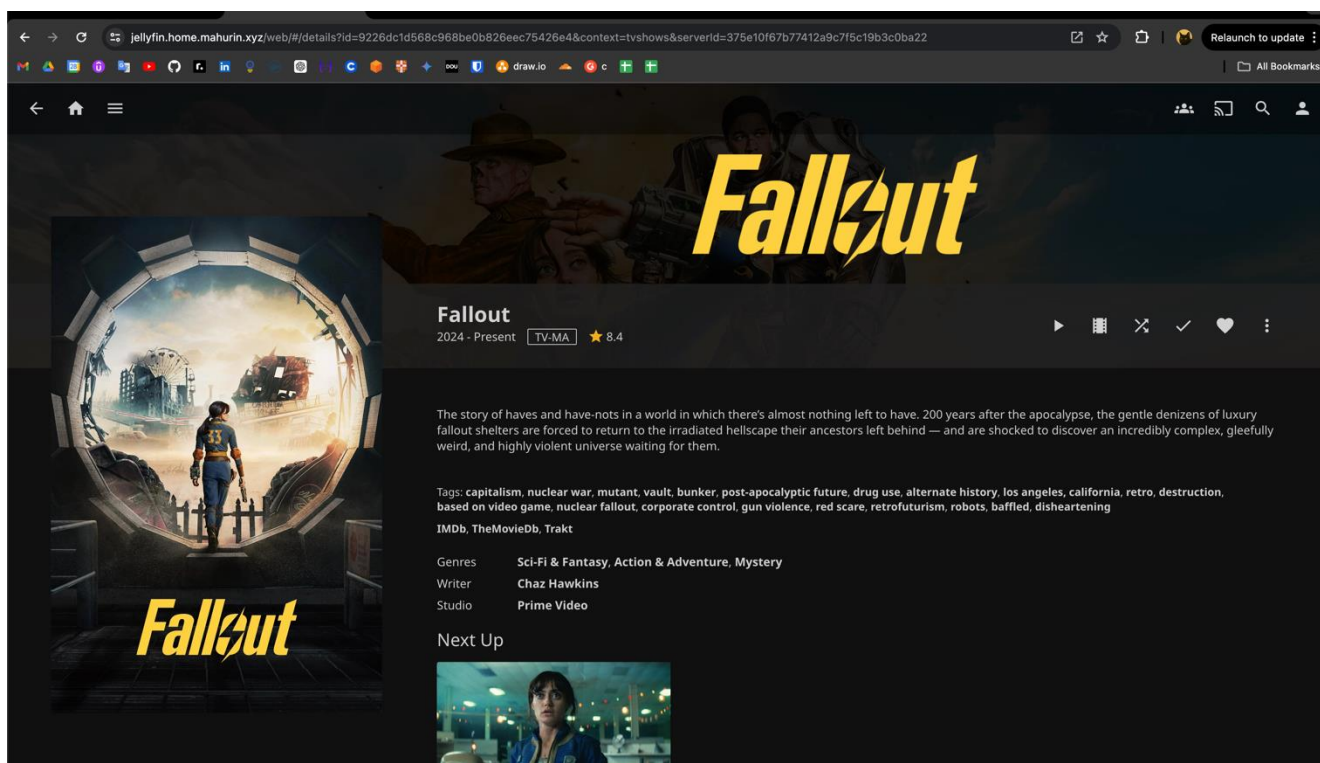


Рис. 3.61 Сторінка фільму jellyfin

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				118
Змн.	Арк.	№ докум.	Підпис	Дата		

