

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Хмарний сервер для зберігання даних на основі Raspberry PI

Назва теми

КвРКІ 210119.21.01.73 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Виконав: студент IV курсу, група KI2-21-1


Підпис

Данило ПОПЕЛІНОВ

Ініціали, прізвище


Керівник


Підпис, дата

В'ячеслав АСКЕРОВ

Ініціали, прізвище

Нормоконтролер


Підпис, дата

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

« 9 » червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

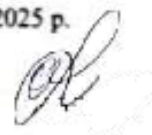
Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.



ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Попелінову Данилу Дмитровичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Хмарний сервер для зберігання даних на основі Raspberry Pi

Керівник проекту (роботи) Аскеров В'ячеслав Васильович, асистент

Прізвище, ім'я, по батькові, науковий ступінь, місце завдання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Хмарний сервер для зберігання даних на основі Raspberry Pi дослідження предметної області і постановка задачі

Проектування системи хмарного сервера для зберігання даних на основі Raspberry Pi

Програмно-апаратна реалізація хмарного сервера для зберігання даних на основі Raspberry Pi

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Блок-схема алгоритм роботи Nextcloud на Raspberry Pi

Блок-схеми архітектури, робота хмарного сервера

Інтерфейс Nextcloud

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – проектування структури хмарного сервера для зберігання даних на основі Raspberry PI	01.04.2025	виконано
5	Робота над розділом 3 – програмно-апаратна реалізація хмарного сервера для зберігання даних на основі Raspberry PI	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Керівник роботи

Підпис Данило ПОПЕЛІНОВ
 Ініціали, прізвище

Підпис В'ячеслав АСКЕРОВ
 Ініціали, прізвище

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л - л и с т і в	№ е к з	П р и м і т к а
			<u>Текстові документи</u>			
1		КвРКІ 210119.21.01.73 ПЗ	Пояснювальна записка	61		
			<u>Графічні матеріали</u>			
2		КвРКІ 210119.21.01.73 Е8	Блок схема алгоритм роботи Nextcloud на Raspberry Pi	1		
3		КвРКІ 210119.21.01.73 Е8	Блок схеми архітектури, робота хмарного сервера	1		
4		КвРКІ 210119.21.01.73 Е8	Інтерфейс Nextcloud	1		

					КвРКІ 210119.21.01.73 ВП			
Зм	Арк	№ докум	Підпис	Дат а	Відомість проекту	Літера	Аркуш	Аркушів
Розробл в		Писанков	<i>[Signature]</i>	05.06.20		У	1	1
Перевір		Аскеров	<i>[Signature]</i>	05.06.20		ХНУ, КІ2-22-1		
Н.контр. Затв.		Кисіль Павлова	<i>[Signature]</i>	05.06.20				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Хмарний сервер для зберігання даних на основі Raspberry Pi».

Автор роботи: Данило ПОПЕЛІНОВ.

Керівник роботи: В'ячеслав АСКЕРОВ.

Пояснювальна записка: 61 с., 22 рис., 6 табл., 3 дод., 49 джерел.

Графічна частина: 3 креслення.

Raspberry Pi, ЗБЕРІГАННЯ ДАНИХ, ХМАРНИЙ СЕРВЕР, БАЗА ДАНИХ, АРХІТЕКТУРА.

Метою дипломної роботи є розробка та впровадження ефективного, економічного та безпечного хмарного сервера для зберігання даних з використанням апаратної платформи Raspberry Pi, що забезпечує доступність, масштабованість та зручність керування даними у локальній або глобальній мережі.

Об'єктом дослідження є процес організації хмарного середовища для зберігання, обробки та передачі даних.

Предметом дослідження є архітектура, програмне забезпечення та технічні рішення для реалізації хмарного серверу зберігання даних на базі одноплатного комп'ютера Raspberry Pi.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.



Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП.....	3
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ.....	4
1.1. Аналіз предметної області і виявлення наявних проблем і завдань	4
1.2 Порівняльний аналіз переваг та недоліків існуючих рішень	9
1.3 Підходи до вирішення задачі за темою дослідження.....	14
1.4 Постановка задачі	16
1.5 Висновки до першого розділу	18
2 ЗАСОБИ ДЛЯ РОЗРОБКИ ХМАРНОГО СЕРЕДОВИЩА.....	19
2.1 Хмарний сервер	19
2.2 Raspberry PI	21
2.3 Raspberry OS	26
2.4 Технологія SSH.....	29
2.5 Apache, PHP, Maria DB	31
2.6 Nextcloud	38
2.7 Висновки до другого розділу	41
3 ВСТАНОВЛЕННЯ НЕОБХІДНИХ КОМПОНЕНТІВ	42
3.1 Вибір та підготовка обладнання.....	42
3.2 Встановлення Raspberry OS	43
3.3 Ввімкнення SSH.....	50
3.4 Встановлення Apache, PHP та Maria DB.....	52
3.5 Встановлення Nextcloud.....	55
3.6. Висновки до третього розділу	60
ВИСНОВКИ	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	64
ДОДАТОК А	69
ДОДАТОК Б	70
ДОДАТОК В	71

КвРКІ.210119.21.01.73 ПЗ				
Зм.	Арк.	№докум.	Підпис	Дата
Виконав		Віктор ПОПЕЛІШ	<i>[Підпис]</i>	06.02.2021
Перевірив		Геннадій АСКЕРОВ	<i>[Підпис]</i>	06.02.21
Н.контр.		Гетяна КИСІЛЬ	<i>[Підпис]</i>	06.02.21
Відвер.		Ольга ПАВЛОВА	<i>[Підпис]</i>	06.02.21
Хмарний сервер для зберігання даних на основі Raspberry PI. Пояснювальна записка			Літера	Арк.ш
			у	2
			Арк.шів	63
ХНУ КІ2-22-1				

ВСТУП

Сучасні інформаційні технології стрімко розвиваються, що зумовлює зростаючу потребу в ефективних, доступних та безпечних системах зберігання і обробки даних. Хмарні технології стали важливою складовою цифрової інфраструктури як для приватних користувачів, так і для комерційних та державних установ. Завдяки можливості централізованого зберігання даних, віддаленого доступу та високої масштабованості, хмарні рішення забезпечують новий рівень ефективності в управлінні інформаційними ресурсами.

Однак традиційні хмарні сервіси, що надаються великими компаніями, мають низку недоліків: високу вартість, залежність від сторонніх постачальників, обмеження у сфері конфіденційності та потребу в постійному підключенні до Інтернету. У цьому контексті актуальним стає пошук альтернативних, економічно доцільних рішень для реалізації персональних або локальних хмарних серверів, які б забезпечували користувача повним контролем над власними даними.

Метою дипломної роботи є розробка та впровадження ефективного, економічного та безпечного хмарного сервера для зберігання даних з використанням апаратної платформи Raspberry Pi, що забезпечує доступність, масштабованість та зручність керування даними у локальній або глобальній мережі.

Об'єктом дослідження є процес організації хмарного середовища для зберігання, обробки та передачі даних

Предметом дослідження є архітектура, програмне забезпечення та технічні рішення для реалізації хмарного серверу зберігання даних на базі однопалатного комп'ютера Raspberry Pi.

					КВРКІ.210119.21.01.73 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1. Аналіз предметної області і виявлення наявних проблем і завдань

Хмарні технології як концепція почали формуватись на межі 1990-х – 2000-х років, коли виникла необхідність масштабного обчислення та зберігання даних без жорсткої прив'язки до конкретного пристрою. Ідея централізації обчислювальних ресурсів спочатку була реалізована у вигляді дата-центрів та віддалених серверів великих компаній, а згодом трансформувалася у модель "інфраструктура як послуга" (IaaS), "платформа як послуга" (PaaS) та "програмне забезпечення як послуга" (SaaS) [21].

Актуальності хмарні рішення набувають у сферах освіти, охорони здоров'я, малого бізнесу, а також у побуті. Наприклад, навчальні заклади використовують хмарні платформи для дистанційного навчання, обміну навчальними матеріалами, організації електронного документообігу [24]. У сфері малого підприємництва хмарні сховища дозволяють знизити витрати на ІТ-інфраструктуру, забезпечити спільний доступ до файлів, а також резервне копіювання без придбання дорогого обладнання. Для пересічного користувача - це можливість зберігати фотографії, документи, відео та інші файли в єдиному цифровому середовищі.

У контексті зростаючих загроз інформаційній безпеці тема створення автономного хмарного середовища стає ще більш актуальною. У наш час все частіше виникають випадки кібератак, масових витоків персональних даних, блокування чи цензурування контенту з боку провайдерів сервісів. За таких умов залежність користувачів від сторонніх платформ несе в собі значні ризики. Наявність власного хмарного сховища дає змогу забезпечити повний контроль над інформацією, гнучкість у налаштуваннях доступу, а також мінімізувати можливість несанкціонованого втручання ззовні [14].

Важливу роль такі рішення відіграють у державному та освітньому секторах. Для невеликих установ, сільських шкіл, бібліотек, місцевих адміністрацій персональний сервер на Raspberry Pi може слугувати економічним варіантом

					КВРКІ.210119.21.01.73 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

локального зберігання документів, резервного копіювання баз даних або організації внутрішнього документообігу[3]. В умовах обмеженого фінансування та необхідності дотримання принципів цифрової суверенності - тобто самостійного управління національними цифровими ресурсами - такі рішення набувають стратегічної важливості.

Варто також відзначити, що в Україні останніми роками спостерігається активний розвиток напрямів ІТ-освіти, підприємництва та хакатонів для молоді. Raspberry Pi є ідеальною платформою для навчання, дослідження та практичного освоєння сучасних технологій з мінімальними витратами. Розробка та розгортання хмарного сервера на його основі дозволяє студентам та початківцям ознайомитися з основами Linux-адміністрування, мережевих протоколів, серверної архітектури, веб-сервісів, інформаційної безпеки та автоматизації [42].

Raspberry Pi - це компактний, енергоефективний пристрій, який забезпечує базові обчислювальні можливості, достатні для роботи веб-сервера, системи зберігання даних, мережевого шлюзу або IoT-рішення. Завдяки відкритому програмному забезпеченню, широкій спільноті користувачів та розвиненій документації Raspberry Pi активно використовується в освітніх, дослідницьких і побутових цілях. При наявності базових знань у галузі комп'ютерних мереж, Linux-систем та серверних технологій Raspberry Pi дозволяє реалізувати повноцінну систему хмарного зберігання з віддаленим доступом, багатокористувацьким інтерфейсом, розмежуванням прав, шифруванням даних, резервним копіюванням тощо. Raspberry Pi як основа для хмарного сховища дозволяє створити ефективну, недорогу та автономну систему, що може бути використана в домашніх умовах, малому бізнесі, навчальних закладах чи навіть в умовах обмежених ресурсів. Зважаючи на низьку вартість обладнання, відкритість програмного забезпечення та підтримку спільноти, така система є надзвичайно привабливою для реалізації.

Ще одним важливим аспектом у контексті аналізу предметної області є міждисциплінарна природа проекту. Побудова хмарного сервера на Raspberry Pi поєднує в собі елементи комп'ютерної інженерії, кібербезпеки, мережевих

					КВРКІ.210119.21.01.73 ПЗ	Арк.
						5
Зм.	Арк.	№ докum.	Підпис	Дата		

технологій, адміністрування UNIX-систем, а також основ веб-розробки[41]. Це дозволяє розглядати дану тему не лише з практичної, а й з навчально-наукової точки зору - як можливість інтегрувати різні галузі знань у рамках одного завдання.

З технічної точки зору важливим є правильний вибір файлової системи (наприклад, ext4, Vtrfs або ZFS), яка впливає на стабільність, швидкодію та надійність зберігання. Також значну роль відіграє організація енергозалежного або енергонезалежного сховища - використання зовнішніх SSD чи HDD накопичувачів через USB 3.0 або SATA-адаптери, з урахуванням споживання струму, нагріву пристрою та обмежень Raspberry Pi по живленню.

Ще одним аспектом є робота в умовах динамічної IP-адреси - проблема, яка виникає у більшості побутових користувачів, підключених до інтернету через звичайних провайдерів. Її вирішенням можуть стати сервіси динамічного DNS (DynDNS, No-IP), або створення тунелю через сторонні сервіси на зразок Cloudflare Tunnel чи ngrok. Також потребує уваги питання створення резервних копій у віддалене сховище або інший фізичний носій, щоб запобігти втраті даних у разі збою обладнання або атаки шкідливого ПЗ [28].

Окремої уваги заслуговує користувацький інтерфейс хмарного сервера. Для успішного впровадження рішення важливо не лише налаштувати функціонал, а й забезпечити зручність користування для кінцевого користувача - зрозумілу навігацію, можливість перегляду файлів, завантаження, пошуку, сортування, зміни доступу. Інтерфейс має бути адаптивним до різних типів пристроїв - від ПК до смартфонів - що вимагає розуміння принципів веб-розробки та front-end технологій.

Таким чином, реалізація персонального хмарного сховища на Raspberry Pi є проектом, що виходить за межі вузькоспеціалізованої інженерної задачі. Вона охоплює широкий спектр дисциплін, сприяє розвитку практичних навичок у сфері сучасних ІТ, а також створює підґрунтя для подальших досліджень і масштабування рішень, наприклад, у напрямі кластерних систем, IoT-інтеграції або розподіленого зберігання даних[44].

					КвРКІ.210119.21.01.73 ПЗ	Арк. 6
Зм.	Арк.	№ докум.	Підпис	Дата		

До типових програмних рішень для реалізації хмарного сервера належать Nextcloud, ownCloud, Seafile, Pydio та інші. Найбільш популярним з них є Nextcloud - безкоштовна платформа з відкритим кодом, що надає можливості синхронізації файлів, керування контактами, календарями, спільного доступу до документів, інтеграції з поштовими сервісами, медіаплеерами, а також модулів безпеки (двофакторна авторизація, шифрування, контроль доступу тощо) [8]. Програмне забезпечення легко встановлюється на сервер на базі Raspberry Pi і підтримує роботу з операційною системою Raspberry Pi OS або іншими дистрибутивами Linux.

Початково операційна система мала назву Raspbian, оскільки була заснована на дистрибутиві Debian Linux, але адаптована спеціально для архітектури ARM, яка використовується в Raspberry Pi. Розробка Raspbian почалася як незалежний проект Майка Томпсона та Пітера Гріна, які оптимізували Debian для використання з процесором ARM11, що застосовувався в перших моделях Raspberry Pi[1].

У травні 2020 року Raspberry Pi Foundation офіційно перейменувала операційну систему на Raspberry Pi OS, щоб краще відображати її статус як офіційної ОС для своїх пристроїв та розширити функціональність за межі простої адаптації Debian. Це перейменування також символізувало зрілість проекту та його перехід від експериментального дистрибутиву до повноцінної операційної системи з власною ідентичністю та екосистемою.

За роки свого існування Raspberry OS пройшла значний еволюційний шлях, з постійними оновленнями та вдосконаленням, що відповідали зростаючим можливостям апаратного забезпечення Raspberry Pi. З кожним новим поколінням Raspberry Pi операційна система отримувала оптимізації для використання додаткової продуктивності та функціональності новітніх моделей.

У процесі розробки хмарного сховища на базі Raspberry Pi виникає ряд проблем, що визначають специфіку предметної області. Однією з основних є обмежені апаратні ресурси цього мікрокомп'ютера. Хоча Raspberry Pi є універсальним і функціональним пристроєм, його обчислювальні можливості не

					КВРКІ.210119.21.01.73 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

можуть конкурувати з повноцінними серверними платформами. Це створює труднощі при обробці великого обсягу запитів, роботі з великими файлами чи при одночасному підключенні багатьох користувачів.

Ще одним важливим аспектом є питання надійності та безпеки зберігання даних. Raspberry Pi не має вбудованих засобів захисту від апаратних збоїв, таких як вихід з ладу накопичувача або раптове відключення живлення. Це суттєво підвищує ризики втрати інформації. Для мінімізації таких ризиків необхідно впроваджувати додаткові механізми - використання зовнішніх SSD-дисків, систем резервного копіювання, шифрування даних, а також забезпечення надійної аутентифікації користувачів.