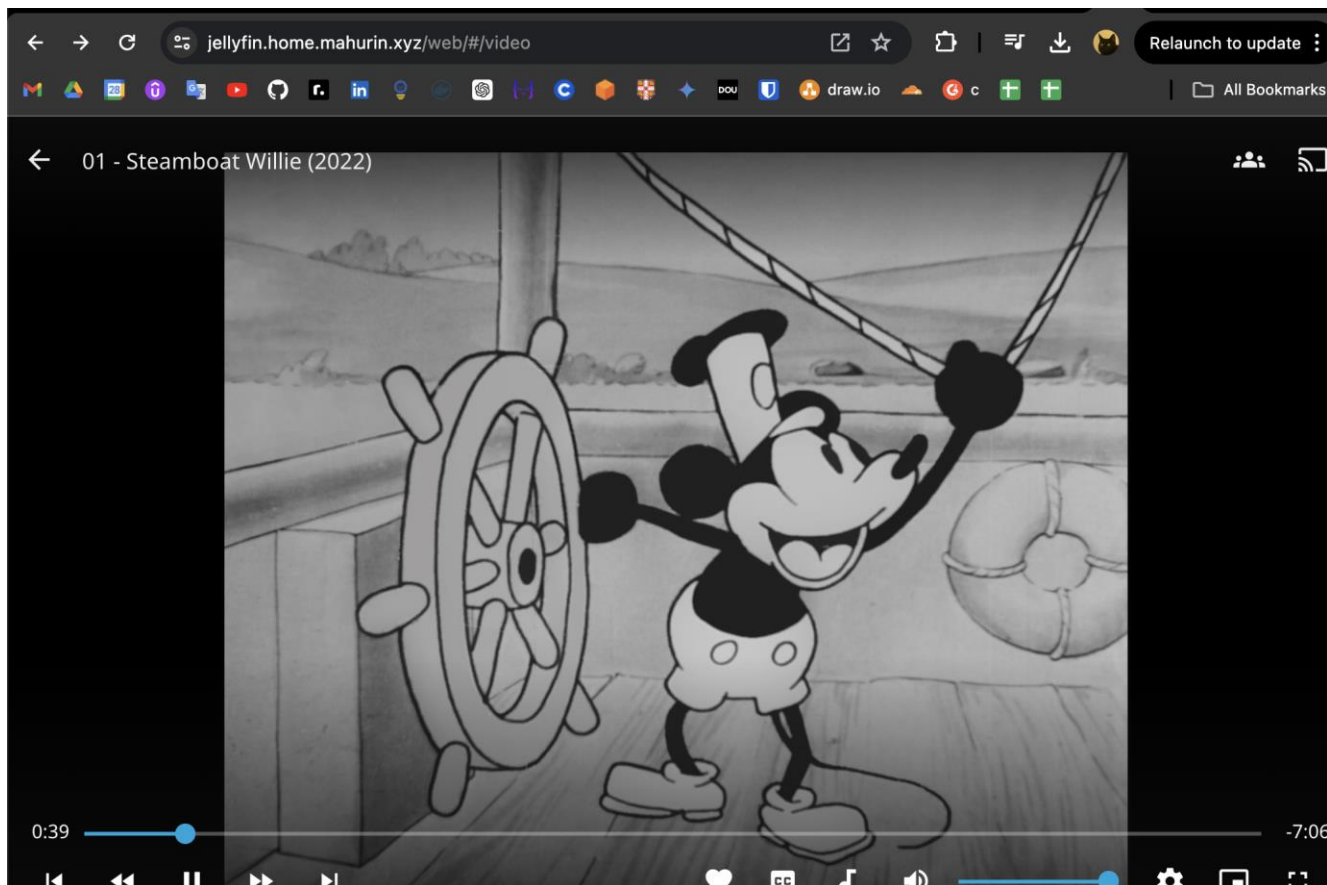


Рис. 3.62 Відеоплеєр jellyfin

В и с н о в к и д о р о з д і л у

У третьому розділі детально описано процес конфігурації та впровадження зовнішніх та внутрішніх сервісів у homelab. Реєстрація домену через Cloudflare, налаштування Nginx Proxy Manager та розробка персональної веб-сторінки стали ключовими кроками у формуванні зовнішньої інфраструктури. Додатково, було встановлено і налаштовано Proxmox для управління віртуальними машинами, включно з сервером Minecraft. Внутрішні сервіси, такі як PiHole, SMB share, і VPN Wireguard, були інтегровані для підвищення функціональності та безпеки системи. Також було здійснено розгортання медіа-стеку, включаючи Jellyfin, qBittorrent, та Filebrowser, що забезпечило ефективне управління медіа ресурсами. Розділ завершується оглядом і перевіркою налаштованої системи, підтверджуючи її готовність до експлуатації.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		119

ВИСНОВКИ

У кваліфакаційній роботі було досліджено концепцію homelab та різні підходи до розгортання IT-сервісів, включаючи SaaS, самотійний хостинг, хмарні та локальні налаштування. Кожен метод був оцінений з точки зору його переваг та обмежень, зазначаючи, що хоча SaaS і хмарні рішення пропонують простоту і масштабованість, вони обмежують можливості кастомізації і створюють ризики для безпеки даних, в той час як самотійне розміщення і локальні рішення забезпечують більший контроль, але вимагають значних інвестицій в ресурси.

У проекті було детально описано важливі кроки, пов'язані з розробкою надійної мережевої інфраструктури, необхідної для роботи homelab. Було підкреслено необхідність добре спланованої мережі не лише як допоміжної структури, але й як основи, що забезпечує ефективність і безпеку всієї системи. Особлива увага була приділена конфігурації як фізичних, так і логічних аспектів мережі для задоволення різноманітних потреб у послугах, доповнена детальними інструкціями RouterOS для покращення навичок управління мережею.

Було детально розглянуто конфігурацію та розгортання як зовнішніх, так і внутрішніх служб. Це включало налаштування зовнішніх служб, таких як реєстрація доменів і Nginx Proxy Manager, а також внутрішніх служб, таких як PiHole для управління DNS, SMB для рішень для зберігання даних і Wireguard VPN для безпечного з'єднання. Крім того, було обговорено розгортання медіа-стеку з Jellyfin, qBittorrent та Filebrowser для покращення управління медіа-ресурсами. Всебічний огляд і перевірка цих налаштувань підтвердили готовність homelab до роботи, довівши корисність проекту в поєднанні теоретичних знань з практичним застосуванням у мережевому та системному адмініструванні.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		120

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Helder. What is a homelab and why should you have one? Linux Handbook. URL: <https://linuxhandbook.com/homelab/> (date of access: 02.02.2024).
2. r/homelab Wiki introduction. www.reddit.com. URL: <https://www.reddit.com/r/homelab/wiki/introduction/> (date of access: 14.02.2024).
3. Join the homelab discord server. Discord. URL: <https://discord.com/invite/homelab> (date of access: 14.02.2024).
4. Join the noted discord server. Discord. URL: <https://discord.com/invite/zA9FT8qW9g> (date of access: 14.02.2024).
5. IaaS vs. paas vs. saas. Red Hat - We make open source technologies for the enterprise. URL: <https://www.redhat.com/en/topics/cloud-computing/iaas-vs-paas-vs-saas> (date of access: 21.02.2024).
6. Інтернет провайдер Corbina Telecom. URL: <https://corbina.ua/uk> (дата звернення: 07.03.2024).
7. RouterOS - RouterOS - MikroTik Documentation. MikroTik Routers and Wireless - Support. URL: <https://help.mikrotik.com/docs/display/ROS/RouterOS> (date of access: 17.03.2024).
8. What is DNS? | How DNS works. Connect, Protect and Build Everywhere | Cloudflare. URL: <http://www.cloudflare.com/learning/dns/what-is-dns> (date of access: 03.04.2024).
9. What is a top-level domain?. Connect, Protect and Build Everywhere | Cloudflare. URL: <http://www.cloudflare.com/learning/dns/top-level-domain/#:~:text=In%20the%20DNS%20hierar-chy,%20a,.com'%20is%20the%20TLD.> (date of access: 03.04.2024).
10. Contributors to Wikimedia projects. Proxy server - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Proxy_server (date of access: 10.04.2024).
11. Nginx Proxy Manager. URL: <https://nginxproxymanager.com/> (date of access: 10.04.2024).

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		121

12. What is a web server? - Learn web development | MDN. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server (date of access: 10.04.2024).
13. Welcome to flask – flask documentation (3.0.x). Welcome to Flask – Flask Documentation (3.0.x). URL: <https://flask.palletsprojects.com/en/3.0.x/> (date of access: 17.04.2024).
14. Home. Docker Documentation. URL: <https://docs.docker.com/> (date of access: 22.04.2024).
15. Proxmox virtual environment. Proxmox. URL: <https://www.proxmox.com/en/proxmox-virtual-environment/overview> (date of access: 25.04.2024).
16. WireGuard: fast, modern, secure VPN tunnel. URL: <https://www.wireguard.com/> (date of access: 02.05.2024).
17. The Free Software Media System | Jellyfin. URL: <https://jellyfin.org/> (date of access: 04.05.2024).
18. qBittorrent Official Website. URL: <https://www.qbittorrent.org/> (date of access: 05.05.2024).
19. Welcome | File Browser. URL: <https://filebrowser.org/> (date of access: 06.05.2024).
20. UNIX and linux system administration handbook (5th edition) / E. Nemeth et al. Addison-Wesley Professional, 2017. 1232 p.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				122
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				123
Змн.	Арк.	№ докум.	Підпис	Дата		

Порівняння виробників мережевого обладнання

- **TP-Link:** TP-Link є добре встановленим брендом в індустрії мережевого обладнання і пропонує широкий спектр бездротових маршрутизаторів. Їхні маршрутизатори відомі своєю доступністю та надійністю. TP-Link пропонує різні варіанти з різними функціями, такими як передові бездротові технології (Wi-Fi 6 та 6E), кілька діапазонів, підтримка MU-MIMO та сумісність з сітчастою мережею. Їхні маршрутизатори часто мають зручні інтерфейси та додатки для легкого управління та налаштування.

- **ASUS:** ASUS славиться своїм високопродуктивним мережевим обладнанням та маршрутизаторами. Вони пропонують широкий вибір бездротових маршрутизаторів, розроблених для різних потреб, включаючи ігри, стрімінг та бізнес-використання. Маршрутизатори ASUS містять передові технології, такі як Wi-Fi 6, сумісність AiMesh для створення сітчастих мереж та розширені опції налаштування через їхнє прошивання та програмне забезпечення. Їх цінують ентузіасти за їх продуктивність та можливості налаштування.