Крім того, робота Raspberry Pi як сервера часто залежить від домашнього інтернет-з'єднання, яке може бути нестабільним, особливо за умов високого навантаження або у разі обмежень, накладених інтернет-провайдером. Відсутність постійної IP-адреси також може ускладнити доступ до хмарного сервісу ззовні.

Не менш суттєвою проблемою є обмеженість у масштабуванні: якщо обсяг даних або кількість користувачів починає зростати, ресурси Raspberry Pi вичерпуються досить швидко. Це потребує переходу до більш складних рішень - інтеграції з іншими системами або апгрейду апаратної частини.

Також слід враховувати, що створення хмарного сховища на Raspberry Pi вимагає технічної підготовки. Пересічному користувачу, який не має досвіду з адміністрування серверів, може бути складно налаштувати відповідне програмне забезпечення, мережеві параметри, реалізувати механізми безпеки, а також проводити тестування та моніторинг системи.

З огляду на зазначені виклики, головна мета практичної частини проекту полягає у створенні надійного, безпечного та водночас доступного для користувача хмарного сховища, побудованого на базі Raspberry Pi. У процесі реалізації передбачається глибоке вивчення апаратних та програмних можливостей пристрою, підбір відповідного програмного забезпечення, такого як Nextcloud, організація зберігання даних з використанням зовнішніх накопичувачів,

					КВРКІ.210119.21.01.73 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

впровадження системи резервного копіювання, реалізація зручного доступу з різних типів пристроїв (ПК, мобільні телефони, браузері), впровадження сучасних механізмів безпеки, а також оцінка продуктивності системи та перспектив її масштабування в майбутньому.

Результатом дослідження має стати повноцінно функціонуючий хмарний сервер на Raspberry Pi, який може задовольнити базові потреби користувача у зберіганні, перегляді та обміні файлами через інтернет. Реалізований проект повинен бути зручним у користуванні, захищеним від зовнішніх загроз та відкритим до розширення функціональності.

Крім того, очікується створення детального опису процесу розгортання такого рішення, що стане корисним як для майбутніх розробників, так і для звичайних користувачів, які прагнуть реалізувати подібну систему у себе вдома або в організації.

1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

У сфері зберігання даних на сьогодні існує чимало рішень, що охоплюють як комерційні, так і локальні інфраструктури. Вибір конкретного рішення залежить від багатьох чинників: бюджету, вимог до безпеки, масштабованості, технічного досвіду користувача та доступності ресурсів. Для кращого розуміння, наскільки доцільним є використання Raspberry Pi у ролі хмарного сервера, розглянемо основні типи існуючих рішень, їхні переваги та недоліки.

Почнемо з комерційних хмарних сервісів, таких як Google Drive, Dropbox чи OneDrive. Їхня популярність пояснюється високим рівнем зручності: користувачеві не потрібно турбуватися про фізичне зберігання даних, створення резервних копій чи обслуговування серверів [26]. Дані зберігаються у віддалених центрах обробки інформації та синхронізуються з усіма пристроями користувача через інтернет. Водночас доступ до файлів можливий практично з будь-якої точки світу. Однак така модель має низку суттєвих недоліків. По-перше, користувач передає свої

					КВРКІ.210119.21.01.73 ПЗ	Арк. 9
Зм.	Арк.	№ докум.	Підпис	Дата		

персональні або конфіденційні дані стороннім компаніям, втрачаючи тим самим контроль над ними. По-друге, навіть великі технологічні гіганти не можуть гарантувати абсолютну безпеку інформації: у минулому вже мали місце витoki персональних даних, злами акаунтів та несанкціонований доступ до файлів. По-третє, безкоштовні тарифи переважно мають обмеження по обсягу даних, швидкості передачі та функціональності. Розширення можливостей пов'язане з додатковими фінансовими витратами. Окрім того, користувач залишається залежним від технічного стану серверів компанії, її внутрішньої політики, змін в інтерфейсі або функціоналі сервісу, а також від стабільності інтернет-з'єднання. Проте використання сторонніх сервісів часто пов'язане з проблемами конфіденційності, залежністю від постачальника послуг та обмеженнями обсягу зберігання у безкоштовних тарифах. Це створює передумови для переходу до локальних хмарних рішень, які дозволяють уникнути вищезгаданих проблем і надати користувачу повний контроль над своїми даними.



Рисунок 1.1 – Логотипи популярних хмарних сервісів

Наступною категорією є мережеві сховища або NAS-системи, такі як Synology, QNAP чи WD MyCloud. Вони забезпечують високий рівень безпеки, стабільності та продуктивності. Такі пристрої мають вбудовані можливості резервного копіювання, підтримують роботу з декількома користувачами, мають розвинені веб-інтерфейси та часто підтримують розширення функціональності

					КВРКІ.210119.21.01.73 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

через додаткові модулі. Проте основним їх недоліком є висока вартість як самого обладнання, так і обслуговування. Крім того, для повного використання функціональності NAS-систем користувачеві також необхідно мати базові технічні знання, а іноді й досвід у налаштуванні мережі[30].



Рисунок 1.2 – Приклад NAS-серверу [45]

Ще одним напрямом є створення локального сервера на базі звичайного персонального комп'ютера або міні-комп'ютера, зокрема Raspberry Pi. Це рішення є економічно вигідним і дуже гнучким. Користувач має змогу самостійно обрати програмне забезпечення, організувати структуру даних, налаштувати доступ і рівень безпеки відповідно до власних потреб. Raspberry Pi вирізняється низьким енергоспоживанням, компактністю та широкою підтримкою спільноти. Проте існують і певні обмеження. Наприклад, апаратні ресурси пристрою обмежені: об'єм оперативної пам'яті, процесорна потужність та пропускну здатність мережевого інтерфейсу не дозволяють обробляти велику кількість паралельних

запитів[31]. Також користувач самостійно відповідає за всі аспекти безпеки: шифрування, бекапи, захист мережі та аутентифікацію.

Raspberry Pi являє собою серію одноплатних комп'ютерів, розроблених британською компанією Raspberry Pi Foundation з метою популяризації основ комп'ютерних наук та програмування серед школярів та студентів. Від моменту випуску першого пристрою в 2012 році, Raspberry Pi перетворився з освітнього інструменту на потужну платформу для різноманітних проектів, включаючи створення домашніх серверних рішень та хмарних сховищ даних.

Ідея створення недорогого комп'ютера, який можна було б використовувати для навчання, належить групі вчених з Кембриджського університету. У 2006 році Ебен Аптон, Роб Маллінс, Джек Ленг та Алан Майкрофт розробили концепцію пристрою, який мав би бути достатньо потужним для базових обчислювальних завдань, але при цьому недорогим і доступним для школярів та студентів. Перший прототип з'явився у 2008 році, а масове виробництво почалося у 2012 році[32].

Основною перевагою Raspberry Pi, яка робить його привабливим для використання у якості основи хмарного сервера, є поєднання доступної ціни, низького енергоспоживання та достатньої продуктивності для базових серверних завдань. Крім того, компактні розміри пристрою дозволяють розмістити його практично у будь-якому місці. Відсутність вентиляторів та інших рухомих частин забезпечує безшумну роботу, що важливо для домашнього середовища. Широкий вибір GPIO-портів надає можливості для підключення додаткових сенсорів та периферійних пристроїв, розширюючи функціональність хмарного сервера. Активна спільнота розробників постійно створює нові рішення та надає підтримку, що робить Raspberry Pi надзвичайно гнучкою платформою для експериментів та навчання.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

ціною та гнучкістю. Це рішення дозволяє реалізувати персональний або локальний сервер зберігання з мінімальними витратами та широкими можливостями подальшої модифікації.

1.3 Підходи до вирішення задачі за темою дослідження

Організація хмарного сховища даних - це багаторівневе завдання, що передбачає врахування як апаратної, так і програмної складової. Зважаючи на те, що в даному дослідженні розглядається створення хмарного сервера на базі міні-комп'ютера Raspberry Pi, основною метою є побудова функціональної, безпечної та доступної системи зберігання даних у домашніх або офісних умовах з мінімальними витратами. Підхід до вирішення задачі передбачає поетапне опрацювання кількох основних аспектів:

Насамперед, слід обрати оптимальну апаратну конфігурацію. Raspberry Pi (моделі 3B+, 4 або 5) є ідеальною основою для створення недорогого, енергоефективного і компактного сервера[10]. Його характеристик достатньо для обробки запитів домашніх або невеликих офісних користувачів. Для забезпечення стабільної роботи системи також передбачено використання зовнішнього накопичувача (жорсткий диск або SSD) з інтерфейсом USB 3.0 або SATA через спеціальний адаптер. Важливою частиною підготовки є забезпечення безперебійного живлення, а також надійне охолодження пристрою, оскільки довготривала робота у режимі навантаження може призвести до перегріву.

Для Raspberry Pi існує кілька придатних операційних систем, але найчастіше використовується Raspberry Pi OS (колишня Raspbian), що має широку підтримку, стабільність та доступ до великої кількості бібліотек і програм[44]. Також можливе використання Ubuntu Server, у випадках, коли необхідна більша гнучкість або звичайна серверна архітектура. Обидві ОС базуються на Linux і мають відкритий код, що дозволяє повністю контролювати роботу системи.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

Одним із рішень є вибір платформи, яка буде забезпечувати хмарну функціональність. Найбільш доцільним вибором у даному контексті є Nextcloud - популярне ПЗ з відкритим кодом, що дозволяє створити повноцінне хмарне середовище зі зручним веб-інтерфейсом, доступом через мобільні додатки, підтримкою синхронізації файлів, календарів, контактів, редагування документів у браузері тощо. Nextcloud також активно підтримується спільнотою, регулярно оновлюється та має високу гнучкість налаштувань[33].

Для забезпечення доступу до хмарного сервера ззовні потрібно налаштувати локальну мережу, пробросити порти на роутері (наприклад, 80 та 443 для HTTP/HTTPS), а також налаштувати динамічний DNS (DynDNS, No-IP або інші сервіси), якщо користувач не має статичної IP-адреси. Це дозволить отримувати доступ до даних навіть за межами домашньої мережі, що значно розширює функціональні можливості системи.

Оскільки система буде доступною з інтернету, важливо впровадити захисні механізми: використання SSL-сертифікатів (наприклад, через Let's Encrypt), обмеження доступу за IP, двофакторну аутентифікацію, регулярне оновлення програмного забезпечення та налаштування фаєрволу. Додатково необхідно реалізувати резервне копіювання даних: або на окремий фізичний диск, або в інший вузол локальної мережі. Також можна налаштувати автоматичне копіювання на інші хмарні сервіси, якщо це не суперечить концепції локального управління даними.

На завершальному етапі важливо перевірити стабільність роботи сервера, швидкість передачі даних, відгук користувацького інтерфейсу, підтримку паралельного підключення декількох користувачів. За потреби виконується оптимізація конфігурацій: кешування, використання протоколів WebDAV, обмеження трафіку або ресурсів для окремих процесів.

1.4 Постановка задачі

Зважаючи на результати аналізу предметної області, недоліки існуючих рішень та технічні можливості реалізації, можна сформулювати конкретну задачу дослідження. Основною метою є розробка, налаштування та тестування хмарного сервера для зберігання та обміну даними, що працює на базі одноплатного комп'ютера Raspberry Pi.

Розв'язання цієї задачі передбачає створення альтернативи комерційним хмарним сервісам, орієнтованої на локальне або персональне використання з високим рівнем контролю над даними. Такий сервер повинен забезпечити зручний доступ до файлів через інтернет, підтримувати багатокористувацький режим, мати базові механізми безпеки, а також можливість масштабування та резервного копіювання.

Для досягнення основної мети - створення надійного хмарного сервера на базі Raspberry Pi - важливо чітко окреслити перелік завдань, які необхідно виконати на кожному етапі реалізації проекту.

Першим етапом є вибір відповідної апаратної конфігурації. Потрібно врахувати не лише технічні характеристики обраної моделі Raspberry Pi (наприклад, обсяг оперативної пам'яті та кількість портів), але й можливість підключення зовнішніх накопичувачів, систем охолодження та додаткових мережевих інтерфейсів, що забезпечать стабільну роботу в умовах тривалого навантаження. Розглядається оптимізація енергоспоживання, що є актуальним у випадках безперервної роботи пристрою.

Наступним кроком є вибір операційної системи та програмного забезпечення, що дозволить реалізувати всі функції сучасного хмарного сервісу. Найпопулярнішим вибором для таких цілей є Nextcloud, оскільки він поєднує зручність використання з потужною функціональністю: зберігання файлів, синхронізація, керування користувачами, спільний доступ до документів тощо.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

Розгортання сервера потребує виконання ряду технічних дій: встановлення ОС (наприклад, Raspberry Pi OS, Ubuntu Server або DietPi), налаштування мережевих параметрів, встановлення веб-сервера (наприклад, Apache або Nginx), бази даних (MariaDB), PHP та інших необхідних компонентів. Після цього виконується інсталяція та конфігурація самого хмарного середовища.

Також увага приділяється організації зовнішнього доступу до хмарного сервера. У цьому контексті реалізуються механізми пробросу портів на домашньому маршрутизаторі, налаштування динамічного DNS для отримання доступу до сервера ззовні без потреби у статичній IP-адресі, а також створення захищеного HTTPS-з'єднання з використанням безкоштовних SSL-сертифікатів (наприклад, Let's Encrypt).

Безпека системи є пріоритетним напрямком. Реалізується багаторівневий захист: шифрування з'єднань, використання VPN, міжмережевий екран, захист від brute-force атак, регулярне резервне копіювання даних, а також налаштування облікових записів з різними рівнями доступу. Додатково встановлюються системи моніторингу, які дозволяють відстежувати стан сервера в реальному часі та вчасно реагувати на можливі загрози чи збої.

Для перевірки працездатності та стабільності проекту проводиться комплексне тестування. Вимірюються швидкість завантаження та передачі файлів, оцінюється стабільність з'єднання при одночасному доступі кількох користувачів, перевіряється поведінка системи під час тривалого навантаження. Результати цих тестів дозволяють зробити обґрунтовані висновки про ефективність обраної архітектури.

Завершальним етапом є аналітична оцінка отриманих результатів. Проводиться аналіз доцільності подальшого масштабування рішення, розглядаються перспективи інтеграції з іншими сервісами та можливість впровадження подібних систем у навчальних закладах, малому бізнесі або в домашніх умовах для особистого користування.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

1.5 Висновки до першого розділу

У першому розділі було проведено аналіз предметної області та досліджено сучасні підходи до організації хмарного зберігання даних. Розглянуто основні характеристики та переваги хмарних технологій, зокрема їх гнучкість, масштабованість, економічність та доступність. Увагу приділено можливостям побудови власного хмарного сховища з використанням недорогих апаратних рішень, зокрема міні-комп'ютера Raspberry Pi.

Встановлено, що Raspberry Pi може бути ефективною основою для створення персонального хмарного сервера за умови правильного підбору апаратного та програмного забезпечення. Оцінено можливості використання операційних систем Linux, таких як Raspberry Pi OS та Ubuntu Server, а також інструментів з відкритим кодом, зокрема платформи Nextcloud, яка забезпечує повноцінну хмарну інфраструктуру.

Також було проаналізовано основні аспекти організації доступу до хмарного сервера, питання безпеки та резервного копіювання даних. Визначено, що належне налаштування мережі, шифрування переданих даних, а також використання сучасних методів автентифікації є важливими умовами для безпечної та стабільної роботи системи.

Таким чином, результати аналізу предметної області створюють теоретичну основу для реалізації практичної частини роботи - розробки і впровадження хмарного сховища на базі Raspberry Pi, що буде розглянуто в наступних розділах.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ЗАСОБИ ДЛЯ РОЗРОБКИ ХМАРНОГО СЕРЕДОВИЩА

2.1 Хмарний сервер

Хмарний сервер являє собою віртуалізований обчислювальний ресурс, що надається через мережу Інтернет і забезпечує користувачам доступ до програмного забезпечення, платформ та інфраструктури як послуги. На відміну від традиційних фізичних серверів, хмарні сервери характеризуються гнучкістю, масштабованістю та можливістю динамічного розподілу ресурсів.