- **Netgear:** Netgear відомий своїми надійними та високопродуктивними бездротовими маршрутизаторами. Вони пропонують різні моделі, які підходять для різних потреб мережі, від базового використання вдома до розширених бізнес-рішень. Маршрутизатори Netgear часто містять функції, такі як Wi-Fi 6, бімформінг та підтримка MU-MIMO. Вони також пропонують зручне програмне забезпечення та додатки для управління та налаштування.

- **Ubiquiti:** Ubiquiti пропонує широкий спектр мережевих рішень, включаючи високопродуктивні та надійні бездротові маршрутизатори. Їх продукція популярна серед професіоналів мережевого обладнання через їх високу продуктивність та передові функції. Маршрутизатори Ubiquiti відомі своєю екосистемою UniFi, яка дозволяє безперервну інтеграцію з іншими мережевими пристроями та забезпечує централізоване управління мережами.

- **MikroTik:** Маршрутизатори MikroTik відомі своїми передовими функціями та широкими можливостями налаштування. Їх маршрутизатори пропонують

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		124

продуктивність промислових пристроїв за доступною ціною, що робить їх популярним вибором серед експертів мережевого обладнання. Маршрутизатори MikroTik часто містять RouterOS, потужну операційну систему, яка дозволяє глибоке управління мережею та налаштування.

- Linksys: Linksys - це надійний бренд, який пропонує різноманітні бездротові маршрутизатори, які балансують продуктивність, надійність та доступність. Їх маршрутизатори задовольняють різні потреби, від домашнього до бізнес-використання, і забезпечують передові функції, такі як сумісність з сітчастою мережею, Wi-Fi 6 та батьківський контроль. Маршрутизатори Linksys часто містять зручне програмне забезпечення для швидкого налаштування та моніторингу.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				125
Змн.	Арк.	№ докум.	Підпис	Дата		

Опис правил фаєрволу

- Rule 0: Special Dummy Rule - це правило просто потрібне для збору статистики мережі.
- Rule 1: Accept Established, Related, Untracked Connections - це правило дозволяє трафіку, який є частиною існуючого з'єднання (встановленого), пов'язаного з існуючим з'єднанням (пов'язаним) або ще не відстеженим системою відстеження з'єднань (невідстеженим), забезпечуючи допуск неперервних та законних з'єднань.
- Rule 2: Drop Invalid Connections - трафік, ідентифікований як маючий неправильний стан з'єднання, відкидається цим правилом, запобігаючи потенційно шкідливим або невірно сформованим пакетам увійти в мережу.
- Rule 3: Accept ICMP - дозволяє трафік ICMP (Internet Control Message Protocol), що використовується для мережевої діагностики та управління.
- Rule 4: Accept Traffic to Local Loopback - це правило дозволяє трафіку, призначеному для локальної адреси петлі (127.0.0.1), сприяючи комунікації між службами, які працюють на самому маршрутизаторі без обмежень.
- Rule 5: Drop Traffic Not Coming from LAN - це правило відкидає трафік, що виникає з інтерфейсів, які не вказані у списку інтерфейсів LAN, підвищуючи безпеку шляхом запобігання несанкціонованому доступу до маршрутизатора зовнішніми мережами.
- Rule 6: Accept Traffic Matching IPsec Policies - це правило дозволяє трафіку, який відповідає політикам IPsec для вхідного трафіку, полегшуючи безпечну комунікацію через з'єднання VPN IPsec.
- Rule 7: Accept Traffic Matching IPsec Policies - схоже на правило 6, це правило застосовується до вихідного трафіку, забезпечуючи двонаправлену безпечну комунікацію через з'єднання VPN IPsec.
- Rule 8: Fasttrack Established, Related Connections - це правило оптимізує обробку встановлених та пов'язаних з'єднань, вивантажуючи їх на апаратне

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				126
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечення, якщо підтримується, для покращення пропускної здатності та зменшення навантаження процесора.

- Rule 9: Accept Established, Related, Untracked Connections - це дублювання правила 1, це правило забезпечує послідовну обробку неперервних, пов'язаних та невідстежених з'єднань для трафіку, що пересилається.
- Rule 10: Drop Invalid Connections - схоже на правило 2, це правило забезпечує послідовну обробку неправильних з'єднань для пересиланого трафіку, запобігаючи потенційно шкідливим пакетам увійти в мережу.
- Rule 11: Drop Traffic from WAN Not Destination NATed - це правило відкидає трафік, що виникає з інтерфейсу WAN, який не пройшов NAT (Network Address Translation) для призначення, підвищуючи безпеку шляхом запобігання непроханому трафіку входу в внутрішню мережу без відповідних перекладів NAT.

		Магурін О.О.			ККІКБ.КР.Б – 123 – 13 – ПЗ	Арк.
		Головня О.С.				
Змн.	Арк.	№ докум.	Підпис	Дата		127