Основна концепція хмарного обчислення базується на наданні обчислювальних ресурсів за принципом "як послуга" (as-a-Service), що дозволяє користувачам отримувати доступ до необхідних технологічних можливостей без необхідності придбання та обслуговування власної ІТ-інфраструктури. Це забезпечує значну економію коштів та спрощує процес управління ІТ-ресурсами.

Хмарні сервери функціонують у віртуалізованому середовищі, де фізичні ресурси одного або декількох серверів розподіляються між множиною віртуальних машин. Така архітектура забезпечує ефективне використання апаратних ресурсів та можливість швидкого масштабування відповідно до потреб користувачів.

Архітектура хмарного сервера базується на сучасних принципах розподілених обчислень і віртуалізації, що дозволяє ефективно використовувати ресурси апаратного забезпечення навіть на пристроях з обмеженою продуктивністю, таких як Raspberry Pi. Одним із елементів цієї архітектури є рівень віртуалізації, який дозволяє абстрагувати фізичні ресурси та створювати віртуальні машини з ізольованими операційними середовищами. Це забезпечує гнучкість і масштабованість під час реалізації різних сервісів.

Інший важливий компонент - рівень управління ресурсами, що відповідає за моніторинг, розподіл і оптимізацію обчислювальних потужностей. Він включає системи автоматичного масштабування, балансування навантаження між віртуальними екземплярами та інструменти керування життєвим циклом віртуальних машин, які дозволяють динамічно реагувати на зміну навантаження.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

Рівень сервісів виконує роль посередника між користувачем та ресурсами хмари. Він надає інтерфейси та API, за допомогою яких можна створювати, налаштовувати та керувати віртуальними машинами, мережевими структурами та системами зберігання даних. Завдяки цьому рівню користувачі отримують зручний доступ до всіх функціональних можливостей хмарного середовища.

Забезпечення безпеки відіграє роль у хмарній архітектурі. Рівень безпеки включає механізми автентифікації, авторизації, шифрування даних, а також забезпечення захисту мережевої інфраструктури. Додатково до цього використовуються системи аудиту та моніторингу безпеки, які дозволяють відстежувати події в системі та своєчасно реагувати на потенційні загрози.

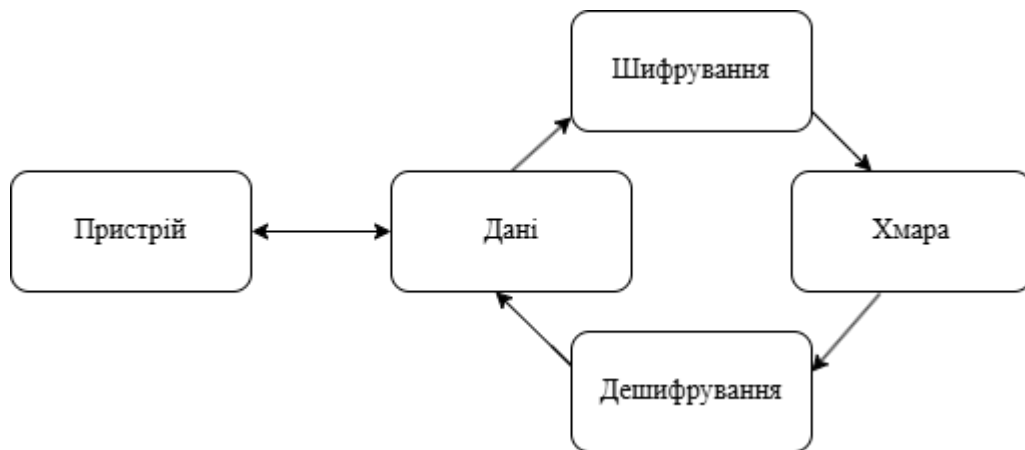


Рисунок 2.1 – Робота хмарного сховища

Використання хмарних серверів надає організаціям ряд суттєвих переваг. Економічна ефективність досягається через модель оплати за фактичне використання ресурсів, що дозволяє уникнути великих капітальних вкладень у придбання та обслуговування власної IT-інфраструктури[21]. Масштабованість забезпечує можливість швидкого збільшення або зменшення обчислювальних ресурсів відповідно до поточних потреб бізнесу.

Гнучкість хмарних рішень дозволяє організаціям швидко адаптуватися до змін у бізнес-вимогах та впроваджувати нові технології без значних затримок.

Високий рівень доступності забезпечується через географічне розподілення дата-центрів та механізми автоматичного відновлення після збоїв[22].

Проте хмарні сервери мають і певні недоліки. Залежність від інтернет-з'єднання може стати у випадку перебоїв у мережі. Питання безпеки та конфіденційності даних залишаються актуальними, особливо для організацій, що працюють з чутливою інформацією[24]. Можливість виникнення vendor lock-in ускладнює міграцію між різними постачальниками хмарних послуг.

Використання одноплатних комп'ютерів, зокрема Raspberry Pi, для створення хмарних серверів відкриває нові можливості для малих організацій та індивідуальних користувачів[1]. Такий підхід дозволяє створювати приватні хмарні рішення з мінімальними капітальними вкладеннями та енергоспоживанням.

Основними перевагами використання Raspberry Pi для хмарного сервера є низька вартість обладнання, мінімальне енергоспоживання та компактність[3]. Ці особливості роблять можливим створення розподілених хмарних інфраструктур навіть у домашніх умовах або малих офісах.

Однак слід враховувати обмеження продуктивності одноплатних комп'ютерів порівняно з серверним обладнанням. Обмежена оперативна пам'ять, швидкість процесора та пропускна здатність мережевого інтерфейсу можуть стати лімітуючими факторами для ресурсомістких додатків.

Для подолання цих обмежень застосовуються технології кластеризації, що дозволяють об'єднати декілька одноплатних комп'ютерів у єдину обчислювальну систему. Такий підхід забезпечує горизонтальне масштабування та підвищує надійність системи через розподілення навантаження.

2.2 Raspberry Pi

Raspberry Pi являє собою серію одноплатних комп'ютерів, розроблених британською компанією Raspberry Pi Foundation з метою популяризації основ комп'ютерних наук та програмування серед школярів та студентів[43]. Від моменту

					КвРКІ.210119.21.01.73 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

випуску першого пристрою в 2012 році, Raspberry Pi перетворився з освітнього інструменту на потужну платформу для різноманітних проектів, включаючи створення домашніх серверних рішень та хмарних сховищ даних.

Ідея створення недорогого комп'ютера, який можна було б використовувати для навчання, належить групі вчених з Кембриджського університету. У 2006 році Ебен Аптон, Роб Маллінс, Джек Ленг та Алан Майкрофт розробили концепцію пристрою, який мав би бути достатньо потужним для базових обчислювальних завдань, але при цьому недорогим і доступним для школярів та студентів. Перший прототип з'явився у 2008 році, а масове виробництво почалося у 2012 році.

Основною перевагою Raspberry Pi, яка робить його привабливим для використання у якості основи хмарного сервера, є поєднання доступної ціни, низького енергоспоживання та достатньої продуктивності для базових серверних завдань[30]. Крім того, компактні розміри пристрою дозволяють розмістити його практично у будь-якому місці.

Для об'єктивної оцінки можливостей Raspberry Pi як платформи для створення хмарного сервера, необхідно детально розглянути технічні характеристики актуальних моделей, доступних на ринку.



Рисунок 2.2 – Raspberry Pi 4 Model B [48]

порівнянні з попередніми моделями. Новий 4-ядерний процесор Broadcom BCM2712 на архітектурі Cortex-A76 із частотою до 2,4 ГГц забезпечив до двох разів вищу обчислювальну потужність, що дозволяє ефективніше виконувати складні завдання, зокрема роботу з хмарними сервісами, обробку даних у реальному часі або запуск віртуальних машин.

Крім того, було впроваджено підтримку вищошвидкісної оперативної пам'яті LPDDR4X-4267, яка забезпечує кращу пропускну здатність і швидший доступ до даних, при роботі з великими обсягами інформації або при одночасному запуску кількох сервісів. Raspberry Pi 5 також підтримує нові відеостандарти, включаючи 8K і 4K @ 120 Гц, що розширює його можливості для мультимедійних та графічних проектів.

Помітним нововведенням стала наявність PCIe-інтерфейсу, який дозволяє підключати NVMe SSD-диски для швидкого зберігання даних, що робить модель ще більш привабливою для створення хмарного сховища чи міні-серверної інфраструктури. Також Raspberry Pi 5 отримав покращене енергоживлення з підтримкою струму до 5 А через USB-C, що забезпечує стабільність навіть при високих навантаженнях.

Завдяки цим удосконаленням, Raspberry Pi 5 є не лише еволюційним розвитком платформи, а й потужним інструментом для створення сучасних мережевих і серверних рішень, включаючи побудову домашнього хмари, IoT-центрів чи навчальних лабораторій.

Таблиця 2.2 – Характеристики Raspberry Pi 5

Компонент	Характеристика
Процесор	Broadcom BCM2712, 4-ядерний Cortex-A76, до 2,4 ГГц
Оперативна пам'ять	4 або 8 ГБ LPDDR4X-4267 SDRAM
Мережеві інтерфейси	Gigabit Ethernet (з PoE+ через NAT), Wi-Fi 2.4/5.0 ГГц (IEEE 802.11ac), Bluetooth 5.0
USB-порти	2 × USB 3.0, 2 × USB 2.0
Відеовиходи	2 × micro-HDMI (до 4K @ 120 Гц або 8K @ 60 Гц)

Таблиця 2.3 – Характеристики Raspberry Pi Zero 2 W

Компонент	Характеристика
Процесор	Broadcom BCM2710A1, 4-ядерний Cortex-A53 (ARM v8) 64-біт, 1 ГГц
Оперативна пам'ять	512 МБ LPDDR2 SDRAM
Мережеві інтерфейси	Wi-Fi 2.4 ГГц (802.11b/g/n), Bluetooth 4.2
USB-порти	1 × micro-USB OTG
Відеовиходи	1 × mini-HDMI порт
Порти GPIO	40-контактний (непаяний)
Накопичувачі	Слот для microSD карти
Живлення	5V DC через micro-USB (рекомендовано 1,2А)
Споживання енергії	~0,4 Вт (без навантаження), до 1,1 Вт (під навантаженням)

2.3 Raspberry OS

Raspberry OS (раніше відома як Raspbian) - це офіційна операційна система, розроблена спеціально для одноплатних комп'ютерів Raspberry Pi[44]. Історія цієї операційної системи тісно пов'язана з розвитком самого пристрою Raspberry Pi, який був створений у 2012 році Raspberry Pi Foundation з метою популяризації комп'ютерних наук та програмування серед студентів та ентузіастів.

Raspberry OS пропонується в кількох варіантах, які розраховані на різні потреби та сценарії використання. Найбільш популярною для серверних рішень є Raspberry OS Lite - мінімалістична версія без графічного інтерфейсу, де пріоритетом є продуктивність та низьке споживання ресурсів. Ця версія ідеально підходить для створення серверів, оскільки всі системні ресурси спрямовуються на виконання основних завдань без витрат на обслуговування графічного середовища.

Для користувачів, які потребують графічного інтерфейсу, доступна Raspberry OS Desktop - повна версія з включеним набором стандартних програм для повсякденного використання. Найбільш функціональною є Raspberry OS Full, яка містить розширений набір додаткових програм, зокрема освітнього призначення, що робить її привабливою для навчальних закладів та початківців у світі одноплатних комп'ютерів.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

Вибір конкретної версії залежить від призначення пристрою та доступних ресурсів - для серверних завдань рекомендується використовувати Lite версію через її оптимізацію та мінімальні системні вимоги. Для створення хмарного сервера найбільш доцільним є використання Raspberry OS Lite, оскільки серверні рішення зазвичай не потребують графічного інтерфейсу, а мінімалістичний підхід забезпечує більш ефективне використання обмежених ресурсів Raspberry Pi [35], але я буду використовувати Raspberry OS Desktop для зручності.

Хоча для серверних завдань графічний інтерфейс не є обов'язковим, використання Desktop версії може бути виправданим на етапі налаштування та тестування системи. Графічне середовище спрощує процес конфігурації, дозволяє легко переглядати логи, редагувати конфігураційні файли через текстові редактори з графічним інтерфейсом та здійснювати моніторинг системи. Після завершення налаштування сервера завжди можна перейти на більш легку версію або просто відключити автозапуск графічного середовища для економії ресурсів.

Для забезпечення виводу графічного інтерфейсу та обробки медіа-даних Raspberry Pi використовує спеціалізований графічний процесор (GPU) - Videocore IV, який є інтегрованим рішенням компанії Broadcom. Графічна підсистема Raspberry Pi підтримує роботу з низкою графічних бібліотек, таких як OpenGL, OpenVG та OpenMAX, через посередництво інтерфейсів, що дозволяють взаємодію між програмами користувача та апаратним забезпеченням.

На наведеній нижче діаграмі рис 2.5 зображено структуру графічної підсистеми Raspberry Pi. Вона ілюструє, як медіа- або 3D-застосунки взаємодіють із графічними бібліотеками, які в свою чергу спілкуються через EGL з ядровими драйверами та графічним процесором. EGL виступає в ролі "моста", що забезпечує узгоджену роботу різних графічних API з GPU.

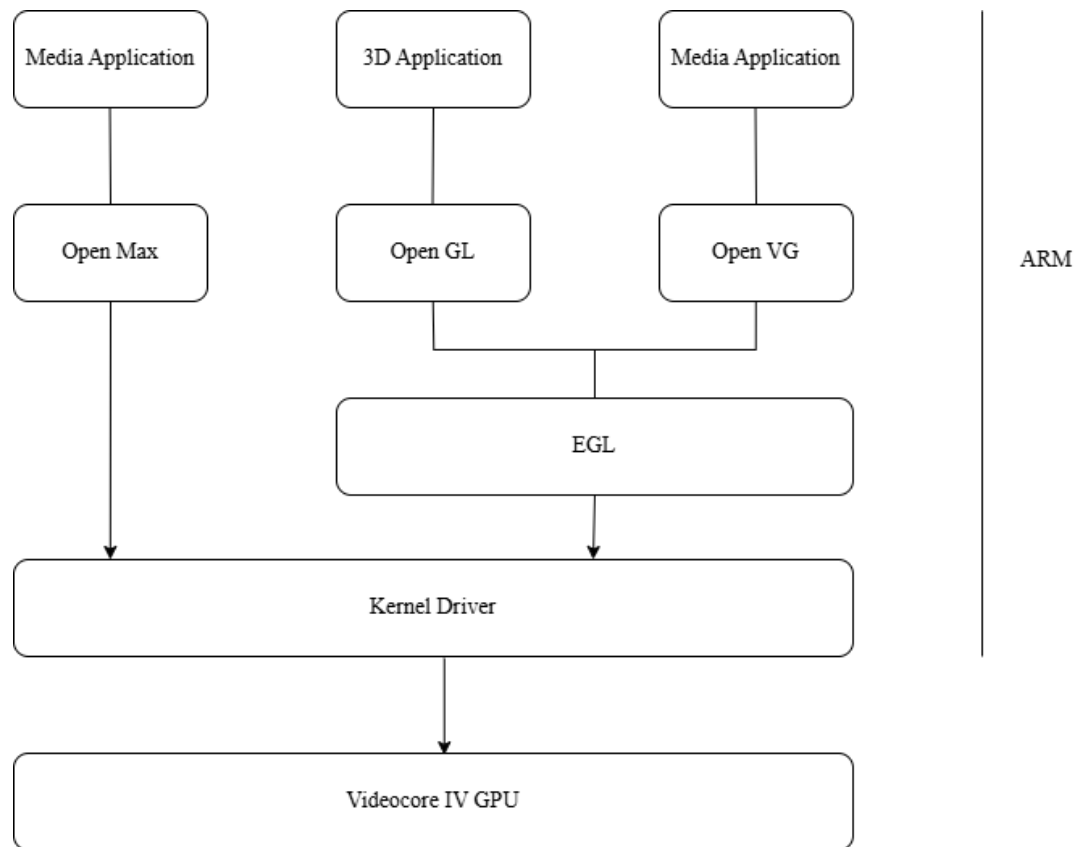


Рисунок 2.5 – Архітектура Raspberry OS

Одним із етапів розгортання хмарного сервера на базі одноплатного комп'ютера Raspberry Pi є інсталяція операційної системи[37]. Від правильного вибору та налаштування ОС залежить стабільність роботи, безпека, а також функціональні можливості майбутнього сервера. Розглянемо покрокову процедуру встановлення операційної системи Raspberry Pi OS, яка є офіційно рекомендованим дистрибутивом.

Перш ніж розпочати встановлення операційної системи на Raspberry Pi, важливо належним чином підготувати як апаратну, так і програмну частину.

Перш за все, необхідно мати сам одноплатний комп'ютер - у даному випадку використовується Raspberry Pi 5, що має потужні технічні характеристики й підходить для побудови хмарного середовища. Для запуску операційної системи та зберігання даних потрібна карта пам'яті microSD обсягом щонайменше 16 Гб, бажано високошвидкісного класу (не нижче класу 10 або UHS-I) для стабільної

роботи з файловими системами. Щоб записати на неї образ ОС, потрібен картрідер або вбудований карт-рідер у ПК чи ноутбукі.

Крім цього, необхідно мати персональний комп'ютер або ноутбук з будь-якою з основних операційних систем (Windows, Linux, macOS), який використовуватиметься для підготовки карти пам'яті та встановлення необхідного ПЗ. Для живлення Raspberry Pi 5 слід використовувати блок живлення з вихідними параметрами не менше 5 В і 3 А, а краще 5 А - це гарантує надійну роботу навіть за умов навантаження та підключення периферії.

Не менш важливо забезпечити доступ до Інтернету, оскільки знадобиться завантаження актуального образу операційної системи, утиліт для запису, а також подальше оновлення системних компонентів після інсталяції.

2.4 Технологія SSH

Secure Shell (SSH) - це криптографічний мережевий протокол, що забезпечує безпечно з'єднання між клієнтом і сервером через захищений канал зв'язку[10]. SSH є технологією для безпечного віддаленого управління серверами, включаючи хмарні системи на базі Raspberry Pi.

Протокол SSH був розроблений у 1995 році Тату Юльонен (Tatu Ylönen) як заміна для небезпечних протоколів telnet, rlogin та rsh. Сучасна версія SSH-2, забезпечує надійний захист від різноманітних атак та витоків даних.

SSH (Secure Shell) є одним з компонентів для безпечного адміністрування хмарного сервера на базі Raspberry Pi[13]. Ця технологія забезпечує шифрований віддалений доступ до командного рядка, що необхідно при розгортанні безмоніторних серверних рішень. SSH це криптографічний мережевий протокол, який забезпечує безпечно віддалене адміністрування та передачу даних через захищені мережі. Принцип роботи SSH базується на клієнт-серверній архітектурі.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

Message Authentication Code), які забезпечують автентифікацію повідомлень і дозволяють виявляти будь-які спроби зміни або підробки даних у трафіку. Технологія SSH є важливою компонентою хмарного сервера на базі Raspberry Pi, забезпечуючи безпечний віддалений доступ та управління[16]. Правильна реалізація SSH включає вибір оптимальних криптографічних алгоритмів для ARM архітектури, налаштування надійних методів автентифікації та впровадження практик безпеки.

Для успішного впровадження SSH для хмарного сервера включають регулярну ротацію ключів, моніторинг безпеки, оптимізацію продуктивності для обмежених ресурсів Raspberry Pi та інтеграцію з іншими сервісами хмарної інфраструктури[19]. Використання SSH як фундаментального рівня безпеки дозволяє створити надійну та масштабовану хмарну систему зберігання даних.

2.5 Apache, PHP, Maria DB

У сучасних веб-системах одним із завдань є створення надійного, безпечного та масштабованого серверного середовища, здатного обробляти запити користувачів, зберігати інформацію, забезпечувати взаємодію з клієнтською частиною та підтримувати динамічний контент. Актуальності це набуває у контексті створення хмарного сервера на базі Raspberry Pi[2]., де ресурси є обмеженими, а потреба в ефективному використанні доступного програмного забезпечення - максимально високою.

У процесі вибору веб-серверного програмного забезпечення важливо враховувати кожного з доступних рішень. Хоча Apache є одним із найпопулярніших варіантів[4]., існує також низка альтернатив, які можуть бути ефективнішими в залежності від сценарію використання, апаратного забезпечення або навантаження.

Серед таких альтернатив - Nginx, LiteSpeed та Microsoft IIS, кожен з яких має свої переваги й обмеження. Для оцінки доцільності використання Apache у проекті

					КвРКІ.210119.21.01.73 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

на базі Raspberry Pi було проведено порівняння його характеристик із вказаними аналогами.

Вибір веб-сервера є важливим рішенням при розгортанні веб-додатків та сайтів. Кожен з популярних серверів має свої унікальні переваги та особливості, які роблять його більш або менш підходящим для конкретних завдань. Правильний вибір веб-сервера впливає не лише на продуктивність системи, але й на безпеку, масштабованість та подальше обслуговування проекту.

У таблиці нижче подано узагальнене порівняння чотирьох найбільш поширених веб-серверів. Порівняння проводиться за критеріями, які впливають на вибір серверного рішення: продуктивність, використання ресурсів, підтримка технологій, простота конфігурації тощо. Ця інформація допоможе системним адміністраторам та розробникам прийняти обґрунтоване рішення відповідно до специфіки їхніх проектів та доступних технічних ресурсів.

Таблиця 2.5 – Порівняння Apache з аналогами

Характеристика	Apache	Nginx	LiteSpeed	Microsoft IIS
Ліцензія	Apache License 2.0 (відкрита)	BSD (відкрита)	Комерційна / OpenLiteSpeed	Пропрієтарна (Microsoft)
ОС	Linux, Unix, Windows	Linux, Unix, Windows	Linux, Unix (OpenLiteSpeed)	Windows
Метод обробки запитів	Процесна	Асинхронна, подієва	Подієва (асинхронна)	Багатопотокова
Продуктивність (статичний контент)	Середня	Висока	Дуже висока	Середня
Продуктивність (динамічний контент)	Висока (з PHP)	Висока (через FastCGI)	Дуже висока (вбудований PHP)	Висока (ASP.NET, PHP)

компонентів, які відіграють роль у роботі хмарного сховища Nextcloud: веб-сервер Apache, мова серверних сценаріїв PHP та система керування базами даних MariaDB.

Apache HTTP Server - один із найпопулярніших і найдавніших веб-серверів у світі, що з моменту свого створення у 1995 році залишається одним із лідерів ринку[32]. Завдяки своїй стабільності, модульності, гнучким налаштуванням і підтримці великої кількості протоколів Apache широко використовується як у великих компаніях, так і у локальних рішеннях[35].

Apache HTTP Server виконує широкий спектр завдань, які забезпечують повноцінне функціонування веб-додатків та сайтів. Розглянемо детальніше основні функції цього потужного веб-сервера. У контексті Raspberry Pi Apache демонструє високу стабільність навіть при помірному навантаженні[36]. Його конфігураційні файли дозволяють точно контролювати доступ до ресурсів, визначати порти, обмеження та умови маршрутизації запитів.

Apache ефективно обробляє вхідні HTTP-запити від браузерів та інших клієнтських додатків. Сервер здатен одночасно обслуговувати тисячі з'єднань, забезпечуючи швидку відповідь на запити користувачів. Підтримуються всі основні HTTP-методи: GET, POST, PUT, DELETE, HEAD та інші, що дозволяє створювати сучасні RESTful API та інтерактивні веб-додатки.

Сервер автоматично аналізує заголовки запитів, визначає тип контенту та застосовує відповідні правила обробки. Вбудовані механізми безпеки перевіряють валідність запитів та захищають від найпоширеніших атак.

Apache забезпечує надійну передачу HTML-сторінок, медіафайлів, сценаріїв та інших веб-ресурсів клієнтам. Сервер автоматично визначає MIME-типи файлів та встановлює відповідні заголовки відповіді, що гарантує коректне відображення контенту в браузерах.

Ефективно Apache працює з динамічним контентом, виконуючи PHP, Python, Perl та інші серверні сценарії. Вбудовані механізми кешування дозволяють

зменшити навантаження на сервер та прискорити завантаження сторінок для користувачів.

Інтеграція Apache із PHP здійснюється через відповідний модуль, який забезпечує обробку PHP-файлів до того, як їх буде відправлено клієнту. Таким чином, веб-сервер передає динамічні сторінки, які формуються в реальному часі на основі дій користувача та даних з бази.

PHP (Hypertext Preprocessor) - одна з найпоширеніших мов програмування для створення динамічних веб-додатків[39]. Її основна перевага полягає в простоті синтаксису, швидкому освоєнні та широкій підтримці з боку спільноти. Вона спеціально створена для генерації HTML-коду на льоту, взаємодії з базами даних і обробки запитів.

PHP забезпечує комплексний набір інструментів для повноцінної веб-розробки. Мова дозволяє ефективно обробляти користувацькі запити через веб-форми, забезпечуючи валідацію даних та їх подальшу обробку. Вбудовані функції роботи з файловою системою надають можливість зберігати завантажені користувачами файли на сервері з контролем безпеки та управлінням правами доступу.

Перевагою PHP є тісна інтеграція з різноманітними системами баз даних. Мова надає вбудовану підтримку MySQL та MariaDB, а також працює з PostgreSQL, SQLite, Oracle та іншими СУБД через уніфіковані інтерфейси PDO та mysqli.

Генерація динамічного HTML-контенту є основною сферою застосування PHP. Мова дозволяє створювати персоналізовані сторінки на основі даних користувача, умов та контексту запиту. Шаблонізатори та фреймворки додатково спрощують процес створення складних інтерфейсів.

PHP надає потужні механізми управління станом додатка через підтримку сесій та cookies. Вбудовані функції шифрування та хешування забезпечують безпечне зберігання паролів та конфіденційних даних користувачів.

Мова також включає розширені можливості для роботи з мультимедійним контентом, дозволяючи обробляти зображення, створювати архіви та здійснювати мережеві запити до зовнішніх API та сервісів.

Завдяки активній спільноті розробників та величезній екосистемі бібліотек, PHP залишається одним з провідних інструментів для створення сучасних веб-додатків, від простих сайтів до складних корпоративних систем.

У межах реалізації хмарного сервера на Raspberry Pi PHP відіграє роль основного інтерпретатора логіки роботи платформи Nextcloud[40]. Саме за допомогою PHP обробляються всі дії користувача: авторизація, створення папок, переміщення файлів, синхронізація тощо.

Крім основного ядра, у PHP активно використовуються додаткові розширення - модулі, які забезпечують розширену функціональність. Наприклад:

- php-mysql - для роботи з MariaDB;
- php-gd - для обробки графічних зображень;
- php-xml, php-zip, php-intl, php-mbstring - для підтримки сучасних форматів і стандартів.

Завдяки активній спільноті, регулярним оновленням та підтримці нових технологій, PHP залишається актуальним інструментом для побудови веб-систем навіть на малопотужних пристроях.

MariaDB - це високопродуктивна, надійна система керування базами даних, яка є форком популярної MySQL[4]. Вона повністю сумісна з MySQL, але має відкриту ліцензію та активніше розвивається спільнотою. MariaDB вибирають за її швидкість, безпеку та масштабованість.

MariaDB повністю підтримує стандарт SQL, забезпечуючи виконання складних запитів для отримання, оновлення та аналізу даних. Система надає потужні інструменти для роботи з транзакціями, гарантуючи цілісність даних навіть при одночасному доступі багатьох користувачів. Механізм ACID-транзакцій забезпечує коректність операцій та можливість відкату змін у разі помилок.

Перевагою MariaDB є вбудовані можливості реплікації баз даних, що дозволяє створювати резервні копії у реальному часі та розподіляти навантаження між кількома серверами. Це забезпечує високу доступність системи та захист від втрати даних.

Надійність зберігання даних досягається через використання сучасних механізмів журналювання та контрольних сум. MariaDB підтримує різні типи індексів для оптимізації швидкості пошуку, а також тригери, події та збережені процедури для автоматизації складних операцій з даними.

Інтеграція з PHP здійснюється за допомогою відповідного драйвера (mysql або mysqli). PHP-скрипти надсилають SQL-запити до MariaDB, отримують результати та відображають їх користувачу через інтерфейс.

Одним з найважливіших кроків є встановлення надійного пароля для адміністративного облікового запису root. Цей пароль повинен бути достатньо складним та унікальним, оскільки адміністратор має повні права доступу до всіх баз даних та налаштувань системи. Використання слабкого або стандартного пароля створює серйозну вразливість для всієї системи[14].

Принцип мінімальних привілеїв реалізується через створення окремих користувачів з обмеженими правами доступу для кожного додатка або сервісу. Замість використання адміністративного облікового запису для повсякденних операцій, створюються спеціалізовані користувачі, які мають доступ лише до необхідних баз даних та операцій. Такий підхід значно знижує потенційні наслідки компрометації облікового запису.

Тестові бази даних та демонстраційні облікові записи, які за замовчуванням присутні у свіжій інсталяції MariaDB, становлять додатковий ризик безпеки. Ці компоненти часто мають відомі паролі або взагалі не захищені, тому їх видалення є обов'язковим кроком забезпечення безпеки.

Мережева безпека забезпечується через обмеження доступу до бази даних із зовнішніх IP-адрес. За замовчуванням MariaDB налаштовується на прийом з'єднань тільки з локального сервера, що запобігає прямим атакам через мережу. Якщо

необхідний віддалений доступ, він повинен бути ретельно налаштований з використанням шифрування та надійної автентифікації.

2.6 Nextcloud

Nextcloud - це безкоштовна платформа з відкритим вихідним кодом для створення власних хмарних сервісів зберігання та обміну файлами[9]. Розроблена у 2016 році як форк проекту ownCloud, Nextcloud швидко стала одним з найпопулярніших рішень для побудови приватних хмарних інфраструктур[2].

Основними принципами Nextcloud є:

- Повний контроль над даними користувача.
- Відкритість та прозорість коду.
- Безпека та приватність.
- Гнучкість та масштабованість.
- Кросплатформенність.

Nextcloud написана на мові програмування PHP і використовує веб-інтерфейс для взаємодії з користувачами. Платформа підтримує різні системи управління базами даних, включаючи MySQL, MariaDB, PostgreSQL та SQLite, що робить її адаптивною до різних апаратних конфігурацій.

Nextcloud надає можливість зберігати файли різних типів та форматів із автоматичною синхронізацією між різними пристроями. Система підтримує версійність файлів, що дозволяє відновлювати попередні версії документів.

Платформа дозволяє створювати публічні та приватні посилання для обміну файлами, налаштовувати права доступу для різних користувачів та груп, встановлювати терміни дії посилань.

Вбудовані додатки для управління календарем та адресною книгою з підтримкою стандартів CalDAV та CardDAV, що забезпечує сумісність з більшістю сучасних пристроїв та додатків[26].

					КВРКІ.210119.21.01.73 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

Для розуміння принципів роботи Nextcloud важливо враховувати призначення та взаємодію її основних компонентів, які відповідають за стабільність і продуктивність системи. На рисунку 2.6 представлено елементи архітектури сервера Nextcloud у разі його розгортання на Raspberry Pi[29].

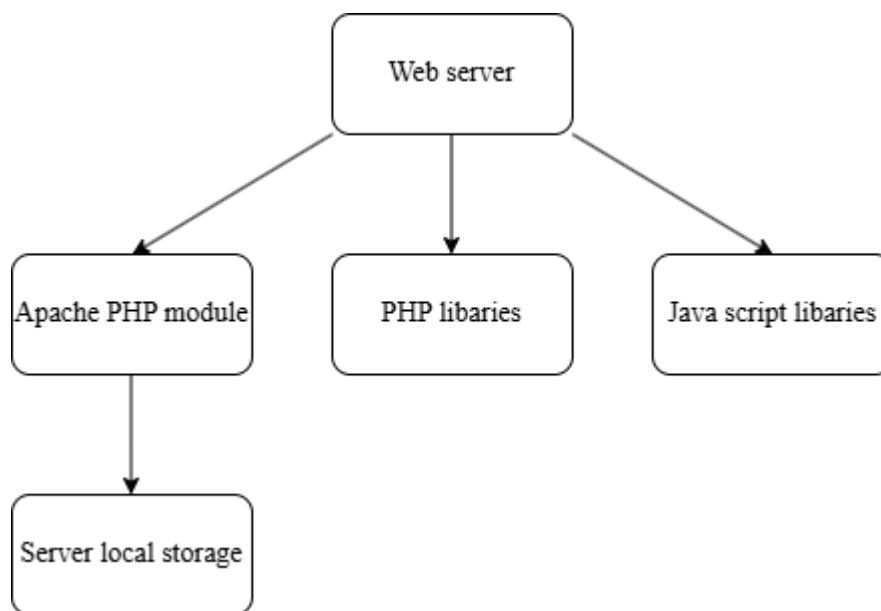


Рисунок 2.6 – Загальний вигляд архітектури сервера Nextcloud

Розуміння архітектури хмарної системи Nextcloud вимагає детального аналізу ролі кожного компонента у загальному процесі обробки користувацьких запитів та забезпечення функціональності платформи.

Веб-сервер Apache виступає як головний шлюз системи, відповідаючи за приймання всіх HTTP-запитів від клієнтських додатків та веб-браузерів користувачів Nextcloud. Коли користувач намагається отримати доступ до файлів, налаштувань або будь-якої іншої функціональності платформи, Apache приймає ці запити та визначає подальший маршрут їх обробки. Після первинного аналізу запити передаються до PHP-модуля для виконання серверної логіки.

Модуль PHP, інтегрований безпосередньо в Apache, служить інтерпретатором серверного коду, який становить основу всієї логіки роботи Nextcloud. Цей компонент відповідає за виконання складних алгоритмів управління файлами, обробку користувацьких сесій, реалізацію правил безпеки та

координацію роботи всіх підсистем платформи. PHP-модуль динамічно генерує HTML-контент на основі запитів користувачів та поточного стану системи.

PHP-бібліотеки розширюють базові можливості інтерпретатора, забезпечуючи виконання операцій платформи. Через ці бібліотеки реалізується взаємодія з базою даних MariaDB для зберігання та отримання метаданих, здійснюється обробка завантажених файлів з контролем їх типів та розмірів, а також забезпечується надійна автентифікація користувачів з підтримкою різних методів входу в систему.

JavaScript-бібліотеки формують клієнтську частину додатка, завантажуючись веб-сервером та виконуючись безпосередньо у браузері користувача. Ці компоненти забезпечують створення динамічного та інтерактивного інтерфейсу, який дозволяє користувачам комфортно працювати з файлами, налаштуваннями та іншими функціями без необхідності постійного перезавантаження сторінок. Інтерактивність включає drag-and-drop операції, контекстні меню, попередній перегляд файлів та багато інших сучасних елементів користувацького досвіду.

Локальне сховище відіграє роль фізичного репозиторію для всіх даних системи, зберігаючи файли користувачів у структурованому вигляді з дотриманням правильних дозволів та організації директорій. Окрім користувацького контенту, локальне сховище містить системні файли Nextcloud, тимчасові дані для обробки запитів, логи системи та резервні копії важливої інформації, що забезпечує стабільну та надійну роботу всієї платформи.

Nextcloud на базі Raspberry Pi представляє собою ефективне рішення для створення власного хмарного сервера зберігання даних. Попри певні технічні обмеження, поєднання цих технологій забезпечує оптимальний баланс між функціональністю, вартістю та простотою впровадження.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

2.7 Висновки до другого розділу

У цьому розділі було проаналізовано компоненти, необхідні для створення персонального хмарного середовища на базі Raspberry Pi. Розглянуто апаратні та програмні засоби, зокрема роль Raspberry Pi як недорогого та енергоефективного пристрою, придатного для реалізації серверних рішень. Увагу приділено платформі Nextcloud, яка забезпечує повноцінну функціональність хмарного сховища та має модульну архітектуру, що дозволяє масштабування та інтеграцію з іншими сервісами.

Аналіз архітектури Nextcloud дав змогу визначити призначення кожного з її компонентів: веб-сервер (Apache), модуль PHP, бібліотеки PHP і JavaScript, а також локальне сховище. Їхня взаємодія забезпечує стабільну роботу системи, обробку запитів користувачів, збереження та захист даних, а також формування зручного інтерфейсу.

Отже, поєднання Raspberry Pi та Nextcloud є доцільним і ефективним рішенням для створення хмарного сервера, що дозволяє користувачам зберігати, керувати та синхронізувати власні дані у межах локальної мережі або з віддаленим доступом, зберігаючи при цьому контроль над конфіденційністю.

					КвРКІ.210119.21.01.73 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

3 ВСТАНОВЛЕННЯ НЕОБХІДНИХ КОМПОНЕНТІВ

3.1 Вибір та підготовка обладнання

Для реалізації хмарного сховища на базі Nextcloud було обрано одноплатний комп'ютер Raspberry Pi 4 Model B з наступними характеристиками:

Таблиця 3.1 – Характеристики Raspberry Pi 4 Model

Компонент	Характеристика
Процесор	Broadcom BCM2711, 4-ядерний Cortex-A72 (ARM v8) 64-біт, 1,5 ГГц
Оперативна пам'ять	8 ГБ LPDDR4-3200 SDRAM
Мережеві інтерфейси	Gigabit Ethernet, Wi-Fi 2.4/5.0 ГГц (IEEE 802.11ac), Bluetooth 5.0
USB-порти	2 × USB 3.0, 2 × USB 2.0
Відеовиходи	2 × micro-HDMI (підтримка до 4K @ 60 Гц)
Порти GPIO	40-контактний GPIO
Накопичувачі	Слот microSD для ОС та зберігання даних
Живлення	5V DC через USB-C (рекомендовано 3A)
Споживання енергії	3,4 Вт (без навантаження), до 7,6 Вт (під навантаженням)

Цей вибір обумовлений оптимальним співвідношенням технічних характеристик та вартості, що робить його ідеальним рішенням для створення персонального хмарного сервера.

Raspberry Pi 4 Model B значно перевершує своїх попередників за продуктивністю завдяки потужному чотирьохядерному процесору Cortex-A72 з тактовою частотою 1.5 GHz та 4 ГБ оперативної пам'яті LPDDR4. Ці характеристики забезпечують достатню обчислювальну потужність для обслуговування кількох одночасних користувачів та виконання ресурсомістких операцій з файлами.

Наявність портів USB 3.0 дозволяє підключати швидкісні зовнішні накопичувачі, що важливо для забезпечення належної швидкості роботи з великими файлами в хмарному сховищі. Вбудований Gigabit Ethernet забезпечує

стабільне та швидке мережеве з'єднання, а підтримка двохдіапазонного Wi-Fi 802.11ac надає гнучкість у налаштуванні мережевої інфраструктури.

Компактні розміри пристрою та низьке енергоспоживання роблять його практичним рішенням для домашнього використання, дозволяючи працювати цілодобово без значного впливу на електроспоживання. Відсутність активного охолодження забезпечує тиху роботу, що може бути помітно у житловому приміщенні.

3.2 Встановлення Raspberry OS

Першим кроком є отримання офіційного програмного забезпечення для запису операційної системи на карту пам'яті. Raspberry Pi Imager - це утиліта, розроблена фондом Raspberry Pi, яка дозволяє швидко й просто записати дистрибутив на microSD-карту.

Для завантаження утиліти необхідно перейти на офіційний сайт та обрати відповідну версію для операційної системи користувача. Після завантаження програма встановлюється у стандартний спосіб.

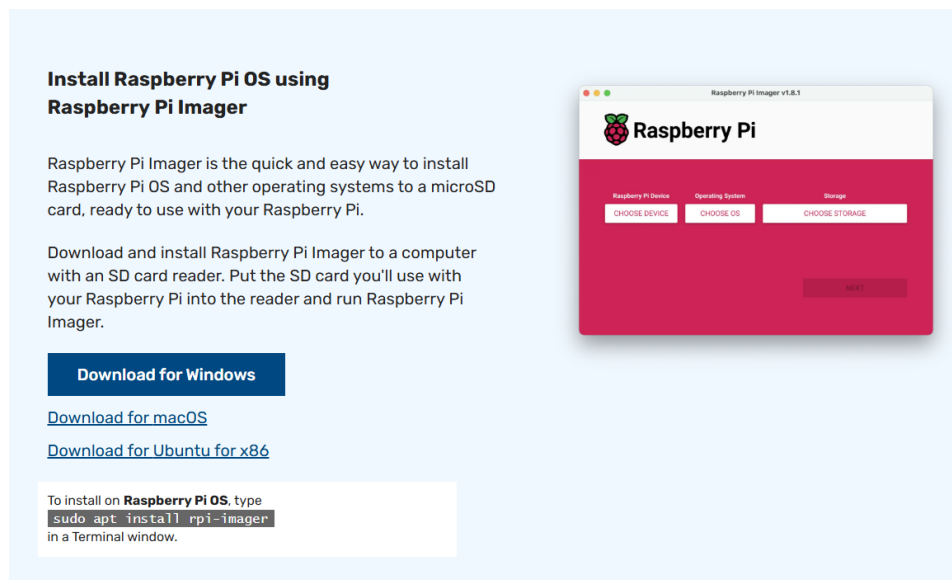


Рисунок 3.1 – Завантаження Raspberry Pi Imager

Після встановлення Raspberry Pi Imager користувач повинен виконати наступні дії:

- Запустити програму та натиснути кнопку “Choose OS”. У відкритому списку обрати дистрибутив.
- Натиснути “Choose Storage” та вибрати відповідний накопичувач (microSD-карту).
- Після цього натиснути “Write”, щоб розпочати процес запису. Після завершення утиліта автоматично перевіряє цілісність записаного образу.

Цей процес повністю стирає дані на microSD-карті, тому заздалегідь варто зробити резервну копію важливої інформації.

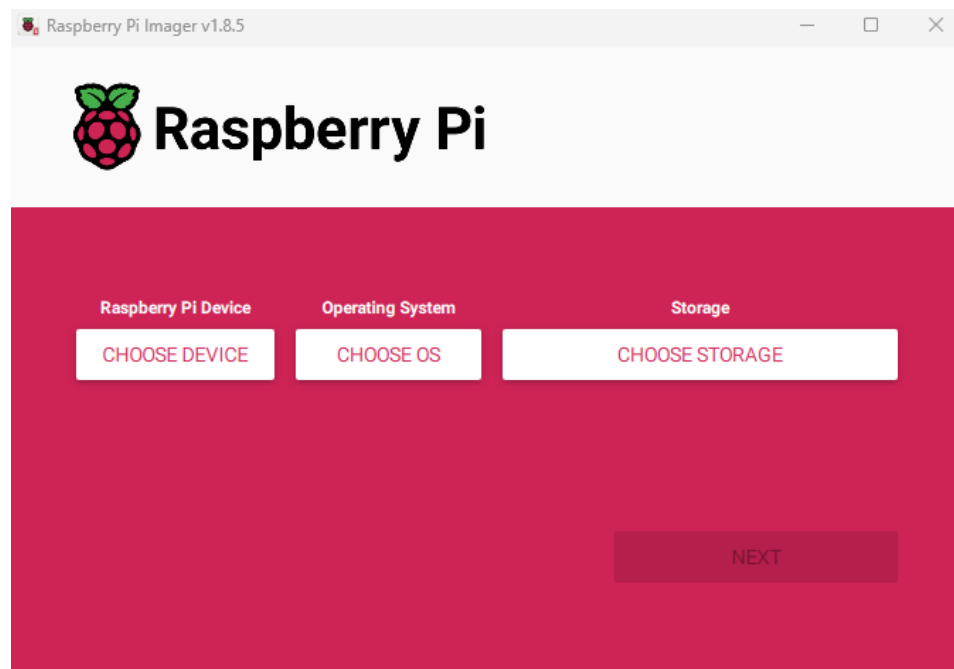


Рисунок 3.2 – Запис образу Raspberry PI OS на носій

Після запуску Raspberry Pi Imager користувачу пропонується обрати одну з доступних операційних систем. Для нашого випадку слід вибрати:

Raspberry Pi OS (32-bit) - рекомендована версія операційної системи, яка базується на Debian Linux.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

Далі користувач обирає такі параметри як країна, мова інтерфейсу, формати дати/часу, а також клавіатурна розкладка. Ці налаштування є важливими як для коректного відображення тексту в системі, так і для нормальної роботи периферійних пристроїв, зокрема клавіатури.

Параметри локалізації безпосередньо визначають формати дати та часу, які використовуються у всій системі. Залежно від обраної локалі, дата може відображатися у форматі день/місяць/рік, місяць/день/рік або рік-місяць-день, а час може використовувати 12-годинний або 24-годинний формат. Ці налаштування впливають не лише на системні повідомлення, але й на роботу веб-додатків, включаючи Nextcloud.

Відображення валюти та одиниць вимірювання також контролюється параметрами локалізації. Система автоматично використовує відповідні символи валют, правила розстановки розділових знаків у числах та метричні або імперські одиниці вимірювання згідно з регіональними стандартами. Це важливо для додатків, які працюють з фінансовими даними або технічними розрахунками.

Часові обчислення та операції з датами суттєво залежать від налаштувань локалі, особливо у контексті часових поясів та переходів на літній час. Правильна конфігурація забезпечує коректну роботу планувальників завдань, систем резервного копіювання та синхронізації даних між різними пристроями користувачів.

Поведінка багатьох програм, які залежать від локалі, адаптується відповідно до встановлених параметрів. Це включає сортування файлів та текстових рядків за національними алфавітами, обробку символів з діакритичними знаками, відображення повідомлень інтерфейсу користувача та форматування виводу команд системи. Це необхідно для веб-серверів та баз даних, які повинні коректно обробляти багатомовний контент та забезпечувати належну підтримку національних символів.

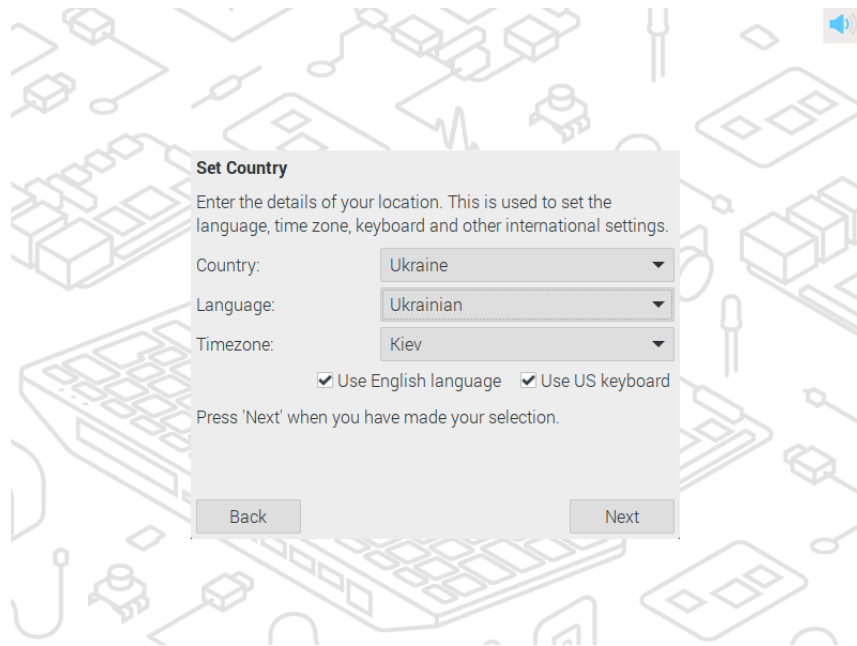


Рисунок 3.4 – Налаштування

Створюється профіль користувача для використання Raspberry OS. Цей етап важливий для безпеки та персоналізації системи, особливо у випадках, коли Raspberry Pi планується використовувати як хмарний сервер або для доступу до нього через мережу. У версіях Raspberry Pi OS, випущених після 2022 року, встановлення системи без створення персонального користувача неможливе. Раніше система за замовчуванням використовувала обліковий запис "pi" з паролем "raspberrypi", однак через загрозу безпеці ця практика була припинена. Створений користувач автоматично додається до основних груп доступу, таких як:

- sudo - для виконання команд з правами адміністратора;
- video, audio, plugdev - для доступу до периферії;
- netdev - для налаштування мережі.

Це діалогове вікно дозволяє або одразу запустити процес оновлення, або пропустити його, якщо немає доступу до Інтернету або потрібно виконати оновлення пізніше вручну.

Рекомендується підтвердити запуск оновлення натисканням кнопки "Next", щоб система завантажила актуальні версії програмних пакетів та оновлення безпеки.

Після перезавантаження Raspberry Pi виконує стандартний процес ініціалізації ядра Linux, підключення служб та запуск графічного середовища LXDE (Lightweight X11 Desktop Environment) або PIXEL - модифікованої оболонки, створеної спеціально для Raspberry Pi OS.

Якщо під час налаштувань було обрано автоматичний вхід (опція "Log in automatically"), користувач одразу потрапляє на головний екран робочого столу. Якщо ж автоматичний вхід не було ввімкнено, система запропонує ввести логін і пароль створеного раніше користувача.

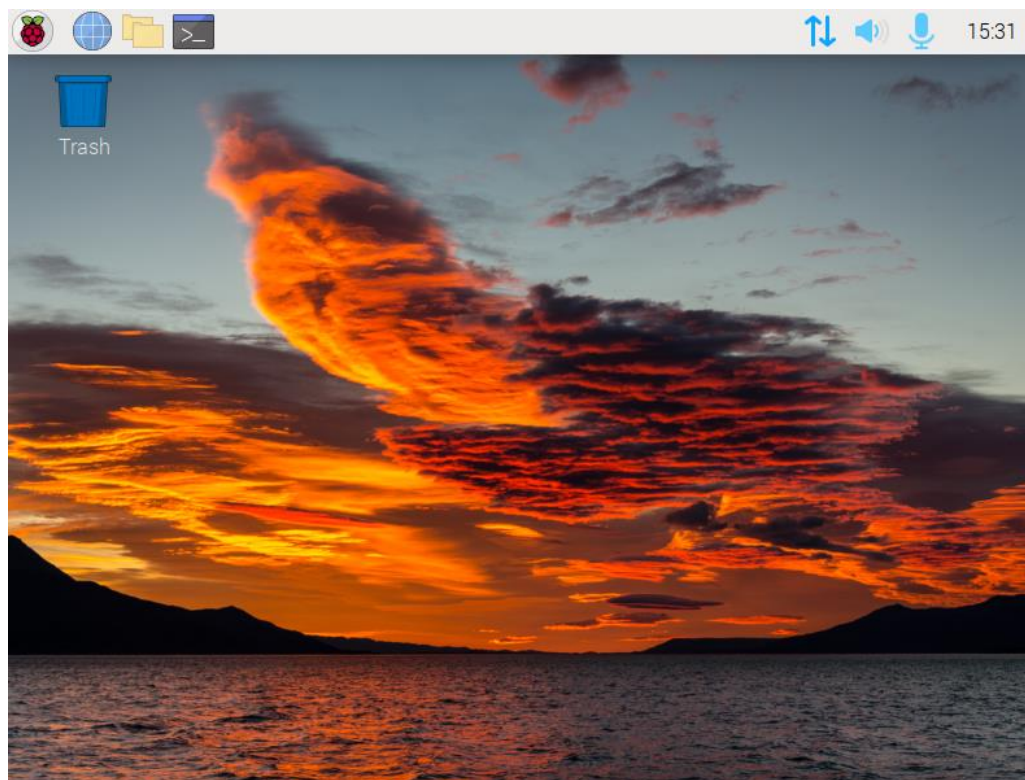


Рисунок 3.7 – Робочий стіл користувача

					КВРКІ.210119.21.01.73 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

Перехід на робочий стіл після встановлення та оновлення Raspberry Pi OS знаменує завершення етапу початкової конфігурації пристрою. Система повністю готова до подальшого налаштування, встановлення серверних рішень або спеціалізованих програмних засобів.

3.3 Ввімкнення SSH

Secure Shell (SSH) - це мережевий протокол, який дозволяє безпечно керувати пристроями через командний рядок за допомогою зашифрованого з'єднання. У контексті реалізації хмарного сервера на Raspberry Pi, використання SSH є важливим, оскільки дає змогу адмініструвати систему віддалено, без необхідності фізичного доступу до пристрою або зовнішнього монітора та клавіатури.

SSH (Secure Shell) надає адміністраторам потужний інструмент для віддаленого управління серверами та робочими станціями через захищене з'єднання. Використовуючи SSH, системні адміністратори можуть повноцінно керувати файловою системою, створювати, редагувати та видаляти файли, налаштовувати права доступу та працювати з директоріями на віддаленій машині так, ніби вони знаходяться безпосередньо за її консоллю.

Протокол дозволяє ефективно управляти системними сервісами, запускати та зупиняти необхідні служби, контролювати їх стан та налаштовувати автоматичний запуск. Адміністратори можуть встановлювати нове програмне забезпечення, оновлювати існуючі пакети, керувати залежностями та підтримувати систему в актуальному стані без фізичної присутності біля сервера.

Налаштування мережевих служб також стає простішим завдяки SSH - можна конфігурувати веб-сервери, бази даних, поштові сервіси, налаштовувати фаєрволи та мережеві інтерфейси. При необхідності адміністратор може безпечно перезапустити систему або окремі сервіси, контролюючи процес завантаження та перевіряючи коректність роботи після перезапуску.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

Завдяки шифруванню всього трафіку SSH забезпечує безпечну передачу конфіденційної інформації, включаючи паролі та системні дані, що робить його незамінним інструментом для адміністрування інфраструктури в сучасних ІТ-середовищах. Найпростішим способом активації SSH для користувачів, які мають доступ до графічного середовища, є використання вбудованого інструмента налаштувань:

- Натиснути "Raspberry" у лівому верхньому куті.
- Перейти до Preferences → Raspberry Pi Configuration.
- У вікні, що відкрилося, перейти на вкладку "Interfaces".
- Знайти пункт SSH і змінити його стан з Disabled на Enabled.
- Натиснути ОК для підтвердження.
- При потребі перезавантажити Raspberry Pi, щоб активувати службу.

Після цього SSH-сервер запускається автоматично під час старту системи, і пристрій можна адмініструвати з іншого комп'ютера в тій самій мережі.

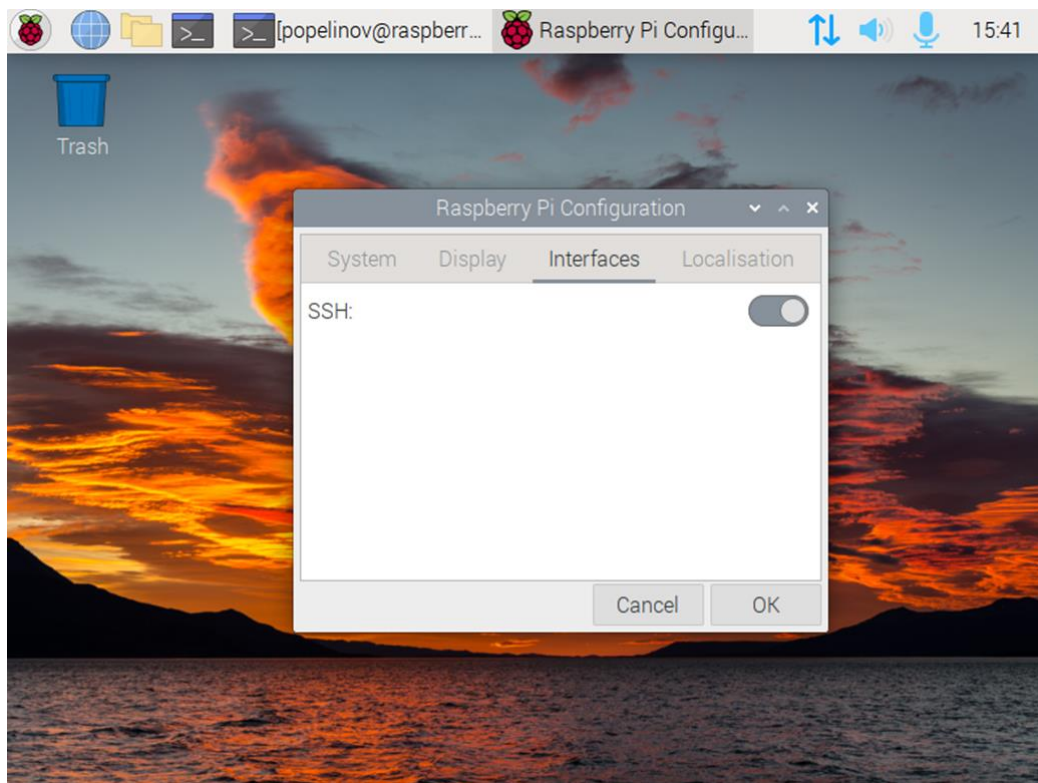


Рисунок 3.8 – Ввімкнення SSH

Якщо користувач працює з Raspberry Pi OS Lite (версія без графічного інтерфейсу), або надає перевагу командному рядку, SSH можна ввімкнути за допомогою консолі.

Після активації служби SSH користувач може підключитися до Raspberry Pi з іншого пристрою. Активація SSH на Raspberry Pi є забезпечує зручне й безпече віддалене адміністрування системи. У контексті побудови хмарного сервера це дозволяє повністю контролювати пристрій без фізичного доступу, що актуально при його розміщенні в складнодоступному місці або при безголовому (headless) використанні. Розуміння принципів роботи SSH, варіантів його активації та засобів захисту є базовими навичками кожного системного адміністратора або розробника в галузі мережевих технологій.

3.4 Встановлення Apache, PHP та Maria DB

Процес встановлення Apache на Raspberry Pi є відносно простим завдяки наявності пакету в офіційному репозиторії операційної системи. На початковому етапі виконується перевірка наявності оновлень та оновлення системних компонентів. Це дозволяє уникнути конфліктів при встановленні нового ПЗ.

Після оновлення системи використовується штатний пакетний менеджер для завантаження і встановлення Apache, PHP та MariaDB. У результаті на Raspberry Pi запускається веб-сервер, який автоматично починає слухати стандартний порт 80 (HTTP).

Для перевірки коректності встановлення достатньо відкрити веб-браузер на будь-якому пристрої, підключеному до тієї ж мережі, і перейти за IP-адресою Raspberry Pi. Якщо все виконано правильно, на екрані з'являється вітальна сторінка Apache із повідомленням про успішне встановлення.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

потреби увімкнути шифрування трафіку за допомогою SSL-сертифікатів (наприклад, від Let's Encrypt).

Крім того, Apache дозволяє гнучко керувати параметрами продуктивності: кількістю одночасних з'єднань, обсягами кешування, використанням пам'яті тощо. У межах даного проекту були застосовані базові параметри, які є достатніми для локального хмарного сервера, що обслуговує кілька користувачів.

Після завершення налаштування було проведено тестування роботи Apache. У результаті запуску Nextcloud у браузері веб-сервер успішно обслуговує запити, передає дані клієнтам та взаємодіє з іншими компонентами системи - PHP-інтерпретатором і базою даних. Було підтверджено, що Apache стабільно працює на Raspberry Pi, не перевантажує систему та відповідає вимогам хмарного середовища.

Встановлення та налаштування веб-сервера Apache на Raspberry Pi є складовою створення локального хмарного сховища. Попри обмежені ресурси пристрою, Apache демонструє високу стабільність та продуктивність у поєднанні з оптимально налаштованим середовищем. Правильна конфігурація веб-сервера забезпечує надійну основу для подальшої роботи системи Nextcloud та гарантує безперебійну роботу збереження та доступу до даних у локальній мережі.

MariaDB є основою для збереження даних хмарного сервера. Вона забезпечує високу швидкість обробки запитів і зручне управління. Завдяки легкості інтеграції з іншими компонентами (Apache, PHP), вона органічно вписується в архітектуру реалізованого рішення.

PHP елемент серверної архітектури Nextcloud. Його правильне встановлення і налаштування забезпечує динамічну взаємодію між користувачем, веб-сервером та базою даних. Завдяки підтримці численних модулів, PHP адаптується до складних завдань і дозволяє розширювати функціональність хмарного сховища без істотних змін у системі.

3.5 Встановлення Nextcloud

Перед безпосереднім встановленням Nextcloud необхідно підготувати операційну систему та програмне середовище Raspberry Pi. У межах даної реалізації використовується Raspberry Pi 4B з операційною системою Raspberry Pi OS, яка була попередньо встановлена та оновлена.

На першому етапі важливо впевнитися, що пристрій має активне з'єднання з мережею, стабільне живлення та достатньо вільного місця для розгортання хмарного сховища. Також потрібно переконатися, що операційна система підтримує роботу з веб-сервером, системою керування базами даних та інтерпретатором PHP, які є основними компонентами для роботи Nextcloud.

Після оновлення системи встановлюються веб-сервер Apache або Nginx, система баз даних (зазвичай використовується MariaDB як відгалуження MySQL), а також різні PHP-модулі, необхідні для коректної роботи Nextcloud. Ці компоненти формують так званий «стек» серверного ПЗ.

Після того як усі серверні компоненти налаштовані, можна переходити до завантаження Nextcloud. Найзручнішим способом є завантаження архіву останньої версії програми з офіційного сайту розробників. Після завантаження програма розпаковується в директорію, доступну для веб-сервера, і налаштовуються відповідні права доступу для забезпечення безпечної роботи.

Увагу слід приділити створенню окремої директорії для зберігання користувацьких файлів - це дозволить відокремити програмний код Nextcloud від даних користувача, що є рекомендованою практикою з точки зору безпеки. Така організація файлової структури спрощує процедури резервного копіювання, оскільки дані користувачів зберігаються окремо від системних файлів додатка, а також полегшує майбутні оновлення платформи без ризику втрати користувацької інформації.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

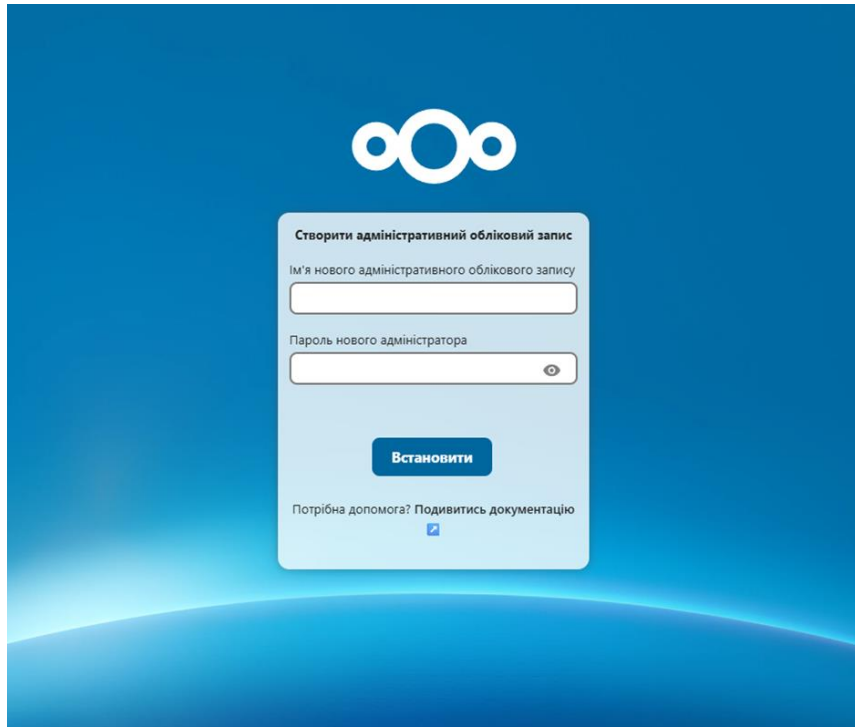


Рисунок 3.11 – Реєстрація користувача у браузері

Після підтвердження введених даних система виконує ініціалізацію: створює структуру таблиць у базі даних, перевіряє середовище та налаштовує базову конфігурацію. Весь цей процес виконується автоматично, і після його завершення користувач потрапляє до головної сторінки хмарного сховища. Варто зазначити, що тривалість ініціалізації залежить від продуктивності сервера та обраної конфігурації бази даних - зазвичай це займає від декількох хвилин до чверті години.

Інтерфейс Nextcloud доволі зручний та інтуїтивно зрозумілий. Він дозволяє не лише зберігати файли, а й переглядати документи, слухати музику, керувати календарями, нотатками, створювати резервні копії та навіть здійснювати відеоконференції - все це доступно завдяки численним додаткам, які можна встановити окремо. Модульна архітектура платформи дозволяє користувачам налаштовувати функціональність відповідно до своїх потреб, при цьому базова версія залишається легкою та швидкою. Адміністратори можуть централізовано керувати доступними додатками для всіх користувачів системи, що забезпечує гнучкість у налаштуванні корпоративного середовища та контроль над встановленим програмним забезпеченням.

					КвРКІ.210119.21.01.73 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

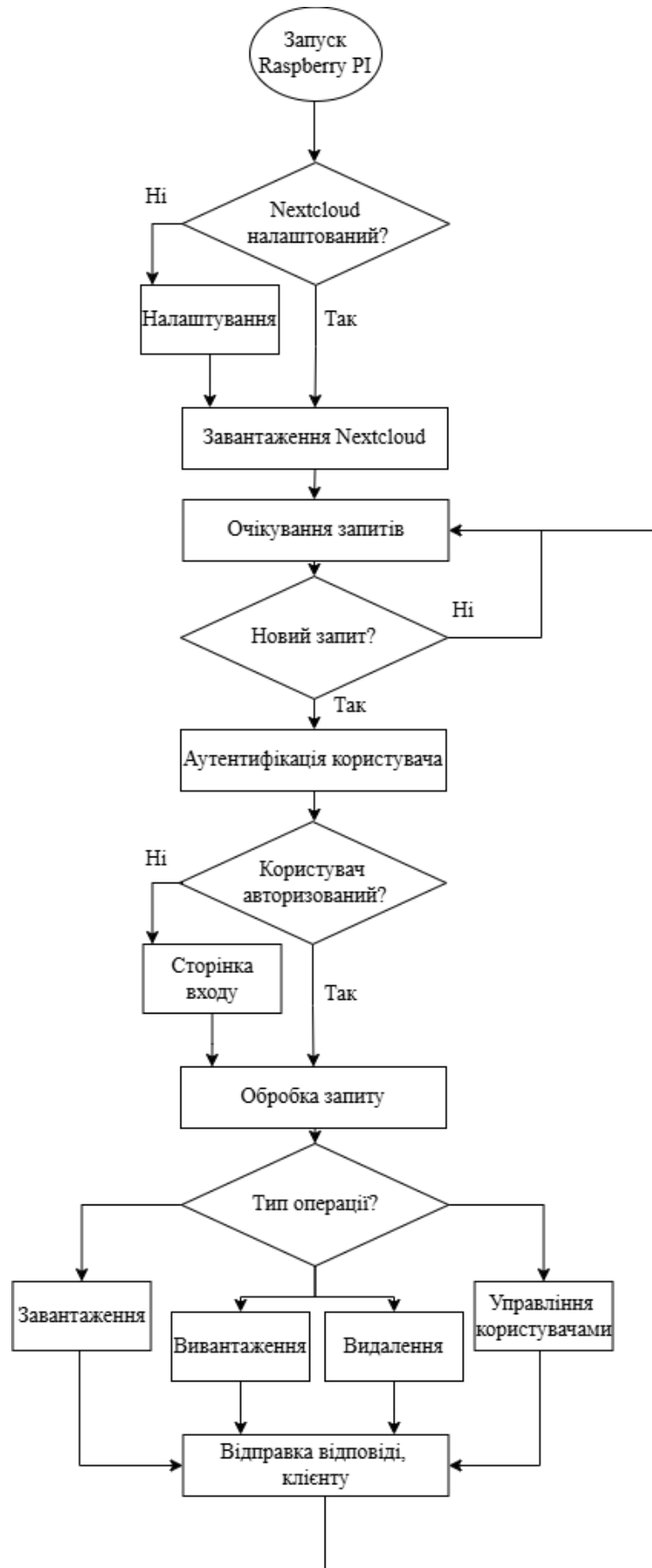


Рисунок 3.13 – Алгоритм роботи Nextcloud на Raspberry Pi

3.6. Висновки до третього розділу

У цьому розділі було детально розглянуто процес реалізації хмарного сервера на базі одноплатного комп'ютера Raspberry Pi 4 Model B. На першому етапі здійснено вибір апаратного забезпечення, яке відповідає вимогам до енергоефективності, портативності та достатньої продуктивності для виконання серверних задач. Було підготовлено все необхідне для встановлення програмного забезпечення: карта пам'яті, живлення, корпус із охолодженням, а також периферійні пристрої для початкового налаштування.

Далі виконано інсталяцію та базову конфігурацію операційної системи Raspberry Pi OS, яка оптимізована для роботи на цій платформі. Увагу приділено оновленню системи, налаштуванню користувача та регіональних параметрів. Увімкнення SSH забезпечило можливість віддаленого адміністрування пристрою, що необхідно для зручної та безперервної роботи в локальній мережі.

На основі стандартного LAMP-стека було налаштовано серверне середовище: встановлено веб-сервер Apache, інтерпретатор сценаріїв PHP та систему керування базами даних MariaDB. Проведено конфігурацію всіх необхідних компонентів, у тому числі створено базу даних і користувача для Nextcloud, налаштовано параметри PHP та перевірено їхню сумісність.

Завершальним етапом стала інсталяція хмарної платформи Nextcloud, яка розгорнута на попередньо налаштованій серверній інфраструктурі. У результаті реалізовано повноцінний хмарний сервер, здатний забезпечити зберігання, синхронізацію та обмін файлами у межах локальної мережі або за її межами.

Таким чином, виконано всі необхідні технічні заходи для створення стабільного, безпечного та функціонального середовища, що відповідає вимогам сучасних хмарних рішень і може бути використане як для особистих, так і для навчальних чи малих корпоративних цілей.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті виконання дипломної роботи на тему «Хмарний сервер для зберігання даних на основі Raspberry Pi» було досягнуто основної мети – створено компактну, доступну та ефективну систему приватного зберігання даних із можливістю розширення та адміністрування. Виконані теоретичні дослідження, практичні реалізації та тестування підтвердили актуальність теми, а також доцільність використання одноплатного комп'ютера Raspberry Pi для таких завдань.

Сьогодні, коли зростає обсяг персональної інформації, питання захисту, автономності та контролю над даними стають. Хмарні сервіси, що надаються великими ІТ-компаніями, мають низку переваг, проте вони не завжди гарантують повну конфіденційність та гнучкість для кінцевого користувача. Тому створення власного хмарного сховища на базі доступного та енергоефективного пристрою є важливим та перспективним напрямком, як у побуті, так і в освітніх або малих корпоративних середовищах.

У першому розділі було проведено ґрунтовний аналіз предметної області. Оцінено сучасний стан технологій хмарного зберігання даних, розглянуто відомі публічні сервіси, такі як Google Drive, Dropbox, iCloud та інші. Виявлено їхні переваги та недоліки, що дозволило обґрунтувати доцільність створення приватного хмарного сервера.

Розглянуто програмне забезпечення Nextcloud як основну платформу для реалізації проекту. Nextcloud надає широкі можливості щодо синхронізації, обміну файлами, створення користувачів, розширень безпеки, резервного копіювання та інтеграції з іншими сервісами. Проведено порівняння з альтернативами (OwnCloud, Seafile, Pydio), і обрано найбільш функціональний та відкритий варіант.

У ході аналізу також визначено вимоги до апаратного забезпечення, параметри енергоспоживання, продуктивності та об'єму зберігання, що дозволило сформулювати технічні вимоги до створення системи.

					КВРКІ.210119.21.01.73 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

У другому розділі виконано практичні дії з підготовки обладнання та програмного середовища. Було розглянуто технічні характеристики одноплатного комп'ютера Raspberry Pi 4 Model B, зокрема: кількість портів USB, підтримка Ethernet і Wi-Fi, підтримка зовнішніх накопичувачів, обсяг оперативної пам'яті.

Здійснено встановлення операційної системи Raspberry Pi OS, з урахуванням актуальних рекомендацій щодо стабільності, безпеки та підтримки апаратного забезпечення. Проведено налаштування мережевих параметрів, облікового запису користувача та параметрів локалізації.

Окрему увагу приділено включенню SSH-доступу, що дозволяє адмініструвати пристрій віддалено. Це стало важливим етапом для забезпечення зручного керування сервером без потреби виводу відео або підключення клавіатури.

Також здійснено встановлення компонентів серверної інфраструктури, які становлять основу функціонування хмарного сховища. Apache виступає як основа веб-сервера, відповідаючи за обробку HTTP-запитів від клієнтів та забезпечуючи стабільну доставку веб-контенту. PHP забезпечує взаємодію з серверною логікою, виконуючи обробку динамічних запитів та реалізуючи функціональність Nextcloud. MariaDB слугує як система управління базами даних для надійного зберігання конфігураційних параметрів системи та користувацьких даних.

У третьому розділі було розглянуто процес реалізації хмарного сервера на базі одноплатного комп'ютера Raspberry Pi 4 Model B. На першому етапі здійснено вибір апаратного забезпечення, яке відповідає вимогам до енергоефективності та достатньої продуктивності для серверних задач.

Виконано інсталяцію та базову конфігурацію операційної системи Raspberry Pi OS з увімкненням SSH для віддаленого адміністрування. На основі LAMP-стека налаштовано серверне середовище з веб-сервером Apache, інтерпретатором PHP та системою керування базами даних MariaDB.

Завершальним етапом стала інсталяція хмарної платформи Nextcloud на попередньо налаштованій серверній інфраструктурі. У результаті реалізовано

					КВРКІ.210119.21.01.73 ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

повноцінний хмарний сервер, здатний забезпечити зберігання, синхронізацію та обмін файлами у локальній мережі.

Таким чином, створено стабільне та функціональне середовище, що відповідає вимогам сучасних хмарних рішень і може використовуватись для особистих, навчальних чи малих корпоративних цілей.

					КВРКІ.210119.21.01.73 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Pi My Life Up. Raspberry Pi OwnCloud. URL: <https://pimylifeup.com/raspberry-pi-owncloud/> (дата звернення: 15.05.2025).
2. Instructables. How to Make a Raspberry Pi Cloud Server. URL: <https://www.instructables.com/How-to-Make-a-Raspberry-Pi-Cloud-Server/> (дата звернення: 15.05.2025).
3. XDA Developers. I built a personal cloud using my Raspberry Pi and here's how you can too. URL: <https://www.xda-developers.com/i-built-a-personal-cloud-using-my-raspberry-pi-and-heres-how-you-can-too/> (дата звернення: 15.05.2025).
4. Personal Cloud Storage using Raspberry PI. ResearchGate. 2023. URL: https://www.researchgate.net/publication/370314210_Personal_Cloud_Storage_using_Raspberry_PI (дата звернення: 15.05.2025).
5. FleetStack. Raspberry Pi Personal Cloud Server NextCloud. URL: <https://fleetstack.io/blog/raspberry-pi-personal-cloud-server-nextcloud> (дата звернення: 15.05.2025).
6. MakeUseOf. Raspberry Pi Cloud Storage Software Solutions. URL: <https://www.makeuseof.com/raspberry-pi-cloud-storage-software-solutions/> (дата звернення: 15.05.2025).
7. Cloud storage on raspberry Pi. Raspberry Pi Forums. URL: <https://forums.raspberrypi.com/viewtopic.php?t=237705> (дата звернення: 15.05.2025).
8. NextCloud on Pi 4. Raspberry Pi Forums. URL: <https://forums.raspberrypi.com/viewtopic.php?t=342466> (дата звернення: 15.05.2025).
9. OpenSource.com. NextCloudPi: NextCloud for Raspberry Pi. URL: <https://opensource.com/article/23/3/nextcloudpi-nextcloud-raspberry-pi> (дата звернення: 15.05.2025).
10. Raspberry Tips. Security Tips for Raspberry Pi. URL: <https://raspberrytips.com/security-tips-raspberry-pi/> (дата звернення: 15.05.2025).
11. Raspberry Pi Guide. Improve Raspberry Pi Security. URL: <https://raspberrypi-guide.github.io/other/Improve-raspberry-pi-security> (дата звернення: 15.05.2025).

					КВРКІ.210119.21.01.73 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

12. Pi My Life Up. Raspberry Pi Security. URL: <https://pimylifeup.com/raspberry-pi-security/> (дата звернення: 15.05.2025).

13. HackerNoon. 4 Privacy-Focused Raspberry Pi Projects. URL: <https://hackernoon.com/4-privacy-focused-raspberry-pi-projects> (дата звернення: 15.05.2025).

14. Cerberus Laboratories. Raspberry Pi Security. URL: https://cerberus-laboratories.com/blog/raspberry_pi_security/ (дата звернення: 15.05.2025).

15. FasterCapital. Raspberry Pi Hacks: Enhancing Security and Privacy. URL: <https://fastercapital.com/content/Raspberry-Pi-Hacks--Enhancing-Security-and-Privacy.html> (дата звернення: 15.05.2025).

16. OpenSource.com. IoT Security Raspberry Pi. URL: <https://opensource.com/article/17/3/iot-security-raspberry-pi> (дата звернення: 15.05.2025).

17. Pestmeester. URL: <https://pestmeester.nl/> (дата звернення: 15.05.2025).

18. Security Best Practices for Pi. Raspberry Pi Forums. URL: <https://forums.raspberrypi.com/viewtopic.php?t=328527> (дата звернення: 15.05.2025).

19. Raspberry Pi Security Hardening Script. GitHub Gist. URL: <https://gist.github.com/boseji/c9e91ff3bd0b3cfb62a5e260fe505374> (дата звернення: 15.05.2025).

20. TechRadar. Best Cloud Storage. URL: <https://www.techradar.com/best/best-cloud-storage> (дата звернення: 22.05.2025).

21. Tom's Guide. Best Cloud Storage Buying Guide. URL: <https://www.tomsguide.com/buying-guide/best-cloud-storage> (дата звернення: 22.05.2025).

22. Cloudwards. Best Cloud Storage. URL: <https://www.cloudwards.net/best-cloud-storage/> (дата звернення: 22.05.2025).

23. Cloudwards. Cloud Storage Comparison. URL: <https://www.cloudwards.net/comparison/> (дата звернення: 22.05.2025).

					КВРКІ.210119.21.01.73 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

24. Zapier. Best Cloud Storage Apps. URL: <https://zapier.com/blog/best-cloud-storage-apps/> (дата звернення: 22.05.2025).

25. CloudZero. Cloud Storage Pricing. URL: <https://www.cloudzero.com/blog/cloud-storage-pricing/> (дата звернення: 22.05.2025).

26. Abijita. The Top Cloud Storage Providers for 2025: In-Depth Review & Comparison. URL: <https://www.abijita.com/the-top-cloud-storage-providers-for-2025-in-depth-review-comparison/> (дата звернення: 22.05.2025).

27. TechRepublic. Cloud Storage Providers. URL: <https://www.techrepublic.com/article/cloud-storage-providers/> (дата звернення: 22.05.2025).

28. Cloudwards. The Top 5 Cloud Companies with Large Free Service Plans. URL: <https://www.cloudwards.net/the-top-5-cloud-companies-with-large-free-service-plans/> (дата звернення: 22.05.2025).

29. TechTarget. The Best Enterprise Data Storage Products of 2024. URL: <https://www.techtarget.com/searchstorage/feature/The-best-enterprise-data-storage-products-of-2024> (дата звернення: 22.05.2025).

30. Pi My Life Up. Raspberry Pi NAS. URL: <https://pimylifeup.com/raspberry-pi-nas/> (дата звернення: 03.06.2025).

31. SunFounder. How to Build an Efficient Raspberry Pi NAS: A Complete Installation and Configuration Guide. URL: <https://www.sunfounder.com/blogs/news/how-to-build-an-efficient-raspberry-pi-nas-a-complete-installation-and-configuration-guide> (дата звернення: 03.06.2025).

32. Instructables. How to Make a Raspberry Pi Cloud Server. URL: <https://www.instructables.com/How-to-Make-a-Raspberry-Pi-Cloud-Server/> (дата звернення: 03.06.2025).

33. OpenSource.com. Host Cloud NAS Raspberry Pi. URL: <https://opensource.com/article/18/9/host-cloud-nas-raspberry-pi> (дата звернення: 03.06.2025).

34. IONOS. Raspberry Pi NAS. URL: <https://www.ionos.com/digitalguide/server/configuration/raspberry-pi-nas/> (дата звернення: 03.06.2025).

35. Core Electronics. How to Raspberry Pi NAS. URL: <https://core-electronics.com.au/guides/how-to-raspberry-pi-nas/> (дата звернення: 03.06.2025).

36. ThePi.io. How to Set Up a Raspberry Pi OwnCloud Server. URL: <https://thepi.io/how-to-set-up-a-raspberry-pi-owncloud-server/> (дата звернення: 03.06.2025).

37. DeepSeaDev. Raspberry Pi NAS Server. URL: <https://www.deepseadev.com/en/blog/raspberry-pi-nas-server/> (дата звернення: 03.06.2025).

38. NAS + VPN. Raspberry Pi Forums. URL: <https://forums.raspberrypi.com/viewtopic.php?t=244224> (дата звернення: 03.06.2025).

39. Instructables. Raspberry Pi 2 NAS With OwnCloud, Samba and MiniDLNA. URL: <https://www.instructables.com/Raspberry-Pi-2-NAS-With-Owncloud-Samba-and-MiniDln/> (дата звернення: 03.06.2025).

40. Jurzik H. Nextcloud on the Raspberry Pi: Set up your own cloud with NextcloudPi. – Yuki likes Snow, 2023. – 114 с

41. Peregrino E. Cloud-Powered Robotics with Raspberry Pi. New Delhi BPB Publications, 2023. 260 с.

42. The Makers of The MagPi magazine. The Official Raspberry Pi Handbook 2025: Astounding projects with Raspberry Pi computers Cambridge Raspberry Pi Press, 2024. 200 с.

43. The Makers of The MagPi magazine. The Official Raspberry Pi Handbook: Astounding projects with Raspberry Pi computers. Cambridge Raspberry Pi Press, 2023. 200 с.

44. Raspberry Pi Foundation. Raspberry OS. URL: <https://www.raspberrypi.org/software/> (дата звернення: 03.06.2025)

					КВРКІ.210119.21.01.73 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

45. Synology DiskStation DS1522+. URL: <https://rozetka.com.ua/ua/synology-ds1522plus/p367501611/> (дата звернення: 04.06.2025).

46. Raspberry Pi 3 Model B. URL: <https://miniboard.com.ua/mini-komputeri/272-raspberry-pi-3-model-b> (дата звернення: 04.06.2025).

47. Мікрокомп'ютер Raspberry Pi 5 Board 4GB. URL: <https://e-server.com.ua/uk/mikrokompiuter-raspberry-pi-5-board-4gb-rpi5-4gb> (дата звернення: 04.06.2025).

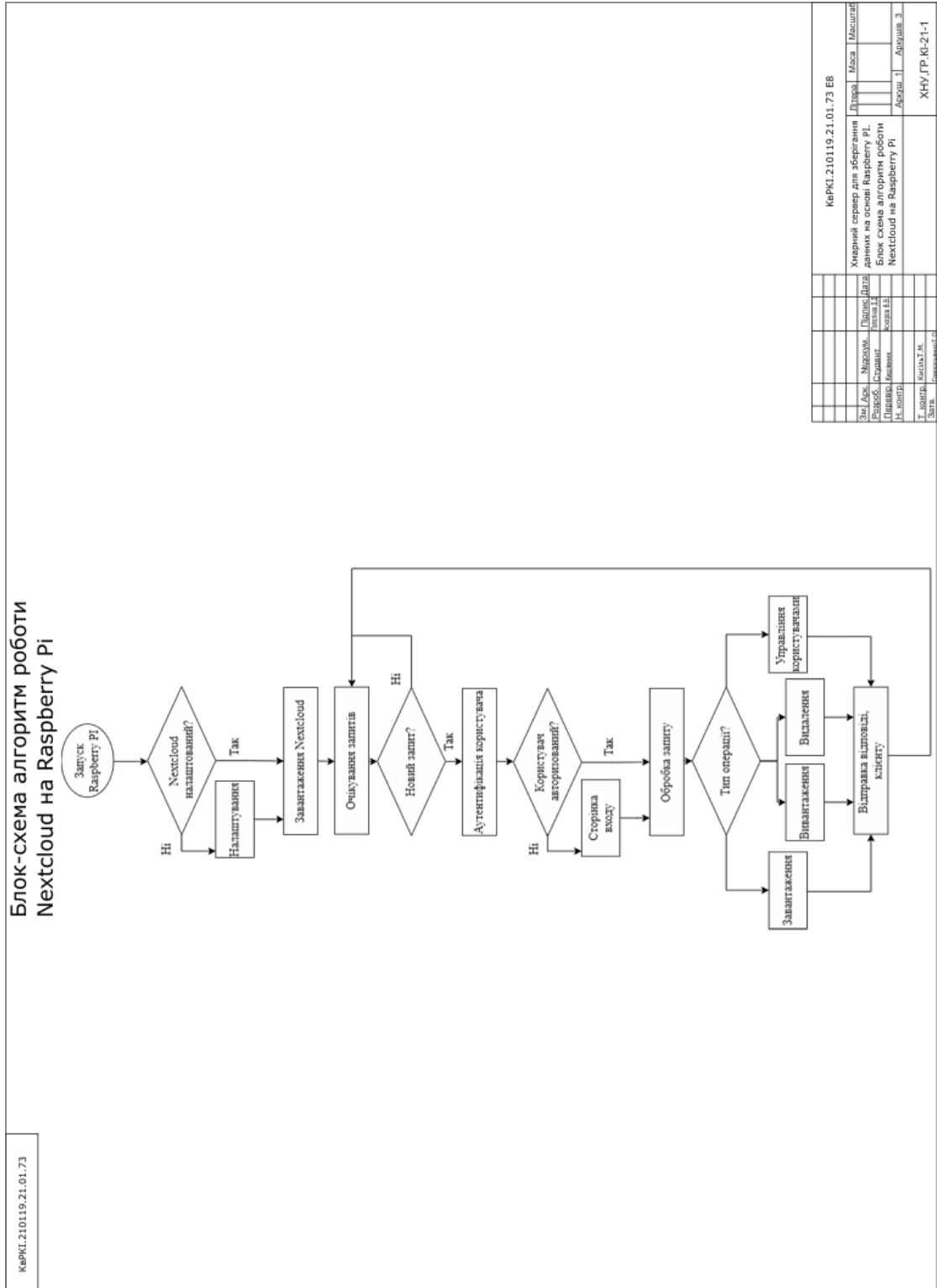
48. Мікрокомп'ютер Raspberry Pi 4 Model B 8GB. URL: <https://evo.net.ua/mikrokomputer-raspberry-pi-4-model-b-8gb/> (дата звернення: 04.06.2025).

49. Raspberry Pi Zero 2 W. URL: <https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/> (дата звернення: 04.06.2025).

					КВРКІ.210119.21.01.73 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

Додаток А (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «БЛОК-СХЕМА АЛГОРИТМ РОБОТИ NEXTCLOUD НА RASPBERRY PI»

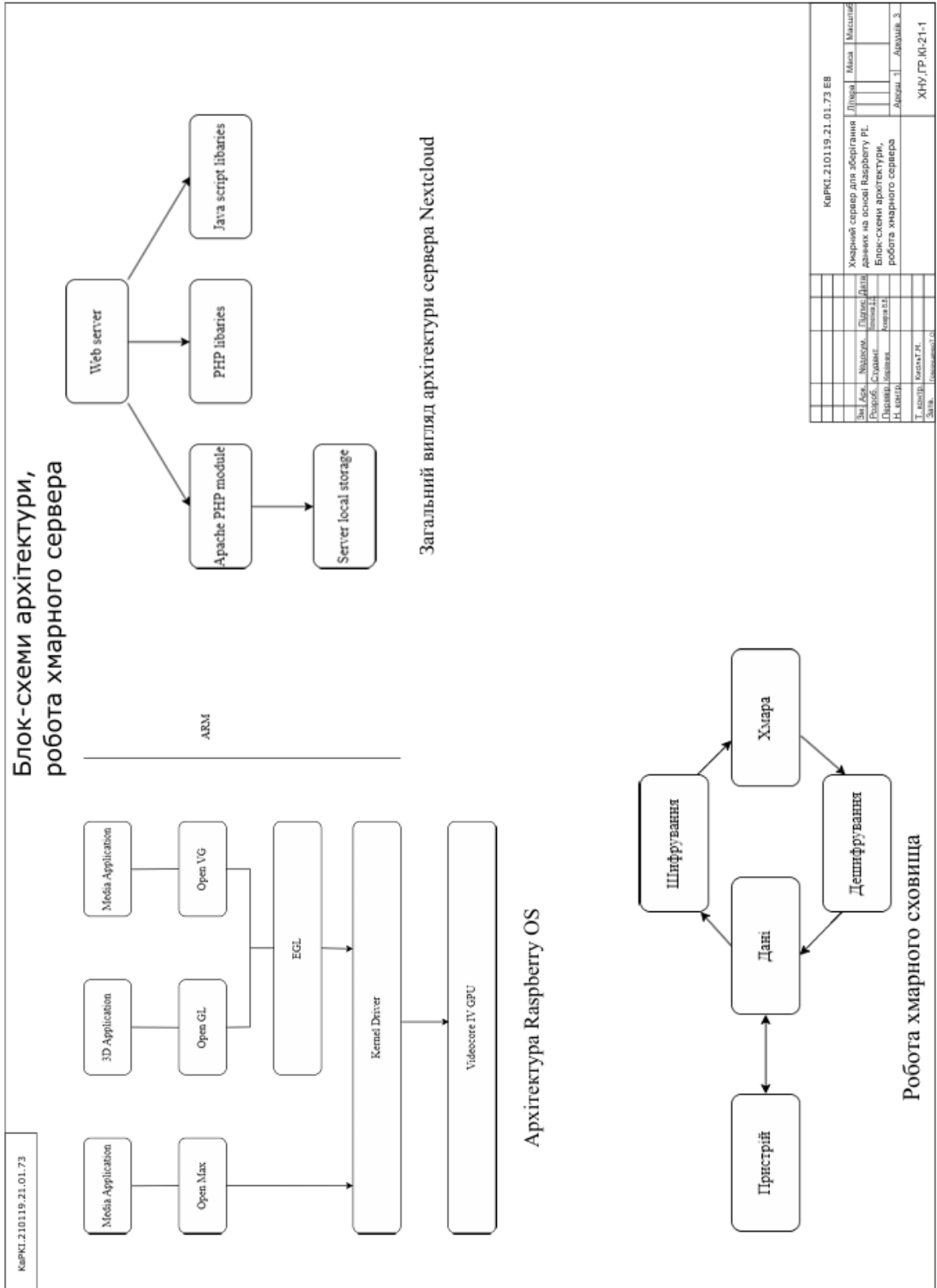


КьРКІ.210119.21.01.73

КьРКІ.210119.21.01.73 ЕВ									
Зм. №	Місяць	Підпис	Діля	Місяць	Місяць				
Розроб.	Сторінка	Код	Код	Код	Код				
Н. код	Місяць	Код	Код	Код	Код				
Т. код	Код	Код	Код	Код	Код				
С. код	Код	Код	Код	Код	Код				
Херний сервер для зберігання даних на основі Raspberry Pi. Блок-схема алгоритму роботи Nextcloud на Raspberry Pi					ХНУ, ГР. КІ-21-1				

Додаток Б
(обов'язковий)

КОПІЯ КРЕСЛЕННЯ «БЛОК-СХЕМИ АРХІТЕКТУРИ, РОБОТА ХМАРНОГО СЕРВЕРА»



Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Данило ПОПЕЛІНОВ

Співавтор:

Назва: Попелінов_Хмарний сервер для зберігання даних на основі Raspberry PI

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:1.5%

Коефіцієнт подібності 2:0.3%

Мікропробіли: 6

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-09 09:15:34.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-09

Дата



Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 11%

ID: 244159 Title: БКР Хмарний сервер для зберігання даних на основі Raspberry PI Added in a DB: 2025-06-09 Authors: Данило ПОПЕЛІНОВ Heads: В'ячеслав АСКЕРОВ Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	90783	659	1193 (1%)	12 (2%)

Plagiarism sources			
ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Попелінов Данило Дмитрович

Тема: Хмарний сервер для зберігання даних на основі Raspberry Pi

Спеціальність: 123 «Комп'ютерна інженерія»

бсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 61

1. Короткий зміст роботи та прийнятих рішень: є розробка хмарного сервера для зберігання даних з використанням апаратної платформи Raspberry Pi

2. Висновок про відповідність роботи дипломному завданню: *Робота повністю відповідає поставленому завданню.*

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи - проведено аналіз сучасних підходів до організації хмарних сервісів зберігання даних, розглянуто принципи їх побудови та функціонування. Здійснено порівняльний аналіз відкритих платформ для створення приватних хмар, зокрема Nextcloud, ownCloud та інших. Розглянуто можливості використання мікрокомп'ютера Raspberry Pi як апаратної основи для реалізації недорогого, енергоефективного хмарного сервера. Обґрунтовано доцільність обраного рішення, сформульовано мету, завдання та основні етапи реалізації проекту. В другому розділі досліджено набір апаратних і програмних компонентів, необхідних для реалізації приватного хмарного сховища. Надано детальний опис апаратної платформи Raspberry Pi, її технічних характеристик, переваг і обмежень. Також охарактеризовано вибрані програмні засоби: операційну систему Raspberry Pi OS, веб-сервер Apache, мову програмування PHP, систему керування базами даних MariaDB та платформу Nextcloud. Проаналізовано вимоги до сумісності, продуктивності та безпеки. В третьому розділі описано практичні кроки розгортання хмарного середовища на базі Raspberry Pi. Послідовно наведено процес

встановлення та налаштування операційної системи, конфігурування мережі, налаштування середовища LAMP (Linux, Apache, MariaDB, PHP), інсталяції платформи Nextcloud. Проведено тестування функціонування серверу в локальній мережі, перевірено доступність з клієнтських пристроїв. У результаті реалізовано повнофункціональний прототип приватної хмарної системи зберігання даних.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага до реалізації апаратної частини проекту.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Аскеров В'ячеслав Васильович, асистент

"9" червня 2025 р.

 (підпис)

Завідувачу кафедри КІС
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Данило ПОПЕЛІНОВ

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

01.06 2025 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Хмарний сервер для зберігання даних на основі Raspberry Pi

Автор: Данило ПОПЕЛІНОВ

Спеціальність: 123- Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: В'ячеслав АСКЕРОВ, асистент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1. Всі запозичені матеріали розміщені виключно в розділах, присвячених аналізу існуючих аналогів та прототипів, що не стосуються безпосередньо авторського дослідження та отриманих результатів;
2. Виявлені запозичення мають фрагментарний характер або супроводжуються належним чином оформленими бібліографічними посиланнями на відповідні джерела;
3. Частина знайдених збігів є загальноживаними словосполученнями або термінами, що підтверджується зазначенням системи перевірки про наявність подібних фрагментів одночасно у 10-40 різних джерелах;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 1.49% і адресується до 21 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС





В'ячеслав АСКЕРОВ

Андрій Нічепорук

Ольга ПАВЛОВА